



**DESENVOLVIMENTO DE NOVAS METODOLOGIAS BASEADAS
EM CLUSTERING PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS HÍBRIDOS NO ESPAÇO DE ESTADOS**

RENATO VILELA LOPES

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

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE NOVAS METODOLOGIAS BASEADAS
EM CLUSTERING PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS HÍBRIDOS NO ESPAÇO DE ESTADOS**

Renato Vilela Lopes

ORIENTADOR: Geovany Araújo Borges
CO-ORIENTADOR: João Yoshiyuki Ishihara

TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS
ELETRÔNICOS E AUTOMAÇÃO

Publicação: PPGEA.TD 088/14
Brasília/DF: Junho-2014

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DESENVOLVIMENTO DE NOVAS METODOLOGIAS BASEADAS
EM CLUSTERING PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS HÍBRIDOS NO ESPAÇO DE ESTADOS

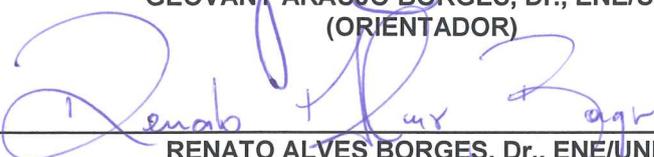
RENATO VILELA LOPES

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA
FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:



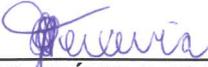
GEOVANY ARAÚJO BORGES, Dr., ENE/UNB
(ORIENTADOR)



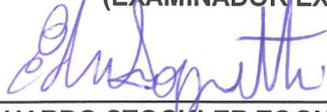
RENATO ALVES BORGES, Dr., ENE/UNB
(EXAMINADOR INTERNO)



DANIEL OLIVEIRA CAJUEIRO, Dr., FACE/UNB
(EXAMINADOR EXTERNO)



BRUNO OTÁVIO S. TEIXEIRA, Dr., DELT/UFMG
(EXAMINADOR EXTERNO)



EDUARDO STOCKLER TOGNETTI, Dr., ENEUNB
(EXAMINADOR INTERNO)

Brasília, 06 de junho de 2014.

FICHA CATALOGRÁFICA

LOPES, RENATO VILELA

Desenvolvimento de Novas Metodologias Baseadas em Clustering para Identificação de Sistemas Dinâmicos Híbridos no Espaço de Estados [Distrito Federal] 2014.

xx, 145p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2014).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Identificação

2. Sistemas Híbridos

3. Espaço de estados

4. Múltiplas entradas - Múltiplas saídas

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

LOPES, R.V. (2014). Desenvolvimento de Novas Metodologias Baseadas em Clustering para Identificação de Sistemas Dinâmicos Híbridos no Espaço de Estados, Tese de Doutorado em Engenharia Elétrica, Publicação PPGEA.TD-088/14, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 145p.

CESSÃO DE DIREITOS

AUTOR: Renato Vilela Lopes

TÍTULO: Desenvolvimento de Novas Metodologias Baseadas em Clustering para Identificação de Sistemas Dinâmicos Híbridos no Espaço de Estados.

GRAU: Doutor ANO: 2014

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

Renato Vilela Lopes

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

"

"

"

"

*A minha mãe, Maria Tereza, e a
minha esposa, Maria Alzira.*

AGRADECIMENTOS

É com imensa alegria e um enorme sentimento de alívio que começo a escrever esta seção de agradecimentos. Alegria, porque é a oportunidade de agradecer a todos que contribuíram direta ou indiretamente para a realização deste trabalho. Alívio, porque finalmente este trabalho de doutorado está chegando ao fim.

Começo agradecendo a Deus e a Nossa Senhora Aparecida que me deram forças e não me deixaram fraquejar nos momentos de maior dificuldade.

Agradeço aos professores Geovany Borges e João Ishihara pela orientação, amizade, paciência e compreensão ao longo de toda esta jornada.

A minha esposa, Maria Alzira, minha maior inspiração e motivação para o término deste trabalho. Agradeço todo o seu amor, carinho, companheirismo e palavras de incentivo nos momentos de medo e dúvidas. Obrigado também por compreender as minhas ausências nos diversos finais de semana e viagens.

A minha mãe, Maria Tereza, meu exemplo de pessoa, minha referência, meu porto seguro, ficam agradecimentos insuficientes por todo amor, carinho, dedicação e compreensão. E é por meio do meu esforço pessoal que procuro retribuir uma pequena parte de tudo que ela fez e faz por mim.

Aos amigos do LARA pelos momentos de riso e descontração que tornaram o meu ambiente de trabalho um ambiente descontraído e agradável. Devo agradecimentos especiais ao Pedro, Henrique, Roberto e Luis pelas discussões técnicas que me ajudaram em diversos momentos do desenvolvimento deste trabalho.

Por fim, mas não menos importante, deixo meus agradecimentos aos meus amigos e familiares, cujos nomes não cito explicitamente com medo de esquecer-me de algum deles, que mesmo distantes me incentivaram a batalhar por mais este sonho.

RESUMO

DESENVOLVIMENTO DE NOVAS METODOLOGIAS BASEADAS EM CLUSTERING PARA IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS HÍBRIDOS NO ESPAÇO DE ESTADOS

Autor: Renato Vilela Lopes

Orientador: Prof. Geovany Araújo Borges, ENE/UnB

Co-orientador: Prof. João Yoshiyuki Ishihara, ENE/UnB

Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e de Automação

Brasília, 06 de junho de 2014

Este trabalho aborda o problema de identificação de sistemas dinâmicos híbridos com modelos descritos no espaço de estados. Mais especificamente é discutido o caso particular de sistemas chaveados lineares a tempo discreto, em que a variável discreta do estado denota o modelo matemático que descreve a forma como os estados contínuos evoluem. As abordagens desenvolvidas utilizam pressupostos gerais de que os instantes de chaveamento, o número de modelos locais, os parâmetros e as ordens destes modelos podem ser simultaneamente desconhecidos. A principal contribuição do trabalho é o desenvolvimento de uma nova estrutura de identificação onde os modelos locais são identificados antes da estimação dos instantes de chaveamento do sistema. Para isso, explora-se o uso de técnicas de identificação em subespaços, clustering, classificação de dados e filtragem híbrida. Essa nova estrutura de identificação resultou no desenvolvimento de seis novos algoritmos. Um importante aspecto dos algoritmos propostos é o fato de ser mantido o controle sobre a base de estados na qual as matrizes dos diferentes modelos locais são calculadas, assim os diferentes sistemas locais podem ser diretamente combinados. A eficiência das abordagens propostas é demonstrada por meio de simulações numéricas e comparações com uma abordagem já existente na literatura.

ABSTRACT

DEVELOPMENT OF NEW METHODOLOGIES BASED IN CLUSTERING FOR IDENTIFICATION OF HYBRID DYNAMIC SYSTEMS IN STATE SPACE FORMS

Author: Renato Vilela Lopes

Advisor: Prof. Geovany Araújo Borges, ENE/UnB

Co-advisor: Prof. João Yoshiyuki Ishihara, ENE/UnB

Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e de Automação

This work addresses the problem of identification of hybrid dynamical systems with models in state space. More specifically, it is discussed the particular case of discrete-time switched linear systems where the discrete state variable denotes the mathematical model that describes how the continuous states evolve. The approaches developed use general assumptions such as the switching times, the number of local models, the parameters and the orders of these models can be simultaneously unknown. The main contribution of this work is the development of a new identification framework where the local models are identified prior to the estimation of the switching times. This task is accomplished by exploring the combined use of subspace identification methods, clustering, data classification and hybrid stochastic filtering. This new framework resulted in the development of six new algorithms. An important feature of the algorithms is that the matrices of the different submodels can be directly combined since they are obtained for the same base states. The efficiency of the proposed approaches is demonstrated by numerical simulations and comparison of the results with a previous literature approach.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA E OBJETIVOS	2
1.3	IDENTIFICAÇÃO DE SISTEMAS HÍBRIDOS: ESTADO DA ARTE	4
1.3.1	SISTEMAS DINÂMICOS CHAVEADOS LINEARES	6
1.4	PROPOSTAS E CONTRIBUIÇÕES	15
1.5	APRESENTAÇÃO DO MANUSCRITO	15
2	FUNDAMENTOS TEÓRICOS	17
2.1	IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS LINEARES	17
2.1.1	ALGORITMO COMBINADO ROBUSTO - ACR	21
2.1.2	MATRIZES DO ESPAÇO DE ESTADOS NA FORMA CANÔNICA	24
2.1.3	ESTIMAÇÃO DA ORDEM DOS MODELOS	30
2.2	CLASSIFICAÇÃO DE DADOS	32
2.2.1	ALGORITMO DE AGLOMERAÇÃO COMPETITIVA ROBUSTA - RCA	34
2.3	FILTRAGEM HÍBRIDA	39
2.3.1	ALGORITMO IMM	42
2.4	VALIDAÇÃO DOS MODELOS	47
3	ABORDAGENS DE IDENTIFICAÇÃO PROPOSTAS	49
3.1	INTRODUÇÃO	49
3.2	DEFINIÇÃO DO PROBLEMA	52
3.3	ALGORITMO 1	53
3.3.1	RESULTADOS NUMÉRICOS	59
3.3.2	AJUSTE DOS PARÂMETROS DO ALGORITMO	66
3.4	ALGORITMO 2	67
3.4.1	RESULTADOS NUMÉRICOS	69
3.5	ALGORITMO 3	73
3.5.1	RESULTADOS NUMÉRICOS	77
3.5.2	AJUSTE DOS PARÂMETROS DO ALGORITMO	82
3.6	ALGORITMO 4	85
3.6.1	RESULTADOS NUMÉRICOS	93
3.7	ALGORITMO 5	96
3.7.1	ALGORITMO PO-MOESP ADAPTADO	99
3.7.2	RESULTADOS NUMÉRICOS	103
3.8	ALGORITMO 6	109
3.8.1	RESULTADOS NUMÉRICOS	112

4	RESULTADOS COMPARATIVOS.....	117
4.1	INTRODUÇÃO	117
4.2	ALGORITMO DO PEKPE.....	118
4.3	RESULTADOS NUMÉRICOS	121
5	CONCLUSÕES	125
5.1	TRABALHOS FUTUROS	127
	REFERÊNCIAS BIBLIOGRÁFICAS.....	129
	ANEXOS	141
A	LISTA DE PUBLICAÇÕES.....	143

LISTA DE FIGURAS

1.1	Diagrama esquemático de um sistema dinâmico híbrido com dois modos de operação.	4
1.2	Representação de um sistema dinâmico chaveado com M modelos locais (Adaptado de [1]).....	6
2.1	Diagrama ilustrando a interação entre os diferentes conceitos utilizados nos algoritmos que serão apresentados no Capítulo 3.....	18
2.2	Sinais utilizados e obtidos no processo de identificação. (a) Sinal de entrada (u_k) em volts. (b) Sinal de saída (y_k) em volts. Nestas figuras, para melhor visualização, apenas um pequeno trecho dos sinais está sendo mostrado.....	29
2.3	Comparação entre a saída real do sistema (proveniente do conjunto de dados de validação) e a estimada pelo modelo. (a) Modelo estimado pelo algoritmo ACR (b) Modelo estimado pelo algoritmo ACR associado com a metodologia descrita na seção 2.1.2 para representar todos os modelos na mesma base de estados.....	30
2.4	Resultados do RCA em um conjunto de dados ruidosos. (a) Conjunto de dados original. (b) Resultado após a primeira iteração. (c) Resultado após a segunda iteração. (d) Resultado após 12 iterações (convergiu). Nesta última figura, os pontos atribuídos a cada cluster estão representados em cores diferentes.	38
2.5	Diagrama do filtro IMM mostrando a interação entre as diferentes etapas do algoritmo.....	45
2.6	Probabilidades dos modos estimadas pelo filtro IMM.....	46
2.7	Sinais utilizados e obtidos na estimação de estados utilizando o filtro IMM.....	46
3.1	Fluxograma da metodologia proposta para identificação de sistemas dinâmicos híbridos chaveados com modelos expressos no espaço de estados. As caixas correspondem às etapas do procedimento, e os rótulos nas bordas descrevem as entradas e as saídas de cada etapa.	50
3.2	Diagrama do Algoritmo 1. Os rótulos ao lado das setas representam as saídas de cada etapa.....	55
3.3	Sinais utilizados e obtidos com o sistema (3.1). (a) Sequencia de chaveamento utilizada na simulação. (b) Pequeno trecho do sinal PRBS aplicado ao sistema. (c) Sinal de saída utilizado para a identificação dos modelos.	61
3.4	Resultado do algoritmo de clustering mostrando que quatro modelos locais foram identificados. As cruzes em vermelho indicam os centros de cada cluster. 63	
3.5	Resultado da estimação dos modos do sistema com um SNR de 35 dB.	63

3.6	Validação dos dados com SNR de 35 dB. Resultado VAF = 98,49%. As linhas verticais tracejadas indicam os instantes de chaveamento.	64
3.7	Histogramas dos valores de VAF (%) obtidos com a validação dos dados com os modelos estimados nos diferentes níveis de ruído considerados. A distribuição de valores de VAF de 0% a 100% foram divididos em conjuntos (bins) de 5%. Para cada bin é mostrado quantas das 100 simulações ficaram dentro daquele bin.	65
3.8	Autovalores das matrizes estimadas A_{m_k} no plano complexo para 100 experimentos com (a) SNR = 10 dB (b) SNR = 20 dB (c) SNR = 30 dB (d) SNR = 35 dB. As cruzes em vermelho correspondem aos pólos reais dos sistemas...	65
3.9	Sinais de chaveamentos utilizados nas simulações. (a) Dinâmica de chaveamento lenta. (b) Dinâmica de chaveamento rápida.	70
3.10	Resultado típico do algoritmo de clustering para o Algoritmo 1. (a) Dinâmica de chaveamento lenta. (b) Dinâmica de chaveamento rápida.	72
3.11	Resultado típico para estimação de modos do sistema com SNR = 35dB. (a) Algoritmo 1. (b) Algoritmo 2. Para melhor visualização é apresentado apenas um pequeno trecho dos sinais de chaveamento estimados.	73
3.12	Diagrama do Algoritmo 3.	75
3.13	O resultado de clustering obtido com o Algoritmo 3, demonstrando que os quatro modelos locais foram identificados. As cruzes em vermelho são os centros de cada cluster e os pontos em marrom representam os pontos dúbios e não utilizados na identificação parcial dos modelos locais.	80
3.14	Resultado típico para a estimação de modos com SNR de 20 dB. (a) Resultado obtido com o Algoritmo 1 (17,92% de erros de classificação). (b) Resultado obtido com o Algoritmo 3 (5,1% de erros de classificação).	81
3.15	Re-dimensionamento dos eixos da Figura 3.14 para ilustrar os detalhes da estimação dos modos. (a) Resultado obtido com o Algoritmo 1. (b) Resultado obtido com o Algoritmo 3.	82
3.16	Distribuição das medidas de VAF nas 100 simulações independentes dos algoritmos com SNR de 20 dB. (a) Distribuição obtida utilizando-se o Algoritmo 1. (b) Distribuição obtida usando-se o Algoritmo 3. Nestas figuras os valores foram divididos em bins de 1%.	83
3.17	Validação de dados com SNR de 20 dB. (a) Resultado obtido com o Algoritmo 1 (VAF = 83,99%). (b) Resultado obtido com o Algoritmo 3 (VAF = 93,38%).	83
3.18	Diagrama do Algoritmo 4.	88
3.19	Sinais utilizados no processo de identificação com o Algoritmo 4. a) Sinal de chaveamento. b) Pequeno trecho do sinal de excitação. c) Saída do sistema.	93

3.20	Comparação das distribuições dos valores de VAF nas 100 simulações realizadas nos diferentes níveis de ruído considerados. (a) Histogramas obtidos com o Algoritmo 3. (b) Histogramas obtidos com o Algoritmo 4.	95
3.21	Resultado do algoritmo de clustering em diferentes iterações do algoritmo. (a) Primeira iteração. (b) Segunda iteração. (c) Terceira iteração. (d) Quarta iteração. No caso apresentado, o algoritmo convergiu após 4 iterações.	96
3.22	Típico resultado para a estimação dos modos. Neste caso, o erro na estimação foi de 0,37%.	106
3.23	Pólos estimados dos sistemas no plano complexo para 100 simulações.	107
3.24	Validação dos dados considerando ruído colorido com SNR = 30 dB. (a) Primeira saída do sistema e o valor de VAF obtido foi de 99,09%. (b) Segunda saída do sistema com VAF = 97,02%. Nesta figura, para melhor visualização, apenas um pequeno trecho do sinal utilizado para validação está sendo apresentado.	108
3.25	Resultado para a estimação dos modos com: (a) SNR of 40 dB (Erro de classificação = 1,80%); (b) SNR of 20 dB (Erro de classificação = 4,45%).	114
4.1	Diagrama de blocos ilustrando a metodologia utilizada no Algoritmo do Pekpe.	118
4.2	Resultado típico para a estimação de modos com SNR = 20 dB: (a) Algoritmo do Pekpe. (b) Algoritmo 6.	124
4.3	Distribuições dos valores de VAF nas 100 simulações independentes realizadas com os dois algoritmos. (a) Algoritmo do Pekpe com SNR = 30 dB. (b) Algoritmo 6 com SNR = 30 dB. (c) Algoritmo do Pekpe com SNR = 20 dB. (d) Algoritmo 6 com SNR = 20 dB. (e) Algoritmo do Pekpe com SNR = 10 dB. (a) Algoritmo 6 com SNR = 10 dB.	124

LISTA DE TABELAS

1.1	Relação de trabalhos publicados em identificação de sistemas dinâmicos chaveados lineares/af ns.	7
1.2	Considerações realizadas em cada abordagem off-line desenvolvida para identificação de sistemas chaveados no espaço de estados.	14
3.1	Comparação entre os pólos reais e estimados com SNR = 35dB.	64
3.2	Desempenho do Algoritmo 1 com diferentes proporções de ruído. Aqui, o termo Sucessos indica o número de vezes que o algoritmo identificou o número de modelos locais corretamente nas 100 simulações.	64
3.3	Comparação de desempenho do Algoritmo 1 com o ACR para o cenário onde apenas um modelo local está ativo e SNR = 20dB.	66
3.4	Comparação de desempenho dos algoritmos com dinâmica lenta de chaveamento. Aqui, o termo Erro representa a média dos erros na estimação dos modos.	71
3.5	Comparação de desempenho dos algoritmos com dinâmica de chaveamento rápida.	72
3.6	Comparação de desempenho dos algoritmos com diferentes níveis de ruído. ...	79
3.7	Desempenho do Algoritmo 3 para diferentes valor do parâmetro $\epsilon_{cluster}$	84
3.8	Comparação de performance dos Algoritmos 3 e 4 com diferentes níveis de ruído.	94
3.9	Desempenho do algoritmo quando os sistemas estão sujeitos somente a ação de ruído branco. Aqui, o termo VAF1 (%) e VAF2 (%) indicam o cálculo do valor de VAF para as duas saídas do sistema. Ordem X (%) indica o número de vezes que o algoritmo estimou corretamente a ordem do sistema X.	105
3.10	Desempenho do algoritmo para cenário onde os sistemas estão sujeitos a ação de ruído colorido.	106
3.11	Resultados do algoritmo com ruído SNR = 10 dB e ordem dos modelos fixada em 5 na Etapa 2 do algoritmo.	108
3.12	Comparação de performance dos Algoritmos 4 e 6 com diferentes níveis de ruído.	112
3.13	Desempenho do Algoritmo 6 na estimação de ordem dos modelos locais com diferentes níveis de ruído. Aqui, o termo Ordem _x representa o número de vezes que o algoritmo estimou corretamente a ordem do modelo local x nas 100 simulações.	113
3.14	Análise de convergência do algoritmo com diferentes valores do parâmetro ξ_{conv} . As simulações foram realizadas com SNR = 20 dB e $\xi_{iterações} = 7$	114

4.1	Comparação de desempenho dos algoritmos com diferentes níveis de ruído. Aqui, os termos VAF_x e MSE_x indicam o cálculo do valor de VAF e MSE, respectivamente, para a saída x do sistema e o termo σ_Z indica o desvio padrão da medida Z	123
-----	---	-----

LISTA DE SÍMBOLOS

Nesta seção estão listados os símbolos mais importantes deste trabalho, seguidos de seus significados mais comuns. Contudo, permitem-se algumas redefinições locais ao longo do texto visando à melhor compreensão de discussões específicas e evitando-se o número excessivo de símbolos na tese.

Símbolos Latinos

k	Número da amostra
u	Vetor de entradas
y	Vetor de medições
m_k	Modo do sistema no instante de tempo k
A, B, C, D	Matrizes do sistema dinâmico
$A_{m_k}, B_{m_k}, C_{m_k}, D_{m_k}$	Matrizes do sistema dinâmico definidas pelo modo m no instante de tempo k
v	Ruído de medição ou vetor de velocidade
w	Ruído de processo
n	Ordem do sistema
x	Vetor de estados contínuos ou vetores do espaço n -dimensional
n_u	Número de entradas do sistema
n_y	Número de saídas do sistema
M	Número de modos do sistema
Q	Matriz de covariância do ruído de processo
R	Matriz de covariância do ruído de medição
S	Matriz de covariância cruzada dos ruídos de processo e medição
\mathbb{R}	Conjunto dos números reais
\mathbb{Z}	Conjunto dos números inteiros
\mathbb{M}	Conjunto de modos do sistema
\mathbb{N}	Conjunto dos números naturais
I	Matriz identidade de dimensões apropriadas
Pr	Probabilidade discreta
r_k	Vetor de regressores no instante de tempo k
u_{qk}	Representa o grau no qual o ponto x_q pertence ao cluster k
i	Número de blocos linhas nas matrizes de block Hankel

s	Quantidade de dados utilizada na identificação em subespaços
X_p^d	Sequencia de estados determinística do passado
X_f^d	Sequencia de estados determinística futuras
H_i^d	Matriz de Toeplitz
U_p e U_p^+	Matriz contendo as entradas passadas $U_{0 i-1}$ e $U_{0 1}$, respectivamente
U_f e U_f^-	Matriz contendo as entradas futuras $U_{i 2i-1}$ e $U_{i+1 2i-1}$, respectivamente
Y_p e Y_p^+	Matriz contendo as saídas passadas $U_{0 i-1}$ e $U_{0 1}$, respectivamente
Y_f e Y_f^-	Matriz contendo as saídas futuras $U_{i 2i-1}$ e $U_{i+1 2i-1}$, respectivamente
W_p	Matriz contendo as entradas e saídas passadas
Z_i	Projeção ortogonal de Y_f sobre W_p e U_f
O_i	Projeção oblíqua de Y_f sobre W_p na direção de U_f
\hat{X}_i e \tilde{X}_f	Sequencias de estados com diferentes condições iniciais
W_1 e W_2	Matrizes de pesos
L	Matriz de permutação que define a base de estados do sistema identificado
p_A	Polinômio característico da matriz A
a_i	Coefficientes do polinômio característico do sistema
J	Função custo utilizada no RCA
U_H	<i>Constrained fuzzy H-partition matrix</i>
H	Quantidade de clusters identificados no RCA
H_q	Matriz de covariância do clusters q
d	Medida de distância
N_s	Cardinalidade robusta
\bar{N}_t	Média ponderada das cardinalidades
$c, c_{min}, c_0, k_0, w_{qk}$	Parâmetros de ajuste do RCA
K_q	Constante de Integração do cluster q no RCA
T_q	Mediana das distâncias em relação ao cluster q
S_q	Mediana dos desvios absolutos do cluster q
X	Conjunto de vetores no espaço n -dimensional
h_q	Centro do cluster q
$\hat{p}^{(p)}$	Probabilidade da hipótese p
$h_k^{(i)}$	Denota a i -ésima hipótese no instante de tempo k

F	Função de penalidade utilizada no algoritmo de estimação de ordem do sistema
T	Quantidade de dados utilizada para determinação da ordem do sistema
W	Tamanho da janela de dados usada na identificação
\mathcal{D}	Conjunto de dados amostrados
LD	Pequenos conjuntos de dados com W elementos
\mathcal{T}	Horizontes de tempo da aquisição de dados
$\{\mathcal{C}_q\}_{q=1}^M$	Coleção de M clusters
$\{\mathcal{C}_q^c\}_{q=1}^M$	Conjunto de dados clusterizados

Símbolos Gregos

Π	Matriz de Transição de Probabilidades
$\pi_{i,j}$	Probabilidade de transição do modo i para o modo j
τ	Período de amostragem
Δ_i^d	Matriz de controlabilidade estendida
φ	Autovalores da matriz dinâmica
λ_j	Números reais diferentes de zero
β	Protótipos que caracterizam os clusters
ρ	Função de perda robusta utilizada no RCA
α	Parâmetro de projeto do RCA
ι	Parâmetro de projeto do RCA
Δ_c	Parâmetro de ajuste do RCA
ϵ_0	Parâmetro que define a eliminação de clusters no RCA
β_q	Centro do cluster q
τ_{dwell}	Tempo de permanência do sistema em cada modo
θ_j	Vetor de parâmetros de Markov
Γ_i	Matriz de observabilidade estendida
$\underline{\Gamma}_i$	Matriz de observabilidade estendida sem as últimas l linhas
σ_i	Valores singulares resultantes da operação de decomposição em valores singulares

Subscritos e Sobrescritos

\bar{x}	Estado do sistema após aplicação de uma transformação de similaridade ou valor predito ou média de x
\hat{x}	Estimativa ou valor corrigido de x
(\bullet^d)	O sobrescrito d denota a parte determinística de (\bullet)

- (\bullet^{st}) O sobrescrito st denota a parte estocástica de (\bullet)
- (\bullet_f) O subscrito f é usado para se referir a valores futuros
- (\bullet_p) O subscrito o é usado para se referir a valores passados

Siglas

ACR	Algoritmo Combinado Robusto (do inglês <i>Robust Combined Algorithm</i>)
AIC	Critério da Informação de Akaike (do inglês <i>Akaike Information Criterion</i>)
ELC	Sistemas Lineares Complementares Estendidos (do inglês <i>Extended Linear Complementarity</i>)
EM	Maximização da Esperança (do inglês <i>Expectation Maximization</i>)
ERA	Algoritmo de Realização de Sistemas (do inglês <i>Eigensystem Realization Algorithm</i>)
FDP	Função Densidade de Probabilidade
FK	Filtro de Kalman
GPB	Pseudo-Bayesiano Generalizado (do inglês <i>Generalized Pseudo-Bayesian</i>)
GPCA	Análise de Componentes Principais Generalizados (do inglês <i>Generalized Principal Components Analysis</i>)
IMM	Interacting Multiple Model
LD	Conjunto de Dados Locais
JL	Saltos Lineares (do inglês <i>Jump Linear</i>)
JML	Saltos Markovianos Lineares (do inglês <i>Jump-Markov Linear</i>)
LC	Sistemas Lineares Complementares (do inglês <i>Linear Complementarity</i>)
MIMO	Múltiplas Entradas e Múltiplas Saídas (do inglês <i>Multiple Input Multiple Output</i>)
MISO	Múltiplas Entradas e Uma Saída (do inglês <i>Multiple Inputs Single Output</i>)
MJLS	Sistemas Lineares com Saltos Markovianos (do inglês <i>Markov Jump Linear Systems</i>)
MLD	Sistemas Mistos Lógicos Dinâmicos (do inglês <i>Mixed Logical Dynamical</i>)
MMPS	<i>Max-Min-Plus-Scaling</i>
MM	Múltiplos Modelos
MOESP	<i>Multivariable Output-Error State Space</i>
MPT	Matriz de Probabilidades de Transição
MSE	Erro Médio Quadrático (do inglês <i>Mean Squared Error</i>)
NIDC	Critério utilizado para estimação da ordem dos modelos locais
PEM	Método de Predição do Erro (do inglês <i>Prediction Error Method</i>)
PRBS	Sinal Binário Pseudo Aleatório (do inglês <i>Pseudo Random Binary Sequence</i>)
PWA	Sistemas Af ns por Partes (do inglês <i>PieceWise Affine</i>)

PWARX	Auto Regressivo Exógeno Af m por Partes (do inglês <i>PieceWise Af-fine Auto-Regressive eXogenous</i>)
PWARMAX	Auto Regressivo Exógeno Af m por Partes com Média Móvel (do inglês <i>PieceWise Autoregressive Moving Average eXogeneous</i>)
RCA	Aglomeraco Competitiva Robusta (do inglês <i>Robust Competitive Agglomeration</i>)
SARX	Chaveados Auto Regressivo Exógeno, (do inglês <i>Switched Auto-Regressive eXogenous</i>)
SISO	Uma Entrada e Uma Saída (do inglês <i>Single Input Single Output</i>)
SMARX	Sistema Chaveado Markoviano Auto Regressivo Exógeno (do inglês <i>Switched Markov ARX System</i>)
SNR	Relaco Sinal Ruído (do inglês <i>Signal to Noise Ratio</i>)
SVD	Decomposico em Valores Singulares (do inglês <i>Singular Value Decomposition</i>)
VAF	<i>Variance-Accounted-For</i>

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Na área de engenharia de controle, a identificação de sistemas é usada para construir modelos matemáticos de processos físicos reais. Esses modelos são análogos matemáticos que apresentam algumas características observadas nos sistemas reais. Tradicionalmente, os modelos utilizados vêm sendo, basicamente, divididos em dois grandes grupos, os modelos a tempo contínuo descritos por equações diferenciais e que representam a evolução do sistema continuamente no tempo, e os modelos a tempo discreto, descritos por equações de diferenças e representam a evolução do sistema em instantes discretos [2].

No entanto, com o desenvolvimento tecnológico, o computador e os dispositivos semicondutores passaram a ter grande importância em diversas aplicações. Com efeito, surgiram sistemas de natureza híbrida que não podem ser facilmente modelados utilizando-se de modelos puramente contínuos ou modelos puramente discretos. Adicionalmente, a conversão destes sistemas em entidades puramente discretas ou puramente contínuas pode levar a uma solução inadequada do problema [3].

Ao mesmo tempo, o crescente desenvolvimento de estudos na área de sistemas autônomos provocou o surgimento de um grande número de sistemas inteligentes e mais capazes. Exemplos desses sistemas incluem as missões espaciais, os veículos aéreos não tripulados, veículos aquáticos não tripulados, robôs autônomos, entre outros. Nestas aplicações a capacidade de lidar com as incertezas inerentes ao ambiente em que estão operando é fundamental para melhorar o desempenho e a precisão do sistema [4].

Desta forma, a utilização de um modelo matemático mais flexível e representativo do ambiente e do próprio sistema constitui um ponto crucial no projeto desses sistemas, uma vez que eles devem ser capazes de lidar com distúrbios, falhas de componentes, desconhecimento completo ou parcial de dinâmicas e incertezas na estimação de posição e/ou obstáculos [4], [5].

Nestes contextos, surgem os sistemas dinâmicos híbridos. Formalmente, os sistemas dinâmicos híbridos são definidos como uma classe de sistemas que são caracterizados pela combinação de comportamentos que são típicos de sistemas dinâmicos a tempo contínuo com comportamentos que são típicos de sistemas dinâmicos a eventos discretos [6]. Eles são ideais para caracterizar a interação entre dispositivos lógicos e processos contínuos. Neste caso, o comportamento contínuo é o resultado da evolução natural do processo físico, enquanto o comportamento (ou evento) discreto é atribuído à presença de interruptores, de fases operacionais, transições, um código de programa de computador, etc.

No entanto, a abordagem híbrida também pode ser utilizada para descrever fenômenos

reais que exibem descontinuidades ou aproximar dinâmicas complexas, lineares ou não-lineares, através da concatenação de modelos lineares mais simples [7]. De fato, muitos sistemas práticos têm suas dinâmicas descritas mais fielmente por um conjunto de modelos matemáticos no lugar de uma abordagem de modelo único. Como exemplo cita-se um circuito eletrônico chaveado, em que voltagens e correntes podem sofrer mudanças bruscas e terem suas dinâmicas alteradas dependendo dos estados lógicos das chaves [1]. Neste contexto, a porção discreta do vetor de estados geralmente denota o modo de operação do sistema e define a forma segundo a qual as variáveis de estado contínuas evoluem.

Desta forma, os formalismos existentes para os sistemas dinâmicos híbridos mostram-se adequados para o tratamento de uma grande variedade de problemas [8]. Conseqüentemente, o paradigma de modelagem híbrida têm sido considerado nas mais diversas aplicações, como por exemplo: modelagem de veículos aéreos não tripuláveis [9], sistemas de localização [1], visão computacional [10], controle de comportamento humano [11], controle de automóveis [12], sistemas de potência [13], sistemas de manufatura [14], controle de tráfego aéreo [15] e evolução da infecção de HIV em pacientes [16]. Como resultado, nas últimas décadas aumentou o interesse nesses sistemas e numerosos resultados em análise, estabilidade, estimação de estados e controle de sistemas dinâmicos híbridos tem aparecido na literatura, alguns exemplos de trabalhos são [1], [17], [18], [19], [20], [21], [22], [23].

No entanto, na maioria desses trabalhos assume-se que o modelo híbrido do sistema está disponível a priori ou que pode ser determinado a partir das leis físicas que regem o processo. A complexidade dos sistemas reais pode comprometer esse pressuposto [24]. Por essa razão é importante ser capaz de identificar os sistemas híbridos a partir dos dados de entrada/saída. Assim, um processo de identificação de sistemas híbridos pode ser utilizado para substituir ou complementar os modelos obtidos através das leis físicas.

1.2 DEFINIÇÃO DO PROBLEMA E OBJETIVOS

Um sistema híbrido a tempo discreto combinando variáveis de estado contínuas e discretas pode ser descrito como [3], [1]:

$$\begin{cases} x_k &= f_{m_{k-1}}(x_{k-1}, u_{k-1}, w_{k-1}), \\ y_k &= h_{m_k}(x_k, u_k, v_k), \end{cases} \quad (1.1)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto, e M é o número de submodelos; $f_{m_{k-1}}$ é uma função de evolução do processo dependente do modo e h_{m_k} é a função de medição do sistema. Os vetores $x_k \in \mathbb{R}^{n_{m_k}}$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos ¹, as entradas e saídas do sistema. Os vetores $w_k \in \mathbb{R}^{n_{m_k}}$ e $v_k \in \mathbb{R}^{n_y}$

¹Embora o sistema (1.1) esteja representado em tempo discreto, o termo estado contínuo é comumente usado na literatura para diferenciar os estados referentes ao processo físico dos modos de operação (estados discretos) do sistema.

são termos de ruído de processo e medida, respectivamente.

O problema de identificação dos sistemas dinâmicos híbridos consiste na determinação da quantidade de modelos locais M e a na obtenção de estimativas das funções $f_{m_{k-1}}$ e h_{m_k} , baseado em uma sequência de dados amostrados $\{u_j, y_j\}_{j=1}^k$. Esse problema é desafiador uma vez que, em geral, os estados contínuos, o estado discreto e as funções de chaveamento são desconhecidas. A principal dificuldade na identificação desses modelos a partir de dados experimentais é causada pelo forte acoplamento entre as dinâmicas contínuas e discretas do sistema [25].

Na maioria dos sistemas na natureza somente os estados ou as saídas das dinâmicas contínuas são observadas. O estado discreto do sistema, ou modo, está oculto na observação, e deve ser inferido a partir dos dados. A classificação dos dados observados em estados discretos pode ser realizada se as dinâmicas contínuas são conhecidas. Da mesma forma, se todos os dados observados são classificados em estados discretos, a dinâmica contínua associada com cada modo pode ser identificada [26].

No entanto, sem o conhecimento a priori das dinâmicas contínuas ou dos estados discretos, o processo de identificação requer simultaneamente a classificação dos dados observados e a identificação da dinâmica associada a cada modo [27], [28], [29]. Trata-se de um difícil problema combinatório, com a complexidade aumentando exponencialmente em função do número de dados amostrados e do número de modos [30].

Diferentes abordagens já foram propostas na literatura. Em [7] é apresentada uma excelente revisão dos métodos de identificação de sistemas híbridos. No entanto, a maioria dos estudos, [31], [32], [33], [28], [29], [34], [35], [36], [37], focam em modelos na forma de entrada-saída tais como modelos Auto Regressivo Exógeno Afim por Partes, PWARX (do inglês, *Piecewise Affine Auto-Regressive Exogenous*) ou modelos Chaveados Auto regressivo Exógeno, SARX (do inglês, *Switched ARX*).

Embora esses modelos possam ser utilizados para descrever sistemas dinâmicos, eles não são os mais adequados para a maioria dos métodos de controle e análises dinâmicas. De fato, modelos no espaço de estados são mais convenientes para descrever sistemas com Múltiplas-Entradas e Múltiplas-Saídas (MIMO), pois proporcionam uma elegante e compacta representação. Além disso, os desenvolvimentos teóricos de observabilidade e controlabilidade foram estudados e são bem estabelecidos com base em modelos na forma de espaço de estados [38].

No entanto, a identificação desses modelos no espaço de estados traz desafios adicionais. Quando as matrizes dos diferentes sub-modelos são identificadas, por exemplo, por técnicas de identificação em subespaços, elas não podem ser combinadas, pois, em geral, estão representadas em bases de estados diferentes. Assim, é necessário transformar todos os sub-modelos para a mesma base de estados [39].

Adicionalmente, em comparação com a estimação de modelos na forma de entrada-saída,

a identificação de sistemas dinâmicos híbridos na forma de modelos no espaço de estados tem a dificuldade de que o estado contínuo é geralmente desconhecido. Portanto, uma vez que o regressor não está totalmente disponível a priori, o particionamento direto do espaço entrada-estado como realizado, por exemplo, nos modelos PWARX, torna-se uma tarefa muito difícil [38].

Desta forma, o principal objetivo deste trabalho é o desenvolvimento de técnicas para identificação de sistemas dinâmicos híbridos a tempo discreto com modelos expressos no espaço de estados e com modos de operação não observáveis. Por tratar-se de um problema bastante complexo, para o qual ainda não há soluções fechadas na literatura, o que se espera apresentar neste estudo é uma metodologia prática para a obtenção de estimativas eficientes da dinâmica contínua e a estimação correta do número de modos do sistema mesmo no cenário onde pouco conhecimento do sistema está disponível a priori. Devido a complexidade teórica inerente, uma prova de convergência dos algoritmos propostos está fora do escopo deste trabalho. No entanto, a convergência dos parâmetros será avaliada empiricamente por meio de simulações numéricas.

1.3 IDENTIFICAÇÃO DE SISTEMAS HÍBRIDOS: ESTADO DA ARTE

Os sistemas híbridos são compostos de dinâmicas contínuas, governadas por leis físicas, e dinâmicas a eventos discretos, controladas por lógicas e regras [35]. O termo sistema híbrido foi utilizado primeiramente para descrever a combinação entre essas dinâmicas em 1966 por Witsenhausen no trabalho [40]. O princípio de funcionamento de um sistema híbrido com dois modos de operação está ilustrado na Figura 1.1.

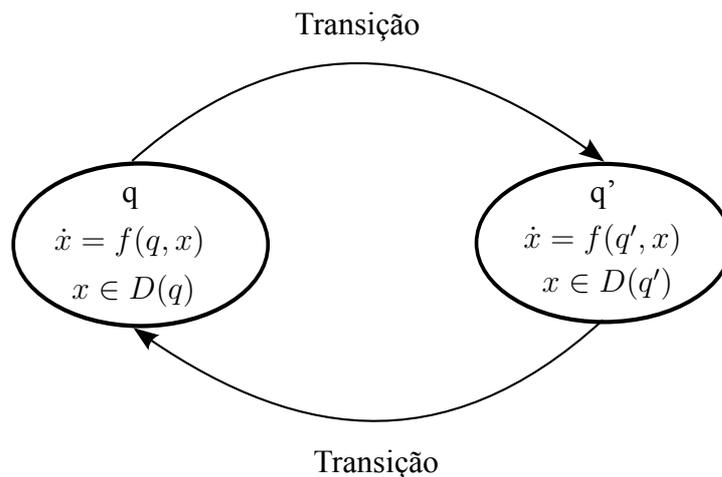


Figura 1.1: Diagrama esquemático de um sistema dinâmico híbrido com dois modos de operação.

Na Figura 1.1, os estados contínuos são representados pela variável x e os dois modos

são representados por q e q' . Quando o sistema está operando no modo q , o estado contínuo x evolui segundo a equação diferencial $\dot{x} = f(q, x)$ no conjunto domínio $x \in D(q)$. Uma transição do modo q para o modo q' pode acontecer e, assim, o estado contínuo passará a evoluir segundo a equação diferencial $\dot{x} = f(q', x)$ com domínio $x \in D(q')$.

As transições entre os modos do sistema podem ser descritas de diferentes formas. Por exemplo, se o sistema estiver trabalhando no modo q , uma transição para o modo q' pode acontecer quando o estado contínuo x atingir uma determinada região de guarda $G(q, q')$. Neste caso, o sistema recebe o nome de Automato Híbrido (do inglês, *Hybrid Automata*) [6]. Por outro lado, a transição entre os modos pode acontecer controlada por uma variável externa, neste caso, o sistema recebe o nome de sistema com Saltos Controlados (do inglês, *Controlled Jumps*) [41]. Recomenda-se a leitura de [41] para uma revisão geral dos diferentes tipos de sistemas híbridos.

Atualmente, os sistemas híbridos têm recebido muita atenção da comunidade científica, principalmente, de pesquisadores das áreas de ciência da computação e controle [35]. No entanto, como métodos tratáveis para analisar sistemas híbridos gerais ainda não estão disponíveis [42], [43], vários autores focam em sub-classes especiais de sistemas dinâmicos híbridos para as quais técnicas de análise e controle estão sendo atualmente desenvolvidas [42]. Alguns exemplos destas sub-classes são os Sistemas Mistos Lógicos Dinâmicos (MLD, do inglês, *Mixed Logical Dynamical*) [44], os sistemas *Max-Min-Plus-Scaling* (MMPS) [45], Sistemas Lineares Complementares (LC, do inglês, *Linear Complementarity*) [46], [47], Sistemas Lineares Complementares Estendidos (ELC, do inglês, *Extended Linear Complementarity*) [48], Sistemas Afins por Partes (PWA, do inglês *Piece Wise Affine*) [49].

Embora haja relações de equivalência entre essas sub-classes [42], cada uma delas apresenta vantagens e desvantagens em relação às outras dependendo da característica que deseja-se estudar. Por exemplo, critérios de estabilidade foram propostos para sistemas PWA, técnicas de controle e verificação para modelos híbridos na forma MLD e condições de existência e unicidade de soluções para sistemas LC [42].

Contudo, do ponto de vista de identificação de sistemas dinâmicos híbridos, a maior parte dos esforços tem sido dedicada a identificação dos sistemas chaveados lineares e sistemas afins por partes [7], [39], [50]. Esse maior interesse é motivado por três razões principais. Primeiro, esses modelos podem ser utilizados para descrever uma vasta gama de processos físicos [51]. Segundo, são adequados para aproximar dinâmicas não-lineares como combinação de vários modelos lineares mais simples [52]. Terceiro, a sua equivalência com várias outras classes de modelos híbridos tornam as técnicas de identificação desenvolvidas para esses modelos adequadas para obter muitos tipos de modelos híbridos a partir dos dados amostrados [7].

Este trabalho abordará a identificação de sistemas dinâmicos chaveados lineares. Desta forma, na próxima seção será apresentada uma pequena revisão sobre o estado da arte na

identificação destes sistemas. Para um estudo mais detalhado e geral recomenda-se a leitura de [28], [7] e [53].

1.3.1 Sistemas dinâmicos chaveados lineares

Os modelos chaveados lineares são definidos como uma coleção de modelos lineares, conectados por chaveamentos que são indexados por uma variável discreta, chamada de estado discreto ou modo. Em outras palavras, a variável discreta denota o modo de operação do sistema e define o modelo matemático que descreve a forma como a porção contínua do vetor de estados evolui, conforme ilustrado na Figura 1.2.

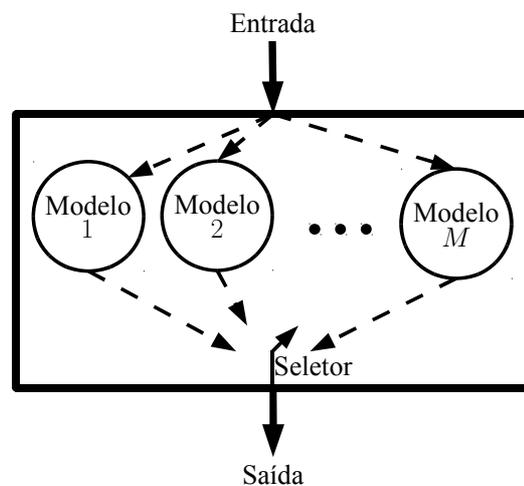


Figura 1.2: Representação de um sistema dinâmico chaveado com M modelos locais (Adaptado de [1]).

A forma de evolução do estado discreto dá origem a várias sub-classes de sistemas híbridos. Por exemplo, em modelos de Saltos Lineares (JL, do inglês *Jump Linear*), o modo é uma entrada desconhecida, determinística e de valor finito. Em modelos de Saltos Markovianos Lineares (JML, do inglês *Jump-Markov Linear*) o estado discreto é modelado como uma cadeia de Markov irredutível governada pela sua matriz de transição de probabilidades. Já nos modelos PWA, a lei de chaveamento entre os diferentes sub-modelos é definida pela forma das regiões do domínio espaço-entrada [7].

Diferentes abordagens já foram desenvolvidas para identificação de sistemas dinâmicos chaveados lineares e podem ser divididas em duas categorias de acordo com a estrutura usada nos modelos [54]: modelos na forma de entrada-saída ou modelos no espaço de estados. A grande maioria dos trabalhos aborda o problema de identificação dos modelos na forma de entrada-saída [7], [38]. A Tabela 1.1 apresenta os trabalhos de identificação mais importantes desenvolvidos considerando essas duas estruturas de modelos.

A seguir será realizado uma breve descrição de alguns dos principais métodos apresentados na Tabela 1.1.

Tabela 1.1: Relação de trabalhos publicados em identificação de sistemas dinâmicos chaveados lineares/af ns.

Forma do Modelo	Trabalhos Publicados
Entrada-saída	[31], [32], [28], [29], [34], [36], [37], [35] [55], [56], [26], [57], [58], [54], [59], [24], [60] [61], [62], [63] [64] [65] [66] [67], [52], [58], [68], [69], [70]
Espaço de estados	[71], [72], [39], [73], [74], [75] [76], [77], [25] [38], [78]

1.3.1.1 Modelos na forma de entrada-saída

A representação de um sistema chaveado na forma entrada-saída é conhecida como SARX, (do inglês, *Switched AutoRegressive Exogenous*) e está apresentada abaixo:

$$y_k = \theta_{m_k}^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k, \quad (1.2)$$

em que $m_k \in \{1, \dots, M\}$ é o estado discreto, M é o número de sub-modelos, θ_{m_k} são as matrizes de parâmetros definida para cada sub-modelo, $y_k \in \mathbb{R}^{n_y}$ é o vetor de saída do sistema, $e_k \in \mathbb{R}^{n_y}$ é um termo de ruído/erro e r_k é o vetor regressão definido por:

$$r_k = \begin{bmatrix} y_{k-1}^T & \cdots & y_{k-n_a}^T & u_k^T & u_{k-1}^T & \cdots & u_{k-n_b}^T \end{bmatrix}^T, \quad (1.3)$$

em que $u_k \in \mathbb{R}^{n_u}$ é o vetor de entrada do sistema no instante de tempo k , n_a e n_b são as ordens dos modelos consideradas fixas.

Uma sub-classe especial destes sistemas é a representação chamada de modelos Auto-regressivos Exógenos Af m por Partes (PWARX, do inglês *Piecewise Autoregressive Exogenous*). Nesta classe, o mecanismo de chaveamento é determinado por partições poliédricas do domínio dos regressores $\mathcal{R} \subseteq \mathbb{R}^d$, em que $d = n_y \cdot n_a + n_u \cdot (n_b + 1)$. Isso significa que o estado discreto é dado por:

$$m_k = i \quad \text{se e somente se} \quad r_k \in \mathcal{R}_i, \quad i = 1, \dots, M, \quad (1.4)$$

em que $\{\mathcal{R}_i\}_{i=1}^M$ é uma partição completa de \mathcal{R} . Essas partições são subconjuntos poliédricos convexos de \mathcal{R} de forma que $\cup_{i=1}^M \mathcal{R}_i = \mathcal{R}$ e $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ para $i \neq j$.

Assim, o modelo PWARX é definido como:

$$y_k = \begin{cases} \theta_1^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k & \text{se } r_k \in \mathcal{R}_1, \\ \vdots & \vdots \\ \theta_M^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k & \text{se } r_k \in \mathcal{R}_M, \end{cases} \quad k = 1, 2, \dots, N. \quad (1.5)$$

Do ponto de vista de identificação, as abordagens para esses dois tipos de sistemas, SARX e PWARX, são iguais, com a única diferença que no caso da primeira representação o passo de identificação das partições do domínio dos regressores não é necessário.

A identificação destes sistemas é um difícil problema, pois envolve a estimação dos parâmetros dos modelos locais e dos coeficientes dos hiperplanos que definem a partição do conjunto de entrada e estados (ou regressores). A principal dificuldade reside no fato que o problema de identificação inclui um problema de classificação em que cada um dos dados amostrados deve ser associado ao modelo local mais adequado.

As diferentes abordagens propostas para este problema podem ser distinguidas em duas categorias de acordo com as considerações feitas sobre as partições; partições fixas a priori ou partições estimadas junto com os sub-modelos [7].

No primeiro caso, partições fixas a priori, a classificação dos dados é muito simples sendo que a estimação dos sub-modelos pode ser realizada por meio de técnicas conhecidas de identificação de sistemas como, por exemplo, o algoritmo de mínimos quadrados. No segundo caso, partições estimadas junto com os modelos locais, a dependência entre classificação dos dados, estimação de parâmetros e estimação das regiões torna o problema de difícil solução. O problema é ainda mais complicado quando o número de sub-modelos também deve ser estimado [34].

Conforme apresentado na Tabela 1.1, diferentes técnicas já foram propostas para o problema de identificação destes modelos. No entanto, as principais técnicas podem ser classificadas em quatro grupos de acordo com a abordagem utilizada [37]: procedimento algébrico, abordagem Bayesiana, abordagem de erro limitado e abordagem por agrupamento ou clustering.

O procedimento algébrico proposto em [26], [65], [55] fornece uma solução de forma fechada para o problema de identificação na ausência de ruído. A ideia da abordagem algébrica é transformar o problema de identificação de vários modelos ARX em um problema de identificação de um único modelo ARX deslocado que simultaneamente engloba todos os modelos locais e não depende da sequência de chaveamento. No entanto, nesta abordagem deve-se conhecer o número de modelos e as respectivas ordens. Esta técnica garante a convergência para o ótimo global, porém, a um alto custo computacional [34].

Na abordagem bayesiana proposta em [29], os parâmetros desconhecidos dos modelos são tratados como variáveis aleatórias e descritos através de suas funções de densidades de probabilidades (FDPs). Assim, o problema de identificação é posto como um problema de se calcular a FDP a posteriori dos parâmetros do modelo dado o conjunto de dados amostrados e algumas informações disponíveis a priori. O problema é então relaxado até ser possível implementá-lo na prática e os parâmetros são facilmente obtidos a partir das FDPs. Para o problema de classificação dos dados é utilizada uma abordagem iterativa sub-ótima posta como um problema de se encontrar a classificação de dados com mais alta probabilidade.

Desta forma, este procedimento pode ser resumido da seguinte forma. Inicialmente, calcula-se o modo mais provável de (y_k, r_k) , usando as FDPs dos parâmetros disponíveis no instante de tempo $k - 1$. Então, os pontos são atribuídos ao modo correspondente e utilizados para atualizar a FDP a posteriori dos parâmetros. Uma desvantagem deste método é a quantidade de informações necessárias a priori que inclui o conhecimento do número de modelos, das respectivas ordens e das FDPs iniciais.

Mais recentemente, em [69] é proposto uma abordagem Bayesiana baseada no método Variacional de Bayes (do inglês, *Variational Bayes*). O método pode ser interpretado como uma generalização Bayesiana do algoritmo EM (do inglês, *Expectation Maximization*) e as estimativas das distribuições dos parâmetros são obtidas de forma determinística, introduzindo um pressuposto de independência entre eles. As principais vantagens da metodologia proposta são a baixa carga computacional, o comportamento determinístico e o fato de não necessitar de informações a priori do sistema. Contudo, não existe garantia de convergência para a distribuição real dos parâmetros.

O procedimento por erro limitado (do inglês *bounded-error procedure*) foi utilizado nos trabalhos [34] e [66]. A principal ideia do método é impor que o erro em (1.2) é limitado por uma dada constante $\delta > 0$ para todas as amostras do conjunto de estimação. Com essa consideração, o conjunto de dados amostrados pode ser classificado pelo particionamento do conjunto de N desigualdades $|y_k - r_k^T \Theta| < \delta$, $k = 1, \dots, N$, em um número mínimo de sub-sistemas viáveis. Trata-se de um método conveniente para lidar com ruído, porém, muito dependente da inicialização dos parâmetros, por exemplo, o número de sub-modelos identificados depende diretamente do limite de erro imposto. Adicionalmente, neste método é necessário conhecer as ordens dos sistemas locais.

A abordagem por agrupamento ou clustering proposta por [32] explora o fato do vetor de regressores estar totalmente disponível e a função de mapeamento que define estes sistemas ser localmente linear, isto significa que os dados em $\{(y_k, r_k)\}_{k=1}^N$ que pertencem ao mesmo modo se encontram próximos uns dos outros. Desta forma, eles podem ser facilmente agrupados. A implementação é composta por quatro passos. Inicialmente, cria-se um conjunto de dados locais \mathcal{C}_k para cada par (y_k, r_k) . Então, é identificado um vetor de parâmetros para cada conjunto de dados locais. Em seguida, esses vetores de parâmetros são agrupados em M clusters utilizando-se o algoritmo K-médias (do inglês *K-means*). O passo seguinte é realizar a classificação dos dados nos clusters identificados. Com os conjuntos de dados obtidos pode-se identificar os parâmetros dos sub-modelos por mínimos quadrados. O passo final é a estimação das partições.

Essa abordagem requer o conhecimento do número de modos, da ordem dos modelos locais e do tamanho dos clusters. No entanto, essa abordagem serviu de estopim motivador para diversos trabalhos mais recentes. Por exemplo, em [35] os autores apresentaram um método simples e mais eficiente baseado em agrupamento por mistura de gaussianas, treinada pelo algoritmo EM. A vantagem da metodologia proposta é que não é necessário o

conhecimento a priori do número de modos. Em [24], a principal diferença é a utilização de um algoritmo de erro de saída afim (do inglês *affine output error*). Já em [9], é proposta a utilização de um novo algoritmo de clustering baseado na metodologia de divisão e fusão (do inglês, *Split and Merge*), enquanto que em [70] é proposta a utilização da técnica Chiu's clustering. Segundo [24] a abordagem por clustering é a que oferece o melhor compromisso entre facilidade de uso e acurácia dos modelos identificados.

Alguns trabalhos mais recentes propõem novas abordagens para solução deste problema de identificação. Em [58] os autores propõem uma abordagem que permite a estimação simultânea dos parâmetros e dos modos de uma sub-classe de sistemas PWA, conhecida como PWARMAX (do inglês, *PieceWise Autoregressive Moving Average eXogeneous*). A proposta utiliza Algoritmos Genéticos para obter os parâmetros de um modelo de misturas de gaussianas e uma variação do algoritmo de Mínimos Quadrados Estendido para estimar os parâmetros dos sub-modelos. Já em [60] os autores abordam o problema para a sub-classe SMARX (do inglês, *Switched Markov ARX System*) e utilizam o Algoritmo EM para estimação dos modos e da dinâmica contínua do sistema.

1.3.1.2 Modelos no espaço de estados

Um modelo chaveado linear geral no espaço de estados é descrito pelo seguinte conjunto de equações:

$$\begin{cases} x_{k+1} &= A_{m_k}x_k + B_{m_k}u_k + w_k, \\ y_k &= C_{m_k}x_k + D_{m_k}u_k + v_k, \end{cases} \quad (1.6)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto, e M é o número de sub-modelos. Os vetores $x_k \in \mathbb{R}^{n_{m_k}}$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos, as entradas e saídas do sistema e n_{m_k} é a ordem do sub-modelo indexado por m_k . As matrizes A_{m_k} , B_{m_k} , C_{m_k} e D_{m_k} tem dimensões apropriadas e descrevem as dinâmicas locais, $w_k \in \mathbb{R}^{n_{m_k}}$, $w_k \sim N(0, Q_k)$ e $v_k \in \mathbb{R}^{n_y}$, com $v_k \sim N(0, R_k)$, são termos de ruído de processo e medida, respectivamente, considerados como sequências de ruído branco descorrelacionados com média zero.

O problema de identificação neste caso consiste em estimar a quantidade e a ordem dos sub-modelos, uma realização $(\hat{A}_{m_k}, \hat{B}_{m_k}, \hat{C}_{m_k}, \hat{D}_{m_k})$ para cada modo e as estimativas dos instantes de chaveamento do sistema. Esse problema é sensivelmente mais difícil quando comparado com a identificação de modelos na forma de entrada-saída por dois motivos. Primeiro porque o estado contínuo é, geralmente, desconhecido. Assim, o vetor de regressão não está totalmente disponível e o particionamento direto do espaço de entrada-estado é uma tarefa muito mais complicada. Segundo, as técnicas geralmente utilizadas para identificação dos modelos locais determinam as matrizes dos diferentes sistemas através de transformações de estados. Essas transformações de estados são, em geral, diferentes para cada modelo local, conforme apresentado em [39]. Com efeito, essas matrizes devem ser transformadas

para a mesma base de estados para que os diferentes sub-modelos possam ser combinados.

A primeira abordagem proposta para identificação de sistemas chaveados lineares no espaço de estados foi proposta em [71]. Neste trabalho, os autores utilizam um método recursivo de identificação em subespaços para a estimação dos instantes de chaveamento do sistema. A técnica de estimação dos mínimos quadrados é aplicada nos conjuntos de dados existentes entre dois instantes de chaveamento para a identificação dos parâmetros de Markov dos modelos locais. Em seguida, realiza-se a procura por modelos semelhantes e realiza-se a fusão destes modelos. Por fim, a identificação final dos sub-modelos é realizada utilizando-se o Algoritmo de Realização de Sistemas (ERA, do inglês *Eigensystem Realization Algorithm*). Os principais problemas deste método são a limitação a sistemas sem ruído de processo e fato de não abordar os problemas de atrasos de detecção.

Em [72], os autores utilizam uma abordagem semelhante, mas que resolve as limitações mencionadas acima. Para isso, é utilizado um método de estimação dos instantes de chaveamento baseado na análise no vetor de resíduos obtidos da projeção das saídas do sistema sobre o espaço ortogonal das entradas. É mostrado que quando um chaveamento ocorre a média deste vetor residual deixa de ser nula.

Em [39] é abordado o problema de identificação assumindo que o estado discreto é conhecido. Assim, o foco é principalmente em determinar a transformação de estados que permite expressar todos os modelos locais na mesma base de estados. Essa transformação é obtida por comparação, em cada transição, do estado do modelo antes da transição com o estado do modelo após a transição. A proposta utiliza várias transições entre os modelos locais para determinar a respectiva transição de estados.

Em [73] é utilizada a abordagem onde os dados de entrada e saída são incorporados em um espaço de dimensão superior. Assim, o problema de estimação dos instantes de chaveamento se torna um problema de segmentação de dados em subespaços distintos. Para realizar essa segmentação é utilizado o algoritmo de análise de componentes principais generalizados (GPCA, do inglês *Generalized Principal Components Analysis*), que permite simultaneamente e automaticamente identificar o número de modelos locais, a ordem desses modelos no espaço de estados, e todos os instantes de chaveamento do sistema a partir dos dados amostrados. No entanto, este trabalho não aborda o problema de estimação das matrizes dos modelos locais.

Uma interessante abordagem iterativa de identificação em subespaços é proposta em [74]. O método integra a estimação dos instantes de chaveamento do sistema com a estimação iterativa dos parâmetros do sistema. Para isso, utiliza-se o algoritmo de identificação dos instantes de chaveamento proposto em [50]. Esse algoritmo é baseado na detecção das variações do posto dos subespaços projetados de pequenos conjuntos de dados de entrada e saída. Esse conjunto de dados é iterativamente atualizado e quando o conjunto contém dados de apenas um modelo local, a dimensão do respectivo subespaço se mantém constante. Já para

a estimação dos parâmetros dos modelos locais é utilizado um algoritmo de identificação em subespaços iterativo. A desvantagem desta abordagem é a necessidade da existência de uma clara distinção entre os valores singulares referentes ao sinal e ao ruído para que seja possível determinar corretamente o posto dos subespaços. Como resultado, o método é somente indicado em situações onde o ruído que afeta o sistema é relativamente baixo.

Para superar essa dificuldade em [75] os autores apresentam uma evolução desta metodologia. Neste novo trabalho, a detecção dos instantes de chaveamento do sistema está relacionada com a dispersão das principais direções dos subespaços. É mostrado que quando o conjunto de dados analisado contém apenas dados gerados pelo mesmo modelo local, o número das principais direções dos subespaços permanece constante. Por outro lado, quando o conjunto de dados contém dados de diferentes modelos locais o número das principais direções aumenta, resultando em uma distorção do subespaços. Em [16], são apresentadas pequenas variações nesta abordagem para melhorar o desempenho do algoritmo.

Em [79] é utilizada uma abordagem que combina métodos em subespaços com uma programação inteira mista para a identificação dos sistemas. Inicialmente os estados do sistema são extraídos a partir dos dados de entrada e saída usando-se métodos em subespaços. Uma vez que os estados são conhecidos, o sistema chaveado é re-escrito como um sistema MLD e os parâmetros são calculados através da programação inteira mista. A desvantagem deste método é que o número de modos e as ordens dos modelos locais devem ser conhecidas a priori.

O trabalho proposto em [38] apresenta uma abordagem que não impõem nenhuma restrição quanto ao tempo de permanência do sistema em cada um dos modos. A metodologia proposta elimina o estado contínuo desconhecido do conjunto de equações do modelo utilizando uma consideração apropriada de observabilidade. Isso permite obter um modelo na forma de entrada-saída que envolve um aumento no número de estados discretos. A estimação desse modelo permite então extrair uma realização do modelo de espaço de estados. A estimação das matrizes do modelos locais é realizada por um algoritmo que alterna entre a classificação dos dados e atualização dos parâmetros via método dos mínimos quadrados recursivos. No entanto, nesta abordagem é necessário o conhecimento a priori do número de modos e da ordem dos modelos locais.

Em [5] os autores apresentam uma interessante abordagem baseada no algoritmo de EM. A proposta aborda o problema de aprendizado para sistemas híbridos onde as transições no estado discreto dependem do estado contínuo e empregam uma aproximação da distribuição a posteriori sobre os estados ocultos e técnicas de otimização locais para atualizar as estimativas dos parâmetros do sistema híbridos. Portanto, a abordagem utiliza uma versão aproximada do algoritmo EM para tornar o problema tratável na prática. Os autores mostram empiricamente que o algoritmo converge na prática. As principais limitações deste método é o fato de precisar de estimativas iniciais das matrizes dos modelos locais (conhecimento prévio dos modelos locais), a forte dependência de boas estimativas iniciais e a possibilidade

de convergência para mínimos locais.

Em [4] os autores estendem o trabalho anterior de modo a evitar a convergência das soluções para mínimos locais. A abordagem proposta identifica as áreas exploradas do espaço de soluções e guia a busca para fora destas áreas. Para isso o método combina o uso do algoritmo EM com o algoritmo de agrupamento K-means. Essa abordagem permite obter a solução ótima ou quase ótima do processo de aprendizagem. No entanto, este método continua dependendo de um conhecimento inicial dos modelos locais do sistema e apresenta uma alta carga computacional na solução do problema.

O trabalho proposto em [78] formula o problema de identificação como um problema de minimizar o número de componentes diferentes de zero em um vetor de erros normalizado. No entanto, como a solução direta deste problema é inviável computacionalmente, os autores relaxam essa primeira formulação e mostram que os diferentes sub-modelos podem ser recuperados através de um problema de otimização convexa. A principal desvantagem desta metodologia é a consideração que o estado contínuo do sistema é completamente mensurável.

Alguns autores [76], [77], [25] abordam o problema de identificação de sistemas chaveados utilizando uma abordagem online. Neste contexto, como o processo de identificação deve ser realizado em tempo real, o problema de identificação se torna muito mais complexo, uma vez que os algoritmos utilizados devem ser capazes de fornecer estimativas dentro de um tempo limitado. Assim, em geral, essas abordagens não podem utilizar algumas manipulações úteis como decomposição em valores singulares ou processamento iterativo devido ao tempo de processamento que essas manipulações requerem. Consequentemente, as abordagens obtidas são relativamente limitadas [51]. Nos trabalhos mencionados, basicamente, é utilizado um algoritmo de identificação em subespaços recursivo e um algoritmo de classificação dinâmica para identificar as mudanças de modo.

Apesar das diversas abordagens desenvolvidas ainda não existe um método em destaque para a identificação de sistemas chaveados no espaço de estados. Cada abordagem apresenta algum tipo de consideração e/ou limitação que dificulta a realização de comparações. De fato, nenhuma das abordagens apresentadas foi comparada com outras técnicas, o que torna ainda mais difícil a definição da melhor técnica ou da técnica referência para os demais trabalhos.

Para exemplificar, apresenta-se na Tabela 1.2 uma comparação com o resumo das considerações de cada uma das abordagens off-line apresentadas nesta seção. Essa tabela inclui ainda as considerações que serão realizadas nos algoritmos propostos neste trabalho (Capítulo 3), com a finalidade de facilitar a visualização do contexto na qual se insere esses algoritmos.

Tabela 1.2: Considerações realizadas em cada abordagem off-line desenvolvida para identificação de sistemas chaveados no espaço de estados.

Trabalhos	Modelos Estáveis e Observáveis	Modos persist. excitados	Um modo ativo a cada instante	Quant. de modos conhecido	Ordens conhecidas	Estados Contínuos Mensuráveis	Estimativa inicial dos modelos	τ_{dwell}^1	Instantes de chav. conhecidos	Validação experimental	Comparação com outros métodos
[39]	X	X	X	X				X	X		
[74]	X	X	X					X			
[75]	X	X	X					X		X	
[50]	X	X	X					X			
[73]	X	X	X					X			
[79]	X	X	X	X	X			X			
[71]	X	X	X					X			
[72]	X	X	X					X			
[38]	X	X	X	X	X						
[5]	X		X	X	X		X				
[4]	X		X	X	X		X				
[78]	X	X	X			X					
Abordagens propostas neste trabalho	X	X	X					X			X

¹Mínimo período de tempo que o sistema permanece em cada modo. Esse parâmetro será melhor definido no Capítulo 3.

1.4 PROPOSTAS E CONTRIBUIÇÕES

Do ponto de vista teórico, é proposta uma nova estrutura (*framework*) para identificação de sistemas dinâmicos híbridos no espaço de estados, mais especificamente é abordado o caso particular de sistemas chaveados lineares a tempo discreto. Essa nova estrutura de identificação resultou no desenvolvimento de seis novos algoritmos. As abordagens desenvolvidas exploram o uso de métodos de identificação de sistemas dinâmicos lineares, clustering, classificação de dados e filtragem. Estas abordagens foram desenvolvidas utilizando-se um conceito modular onde é possível utilizar uma grande variedade de algoritmos para implementar cada um dos métodos mencionados. Essa é uma importante característica, uma vez que é possível escolher os algoritmos em função da aplicação de modo a obter melhores resultados.

Diferentemente dos principais métodos de identificação de sistemas híbridos existentes na literatura onde primeiro é identificado os instantes de chaveamento do sistema, as abordagens propostas neste trabalho, inicialmente identificam os modelos locais através da utilização de técnicas clássicas de identificação em subespaços aplicadas a pequenos conjunto de dados. Os instantes de chaveamento são estimados em etapa posterior com a aplicação de um algoritmo de clustering robusto. Durante o desenvolvimento dos seis algoritmos propostos, a principal preocupação era torná-los capazes de fornecer estimativas de qualidade mesmo nos cenários onde o sistema está sujeito a ação de elevados níveis de ruído e pouco conhecimento do sistema está disponível a priori.

Contribuições adicionais deste trabalho refere-se à utilização, no contexto de identificação de sistemas, do algoritmo de clustering RCA e de um algoritmo desenvolvido para estimação de estados de sistemas híbridos (filtragem híbrida) conhecido na literatura como *Interacting Multiple Model* (IMM) [80], [81]. Acrescenta-se ainda como contribuição deste trabalho a comparação da metodologia desenvolvida com uma abordagem existente na literatura, uma vez que a revisão bibliográfica mostrou a ausência de comparações entre as diferentes abordagens já desenvolvidas.

Os artigos científicos publicados durante o desenvolvimento deste trabalho de doutorado estão listados no Anexo A.

1.5 APRESENTAÇÃO DO MANUSCRITO

O primeiro capítulo apresenta a introdução, a motivação do estudo e a revisão bibliográfica das principais fontes para a elaboração desta tese e constitui o estado da arte na área de identificação de sistemas híbridos. O segundo capítulo apresenta os conceitos teóricos e ferramentas utilizadas durante o desenvolvimento dos novos algoritmos de identificação propostos. O terceiro capítulo contém as contribuições deste trabalho, onde são apresenta-

dos seis novos algoritmos para a identificação dos sistemas híbridos. Este capítulo apresenta ainda o desempenho desses novos algoritmos por meio de simulações numéricas. No capítulo quatro é realizada a comparação da abordagem mais completa desenvolvida com uma abordagem existente na literatura. Por fim, as conclusões e as propostas para trabalhos futuros são apresentadas no quinto capítulo.

2 FUNDAMENTOS TEÓRICOS

Este trabalho abordará o problema de identificação de sistemas chaveados lineares no espaço de estados, definidos na Seção 1.3. Conforme mencionado anteriormente, trata-se de um difícil problema que envolve a estimação dos parâmetros do modelo contínuo, do estado discreto e das funções de chaveamento a partir do conjunto de dados de entrada e saída amostrados.

Neste contexto, abordagens eficientes de identificação consistem na separação dos dados gerados por diferentes sistemas dinâmicos e na aplicação de técnicas de identificação de sistemas lineares nos diferentes conjuntos obtidos. As metodologias de identificação que serão apresentadas no Capítulo 3 abordam esse problema dividindo-o em etapas que envolvem conceitos de identificação de sistemas, clustering, classificação de dados e filtragem híbrida. Na Figura 2 é apresentado um fluxograma geral ilustrando a interação entre esses conceitos. As estruturas específicas de cada algoritmo serão apresentadas no Capítulo 3.

Os algoritmos desenvolvidos apresentam um conceito modular e permitem que diferentes métodos (ou algoritmos) sejam utilizados na implementação de cada um dos conceitos mencionados. Por exemplo, na etapa de identificação dos sistemas, pode-se utilizar qualquer método de identificação de sistemas lineares que forneça uma representação do sistema no espaço de estados, ou na etapa de clustering pode-se utilizar qualquer algoritmo de clustering existente na literatura.

Desta forma, este capítulo tem a finalidade de apresentar uma breve introdução aos conceitos teóricos e algoritmos utilizados nas diferentes etapas das metodologias que serão apresentadas no próximo capítulo. Detalhes matemáticos e demonstrações formais desnecessárias para o entendimento dos conceitos e métodos não serão apresentados. O leitor interessado em um estudo mais profundo deve consultar os trabalhos originais citados ao longo do texto.

2.1 IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS LINEARES

A identificação de sistemas dinâmicos aborda o problema de construção de modelos matemáticos de maneira empírica, baseada em experimentos e coletas de dados. A teoria de identificação de sistemas dinâmicos começou a se fortalecer no campo da engenharia a partir da década de 60. Desde então, diversas abordagens vem sendo desenvolvidas e apresentam como característica o fato de requererem pouco ou nenhum conhecimento prévio do sistema [2].

Dentre essas diversas abordagens, as de interesse neste trabalho são aquelas que forneçam

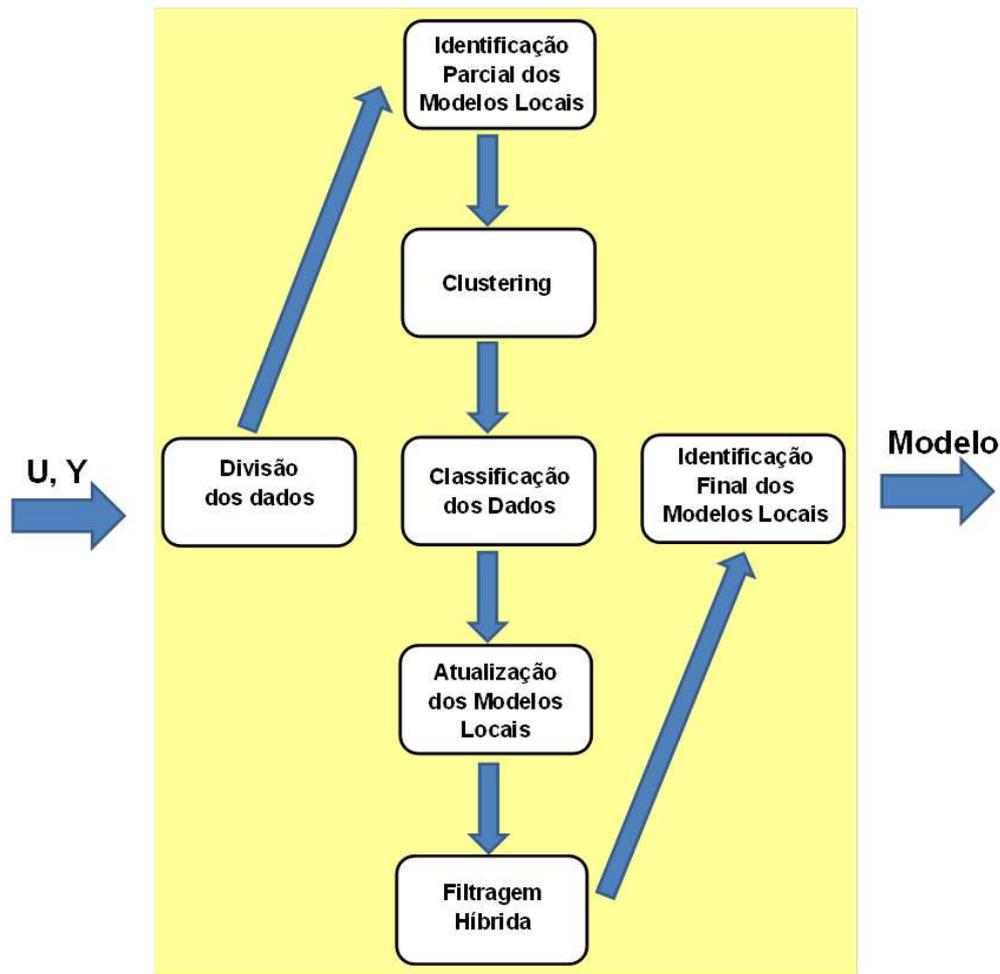


Figura 2.1: Diagrama ilustrando a interação entre os diferentes conceitos utilizados nos algoritmos que serão apresentados no Capítulo 3.

uma representação do sistema dinâmico na forma de espaço de estados. Segundo [82], essas abordagens podem ser divididas em três diferentes categorias.

1. Métodos clássicos de identificação paramétrica: Consistem na estimação de parâmetros através da adaptação de métodos clássicos (Mínimos Quadrados, Método de Erro de Predição, Variáveis Instrumentais) para modelos no espaço de estados. Nestes métodos os parâmetros são estimados pela resolução de um problema de otimização que consiste na minimização de uma função custo que, geralmente, é formulada em termos da diferença entre a saída medida e a saída estimada. Para maiores detalhes destes métodos, consultar os trabalhos [83] e [2].
2. Métodos da teoria das realizações: Nestes métodos as matrizes do modelo no espaço de estados são obtidas a partir dos coeficientes da resposta ao impulso. Os principais algoritmos desta categoria foram propostos nos trabalhos [84] e [85].
3. Métodos de identificação em subespaços: Esses métodos têm suas origens na teoria das

realizações descrita na categoria anterior e são caracterizados por obter um modelo no espaço de estados através dos subespaços gerados pelas linhas ou colunas de matrizes formadas pelos dados de entrada - saída.

Na comparação destes métodos, as abordagens da segunda categoria apresentam uma séria desvantagem que é a dificuldade de se obter uma estimativa não paramétrica confiável da resposta impulsiva [82]. Em relação as outras duas categorias, observa-se que, em geral, os métodos clássicos produzem estimativas mais precisas quando comparados com os métodos em subespaços [86]. Apesar disso, nos últimos 15 anos, a teoria de identificação em subespaços tem atraído a atenção da comunidade científica internacional devido as várias vantagens apresentadas em relação às técnicas clássicas. Algumas destas vantagens são:

- Os algoritmos de identificação em subespaços são mais rápidos do que os algoritmos clássicos, pois não são iterativos [87];
- As técnicas de identificação em subespaços não possuem problemas de otimização não convexas como as técnicas tradicionais, como por exemplo, os modelos de erros de predição [39];
- Não possuem problemas de convergência [87];
- São numericamente robustos, pois utilizam algoritmos bem estabelecidos da álgebra linear numérica [87];
- Na identificação em subespaços, os modelos reduzidos podem ser obtidos diretamente, sem a necessidade de calcular primeiro modelos de ordem superior [87].
- Não apresentam a premissa de que todos os estados precisam ser diretamente mensuráveis.

Motivado por essas vantagens, neste trabalho optou-se pela utilização de um método de identificação em subespaços para a identificação dos modelos locais constituintes do sistema híbrido. Assim, na sequência será realizada uma discussão um pouco mais detalhada sobre esses métodos e em seguida apresentado o algoritmo utilizado neste trabalho.

Matematicamente, a identificação em subespaços considera modelos descritos pelo seguinte conjunto de equações:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k, \\ y_k = Cx_k + Du_k + v_k, \end{cases} \quad (2.1)$$

com

$$E \left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix} \right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \geq 0, \quad (2.2)$$

em que $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são as medidas de entrada e saída, respectivamente, no instante de tempo k . O vetor $x_k \in \mathbb{R}^n$ é o vetor de estados do processo. As matrizes A , B , C e D possuem dimensões apropriadas e descrevem a dinâmica do sistema de ordem n . Os vetores $w_k \in \mathbb{R}^n$ e $v_k \in \mathbb{R}^{n_y}$ são termos de ruídos branco, assumidos serem de média zero, e estacionários.

Basicamente, a partir de um conjunto de dados observados de entrada e saída, os métodos de identificação em subespaços determinam qual a realização apresenta a maior probabilidade de gerar as saídas (ou a realização que tem o menor erro quadrático na saída) dadas as entradas do sistema [88]. Para isso, inicialmente estima-se o vetor de estados x_k ao longo de uma janela de observação e então determina-se as matrizes do sistemas A , B , C , D por um procedimento de regressão via mínimos quadrados. Essa é uma importante diferença entre os métodos de identificação em subespaços e os métodos clássicos como, por exemplo, o método de predição do erro (PEM, do inglês *Prediction Error Method*).

A análise das estatísticas do resíduo obtido na regressão pode ser empregada para estimar as matrizes de covariância do sistema Q , S e R . A estimação do vetor de estados é obtida por meio de uma combinação linear dos preditores da saída y_k . Esses preditores são obtidos através de operações de projeções envolvendo as matrizes de dados de entrada e saída disponíveis [87], [83].

O resultado da projeção é dividido, mediante decomposição em valores singulares, em uma matriz de observabilidade e uma matriz que contém a evolução dos estados estimados. Essa é a essência do porquê esses algoritmos são chamados de subespaços. Em outras palavras, eles recuperam as matrizes relacionadas ao sistema como subespaços da projeção das matrizes de dados [87]. O método permite ainda determinar a ordem do sistema através da inspeção dos valores singulares diferentes de zero.

Existem diferentes abordagens de identificação em subespaços na literatura. No entanto, essas abordagens variam apenas na forma de geração e análise das possíveis realizações das matrizes do sistema. Neste trabalho, optou-se por uma abordagem unificada conhecida como Algoritmo Combinado Robusto - ACR (do inglês, *Robust Combined Algorithm*) apresentada em [87]. Segundo os autores, o termo robusto é usado no sentido que o algoritmo apresenta bons resultados para diversas aplicações práticas. Este último aspecto, alinhado com o fato do algoritmo poder ser implementado de uma maneira numericamente estável e eficiente foram os principais motivos para sua escolha neste trabalho.

A seguir será realizada uma breve descrição desta metodologia. A ideia é mostrar intuitivamente o procedimento realizado pelo algoritmo. O leitor interessado em um estudo mais profundo deve consultar o trabalho original [87].

2.1.1 Algoritmo Combinado Robusto - ACR

O modelo (2.1) pode ser reformulado e colocado em forma de equações matriciais como:

$$Y_p = \Gamma_i X_p^d + H_i^d U_p + Y_p^{st}, \quad (2.3)$$

$$Y_f = \Gamma_i X_f^d + H_i^d U_f + Y_f^{st}, \quad (2.4)$$

$$X_f^d = A^i X_p^d + \Delta_i^d U_p, \quad (2.5)$$

as notações (\bullet^d) e (\bullet^{st}) referem-se as partes determinísticas e estocásticas, respectivamente, o sub-escrito f denota "futuro" e p denota o "passado". O parâmetro i é um índice definido pelo usuário e conhecido como o número de blocos linhas. Esse número deve ser maior do que a máxima ordem do sistema a ser identificado. Os outros termos nas equações são definidos como segue:

- Γ_i é uma matriz de observabilidade estendida de ordem i definida como

$$\Gamma_i = \begin{bmatrix} C & CA & \dots & CA^{i-1} \end{bmatrix}^T \in \mathbb{R}^{in_y \times n}. \quad (2.6)$$

- Δ_i^d é uma matriz de controlabilidade estendida de ordem i .

$$\Delta_i^d = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix} \in \mathbb{R}^{n \times pi}. \quad (2.7)$$

- H_i^d é uma matriz triangular inferior de Toeplitz

$$H_i^d = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \dots & \dots & 0 \\ CA^{(i-2)}B & CA^{(i-3)}B & \dots & CB & D \end{bmatrix} \in \mathbb{R}^{in_y \times in_u}. \quad (2.8)$$

- A matriz de *block Hankel* de entrada é definida como:

$$\begin{aligned} U_{0|2i-1} &= \begin{bmatrix} u_0 & u_1 & u_2 & \dots & u_{t-1} \\ u_1 & u_2 & u_3 & \dots & u_t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \dots & u_{i+t-2} \\ u_i & u_{i+1} & u_{i+2} & \dots & u_{i+t-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \dots & u_{i+t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \dots & u_{2i+t-2} \end{bmatrix}, \\ &= \begin{bmatrix} U_{0|i-1} \\ U_{i|2i-1} \end{bmatrix} = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \end{aligned} \quad (2.9)$$

em que o número de colunas t é tipicamente igual a $s - 2i + 1$, sendo s a quantidade de dados utilizada para a identificação.

A partir desta matriz pode-se definir outras duas importantes matrizes, U_p^+ e U_f^- , que são construídas da seguinte forma:

$$U_p^+ = U_{0|i} = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{t-1} \\ u_1 & u_2 & u_3 & \cdots & u_t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+t-2} \\ u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+t-1} \end{bmatrix}. \quad (2.10)$$

$$U_f^- = U_{i+1|2i-1} = \begin{bmatrix} u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+t-2} \end{bmatrix}. \quad (2.11)$$

As matrizes Y_p , Y_p^+ , Y_f e Y_f^- são definidas de maneira similar para as saídas y_k . A matriz contendo as entradas U_p e saídas Y_p será denominada W_p e $W_p \doteq \begin{bmatrix} U_p & Y_p \end{bmatrix}^T$.

Finalmente a sequência de estados X_i é definida como:

$$X_i = \begin{bmatrix} x_i & x_{i+1} & \cdots & x_{i+t-1}^d \end{bmatrix} \in \mathbb{R}^{n \times t}. \quad (2.12)$$

Definição 2.1.1. A projeção ortogonal do espaço linha de A sobre o espaço linha de B é definida como [87]:

$$A/B = AB^T(BB^T)^\dagger B, \quad (2.13)$$

em que $(\bullet)^\dagger$ denota a matriz pseudo-inversa de Moore-Penrose da matriz (\bullet) . \square

Definição 2.1.2. A projeção oblíqua do espaço linha de A na direção do espaço linha de B sobre o espaço linha de C é denotada por $A/_B C$ e definida como:

$$A/_B C = [A/B^\perp][C/B^\perp]^\dagger C, \quad (2.14)$$

em que B^\perp representa o complemento ortogonal de B . \square

Assim, sobre as hipóteses que:

H1. A entrada do sistema u_k é persistentemente excitante de ordem $2i$;

H2. Os ruídos de processo e medida são descorrelacionados;

pode-se obter as seguintes relações [87]:

$$Z_i = Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix} = \Gamma_i \hat{X}_i + H_i U_f. \quad (2.15)$$

$$Z_{i+1} = Y_f^- / \begin{pmatrix} W_p^+ \\ U_f^- \end{pmatrix} = \Gamma_{i-1} \hat{X}_{i+1} + H_{i-1} U_f^-. \quad (2.16)$$

$$\mathcal{O}_i = Y_f / U_f W_p = \Gamma_i \tilde{X}_f. \quad (2.17)$$

em que \hat{X}_i e \tilde{X}_f são sequencias de estados diferentes, no sentido de que cada uma delas utiliza um estado inicial diferente.

Observe que, a equação (2.15) mostra a relação da sequencia de estados \hat{X}_i com o conjunto de dados amostrados. A matriz Z_i é computada a partir dos dados amostrados, sem o conhecimento prévio das matrizes do sistema. Essa projeção pode ser interpretada como a predição ótima dos dados de saída futuro Y_f , a partir dos dados de entrada e saída passados W_p e as entradas futuras U_f . Já a equação (2.17) permite o cálculo do espaço linha da sequência de estados \tilde{X}_f e o do espaço coluna da matriz de observabilidade estendida \mathcal{O}_i diretamente dos dados amostrados, sem o conhecimento das matrizes do sistema.

Desta forma, o algoritmo de identificação em subespaços segue os seguintes passos:

1. Calcular as seguintes projeções ortogonais e oblíquas:

$$Z_i = Y_f / \begin{pmatrix} W_p \\ U_f \end{pmatrix}, \quad Z_{i+1} = Y_f^- / \begin{pmatrix} W_p^+ \\ U_f^- \end{pmatrix}, \quad \mathcal{O}_i = Y_f / U_f W_p.$$

2. Determinar a SVD da projeção oblíqua ponderada:

$$W_1 \mathcal{O}_i W_2 = U S V^T, \quad (2.18)$$

$W_1 \in \mathbb{R}^{in_y \times in_y}$ e $W_2 \in \mathbb{R}^{t \times t}$ são matrizes de pesos, definidas de forma que W_1 é de posto completo e W_2 obedece a relação $\text{posto}(W_p) = \text{posto}(W_p W_2)$.

3. Determinar a ordem através da inspeção dos valores singulares em S e particionar adequadamente a SVD para obter as matrizes U_1 e S_1 . Isso é obtido da seguinte forma:

$$W_1 \mathcal{O}_i W_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \quad (2.19)$$

É importante destacar que, a ordem n do sistema é igual ao número de valores singulares de S em (2.18) diferentes de zero. Contudo, esta não é uma tarefa trivial e será discutida mais detalhadamente na seção 2.1.3.

4. Determinar Γ_i e Γ_{i-1} da seguinte forma:

$$\Gamma_i = U_1 S_1^{1/2} T, \quad \Gamma_{i-1} = \underline{\Gamma}_i, \quad (2.20)$$

em que $\underline{\Gamma}_i$ é a matriz Γ_i sem a última linha e $T \in \mathbb{R}^{n \times n}$ é uma transformação não-singular arbitrária.

5. Obter as matrizes A e C através da resolução do seguinte conjunto de equações:

$$\begin{bmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \Gamma_i^\dagger Z_i + \mathcal{K}U_f + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}, \quad (2.21)$$

em que ρ_w e ρ_v são termos de resíduos e \mathcal{K} pode ser definida como:

$$\mathcal{K} = \begin{bmatrix} \left[\begin{array}{c|c} B & \Gamma_{i-1}^\dagger H_{i-1}^d \\ \hline D & 0 \end{array} \right] - A\Gamma_i^\dagger H_i^d \\ - C\Gamma_i^\dagger H_i^d \end{bmatrix}. \quad (2.22)$$

Recalcular Γ_i e Γ_{i-1} a partir de A e C .

6. Obtenha as matrizes B e D a partir da equação:

$$B, D = \arg \min_{B, D} \left\| \begin{bmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} A \\ C \end{bmatrix} \Gamma_i^\dagger Z_i + \mathcal{K}U_f \right\|_F^2, \quad (2.23)$$

em que $\|\bullet\|_F$ denota a norma de Frobenius.

7. Estimar Q , S e R . Segundo [87] essas matrizes satisfazem:

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = E_t \left[\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \begin{pmatrix} \rho_w^T & \rho_v^T \end{pmatrix} \right], \quad (2.24)$$

em que

$$E_t[\bullet] \doteq \lim_{t \rightarrow \infty} \frac{1}{t} [\bullet], \quad (2.25)$$

ou seja, as matrizes Q , S e R são estimadas a partir da média temporal dos resíduos da regressão. Outras formas para determinação dessas matrizes podem ser utilizadas resultando em diferentes propriedades estatísticas [87].

Em [87], os autores mostram que diferentes algoritmos existentes na literatura podem ser obtidos através da escolha adequada das matrizes de pesos W_1 e W_2 . Neste trabalho, adotou-se $W_1 = I_{in_y}$ e $W_2 = \Pi_{U_f^\perp} = I_t - U^T(UU^T)^\dagger U$, o que torna a abordagem equivalente ao algoritmo MOESP (do inglês, *Multivariable Output-Error State Space*). Para maiores detalhes a respeito dos métodos de identificação em subespaços recomenda-se a leitura de três livros publicados sobre esse assunto: [89], [90], [87].

2.1.2 Matrizes do Espaço de Estados na Forma Canônica

A metodologia descrita na seção anterior permite obter uma estimativa das matrizes (A , B , C , D) do sistema. No entanto, deve-se notar que não existe apenas um conjunto de matrizes que são compatíveis com os dados amostrados $\{(u_k, y_k)\}_{k=1}^N$. De fato, para cada matriz não-singular T , chamada de matriz de transformação de estados, o conjunto $(TAT^{-1}$, TB , CT^{-1} , D) também descreve o comportamento do sistema.

No contexto de identificação de sistemas dinâmicos híbridos isso pode se tornar um problema, uma vez que as matrizes dos diferentes modelos locais podem ser obtidas em bases de estados diferentes. Como resultado, os diferentes modelos locais não podem ser diretamente combinados. Assim, é importante garantir que essa transformação de estados seja a mesma para todos os modelos locais [39].

Neste trabalho, utilizou-se a abordagem proposta em [91] e [25] para manter o controle da base do estado nas quais as matrizes do sistema são calculadas. Esta abordagem utiliza uma transformação especial que permite a manipulação do sistema MIMO em uma base canônica adequada. Formas canônicas são úteis para descrever sistemas dinâmicos lineares, principalmente porque proporcionam uma parametrização única do sistema em estudo. Essa propriedade pode ser explorada no contexto de identificação de sistemas pois, em geral, dão origem a estruturas de modelos identificáveis. Adicionalmente, os parâmetros das formas canônicas podem ter uma interpretação direta com os coeficientes de uma função de transferência particular [91].

A ideia central da metodologia é englobar a observabilidade dos estados, inicialmente distribuída sobre todas as saídas do sistema, em uma saída auxiliar definida como uma combinação de todas as saídas. Desta forma, é possível definir uma matriz de permutação que determina a base na qual as matrizes serão calculadas.

Para entender esse procedimento considere inicialmente um sistema MISO (do inglês, *Multiple Inputs Single Output*). Para esse sistema a matriz de observabilidade $\Gamma_n = \Gamma_n(A, C)$, com $C \in \mathbb{R}^{1 \times n}$, é uma matriz quadrada e não singular. Utilizando essa matriz, uma transformação de similaridade pode ser realizada através de $x_k \leftarrow \bar{x} = \Gamma_n x_k$. Pode-se mostrar facilmente que a matriz da dinâmica do sistema $A = \Gamma_n A \Gamma_n^{-1}$ é uma matriz na forma companheira (do inglês, *companion form*) após uma operação de transposição e que A e C tem as seguintes formas:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}, \quad (2.26)$$

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad (2.27)$$

em que a_i são os coeficientes do polinômio característico (p_A) de A ,

$$p_A(z) = \det(zI - A) = a_0 + a_1 z + \dots + a_{n-1} z^{n-1} + z^n.$$

Contudo, para os casos onde os sistemas são MIMO, a forma companheira não é tão direta de se obter. No entanto, em [91] e [25] os autores descrevem um conjunto de transformações que torna possível descrever um sistema MIMO na forma canônica companheira.

Novamente detalhes matemáticos e demonstrações formais desnecessárias para o entendimento do método não serão apresentados.

A ideia básica da abordagem é realizar uma reformulação do sistema de tal modo que seja possível representá-lo através de uma única saída y_{j_k} . Para isso, considere uma saída auxiliar construída como uma combinação linear das saídas do sistema, definida como:

$$y_{a_k} = \sum_{j=1}^{n_y} \lambda_j y_{j_k}, \quad (2.28)$$

em que y_{j_k} são componentes do vetor de saída y_k e λ_j são números reais diferentes de zero.

Se a matriz A for não-depreciativa (do inglês, *nonderogatory*), os λ_j 's podem ser selecionados de tal modo que todas as dinâmicas do sistema sejam observadas somente de y_{a_k} . Segundo [91], uma matriz A é não-depreciativa se e somente se uma das seguintes condições são satisfeitas:

- A é similar a uma matriz companheira;
- O polinômio característico de A é igual ao polinômio mínimo, onde os dois polinômios são considerados como sendo unitário;
- Existe um vetor $v \in \mathbb{R}^n$ tal que a matriz $\begin{bmatrix} v & Av & \cdots & A^{n-1}v \end{bmatrix}$ tem posto completo;
- Para qualquer autovalor φ de A , $\dim\{\ker\{A - \varphi I_n\}\} = 1$;

Assim, substituindo um dos componentes do vetor de saída, por exemplo o primeiro componente, por y_{a_k} , obtém-se:

$$\bar{y}_k = Ky_k, \quad (2.29)$$

com

$$K = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{n_y} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \quad (2.30)$$

Denotando as novas matrizes como $\bar{C} = KC$, $\bar{D} = KD$, $\bar{\Gamma}_i = \Gamma_i(A, \bar{C})$ e $\bar{H}_i^d = H_i^d(A, B, \bar{C}, \bar{D})$, a equação (2.5) pode ser re-escrita como:

$$\bar{Y}_f = \bar{\Gamma}_i X_f^d + \bar{H}_i^d U_f + \bar{Y}_f^{st}, \quad (2.31)$$

em que \bar{Y}_f e \bar{Y}_f^{st} são as matrizes de *block Hankel* construídas de \bar{y}_k .

A ideia dessa transformação é obter uma saída $y_{a_k} \in \mathbb{R}$ a partir da qual todos os pólos do sistema (2.1) sejam observáveis. Isso é equivalente a $\Gamma_n(A, c_1^T)$ ter posto completo, com $c_1^T = \lambda^T C$, $\lambda = \begin{bmatrix} \lambda_1 & \cdots & \lambda_{n_y} \end{bmatrix}^T$. Se isso for verdade, é possível obter diretamente,

através da transformação $\bar{x}_k \leftarrow \Gamma_n(A, c_j^T)x_k$, as matrizes do sistema (2.1) na forma canônica de forma similar ao caso MISO. Desta forma, o problema se torna em como determinar os valores de $\lambda_1, \dots, \lambda_{n_y}$ que tornam essa exigência verdadeira.

Em [91] e [25] os autores demonstram que, se $\lambda = \begin{bmatrix} \lambda_1 & \dots & \lambda_{n_y} \end{bmatrix}^T \in \mathbb{R}^{n_y}$ for um vetor gerado aleatoriamente e se o par (A, C) do sistema (2.1) é observável e a matriz A é não-depreciativa, é seguro com probabilidade um que:

$$\text{posto}(\Gamma_n(A, \lambda^T C)) = n. \quad (2.32)$$

Desta forma, é possível obter uma certa representação no espaço de estado na forma canônica companheira de um sistema MIMO diretamente no processo de identificação. Para isto, basta que, em cada caso, seja utilizada uma matriz não singular T para alterar a base do espaço de estados na equação (2.5). Para essa finalidade, defini-se uma matriz de permutação L como:

$$L = \begin{bmatrix} d_1 & \dots & d_{n_y} \end{bmatrix} \in \mathbb{R}^{i n_y \times i n_y}, \quad (2.33)$$

em que $d_z = \begin{bmatrix} e_z & e_{z+n_y} & \dots & e_{z+(i-1)n_y} \end{bmatrix}$ com $e_z \in \mathbb{R}^{i n_y}$ um vetor composto de zeros e 1 na posição z .

Assim, considerando $T = \Gamma_n(A, c_1^T)$ e usando-se a matriz de (2.33), pode-se obter:

$$L\bar{\Gamma}_i = \begin{bmatrix} \bar{c}_1^T \\ \bar{c}_1^T A \\ \vdots \\ \bar{c}_1^T A^{i-1} \\ \bar{c}_2^T \\ \vdots \\ \bar{c}_{n_y}^T A^{i-1} \end{bmatrix} = \begin{bmatrix} \Gamma_n(A, \bar{c}_1^T) \\ \Psi \end{bmatrix}. \quad (2.34)$$

Por construção, o $\text{posto}(\Gamma_n(A, \bar{c}_1^T)) = n$. Consequentemente, $\Gamma_n(A, \bar{c}_1^T)$ pode ser utilizada como uma matriz de transformação T e a escolha da matriz L , definida em (2.33), determina uma certa base de estados na qual as matrizes do sistema são recuperadas. Existe diferentes maneiras de ajustar a matriz de permutação L . O único requisito é que as primeiras n linhas de $L\bar{\Gamma}_i$ sejam linearmente independente. Após essas transformações as matrizes

do sistema são obtidas na seguinte forma:

$$\mathcal{A} = TAT^{-1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} \end{bmatrix}, \quad (2.35a)$$

$$\mathcal{B} = TB = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \\ b_{n1} & \cdots & b_{n(p-1)} & b_{np} \end{bmatrix}, \quad (2.35b)$$

$$\mathcal{C} = KCT^{-1} = \begin{bmatrix} c_1^T \\ \vdots \\ c_l^T \end{bmatrix} \text{ com } c_1^T = e_1^T, \quad (2.35c)$$

$$\mathcal{D} = KD. \quad (2.35d)$$

Segundo [91] a parametrização introduzida acima tem as seguintes características:

- A parametrização é unicamente caracterizada pelos n_y valores de λ_i gerados pelo usuário;
- A parametrização obtida tem o menor número de parâmetros ($n(n_u + n_y) + n_u n_y$);
- A estrutura (2.35) é globalmente identifável. Segundo [83], a estrutura de um modelo é globalmente identifável no vetor de parâmetros θ se $\mathcal{M}(\theta) = \mathcal{M}(\theta^*) \implies \theta = \theta^*$

Portanto, o algoritmo de identificação utilizado neste trabalho é formado pela combinação da metodologia de identificação em subespaços descrita na seção anterior com a abordagem descrita nesta seção para manter o controle da base de estados nas quais as matrizes do sistema são calculadas.

Exemplo 2.1: Para ilustrar a metodologia de identificação, será considerado a identificação de um circuito RLC série discutido em [86]. Este circuito consiste de uma fonte de tensão ligada em série com um resistor (R), um capacitor (C) e um indutor (L). O sinal de entrada u é o sinal proveniente da fonte de tensão e a tensão sobre o indutor é escolhida como o sinal de saída y . Este sistema pode ser representado no espaço de estados com as seguintes matrizes:

$$A = \begin{bmatrix} 0,9877 & 4,8586 \\ -0,0049 & 0,9392 \end{bmatrix}, B = \begin{bmatrix} 0,0123 \\ 0,0049 \end{bmatrix}, C = \begin{bmatrix} -1 & -10 \end{bmatrix}, D = 1,$$

essas matrizes foram obtidas considerando a discretização do sistema com período de amostragem de 0,01s e escolhendo-se $R = 10\Omega$, $C = 0,02F$ e $L = 20H$.

Para identificação do sistema, utilizou-se como sinal de excitação do sistema um ruído branco de média nula e desvio padrão $\sigma = 20$. O sistema foi simulado e adicionado ao sinal de saída uma sequência de ruído branco de média nula na proporção de 40dB. Não foi considerado ruído de processo. Na simulação foram geradas 1500 amostras, sendo que as primeiras 1000 amostras foram utilizadas para identificação do sistema e as últimas 500 amostras utilizadas para a validação. No processo de identificação a ordem do sistema foi considerada conhecida a priori. Um pequeno trecho dos sinais utilizados na identificação estão mostrados nas Figuras 2.2a e 2.2b.

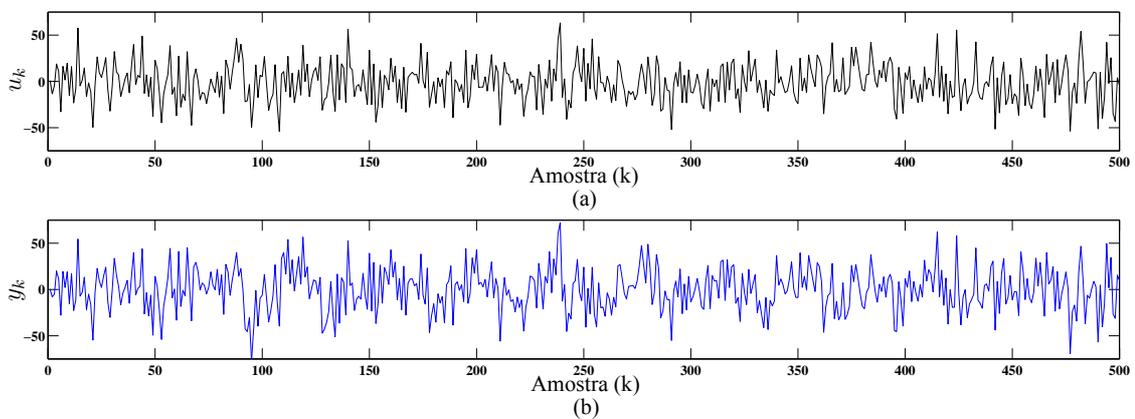


Figura 2.2: Sinais utilizados e obtidos no processo de identificação. (a) Sinal de entrada (u_k) em volts. (b) Sinal de saída (y_k) em volts. Nestas figuras, para melhor visualização, apenas um pequeno trecho dos sinais está sendo mostrado.

A aplicação do algoritmo ACR descrito na seção 2.1.1 permite obter uma estimativa das matrizes do sistema. No presente exemplo, as matrizes obtidas foram:

$$\hat{A} = \begin{bmatrix} 0,9569 & -0,3281 \\ 0,0721 & 0,9691 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0,3319 \\ -0,1072 \end{bmatrix}, \hat{C} = \begin{bmatrix} -0,3256 & -0,4833 \end{bmatrix}, \hat{D} = 0,9503,$$

Esse modelo estimado constitui uma boa aproximação do sistema real, uma vez que consegue reconstruir a saída real do sistema de maneira satisfatória, como pode ser observado na Figura 2.3a. De fato, os autovalores estimados ($0,9630 \pm 0,1537j$) são muito próximos dos autovalores reais do sistema ($0,9635 \pm 0,1523j$).

No entanto, as matrizes do sistema foram estimadas em relação a uma transformação de similaridade T . Conforme já discutido na seção 2.1.2, no contexto de identificação de sistemas dinâmicos híbridos, essa é uma questão delicada, pois essa matriz T pode ser diferente

para cada modelo local do sistema híbrido. Assim, o algoritmo ACR é combinado com a metodologia descrita na seção 2.1.2 para representar as matrizes do sistema na forma canônica. A aplicação conjunta destas metodologias permite obter a seguinte matriz de transformação:

$$T = \begin{bmatrix} -0,3256 & -0,4833 \\ -0,3464 & -0,3616 \end{bmatrix}$$

Essa matriz de transformação leva à seguintes matrizes transformadas:

$$\hat{A}_c = \begin{bmatrix} 0 & 1 \\ -0,9509 & 1,9260 \end{bmatrix}, \hat{B}_c = \begin{bmatrix} -0,0562 \\ -0,0762 \end{bmatrix}, \hat{C}_c = \begin{bmatrix} 1 & 0 \end{bmatrix}, \hat{D}_c = 0,9974,$$

em que o sub-escrito c foi utilizado para representar as matrizes na forma canônica.

Esse novo modelo também consegue reconstruir a saída real do sistema de maneira satisfatória, como pode ser observado na Figura 2.3b.

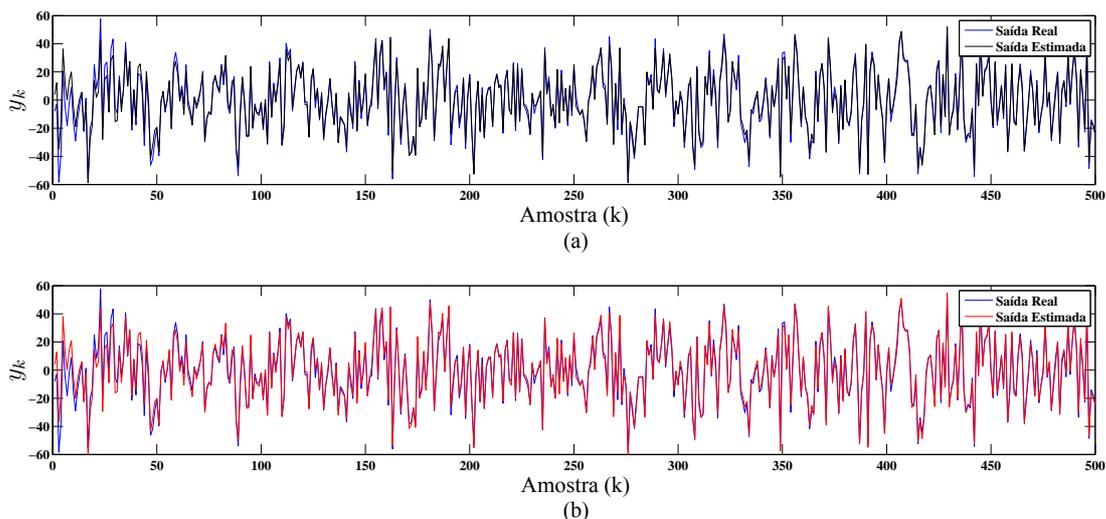


Figura 2.3: Comparação entre a saída real do sistema (proveniente do conjunto de dados de validação) e a estimada pelo modelo. (a) Modelo estimado pelo algoritmo ACR (b) Modelo estimado pelo algoritmo ACR associado com a metodologia descrita na seção 2.1.2 para representar todos os modelos na mesma base de estados.

2.1.3 Estimação da ordem dos modelos

A metodologia de identificação em subespaços utilizada neste trabalho permite a estimação da ordem dos modelos locais através da inspeção dos valores singulares diferentes de zero na equação (2.18). No entanto, quando o sistema está corrompido com altos níveis

de ruído, ou quando o número de dados amostrados não é suficientemente grande ou ainda quando o sistema real é não-linear, a inspeção desses valores singulares não é trivial, uma vez que todos os valores singulares são diferentes de zero [86], [92]. Nesses casos, deve-se estimar a ordem do sistema como o número de valores singulares dominantes. O problema é que esses valores singulares dominantes podem não ser evidentes.

Existe uma extensa literatura de algoritmos de estimação de ordem para sistemas dinâmicos [93], [94], [95], [96], [97], [98]. Segundo [98], as diferentes técnicas de estimação de ordem desenvolvidas podem ser divididas em três grupos; 1) aquelas que estimam a ordem e os parâmetros dos modelos simultaneamente, 2) aquelas que necessitam de estimativas a priori dos parâmetros do modelo e; 3) as que não precisam dessas estimativas.

O primeiro grupo de técnicas consiste em estimar a ordem e os parâmetros do modelo simultaneamente. Uma abordagem Bayesiana oferece essa possibilidade através de um processo adaptativo que consiste em atualizar os parâmetros do modelo e alterar a ordem modelo baseado na natureza dos dados que estão sendo modeladas [99]. A principal desvantagem desses métodos refere-se custo computacional.

No segundo grupo, as abordagens consistem em encontrar um equilíbrio entre a qualidade do ajuste e complexidade do modelo. Em geral, essa tarefa é realizada através de um procedimento recursivo que avalia uma gama de diferentes valores de ordens. Para cada valor de ordem é estimado um modelo e este é usado para calcular a variância do erro de predição. Por fim, o modelo que minimiza esse erro é escolhido como o melhor modelo a partir do conjunto de modelos disponível. As técnicas mais utilizadas e importantes estão classificadas neste grupo, como exemplo destacam-se o trabalho pioneiro [100] que propõem o *Akaike information criterion* (AIC) e suas extensões AICC [101] and AIC_i [96]. O critério AIC também motivou muitas outras propostas, como por exemplo [102] e [103]. Contudo, a desvantagem destas técnicas é a complexidade computacional, uma vez que diversos modelos devem ser estimados até que a ordem ideal seja determinada [93], [98].

O terceiro grupo inclui uma enorme gama de propostas e se caracterizam por não necessitar de estimativas a priori dos parâmetros do modelo. Em geral, a maioria das técnicas focam na análise dos autovalores ou dos autovetores da matriz de covariância dos dados de entrada/saída para determinar a ordem ótima do modelo. Destas abordagens destacam-se os trabalhos [104], [105] e [98]. Alguns autores abordam esse problema utilizando critérios baseados em subespaços os quais se baseiam na análise do maior valor singular negligenciado na etapa de decomposição em valores singulares. Neste contexto, os trabalhos de maior destaque são [106], [93] e [97].

No presente trabalho, para a escolha do algoritmo de estimação de ordem, eliminou-se os trabalhos dos dois primeiros grupos devido a alta carga computacional necessária. Assim, dentre os trabalhos do terceiro grupo escolheu-se a abordagem proposta em [97]. Essa metodologia chamada de critério NIDC é baseada na informação contida nos valores

singulares estimados do sistema e apresenta bons resultados mesmo nas situações onde o conjunto de dados é relativamente pequeno. Esse algoritmo é implementado no toolbox do Matlab chamado de E⁴ [97]. O critério NIDC é definido como:

$$NIDC(n) = \hat{\sigma}_{n+1}^2 + F(T, i)d(n), \quad (2.36)$$

em que σ_{n+1} é o $n+1$ valor singular, $d(n) = 2nn_u$ denota o número de parâmetros, $n = 0, 1, \dots, i-1$, são os diferentes valores de ordem avaliadas e $F(T, i)$ é uma função de penalidade definida como $F(T, i) = \exp^{-2} T^{-0,9} i^{1,5}$. O parâmetro i é o número de blocos linha nas matrizes de *block-Hankel* e T é o número de dados utilizado na análise.

Em (2.36), os autovalores considerados são obtidos de:

$$W_1 Y_f W_2 \quad (2.37)$$

em que Y_f é a matriz de Block-Hankel definida de maneira similar a (2.9), W_1 e W_2 são duas matrizes de pesos. A matriz W_1 é definida como $W_1 = (\tilde{Y}_f \tilde{Y}_f^T)^{-0,5}$, em que \tilde{Y}_f contém os resíduos da regressão de Y_f sobre Y_p . A matriz W_2 é definida como $W_2 = Y_p^T (Y_p Y_p^T)^{-0,5}$.

Assim, a ordem estimada é obtida como o argumento que minimiza (2.36). Para maiores detalhes veja [97].

2.2 CLASSIFICAÇÃO DE DADOS

Conforme mencionado no início deste capítulo, o problema de identificação híbrida inclui um problema de classificação de dados onde cada par de dado amostrado $\{(u_k, y_k)\}_{k=1}^N$ deve ser associado ao modelo local mais adequado. Para essa classificação existem duas abordagens: 1) as partições são fixas a priori e, 2) as partições são estimadas junto com os sub-modelos. No primeiro caso, a classificação de dados é muito simples, e a estimação dos sub-modelos pode ser realizada recorrendo-se a técnicas clássicas de identificação de sistemas lineares [7]. O segundo caso aborda um difícil problema, uma vez que há uma estrita relação entre a classificação dos dados e estimação dos parâmetros. O problema se torna ainda mais difícil quando o número de sub-modelos deve ser estimado.

Desta forma, a classificação de dados tem um papel fundamental no processo de identificação de sistemas híbridos. A classificação ou agrupamento de dados em categorias distintas baseado em algum critério de similaridade é uma importante atividade na área de análise de dados [107]. A classificação desempenha um papel importante e indispensável ao longo da história do desenvolvimento humano. A fim de aprender sobre um novo objeto ou entender um novo fenômeno, as pessoas sempre tentam identificar e comparar suas características descritivas com as de objetos ou fenômenos conhecidos, com base em sua similaridade ou dissimilaridade de acordo com algumas normas ou regras determinadas [108].

Basicamente, os sistemas de classificação são divididos em supervisionados ou não-supervisionados. Na classificação supervisionada a estrutura de dados é previamente estabelecida e a tarefa é decidir a qual grupo cada elemento pertence, em geral, o mapeamento de um conjunto de dados de entrada em um conjunto finito de clusters é modelado em termos de alguma função matemática [108]. Como cluster entenda-se um grupo de objetos que são similares entre si e dissimilares com objetos de outros grupos [95].

Já na classificação não-supervisionada, também chamada de clustering ou análise de dados exploratória, não há nenhuma estrutura de grupos previamente definida. O objetivo do clustering é, portanto, criar uma estrutura de grupos baseado nas características dos elementos. Trata-se de um problema complexo que envolve a determinação de conceitos abstratos de similaridades e que podem não estar evidentes. O conceito de similaridade deve ser adequado e definido para cada aplicação.

O interesse no desenvolvimento de novas técnicas de clustering aumentou consideravelmente nos últimos anos. Atualmente, já existem uma enorme quantidade de algoritmos de clustering para as mais diversas aplicações [9]. É importante destacar que, diferentes abordagens de clustering geralmente levam a diferentes conjuntos de dados, inclusive em um mesmo algoritmo, a seleção de um parâmetro ou a ordem de apresentação dos padrões de entrada podem afetar os resultados finais.

Uma extensa e excelente revisão sobre as diferentes abordagens de clustering já desenvolvidas é apresentada em [108] e [107], para um aprofundamento maior no assunto, sugere-se ainda os trabalhos [109], [110], [111], [37]. Do ponto de vista de identificação de sistemas dinâmicos híbridos, a maioria das abordagens baseadas em clustering utilizam algoritmos clássicos de clustering e suas derivações [9]. Esses trabalhos utilizam os seguintes algoritmos de clustering: o *K-Means* [32], [67], o *Single Linkage* [112], o *Expectation Maximization* [35] e o *Split and Merge* [37].

O algoritmo *K-means* [113], [114] é um dos algoritmos de clustering mais conhecido e utilizado [107]. Sua popularidade deve-se ao fato dele ser um algoritmo muito simples e pode ser facilmente implementado na resolução de muitos problemas práticos. Esse algoritmo busca a partição ótima dos dados através da minimização de uma função custo através de um procedimento de otimização iterativo. A função custo, em geral, utiliza um critério de erro médio quadrático.

As características e desvantagens desse algoritmo já são bem estudadas, e como resultado, muitas variantes desse algoritmo surgiram para tentar superar essas desvantagens. A mais importante delas é o algoritmo *Fuzzy C-Means* [115]. As principais desvantagens desse algoritmo são: não garantia de convergência para o máximo global, é muito sensível a *outliers* (dados de entrada que não são similares a nenhum dos clusters encontrados) e ruído e não há um método eficiente e universal para identificar o número de clusters [108].

O algoritmo *Single Linkage*, também conhecido como algoritmo do vizinho mais pró-

ximo (do inglês *nearest neighbor*), é um método hierárquico que se baseia no agrupamento dos elementos próximos uns aos outros. No início do processo é criado um cluster para cada elemento do conjunto total de dados. Assim, a cada iteração os clusters mais próximos são unidos para formar novos clusters. Esse procedimento é repetido até que algum critério de parada seja satisfeito. Uma vantagem deste algoritmo é a possibilidade de detectar automaticamente o número de clusters. Contudo, observa-se que seu desempenho está diretamente relacionado com a escolha adequada do critério de parada. As outras desvantagens deste método são que grupos muito próximos podem não ser identificados e o fato de apresentar pouca tolerância a ruído, uma vez que possui a tendência de incorporar os ruídos em um grupo já conhecido.

O algoritmo *Expectation Maximization* introduziu um novo conceito ao considerar cada cluster como uma FDP. Assim, o agrupamento dos elementos em diferentes clusters é realizado de acordo com as características da FDP que representa cada cluster. Esse algoritmo tem duas desvantagens. Primeiro, é a ausência da garantia de convergência para o máximo global. De fato, o algoritmo pode convergir para máximos locais e até mesmo para mínimos locais. Segundo, é muito dependente da inicialização, ou seja, um bom desempenho depende de uma boa inicialização.

O algoritmo *Split and Merge* proposto em [37] utiliza o procedimento de Análise dos Componentes Principais para dividir o conjunto de dados e usa o índice de silhueta para avaliar a configuração resultante. Desta forma, esta metodologia procura a melhor configuração de agrupamento do conjunto de dados. O algoritmo proposto detecta automaticamente o número de sub-modelos e requer o ajuste de poucos parâmetros na inicialização. Contudo, assim como os demais algoritmos, o método também apresenta pouca tolerância ao ruído.

Desta forma, neste trabalho, escolheu-se o algoritmo Aglomeração Competitiva Robusta - RCA (do inglês, *Robust Competitive Agglomeration*) proposto por [116]. A escolha do RCA deve-se ao fato dele abordar os três principais problemas associados com as técnicas de clustering, quais sejam, a dificuldade em determinar o número de clusters, a sensibilidade à inicialização e a sensibilidade a outliers. As ideias envolvidas neste algoritmo serão discutidas a seguir.

2.2.1 Algoritmo de Aglomeração Competitiva Robusta - RCA

No RCA o número ótimo de clusters é determinado usando um processo de aglomeração competitiva, enquanto que o conhecimento sobre a forma global dos clusters é incorporado através da utilização de protótipos. Para superar a sensibilidade aos outliers são incorporados conceitos da estatística robusta e a sobreposição de clusters é tratada pelo uso de conjuntos nebulosos (do inglês, *fuzzy memberships*).

O algoritmo se inicia particionando o conjunto de dados em um grande número de pequenos clusters. Isso tem a função de reduzir a sensibilidade à inicialização. A medida que

o algoritmo progride, clusters adjacentes começam a competir por pontos, e os clusters que perdem a competição vão desaparecendo gradualmente. Neste processo, os pontos podem se mover de um cluster para outro.

As estimativas robustas dos protótipos de clusters são obtidas através da utilização de dois diferentes conjuntos de pesos para cada ponto. O primeiro conjunto representa o grau de compartilhamento entre os clusters e o segundo representa o grau de semelhança dos pontos com relação aos clusters. A seguir será realizada uma breve apresentação deste algoritmo.

A ideia do algoritmo é minimizar a seguinte função custo:

$$J(\mathcal{B}, U_H; X) = \sum_{q=1}^H \sum_{k=1}^{\mathcal{N}} (u_{qk})^2 \rho_q(d_{qk}^2) - \alpha \sum_{q=1}^H \left[\sum_{k=1}^{\mathcal{N}} w_{qk} u_{qk} \right]^2, \quad (2.38)$$

sujeito a:

$$\sum_{q=1}^H u_{qk} = 1 \quad \text{para} \quad 1 \leq k \leq \mathcal{N}, \quad (2.39)$$

em que, $\mathcal{B} = (\beta_1, \dots, \beta_H)$ representa os protótipos que caracterizam cada um dos H clusters, $X = \{x_k \mid k = 1, \dots, \mathcal{N}\}$ é um conjunto de \mathcal{N} vetores no espaço n -dimensional, $u_{q,k}$ representa o grau no qual x_k pertence ao cluster q , $U_H = [u_{qk}]$ é uma matriz $H \times \mathcal{N}$ chamada *constrained fuzzy H-partition matrix* e d_{qk} representa a distância do vetor x_k até o protótipo β_q . A variável α é um parâmetro de projeto utilizada para balancear os dois termos da equação de forma a encontrar uma solução do número de clusters H .

O primeiro termo na equação (2.38) é, essencialmente, a soma das distâncias intra-clusters. Esse termo tem uma tendência monotônica e tem seu valor mínimo igual a zero quando $H = \mathcal{N}$. Isso não é útil para a determinação automática de H . Assim, para superar esse inconveniente, adicionou-se o segundo termo da equação, que é um termo de regularização que visa prevenir o superajuste do número de clusters. Observe que, esse termo é o negativo da soma dos quadrados da cardinalidade dos clusters e é minimizado quando a cardinalidade de um dos clusters é igual a \mathcal{N} e os demais clusters estão vazios. Desta forma, com uma escolha adequada de α é possível balancear os dois termos para encontrar a solução de H .

As estimativas robustas são obtidas adicionando-se a função de perda robusta $\rho_q(\cdot)$ associada ao cluster q e a função de pesos $w(\cdot)$. A função $w_{qk} = w_q(d_{qk}^2) = \partial \rho_q(d_{qk}^2) / \partial d_{qk}^2$ representa o grau de semelhança do ponto x_k com relação ao clusters q . Isso permite obter a robustez nas estimativas, mantendo a complexidade computacional baixa. A função de perda robusta reduz os efeitos dos outliers no primeiro termo, e a função de pesos desconta os outliers enquanto calcula a cardinalidade. Desta forma, pela seleção de d_{qk} e α cuidadosamente, a função J pode ser utilizada para encontrar clusters compactos de vários tipos. Neste trabalho, foi utilizado a seguinte medida de distância:

$$d_{qk} = \sqrt{(x_k - h_q)^2}, \quad (2.40)$$

em que h_q é o centro do cluster β_q .

A minimização da função J leva a seguinte equação de atualização:

$$u_{st} = \frac{1/\rho_s(d_{st}^2)}{\sum_{k=1}^H 1/\rho_k(d_{st}^2)} + \frac{\alpha}{\rho_k(d_{st}^2)}(N_s - \bar{N}_t), \quad (2.41)$$

em que

$$N_s = \sum_{k=1}^N w_{sk} u_{sk}, \quad \bar{N}_t = \frac{\sum_{k=1}^H \frac{1}{\rho_k(d_{kt}^2)} N_k}{\sum_{k=1}^H \frac{1}{\rho_k(d_{kt}^2)}}. \quad (2.42)$$

Ao longo das iterações do algoritmo são descartados os clusters que apresentarem a cardinalidade robusta N_s menor que um valor ϵ_0 definido pelo usuário. Assim, o processo de aglomeração, controlado por α , deve ser lento no começo para encorajar a formação de pequenos clusters. Então ele deve ser aumentado gradualmente para promover a aglomeração. Após algumas iterações, quando o número de clusters torna-se perto do ótimo, o valor de α deve voltar a decair lentamente para permitir a convergência do algoritmo. Em [116], os autores propõem a utilização da seguinte função:

$$\alpha(k) = \eta(k) \frac{\sum_{q=1}^H \sum_{j=1}^N (u_{qj}^{(k-1)})^2 (\rho_q(d_{qj}^2))^{(k-1)}}{\sum_{q=1}^H \left[\sum_{j=1}^N w_{qj}^{(k-1)} u_{qj}^{(k-1)} \right]^2}. \quad (2.43)$$

em que o sobrescrito $(k-1)$ é usado para denotar os valores na iteração anterior. Uma boa escolha para o parâmetro η é:

$$\eta(k) = \begin{cases} \eta_0 \exp \frac{-|k_0-k|}{\iota} & \text{se } k > 0, \\ 0 & \text{se } k = 0, \end{cases} \quad (2.44)$$

em que η_0 é o valor inicial, ι é a constante de tempo e k_0 é o número da iteração na qual a função η começa a decair.

Para a função de pesos foi utilizada uma função monotônica decrescente, $w_q(d^2) : \mathcal{R}^+ \rightarrow [0, 1]$ tal que $w_q(d^2) = 0$ para $d^2 > T_q + cS_q$, em que c é uma constante, e T_q e S_q são dados por:

$$T_q = \text{Med}_q(d_{qk}^2) \quad \text{e} \quad S_q = \text{MAD}_q(d_{qk}^2) \quad \text{para } q = 1, \dots, H, \quad (2.45)$$

em que Med_q representa a mediana das distâncias em relação ao cluster q e MAD_q é a mediana dos desvios absolutos do clusters q . O parâmetro c é um parâmetro de ajuste do usuário normalmente escolhido entre quatro e doze. Quando c é muito grande, muitos outliers terão pesos diferentes de zeros, consequentemente afetando os parâmetros estimados. Por outro lado, se c é pequeno, somente um sub-conjunto dos dados será visível no processo de estimação, tornando a convergência para um mínimo local mais provável. Em [116], os autores sugerem iniciar o processo de estimação com um alto valor de c e então ir diminuindo gradualmente como função do número da iteração.

Escolhendo $w_q(0) = 1$, $w_q(T_q) = 0.5$, e $w'_q(0) = 0$, resulta na seguinte função de pesos:

$$w_q(d^2) = \begin{cases} 1 - \frac{d^4}{2T_q^2} & \text{se } d^2 \in [0, T_q], \\ \frac{[d^2 - (T_q + cS_q)]^2}{2c^2S_q^2} & \text{se } d^2 \in [T_q, T_q + cS_q], \\ 0 & \text{se } d^2 > T_q + cS_q. \end{cases} \quad (2.46)$$

A correspondente função de perda é:

$$\rho_q(d^2) = \begin{cases} d^2 - \frac{d^6}{6T_q^2} & \text{se } d^2 \in [0, T_q], \\ \frac{[d^2 - (T_q + cS_q)]^3}{6c^2S_q^2} + \frac{5T_q + cS_q}{6} & \text{se } d^2 \in [T_q, T_q + cS_q], \\ \frac{5T_q + cS_q}{6} + K_q & \text{se } d^2 > T_q + cS_q. \end{cases} \quad (2.47)$$

em que K_q é uma constante de integração definida como:

$$K_q = \max_{1 \leq k \leq H} \left\{ \frac{5T_k + cS_k}{6} \right\} - \frac{5T_q + cS_q}{6} \quad \text{para } q = 1, \dots, H. \quad (2.48)$$

A eficiência do RCA para determinação automática do número de clusters e segmentação dos dados será ilustrada através de um simples exemplo apresentado a seguir.

Exemplo 2.2: Neste exemplo, o RCA é utilizado para segmentar os dados mostrados na Figura 2.4a. Esses dados, que formam dois clusters, foram gerados através das seguintes equações:

$$y_1 = 1, 2x; \quad (2.49)$$

$$y_2 = -0, 8x; \quad (2.50)$$

em que $x \in [0, 10]$.

No total foi considerado 250 pontos de cada equação e adicionado aos sinais de saída um termo de ruído branco de média zero e variância de 0, 25. Para dificultar a tarefa do RCA, foi adicionado ao conjunto de dados uma sequência de ruído (denominado y_3) uniformemente distribuído constituindo 40% do número total de pontos. Considerou-se o número inicial de clusters igual a 10. O conjunto total de dados utilizado é representado por $Y = [y_1 \ y_2 \ y_3]$.

A Figura 2.4b apresenta o resultado após a primeira iteração, onde percebe-se que apenas um cluster foi descartado. Nesta figura, as cruzes em vermelho indicam o centro de cada cluster. Na segunda iteração o número de cluster é reduzido para quatro, como mostrado na Figura 2.4c. O número correto de clusters é identificado após a terceira iteração. O resultado final após um total de 12 iterações é mostrado na Figura 2.4d.

Além dos centros de cada cluster identificado, o RCA fornece como resultado a matriz *constrained fuzzy H-partition matrix*, U_H , que fornece o grau de confiança de cada ponto pertencer a cada um dos clusters identificados. Assim, essa matriz permite associar cada um dos pontos ao cluster com maior grau de confiança. Por exemplo, considere a primeira coluna da matriz U_H

$$U_H(:, 1) = \begin{bmatrix} 0,8057 \\ 0,1943 \end{bmatrix},$$

em que utilizou-se a notação como no Matlab $(:, 1)$ para se referir a todas as linhas da primeira coluna da matriz.

Esta coluna corresponde ao par de dados $(x = 0, y_1 = -0,3043)$. Os valores desta coluna indicam que tem-se 80,57% de confiança do ponto pertencer ao cluster 1 e 19,43% de confiança do ponto pertencer ao cluster 2. Desta forma, o ponto é associado ao cluster 1, pois apresenta maior grau de confiança. O processo de classificação é concluído repetindo-se esse procedimento para todos os pontos do conjunto de dados. Na Figura 2.4d é apresentado o resultado final da classificação, onde os dados pertencentes a cada um dos clusters são apresentados com cores diferentes. Observe que, no exemplo aqui considerado, existe uma sobreposição dos clusters e uma enorme quantidade de ruído adicionado ao conjunto de dados e, mesmo neste cenário, o RCA consegue classificar corretamente a maioria dos pontos existentes no conjunto total de dados.

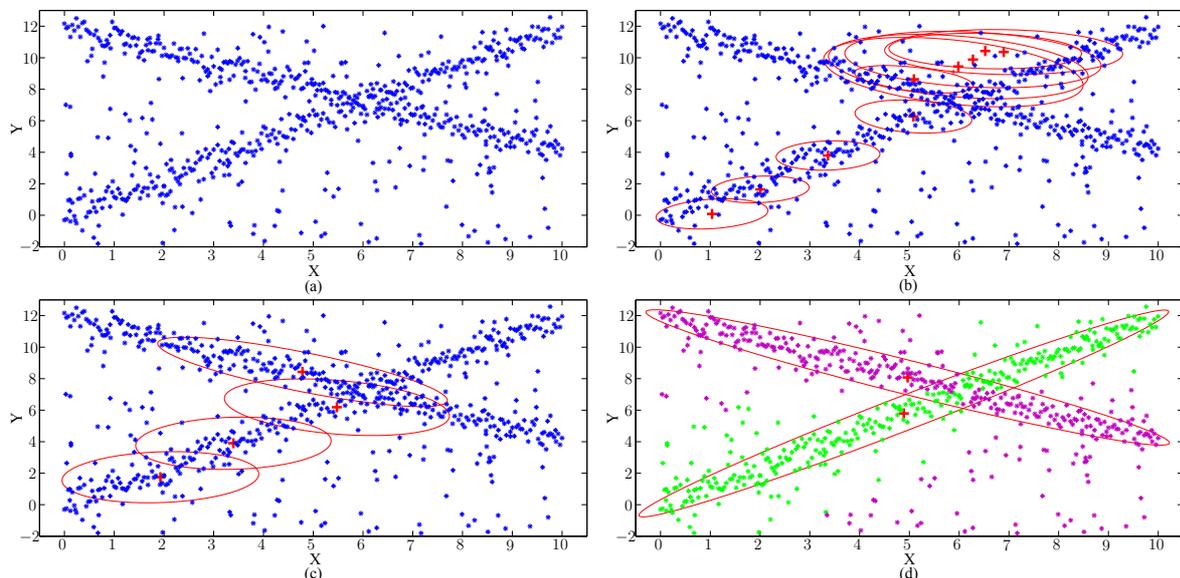


Figura 2.4: Resultados do RCA em um conjunto de dados ruidosos. (a) Conjunto de dados original. (b) Resultado após a primeira iteração. (c) Resultado após a segunda iteração. (d) Resultado após 12 iterações (convergiu). Nesta última figura, os pontos atribuídos a cada cluster estão representados em cores diferentes.

2.3 FILTRAGEM HÍBRIDA

No Capítulo 3 serão apresentados dois algoritmos que possuem em seu procedimento uma etapa nomeada de atualização da segmentação dos dados. Nesta etapa, já existem estimativas parciais dos modelos locais e o problema consiste em associar cada par de dado amostrado $\{(u_k, y_k)\}_{k=1}^N$ ao sub-modelo mais adequado. Desta forma, esta tarefa pode ser vista como um problema de estimação de estados em sistemas dinâmicos híbridos, também conhecido como filtragem híbrida. A principal diferença entre as técnicas de filtragem tradicional e as técnicas de filtragem híbrida é a necessidade, na abordagem híbrida, da estimação dos estados discretos do sistema. Neste trabalho, o principal interesse nas técnicas de filtragem híbrida é apenas na estimação do estado discreto do sistema.

Para descrever o problema de filtragem híbrida será considerado uma formulação de sistemas híbridos mais geral do que a apresentada na Seção 1.3. Esta formulação está apresentada abaixo.

$$\begin{cases} x_k &= f_{m_{k-1}}(x_{k-1}, u_{k-1}) + w_{k-1}, \\ y_k &= h_{m_k}(x_k) + v_k, \end{cases} \quad (2.51)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto, e M é o número de sub-modelos; $f_{m_{k-1}}$ é uma função de evolução do processo dependente do modo e h_{m_k} é a função de medição do sistema, onde considerou-se para maior clareza de notação e sem perda de generalidade que não depende dos sinais de entrada u_k . Os vetores $x_k \in \mathbb{R}^{n_{m_k}}$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos, as entradas e saídas do sistema. Os vetores $w_k \in \mathbb{R}^{n_{m_k}}$ e $v_k \in \mathbb{R}^{n_y}$ são termos de ruído de processo e medida, respectivamente.

Desta forma, o problema consiste em obter estimativas dos estados discretos (m_k) e contínuos (x_k) do sistema baseado em uma sequência de dados amostrados $\{y_j\}_{j=1}^k$. Este problema pode ser posto como um problema de se obter a FDP a posteriori conjunta de x_k e m_k . Esta FDP é definida como:

$$p(x_k, m_k | \{y_j\}_{j=1}^k). \quad (2.52)$$

Geralmente, o modo de operação e os estados contínuos não são mensuráveis. Assim, a estimação de (2.52) é uma tarefa normalmente muito difícil. No entanto, este problema pode ser dividido em dois através da fatoração da PDF em (2.52) da seguinte forma:

$$p(x_k, m_k | \{y_j\}_{j=1}^k) = p(x_k | m_k, \{y_j\}_{j=1}^k) \Pr(m_k | \{y_j\}_{j=1}^k). \quad (2.53)$$

Assim, o problema agora consiste em estimar a FDP a posteriori de $p(x_k | m_k, \{y_j\}_{j=1}^k)$ de x_k assumindo que o modo é conhecido; e estimar a probabilidade condicional modal discreta $\Pr(m_k | \{y_j\}_{j=1}^k)$ que não depende do estado contínuo. O cálculo deste segundo termo é relativamente fácil de se realizar. Para isso, pode-se aplicar o Teorema de Bayes da seguinte forma.

$$\Pr(m_k | \{y_j\}_{j=1}^k) = \frac{p(y_k | m_k, \{y_j\}_{j=1}^{k-1}) \Pr(m_k | \{y_j\}_{j=1}^{k-1})}{c}, \quad (2.54)$$

em que c é fator independente tanto de x_k quanto de m_k e pode ser excluído da análise.

O primeiro termo em (2.54) é a verossimilhança da medida y_k assumindo-se que o modo m_k está ativo. A verossimilhança de y_k é uma medida de quão próxima esta medição experimental está do valor predito usando a informação do modelo matemático [1]. Se o ruído de medição v_k em (2.51) for suposto Gaussiano, a expressão desta verossimilhança torna-se a conhecida expressão da FDP exponencial Normal:

$$p(y_k|m_k, \{y_j\}_{j=1}^{k-1}) = \frac{1}{(2\pi)^{\frac{n_y}{2}} \det(\Sigma_\vartheta)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\vartheta)^T \Sigma_\vartheta^{-1}(\vartheta)\right\}, \quad (2.55)$$

em que

$$\vartheta = y_k - \bar{y}_k, \quad (2.56)$$

e \bar{y}_k é a saída predita pelo modelo e definida como:

$$\bar{y}_k = h_{m_k}(\bar{x}_k) + E\{v_k\} = h_{m_k}(\bar{x}_k), \quad (2.57)$$

em que $\bar{x}_k = E\{x_k|y_{1:k-1}\}$ é o valor do estado contínuo predito pelo modelo. Em (2.57) usou-se a consideração que o ruído de medição v_k é Gaussiano de média nula.

O termo Σ_ϑ em (2.55) é a matriz de covariância definida como:

$$\Sigma_\vartheta = \left(\left. \frac{\partial h'_{m_k}(x_k)}{\partial x_k} \right|_{\bar{x}_k} \right) \bar{P}_k \left(\left. \frac{\partial h'_{m_k}(x_k)}{\partial x_k} \right|_{\bar{x}_k} \right)^T + R_{v_k},$$

em que $\partial h'_{m_k}(x_k)/\partial x_k$ é a matriz Jacobiana da função de medição avaliada na estimativa predita de estado \bar{x}_k e \bar{P}_k é a matriz de covariância associada a esta estimativa. Para maiores detalhes dessas equações e sobre métodos de filtragem não-linear sugere-se a leitura de [1].

O segundo termo em (2.54) é a probabilidade modal discreta $\Pr(m_k|\{y_j\}_{j=1}^{k-1})$. Para o cálculo deste termo, será necessário impor a condição que as transições entre os modos ocorrem segundo uma Cadeia de Markov. Neste caso, as probabilidades de transição entre os modos dependem apenas do modo do sistema no instante de tempo atual. As probabilidades de transições entre os diferentes modos são definidos por uma Matriz de Probabilidades de Transição (MPT) e que pode variar com o tempo. Esta MPT é definida como:

$$\Pi_k = \{\pi_{i,j}\}, \pi_{i,j} = \Pr\{m_k=j|m_{k-1}=i\}, i, j \in \mathbb{M}, \forall k \in \mathbb{N}, \quad (2.58)$$

é importante notar que esta matriz define o quão provável é o salto do modelo $m_{k-1}=i$ no instante $k-1$ para o modelo $m_k=j$ no instante k .

Assim, considerando que as transições entre os modos segue uma Cadeia de Markov com modelo matemático definido por (2.58), o cálculo de $\Pr(m_k|\{y_j\}_{j=1}^{k-1})$ necessita apenas da MPT do instante de tempo anterior Π_{k-1} , e do vetor de probabilidades dos modos no instante anterior. Este vetor é definido por:

$$p(m_{k-1}) = [\Pr(m_{k-1}=1|y_{1:k-1}), \Pr(m_{k-1}=2|y_{1:k-1}), \dots, \Pr(m_{k-1}=M|y_{1:k-1})]^T.$$

Desta forma, a seguinte expressão permite calcular de uma só vez todos os termos de $\Pr(m_k | y_{1:k-1})$:

$$\bar{p}(m_k) = (\Pi_{k-1})^T p(m_{k-1}), \quad (2.59)$$

em que $\bar{p}(m_k)$ é o vetor de probabilidades preditas dos modos no instante atual.

Portanto, as expressões (2.55) e (2.59) fornecem uma forma fechada para o cálculo do termo $\Pr(m_k | \{y_j\}_{j=1}^{k-1})$. Contudo, a principal dificuldade na estimação de (2.53) está no cálculo do termo $p(x_k | m_k, \{y_j\}_{j=1}^k)$. O problema na estimação deste termo é que uma estimativa ótima consiste em monitorar o conjunto de todas possíveis trajetórias dos modos discretos do sistema. Seguir todas essas hipóteses de trajetórias é quase sempre intratável porque o número de hipóteses se torna muito grande após alguns instantes de tempo. Como exemplo, considere o estimador ótimo, no sentido de mínimos quadrados, obtido em [117] para o caso em que todos os modelos locais são lineares, a MPT é conhecida e os processos de ruído são Gaussianos. Este estimador é dado por:

$$\tilde{x}_k = \sum_{i=1}^{M^k} \tilde{x}_k^{(i)} \Pr(h_k^{(i)} | y_{1:k}), \quad (2.60)$$

em que $h_k^{(i)} \in \Omega_k = \{h_k^{(i)} | i = 1, 2, \dots, M^k\}$ denota a i -ésima hipótese do conjunto de todas as possíveis sequências de modos Ω_k ; e $\tilde{x}_k^{(i)} = E\{x_k | h_k^{(i)}, y_{1:k}\}$ é a estimativa de estado dada por um Filtro de Kalman (FK) seguindo a hipótese $h_k^{(i)}$.

A equação (2.60) torna clara a necessidade de monitorar todas as possíveis trajetórias dos modos discretos para obter a estimativa ótima do estado contínuo. Como consequência, é inevitável a utilização de abordagens subótimas que fazem uso de métodos de gerenciamento de hipóteses, tais como a fusão de hipóteses similares ou eliminação daquelas que são improváveis, para manter o número de hipóteses em avaliação dentro de um determinado limite.

Em geral, as abordagens utilizam um conjunto de filtros, tipicamente um para cada modo do sistema, que são executados simultaneamente e a estimativa final é resultado da combinação das estimativas dos filtros individuais de acordo com algum esquema de ponderação específico. Alguns exemplos das abordagens já desenvolvidas são o algoritmo *Generalized Pseudo-Bayesian* (GPB) [117], o algoritmo *detection-estimation* [118], o algoritmo *residual correlation Kalman filter bank* [119], o algoritmo *Interacting Multiple-Model* (IMM) [80], [81] e o filtro M³H [117]. Destas abordagens, as mais importantes são os filtros GPB e o IMM [1]. A estratégia utilizada no primeiro para lidar com a explosão exponencial do número de hipóteses é realizar uma combinação ponderada de hipóteses baseada em suas probabilidades após um número fixo de passos. A primeira versão do algoritmo, chamada de GPB1, considera a estimação com apenas um passo. Mas em sua forma generalizada o algoritmo GPBn calcula todas as hipóteses de trajetória para n passos.

Já o algoritmo IMM executa uma combinação (“mistura”) das estimativas do passo anterior de forma a gerar novas estimativas iniciais para o seu banco de FK a cada instante

de tempo. O filtro IMM é o estimador híbrido sub-ótimo mais utilizado e que apresenta o melhor compromisso entre performance e complexidade [120]. Em particular, o IMM exibe requisitos computacionais que crescem de forma aproximadamente linear com o número de modos, enquanto que seu desempenho é comparável com dos algoritmos de complexidade quadrática. Neste sentido, o IMM é uma das melhores escolhas para estimação de estados de sistemas híbridos [1]. Por estes motivos foi o filtro escolhido para a ser implementado neste trabalho.

A seguir será realizada uma breve descrição deste algoritmo. Desenvolvimentos e provas matemáticas serão omitidas e o leitor mais interessado deverá se referir aos trabalhos originais [80], [81]. Recomenda-se também a leitura de [1] para uma discussão mais detalhada sobre os métodos de filtragem híbrida.

2.3.1 Algoritmo IMM

O filtro IMM que será descrito aborda o problema de filtragem para Sistemas Lineares com Saltos Markovianos (MJLS, do inglês *Markov Jump Linear Systems*) descritos por:

$$\begin{cases} x_{k+1} &= A_{m_k} x_k + B_{m_k} u_k + w_k, \\ y_k &= C_{m_k} x_k + D_{m_k} u_k + v_k, \end{cases} \quad (2.61)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto e assume-se que segue uma Cadeia de Markov com MPT $\Pi : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^M$, podendo assumir M diferentes valores. Os vetores $x_k \in \mathbb{R}^{n_{m_k}}$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos, as entradas e saídas do sistema e n_{m_k} é a ordem do sub-modelo indexado por m_k . As matrizes A_{m_k} , B_{m_k} , C_{m_k} e D_{m_k} tem dimensões apropriadas e descrevem as dinâmicas afns, $w_k \in \mathbb{R}^{n_{m_k}}$, $w_k \sim N(0, Q_k)$ e $v_k \in \mathbb{R}^{n_y}$, com $v_k \sim N(0, R_k)$, são termos de ruído de processo e medida, respectivamente, considerados como sequencias de ruído branco descorrelacionados com média zero.

Para obter as estimativas dos estados do sistema o filtro IMM utiliza um conjunto de Filtros de Kalman operando em paralelo. Os filtros individuais são inicializados usando a mistura das estimativas de passos anteriores do filtro. A saída do algoritmo, a estimativa global do estado, é também uma mistura das estimativas dos filtros individuais.

Assim, o filtro IMM precisa de três itens básicos. O primeiro é um conjunto de Filtros de Kalman, um para cada M modelos ou modos de operação do sistema. O segundo é o vetor de probabilidades $\hat{p}^{(q)}(m_k) = \Pr(m_k = q | y_{1:k})$, que contém a probabilidade do modelo $m_k = q$ estar ativo no instante de tempo k , e o terceiro é a MPT. Desta forma, considerando o sistema (2.61) e que $\hat{x}_k^{(q)}$ e $\hat{P}_k^{(q)}$, $q \in \{1, 2, \dots, M\}$, sejam o vetor de estados e a matriz de covariância associada correspondendo ao filtro seguindo o modo $m_k = q$ no k -ésimo instante amostral e assumindo-se as seguintes condições iniciais:

$$\hat{p}^{(q)}(m_0), \hat{x}_0^{(q)}, \hat{P}_0^{(q)}, q \in \{1, 2, \dots, M\},$$

um conjunto de M filtros é usado para estimar os estados do sistema (2.61). Cada um dos filtros segue um modo diferente do sistema. Assim, o filtro IMM é dado pelas seis etapas seguintes:

i Predição das probabilidades dos modos:

$$\Pr(m_{k-1}|y_{1:k-1}) \xrightarrow{\text{Modelo da CM}} \Pr(m_k|y_{1:k-1}). \quad (2.62)$$

A probabilidade dos modos é estimada por predição da Cadeia de Markov através da seguinte expressão:

$$\bar{p}^{(q)}(m_k) = \sum_{j=1}^M \pi_{j,q} \hat{p}^{(j)}(m_{k-1}). \quad (2.63)$$

ii Mistura das estimativas:

$$p(x_{k-1}|m_{k-1}, y_{1:k-1}) \xrightarrow{\text{"Mistura" de estimativas}} p(x_{k-1}|m_k, y_{1:k-1}).$$

Nesta etapa a distribuição de x_{k-1} é atualizada a partir da atualização de m_k . Isso é feito através da mistura das estimativas e utiliza as seguintes expressões:

$$\underline{x}_{k-1}^{(q)} = \sum_{j=1}^M \frac{\pi_{j,q} \hat{p}^{(j)}(m_{k-1}) \hat{x}_{k-1}^{(j)}}{\bar{p}^{(q)}(m_k)}, \quad (2.64)$$

$$\underline{P}_{k-1}^{(q)} = \sum_{j=1}^M \frac{\pi_{j,q} \hat{p}^{(j)}(m_{k-1}) \left[\hat{P}_{k-1}^{(j)} + \left(\hat{x}_{k-1}^{(j)} - \underline{x}_{k-1}^{(q)} \right) \left(\hat{x}_{k-1}^{(j)} - \underline{x}_{k-1}^{(q)} \right)^T \right]}{\bar{p}^{(q)}(m_k)}. \quad (2.65)$$

iii Predição das estimativas

$$p(x_{k-1}|m_k, y_{1:k-1}) \xrightarrow{\text{Predição do FK}} p(x_k|m_k, y_{1:k-1}).$$

Etapla responsável por propagar a estimativa x_{k-1} por predição. Para isso utiliza-se a etapa de predição do Filtro de Kalman:

$$\begin{aligned} \bar{x}_k^{(q)} &= A_{m_{k-1}} \underline{x}_{k-1}^{(q)} + B_{m_{k-1}} u_{k-1}, \\ \bar{P}_k^{(q)} &= A_{m_{k-1}} \underline{P}_{k-1}^{(q)} A_{m_{k-1}}^T + Q_{m_{k-1}}. \end{aligned}$$

iv Correção das estimativas

$$p(x_k|m_k, y_{1:k-1}) \xrightarrow{\text{Correção do FK}} p(x_k|m_k, y_{1:k}).$$

Nesta etapa é realizada a correção da estimativa x_k por meio da nova medição:

$$\begin{aligned} K_k^{(q)} &= \bar{P}_k^{(q)} C_{m_k}^T \left(C_{m_k} \bar{P}_k^{(q)} C_{m_k}^T + R_{m_k} \right)^{-1}, \\ \hat{x}_k^{(q)} &= \bar{x}_k^{(q)} + K_k^{(q)} \left(y_k - C_{m_k} \bar{x}_k^{(q)} - D_{m_k} u_k \right), \\ \hat{P}_k^{(q)} &= (\mathbb{I} - K_k^{(q)} C_{m_k}) \bar{P}_k^{(q)} (\mathbb{I} - K_k^{(q)} C_{m_k})^T + \left(K_k^{(q)} \right) R_{m_k} \left(K_k^{(q)} \right)^T, \end{aligned}$$

em que \mathbb{I} é a matriz identidade de dimensões apropriadas.

v Correção das probabilidades dos modos

$$\Pr(m_k | y_{1:k-1}) \xrightarrow{\text{Teorema de Bayes}} \Pr(m_k | y_{1:k}).$$

Etapa que trata da correção das probabilidades dos modos por meio da nova medição. É utilizado o teorema de Bayes para realizar esta correção:

For $q=1, 2, \dots, M$

$$\begin{aligned} \vartheta_k^{(q)} &= y_k - C_{m_k} \bar{x}_k^{(q)} - D_{m_k} u_k, \\ \Sigma_{\vartheta_k^{(q)}} &= C_{m_k} \bar{P}_k^{(q)} C_{m_k}^T + R_{m_k}, \\ \hat{p}^{(q)}(m_k) &= \frac{\bar{p}^{(q)}(m_k)}{c_q \det \left(\Sigma_{\vartheta_k^{(q)}} \right)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} \vartheta_k^{(q)T} \left(\Sigma_{\vartheta_k^{(q)}} \right)^{-1} \vartheta_k^{(q)} \right\}, \end{aligned}$$

end For

$$\hat{p}(m_k) = [\hat{p}^{(1)}(m_k), \dots, \hat{p}^{(M)}(m_k)]^T \left(\frac{1}{\sum_{j=1}^M \hat{p}^{(j)}(m_k)} \right),$$

em que c_q é uma constante de normalização que não precisa ser calculada.

vi Geração das saídas: Esta última etapa é a responsável pela geração das saídas do filtro:

$$\begin{aligned} \hat{x}_k &= \sum_{q=1}^M \hat{p}^{(q)}(m_k) \hat{x}_k^{(q)}, \\ \hat{P}_k &= \sum_{q=1}^M \hat{p}^{(q)}(m_k) \left[\hat{P}_k^{(q)} + \left(\hat{x}_k^{(q)} - \hat{x}_k \right) \left(\hat{x}_k^{(q)} - \hat{x}_k \right)^T \right]. \end{aligned}$$

Na Figura 2.5 é apresentado o diagrama de blocos mostrando a interação entre as diferentes etapas do algoritmo. Em seguida, no Exemplo 2.3, é apresentado uma simulação para ilustrar o desempenho do filtro IMM.

Exemplo 2.3: Neste exemplo, para demonstrar a eficiência do algoritmo IMM é considerado a simulação de um robô planetário (do inglês *planetary rover*) descrito em [5]. Será considerado apenas o subsistema constituído de um motor e uma roda. Neste subsistema, uma falha intermitente causa o aumento do atrito da roda. Este subsistema pode ser modelado como um sistema híbrido com dois modos de operação. No modo 1 a roda opera

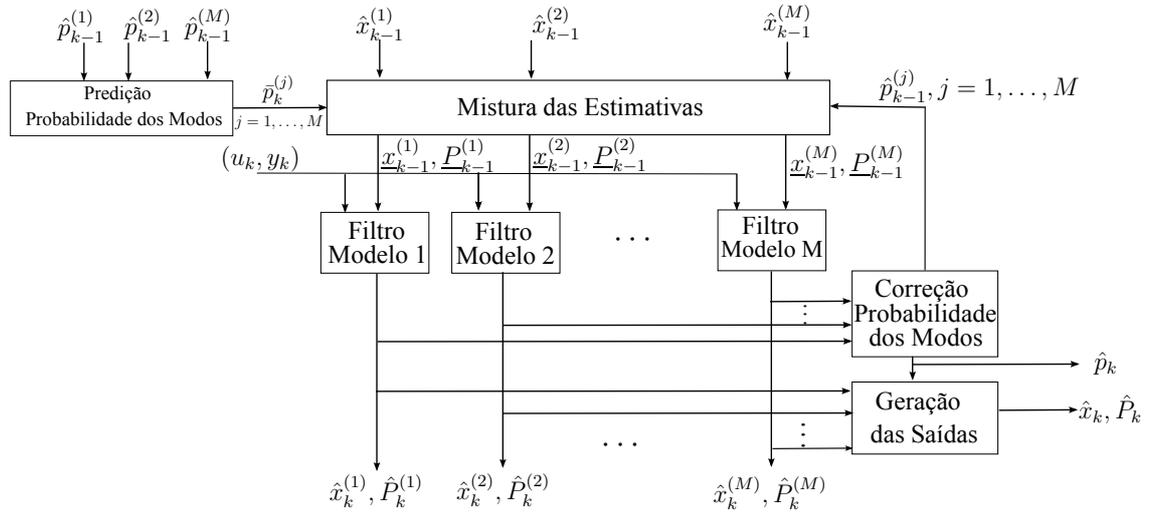


Figura 2.5: Diagrama do filtro IMM mostrando a interação entre as diferentes etapas do algoritmo.

normalmente, enquanto que no modo 2 o sistema opera com problema onde o atrito na roda é maior.

Neste exemplo, o estado contínuo é $[i \ \dot{\theta}]^T$, onde i é a corrente no motor e $\dot{\theta}$ é a velocidade angular da roda. A saída observada no sistema y é a velocidade da roda medida através de um encoder e a entrada do sistema u é a voltagem aplicada ao sistema.

Esses dois modos de operação podem ser descritos através de um modelo no espaço de estados com as seguintes matrizes:

$$A_1 = \begin{bmatrix} -0,0044 & -0,0203 \\ 0,0366 & 0,1665 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0,92 \\ 0,81 \end{bmatrix} \quad (2.66)$$

$$A_2 = \begin{bmatrix} -0,0032 & -0,0142 \\ 0,0256 & 0,1106 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0,93 \\ 0,71 \end{bmatrix} \quad (2.67)$$

$$C_1 = C_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad D_1 = D_2 = 0. \quad (2.68)$$

A probabilidade de ocorrência do modo 2 é diferente dependendo se a roda está sendo impulsionada para a frente ou para trás. Contudo, neste exemplo, para melhor compreensão dos resultados será considerado que durante a primeira metade da simulação o sistema trabalha sem falhas e durante a segunda metade da simulação ocorre o problema.

O sistema foi simulado considerando a aplicação de uma tensão de entrada constante de 1V. Ao sinal de saída foi adicionado um sinal de ruído branco de média zero e variância de 0,016. O filtro IMM descrito anteriormente foi utilizado para a estimação dos estados contínuos e discreto do sistema. Os resultados obtidos são apresentados nas figuras 2.6 e 2.7. Na Figura 2.6 é mostrado as probabilidades estimadas de cada um dos modos. Essas probabilidades são utilizadas para determinar qual o modo ativo a cada instante de tempo. O

modo com mais alta probabilidade estimada no instante de tempo k é assumido como o modo ativo. Por exemplo, no instante de tempo $k = 341$ o modo 1 tem a probabilidade estimada de 0,2391, enquanto que o modo 2 tem a probabilidade de 0,7609. Assim, neste instante de tempo é assumido que o sistema está operando no modo 2. Este procedimento repetido para todos os pontos amostrados permite obter uma estimativa dos instantes de chaveamento do sistema, conforme mostrado no último gráfico da Figura 2.7. Esta figura também apresenta os resultados obtidos para a estimação dos estados contínuos do sistema. Observe que o filtro IMM foi capaz de identificar corretamente os modos do sistema em praticamente toda a simulação e que o algoritmo estima adequadamente os estados contínuos.

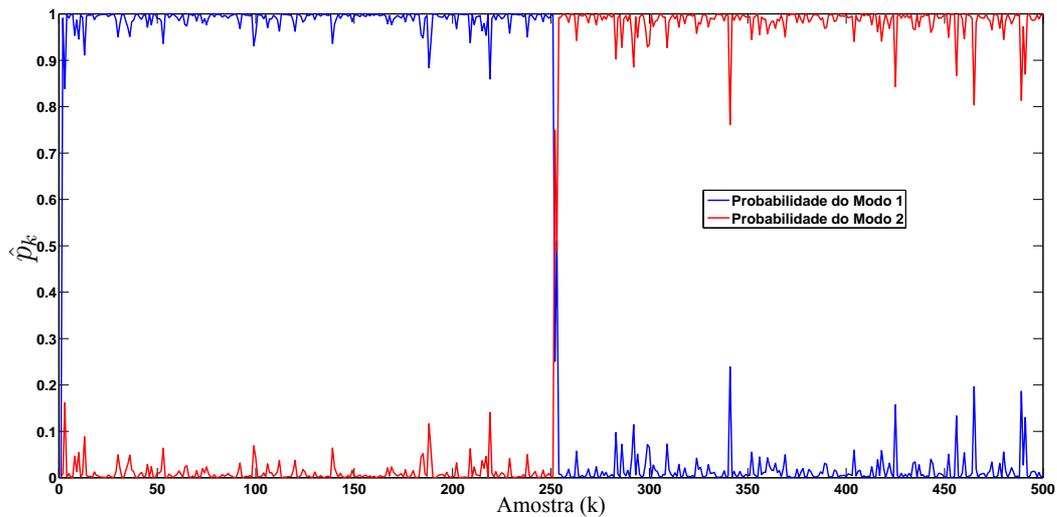


Figura 2.6: Probabilidades dos modos estimadas pelo filtro IMM.

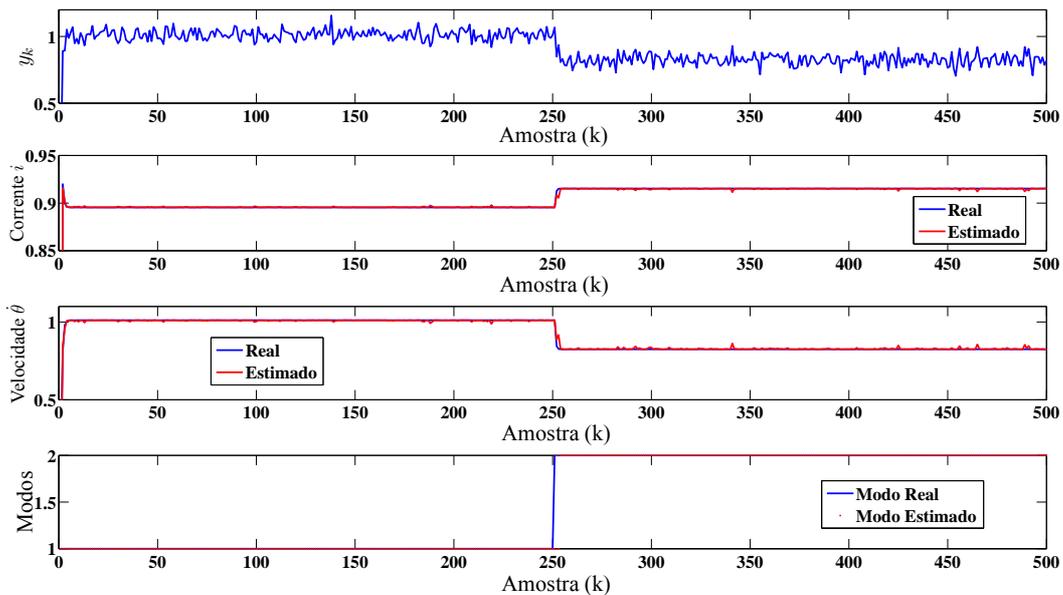


Figura 2.7: Sinais utilizados e obtidos na estimação de estados utilizando o filtro IMM.

2.4 VALIDAÇÃO DOS MODELOS

Após a aplicação dos algoritmos de identificação que serão descritos no Capítulo 3 chega-se ao produto final que é o modelo matemático do sistema híbrido. A pergunta que naturalmente surge é: O modelo é suficientemente bom?

As técnicas de validação do modelo oferecem suporte para responder essa pergunta. Trata-se de um conjunto de ferramentas para avaliar a forma como o modelo se refere aos dados observados. Isso permite avaliar se ele incorpora, ou não, as características requeridas do sistema. Neste sentido, modelos que apresentam comportamentos deficientes devem ser rejeitados, enquanto que os que apresentam bom desempenho proporcionam um certo grau de confiança.

Assim, o objetivo da validação dos modelos nunca é aceitar um modelo como sendo verdadeiro ou correto e sim descartar os modelos incorretos em relação alguns aspectos [66]. No entanto, é importante ressaltar o caráter subjetivo na validação do modelo. Ferramentas de validação de modelo só devem ser vistas como assessores para o usuário. É este usuário que chega a conclusão final sobre aceitar ou não um determinado modelo [66].

Na identificação de sistemas, a entidade mais natural com que o modelo deve ser comparado são os próprios dados [2], [66]. Neste caso, o interesse é determinar se o modelo é capaz de reproduzir os dados observados ao longo do tempo [2]. Trata-se de um procedimento relativamente simples que permite avaliar a qualidade do modelo através da quantificação da concordância entre o modelo e os dados medidos do sistema.

Segundo [2], um cuidado básico a se tomar é o de não usar no processo de validação os mesmos dados utilizados no procedimento de identificação do modelo. Assim, o procedimento ideal é a utilização de dois conjuntos de dados independentes, um deles usado para a identificação do modelo e outro para a validação. Esse cuidado deve-se a necessidade de saber o quão geral é o modelo obtido.

Contudo, no contexto de identificação de sistemas dinâmicos híbridos chaveados este procedimento de validação não é trivial. Essa dificuldade deve-se ao fato de não se ter garantias que o sinal de chaveamento do sistema híbrido será o mesmo entre dois experimentos independentes. Assim, não é possível realizar a comparação entre a saída real do sistema e a saída simulada pelo modelo. Os principais trabalhos existentes na literatura para identificação deste tipo de sistema no espaço de estados (descritos na seção 1.3.1.2) não abordam esse assunto.

Neste trabalho, para resolver este problema é assumido, apenas no processo de validação dos modelos, que o sinal de chaveamento do sistema é conhecido. Assim, é possível realizar a comparação entre a saída real e a saída obtida com o modelo identificado em um conjunto de dados diferente daquele utilizado no processo de identificação dos modelos locais. Destaca-se que esse procedimento não é restritivo, uma vez que a intenção é apenas verificar

se os modelos locais constituintes do sistema híbrido foram corretamente identificados.

No processo de validação, muitas vezes são utilizados índices de qualidade para quantificar a qualidade do ajuste do modelo identificado aos dados experimentais do sistema. Um índice de qualidade adequado e que será usado nesta tese é o valor de VAF (do inglês, *Variance-Accounted-For*). Este é definido como [39]:

$$VAF = \max \left\{ 1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}, 0 \right\} \times 100\%, \quad (2.69)$$

em que \hat{y} denota o sinal de saída obtido através da simulação do modelo identificado e y é o valor do sinal medido.

O erro médio quadrático (MSE, do inglês, *Mean Squared Error*), definido em (2.70) será utilizado como uma medida de desempenho adicional em alguns momentos deste trabalho.

$$\text{MSE} \triangleq \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2, \quad (2.70)$$

em que N denota a quantidade de dados amostrados.

Para fins de análise, quanto mais próximo de 100% for o valor de VAF mais próximos estão o sinal medido y e o sinal estimado \hat{y} . Por outro lado, quanto menor for o valor do MSE, maior será a eficiência do modelo na recuperação das características dinâmicas do sistema.

3 ABORDAGENS DE IDENTIFICAÇÃO PROPOSTAS

3.1 INTRODUÇÃO

Este capítulo representa o cerne desta tese, onde são apresentados os novos algoritmos para identificação off-line de sistemas dinâmicos híbridos e suas inovações com relação às abordagens tradicionais encontradas na literatura. A fundamentação teórica necessária para o desenvolvimento destes algoritmos foi apresentada no Capítulo 2.

Esses algoritmos são resultados da proposição de uma nova estrutura (*framework*) de identificação de sistemas híbridos. Esta nova estrutura aborda o problema de identificação dividindo-o em diferentes etapas. Em cada etapa é explorado um conceito de identificação de sistemas dinâmicos lineares, clustering, classificação de dados ou de filtragem híbrida. A estrutura proposta apresenta como importante característica o conceito modular que permite a utilização de diferentes métodos para implementação de cada um dos conceitos mencionados. Assim, é possível escolher cada um dos métodos em função da aplicação de modo a obter os melhores resultados. O fluxograma completo desta metodologia está ilustrado na Figura 3.1. A proposição desta nova estrutura para a identificação de sistemas híbridos constitui a principal contribuição deste trabalho.

A principal diferença entre a metodologia proposta e as abordagens já existentes na literatura é o fato da identificação dos modelos locais ser realizada antes da estimação dos instantes de chaveamento do sistema. Conforme será mostrado neste capítulo, em cenários críticos, onde ocorre muitas transições entre os diferentes modos e o sistema está corrompido com elevados níveis de ruído, o conhecimento das estimativas dos modelos locais possibilita a redução do erro de classificação dos pontos através da utilização de técnicas clássicas de filtragem híbrida.

O fluxograma apresentado na Figura 3.1 resume a metodologia final desenvolvida. Contudo, esse desenvolvimento foi realizado de forma gradual. Consequentemente, os primeiros algoritmos obtidos são baseados em apenas algumas etapas desta metodologia. Desta forma, a sequência dos resultados aqui apresentados segue a ordem cronológica de seu desenvolvimento, tornando fácil perceber como trabalhos posteriores encontraram seus alicerces naqueles que os antecederam até chegar na metodologia completa apresentada na Figura 3.1.

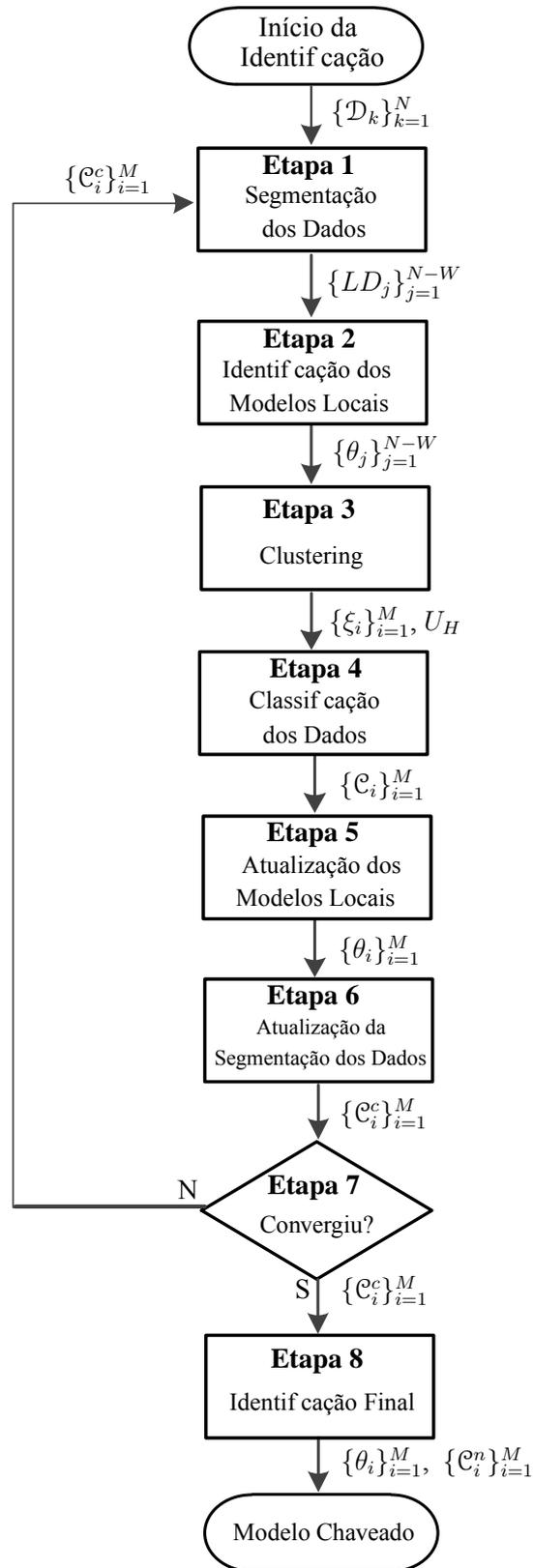


Figura 3.1: Fluxograma da metodologia proposta para identificação de sistemas dinâmicos híbridos chaveados com modelos expressos no espaço de estados. As caixas correspondem às etapas do procedimento, e os rótulos nas bordas descrevem as entradas e as saídas de cada etapa.

O primeiro algoritmo proposto, e publicado em [121], introduziu o conceito de identificar os modelos locais antes da estimação dos instantes de chaveamento do sistema. Para isso, o algoritmo é composto pelas etapas 1, 2, 3, 4 e 8. A etapa de classificação dos dados (etapa 4), é baseada apenas nas informações fornecidas pelo algoritmo de clustering. O segundo algoritmo é uma versão mais eficiente do ponto de vista computacional, no sentido que fornece uma estimativa do sistema híbrido em menor tempo. Para isso, o algoritmo é composto apenas das etapas 1 a 3, e a representação das matrizes dos modelos locais são obtidas a partir das estimativas fornecidas pelo algoritmo de clustering. Esta nova versão do algoritmo foi publicada em [122]. A principal desvantagem desses dois algoritmos é a elevada sensibilidade à outliers.

Esses dois algoritmos motivaram o desenvolvimento dos demais. O terceiro algoritmo é uma evolução do primeiro e conseguiu melhorar as estimativas finais dos modelos locais através da introdução de um passo de atualização da segmentação dos dados baseada nas estimativas parciais dos modelos locais. Essa atualização da segmentação dos dados é realizada utilizando-se uma técnica de filtragem híbrida. Este algoritmo é composto das etapas 1 a 6 e etapa 8.

O quarto algoritmo proposto foi inspirado nas ideias do algoritmo de *Expectation Maximization* [123]. Nesta abordagem o problema de identificação consiste em maximizar a FDP a posteriori conjunta dos parâmetros e da segmentação. Essa maximização é realizada através de iterações alternadas buscando-se maximizar a FDP a posteriori dos parâmetros contínuos seguida da maximização da FDP a posteriori dos parâmetros discretos. O processo termina quando os parâmetros do sistema não sofrem mais alterações significativas entre duas iterações consecutivas. Esta metodologia é exatamente a apresentada na Figura 3.1.

No quinto algoritmo proposto, a metodologia desenvolvida para a identificação de sistemas dinâmicos híbridos chaveados é adaptada para a identificação de sistemas híbridos do tipo PWA. Essa adaptação constitui uma importante contribuição à literatura, uma vez que, ao melhor conhecimento do autor, o único trabalho a abordar a identificação de sistemas PWA no espaço de estados foi publicado em [75]. Contudo, esta abordagem prévia requer um número suficientemente grande de transições entre os diferentes modos para que seja possível representar todas as matrizes dos sub-modelos na mesma base de estados.

O último algoritmo desenvolvido possui a mesma estrutura do quarto algoritmo. No entanto, são realizadas algumas modificações de modo a obter menor custo computacional.

Um importante aspecto a ser considerado no desenvolvimento de abordagens de identificação é a análise da convergência do algoritmo proposto. Em um cenário ideal, deve-se fornecer provas matemáticas, ou pelo menos um conjunto de condições, que garanta a convergência do algoritmo para uma solução válida. No entanto, a complexidade teórica inerente ao processo de identificação de sistemas dinâmicos híbridos torna essa tarefa muito difícil. De fato, este é um problema que continua em aberto na literatura [64]. Observou-se que

as principais abordagens encontradas na literatura não abordam o problema da convergência dos algoritmos propostos. As metodologias propostas neste trabalho também não fornecem prova de convergência, e o desenvolvimento de bases teóricas formais que garantam essa convergência está fora do escopo do trabalho. No entanto, a convergência dos parâmetros será avaliada empiricamente por meio de simulações numéricas.

O restante deste capítulo está organizado como segue. Inicialmente será realizada a formulação matemática do problema de identificação híbrida. Em seguida os algoritmos e os principais resultados de simulações serão descritos seguindo a ordem cronológica de seu desenvolvimento.

3.2 DEFINIÇÃO DO PROBLEMA

Neste trabalho, será considerado o problema de identificação de sistemas chaveados lineares de tempo discreto. Esses são definidos como uma coleção de modelos lineares, conectados por chaveamentos que são indexados por uma variável discreta adicional, chamada de estado discreto [7]. Estes sistemas são descritos pelo seguinte modelo no espaço de estados:

$$\begin{cases} x_{k+1} &= A_{m_k}x_k + B_{m_k}u_k + w_k, \\ y_k &= C_{m_k}x_k + D_{m_k}u_k + v_k, \end{cases} \quad (3.1)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto, e M é o número de submodelos. Os vetores $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos, as entradas e saídas do sistema e n é a ordem dos modelos. As matrizes A_{m_k} , B_{m_k} , C_{m_k} e D_{m_k} tem dimensões apropriadas e descrevem as dinâmicas dos sistemas locais, $w_k \in \mathbb{R}^n$, $w_k \sim N(0, Q_k)$ e $v_k \in \mathbb{R}^{n_y}$, com $v_k \sim N(0, R_k)$, são termos de ruído de processo e medida, respectivamente, considerados como sequências de ruído branco descorrelacionados com média zero.

Problema 3.1: Assumindo-se que o conjunto de dados $\mathcal{S} = \{(u_k, y_k)\}_{k=1}^N$ é gerado pelo sistema (3.1), deve-se determinar:

1. O número de sub-modelos M ;
2. A ordem n dos modelos locais,
3. Uma realização $(\hat{A}_{m_k}, \hat{B}_{m_k}, \hat{C}_{m_k}, \hat{D}_{m_k})$ de cada modo local;
4. As estimativas dos instantes de chaveamento.

Hipóteses: Para resolução deste problema serão assumidas as seguintes considerações:

1. Somente um modelo local está ativo a cada instante de tempo;

2. O sistema permanece em cada modo por pelo menos um mínimo período de tempo. Esse tempo é referido na literatura como tempo de permanência (*dwelling time* - τ_{dwell}). Note que na prática, a hipótese do mínimo tempo de permanência não é excessivamente restritiva, uma vez que comutações que ocorrem muito rapidamente podem resultar em sérios problemas de estabilidade [25];
3. Cada um dos modos é persistentemente excitado pela sequência de entrada u_k ;
4. Os modelos locais são assumidamente estáveis e observáveis;

3.3 ALGORITMO 1

A principal dificuldade na solução do Problema 3.1 reside no fato de que o estado discreto, o estado contínuo e os parâmetros dos diferentes sub-modelos são acoplados e são, infelizmente, desconhecidos. Conforme apresentado na Seção 1.3, a solução deste problema para modelos no espaço de estados ainda tem recebido pouca atenção e as principais abordagens desenvolvidas [71], [72], [73], [39], [50], [124], [25], [38], basicamente, dividem o problema em duas etapas sequenciais;

- Etapa 1: Identificação dos instantes de chaveamento do sistema;
- Etapa 2: Identificação dos modelos locais.

Este trabalho propõe o desenvolvimento de uma abordagem alternativa às existentes, na qual, essencialmente, a identificação dos modelos locais é realizada antes da estimação dos instantes de chaveamento do sistema, ou seja, propõe a inversão das duas etapas mencionadas acima. No entanto, a pergunta a ser respondida neste caso é: Como realizar a identificação dos modelos locais sem o conhecimento da quantidade destes sub-modelos e a qual modo pertence cada um dos dados amostrados?

A solução proposta foi a divisão dos dados amostrados em pequenos subconjuntos de dados. Para cada subconjunto formado é aplicado um método de identificação em subespaços. O conjunto de modelos estimados é então enviado a um algoritmo de clustering que realiza o agrupamento dos modelos semelhantes e permite identificar a quantidade de modos existentes no sistema. Esse processo permite ainda a determinação de qual modelo local gerou cada um dos dados amostrados e, conseqüentemente, possibilita a estimação dos instantes de chaveamento do sistema. O conhecimento a respeito dos instantes de chaveamento é utilizado posteriormente para refinar as estimativas dos modelos locais identificados.

É importante destacar que, esta abordagem foi inspirada nas propostas de [32] e [35]. No entanto, nestes trabalhos os autores abordam o problema de identificação de sistemas PWA MISO com modelos expressos na forma de entrada e saída. Desta forma, o problema

de identificação se torna relativamente mais simples do que a identificação de modelos no espaço de estados, uma vez que os autores podem explorar o fato do vetor de regressores (formado pelos dados de entrada e saída) ser totalmente disponível a priori e o fato da função de mapeamento que define estes sistemas ser localmente linear. Isso significa que, pode-se facilmente agrupar os dados com maior probabilidade de terem sido gerados pelo mesmo sub-modelo, uma vez que os vetores de regressores gerados pelo mesmo modelo local estarão em regiões próximas do plano.

Portanto, o presente trabalho estende essas abordagens para sistemas MIMO no espaço de estados, onde a principal dificuldade reside no fato que o regressor não está totalmente disponível, portanto não é possível realizar o particionamento direto do espaço de entrada/estados como nos trabalhos de [32] e [35].

A solução encontrada para contornar esta dificuldade foi a divisão dos dados no tempo. Essa divisão inicial não segue nenhum critério de classificação. A separação dos dados nos diferentes modos do sistema é realizada somente após a estimação dos modelos locais. Para isso, utiliza-se um algoritmo de clustering nos vetores dos parâmetros de Markov de cada modelo identificado. Este algoritmo de clustering possibilita identificar a quantidade de modos do sistema e realizar a classificação dos pontos nos diferentes modos identificados. Após a separação dos dados um novo processo de identificação é realizado para a estimação final dos modelos locais.

O algoritmo proposto pode ser dividido em cinco etapas. Essas etapas estão ilustradas no diagrama didático apresentado na Figura 3.2 e serão descritas detalhadamente a seguir.

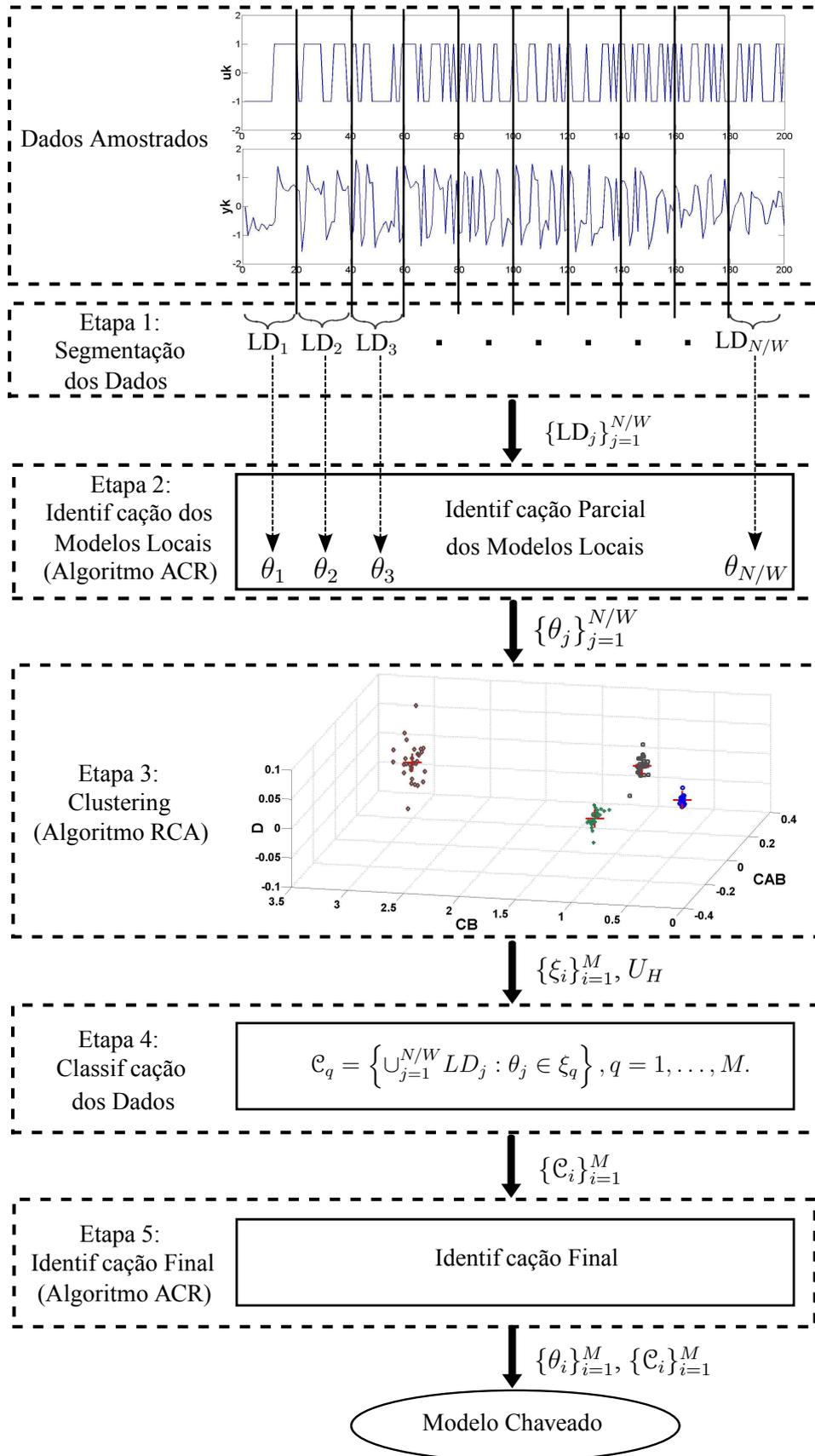


Figura 3.2: Diagrama do Algoritmo 1. Os rótulos ao lado das setas representam as saídas de cada etapa.

Etapa 1: Segmentação dos Dados:

O primeiro passo da metodologia é a formação de pequenos conjuntos de dados que serão utilizados para identificar os modelos locais. Esses pequenos conjuntos serão denominados conjuntos de dados locais (LD) e são formados dividindo-se o conjunto de dados amostrados no tempo. Para entender esse processo, considere que o conjunto total de dados amostrados é representado pelo seguinte conjunto:

$$\mathcal{D} = \{(u_k, y_k) : k \in \mathcal{T}\} \quad \text{para} \quad \mathcal{T} = \{k\}_{k=1}^N, \quad (3.2)$$

em que \mathcal{T} denota um horizonte de tempo e N é o número de dados amostrados.

A divisão dos dados no tempo consiste em dividir o conjunto \mathcal{T} em N/W conjuntos sequenciais menores $\{\mathcal{T}_j\}_{j=1}^{N/W}$ contendo W elementos, de forma que:

$$\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \quad \forall j \neq q. \quad (3.3)$$

Os índices de tempo em cada um dos conjuntos \mathcal{T}_j são utilizados para formação dos pequenos conjuntos de dados $\{\text{LD}_j\}_{j=1}^{N/W}$, de forma que:

$$\text{LD}_j = \{(u_q, y_q) \in \mathcal{D} : q \in \mathcal{T}_j\}. \quad (3.4)$$

Este processo está ilustrado graficamente na Figura 3.2. Nesta figura, os dois primeiros gráficos referem-se aos dados amostrados de entrada e saída de um sistema. As linhas verticais nestes gráficos indicam as divisões realizadas e, na sequência, é mostrado os pequenos conjuntos de dados formados.

Neste processo de divisão, alguns LDs serão formados somente por dados gerados por um único modelo local. Estes conjuntos serão denominados de conjuntos de dados locais puros. Por outro lado, alguns conjuntos poderão conter dados gerados por diferentes sub-modelos e serão chamados de conjunto de dados locais mistos.

Uma vez que os LDs mistos contém dados gerados por diferentes modelos locais, a aplicação de uma técnica de identificação nestes conjuntos resulta em modelos que não são semelhantes a nenhum sub-modelo do sistema híbrido. Desta forma, estes modelos, quando enviados ao algoritmo de clustering para segmentação serão considerados outliers. Obviamente, a formação de muitos conjuntos mistos prejudicará o desempenho do algoritmo de clustering e o efeito será a estimação de modelos menos precisos.

Desta forma, a definição do tamanho W na formação dos LDs tem um papel fundamental no algoritmo de identificação. Esses conjuntos devem conter uma quantidade de dados suficiente (D_{suf}) para que seja possível realizar a identificação dos modelos e, ao mesmo tempo, ter uma duração menor ou igual ao tempo de permanência do sistema em cada modo, ou seja, $D_{\text{suf}} \leq W \leq \tau_{\text{dwell}}$. Observe que, se a restrição $W \leq \tau_{\text{dwell}}$ não for atendida, poderá ocorrer apenas a formação de LDs mistos e, neste caso, não será possível identificar corretamente os modelos locais do sistema.

A definição do parâmetro W é um problema comum a todas abordagens que utilizam métodos de identificação em subespaços para identificação de sistemas híbridos [74], [75], [71], [73]. Ao melhor conhecimento do autor, este problema permanece em aberto na literatura. Assim, estudos teóricos para determinar valores necessários e suficientes para este parâmetro não serão discutidos neste trabalho.

Etapa 2: Identificação dos Modelos Locais:

Esta etapa é a responsável pela identificação dos modelos locais. Para isso, para cada LD_j formado é identificado um modelo j através de uma técnica de identificação de sistemas que forneça uma representação do modelo na forma de espaço de estados (A_j, B_j, C_j, D_j) .

Conforme já mencionado, para esta tarefa pode-se utilizar qualquer técnica de identificação que forneça uma representação do sistema na forma de espaço de estados. Neste trabalho, optou-se pela técnica de identificação em subespaços apresentada na seção 2.1.1. Este algoritmo foi combinado com a abordagem apresentada na seção 2.1.2 para identificação das matrizes dos modelos locais na forma canônica e na mesma base de estados. Esse é um importante aspecto do presente algoritmo, pois mantendo o controle sobre a base de estados na qual as matrizes são calculadas é possível associar as matrizes dos diferentes sistemas, como será descrito na próxima fase do algoritmo.

Cada modelo identificado constituirá uma observação que será enviada ao algoritmo de cluster para segmentação. Assim, para representar cada modelo será utilizado o seu vetor de parâmetros de Markov θ_j :

$$\theta_j = \text{vec} \left(\begin{bmatrix} C_j A_j^{p-2} B_j & \cdots & C_j B_j & D_j \end{bmatrix} \right) \in \mathbb{R}^{n_y \cdot n_u \cdot p \times 1}, \quad (3.5)$$

em que $\text{vec}(\bullet)$ é o operador de vetorização e p o parâmetro que define a quantidade de parâmetros de Markov utilizado na representação do sistema.

O vetor de parâmetros de Markov consiste de todos os parâmetros monitorados no modelo. De acordo com a complexidade da aplicação, a dimensão deste vetor pode ser reduzida (redução ou seleção dos parâmetros de Markov) ou aumentada [77]. Uma vantagem adicional no uso dos parâmetros de Markov, é a possibilidade de estimação de sistemas de diferentes ordens mantendo-se a mesma dimensão dos vetores θ_j .

Embora a quantidade de parâmetros de Markov possa ser ajustada de acordo com a complexidade da aplicação, existe um valor mínimo teórico para essa variável que é suficiente para caracterizar completamente o sistema. Segundo [125], esse valor mínimo deve ser igual a ordem do sistema mais um. Desta forma, tem-se que:

$$p = n + 1. \quad (3.6)$$

Assim, se a ordem dos modelos locais for conhecida, os parâmetros que caracterizam cada subsistema identificado são codificados por (3.5) com o parâmetro p escolhido como (3.6).

Fase 3: Clustering:

O objetivo desta etapa é agrupar os vetores dos parâmetros de Markov (θ_j) estimados na etapa anterior em vários clusters utilizando uma técnica de agrupamento de dados. Espera-se que o número de clusters identificados seja o número M de sub-modelos do sistema.

Para esse propósito, em princípio, pode-se utilizar qualquer algoritmo de clustering [107], [126], [37]. No entanto, o desempenho da maioria dos algoritmos clássicos é frequentemente comprometido pela existência de outliers e por problemas de inicialização que podem levar a solução a mínimos locais. Para minimizar esses problemas, neste trabalho optou-se pelo RCA apresentado na seção 2.2. Este algoritmo apresenta baixa sensibilidade a inicialização e permite identificar o número de clusters mesmo no cenário onde os dados estão contaminados com elevado nível de ruído. Ao melhor conhecimento do autor, esse algoritmo nunca foi utilizado no contexto de identificação de sistemas dinâmicos híbridos.

Fase 4: Classificação dos Dados:

Uma vez que o número de modelos locais foi determinado é possível obter uma representação destes modelos no espaço de estados. Para isso, pode-se utilizar duas diferentes abordagens. A primeira, utiliza informações fornecidas pelo algoritmo de clustering para realizar a classificação dos dados seguida de um novo processo de identificação. Este novo processo de identificação é realizado com uma quantidade maior de dados. A segunda abordagem obtém uma representação das matrizes dos modelos locais do sistema a partir dos centros dos clusters identificados pelo algoritmo de clustering.

Neste trabalho, serão avaliadas essas duas possibilidades. A primeira abordagem será explorada no Algoritmo 1, enquanto que a segunda abordagem será apresentada no Algoritmo 2. Em favor da primeira abordagem, pode-se argumentar que as estimativas provenientes dos centros dos clusters podem ser imprecisas [127], especialmente na existência de muitas transições entre os modos do sistema, o que resulta no aparecimento de muitos outliers. Adicionalmente, deve-se observar que os dados utilizados na segmentação são provenientes da identificação em uma pequena janela de tempo, com poucos dados, de forma que o efeito dos ruídos de medida/processo sobre os modelos parciais identificados pode ser significativo, comprometendo a acurácia destes modelos.

Assim, é importante identificar quais são os pontos do conjunto de dados amostrados \mathcal{D} que foram gerados pelo mesmo sub-modelo. Isso é um processo de classificação, onde os dados disponíveis são classificados em clusters $\{\mathcal{C}_i\}_{i=1}^M$ de forma que $(u_k, y_k) \in \mathcal{C}_i$ se e somente se (u_k, y_k) é atribuído ao i -ésimo modo.

Para realizar esta tarefa é utilizada a matriz U_H (*constrained fuzzy H-partition matrix*) fornecida pelo RCA, que indica o grau no qual o ponto θ_j pertence ao cluster q . Uma discussão detalhada desta matriz foi realizada no exemplo 2.2 do Capítulo 2. Os clusters

identificados pelo RCA são representados por $\{\xi_i\}_{i=1}^M$ e definidos como:

$$\xi_q = \left\{ \bigcup_{j=1}^{N/W} \theta_j : u_{q,j} > u_{i,j} \forall i \in (1, \dots, M) \right\} \quad (3.7)$$

Cada ponto θ_j é atribuído ao cluster no qual ele apresenta maior grau de pertinência. Desta forma, como cada sub-modelo θ_j corresponde a um subconjunto de dados LD_j , a clusterização dos sub-modelos a um mesmo modo q leva a uma clusterização dos subconjuntos de dados que correspondem ao mesmo modo.

Assim, é possível criar conjuntos de dados clusterizados correspondentes ao mesmo modo q . Esses conjuntos serão denotados por $\{\mathcal{C}_i\}_{i=1}^M$, e a classificação pode ser definida como:

$$\mathcal{C}_q = \left\{ \bigcup_{j=1}^{N/W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M. \quad (3.8)$$

Com isto, o conjunto de dados amostrados \mathcal{D} pode ser representado por $\mathcal{D} = \bigcup_{q=1}^M \{\mathcal{C}_q\}$. Observe que a primeira partição de dados $\mathcal{D} = \bigcup_{j=1}^{N/W} \{LD_j\}$ foi feita de forma bastante arbitrária e consiste em uma aproximação grosseira das reais janelas de tempo para cada subsistema. A clusterização proposta em (3.8) permite obter uma correção ou aproximação corrigida das janelas de tempo em que cada subsistema esteve ativo. As novas janelas são necessariamente maiores que as não corrigidas e assim, uma eventual identificação considerando as janelas corrigidas devem apresentar resultados com melhor proximidade aos sistemas originais.

Adicionalmente, deve ser notado que este procedimento permite obter uma estimativa dos instantes de chaveamento do sistema. Essas estimativas são obtidas por inspeção do conjunto de dados amostrados, ou seja, são obtidas procurando-se, no conjunto de dados \mathcal{D} e \mathcal{T} , pontos consecutivos que pertencem a modelos locais diferentes.

Fase 5: Identificação Final dos Modelos Locais:

Com os dados originais classificados, é possível realizar uma nova identificação contendo uma maior quantidade de dados. Para isso, utilizou-se a mesma abordagem de identificação em subespaços descrita anteriormente para a obtenção dos modelos finais na forma de espaço de estados.

A metodologia proposta está resumida no Algoritmo 1.

3.3.1 Resultados Numéricos

Nesta seção, será apresentada a avaliação da metodologia proposta em um exemplo numérico. Utilizou-se o software Matlab[®] para implementação do algoritmo e realização de todas as simulações apresentadas. O sistema híbrido escolhido foi o mesmo utilizado em

Algoritmo 1:

1: Segmentação dos Dados:

- a) Crie pequenos conjuntos de dados $\{LD_j\}_{j=1}^{N/W}$, de forma que $LD_j = \{(u_q, y_q) \in \mathcal{D} : q \in \mathcal{T}_j\}$, e \mathcal{T}_j é formado pela divisão do horizonte de tempo \mathcal{T} em conjuntos menores tais que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$

2: Identificação dos Modelos Locais:

- a) Determine θ_j para cada conjunto de dados em $\{LD_j\}_{j=1}^{N/W}$ usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.

3: Clustering:

- a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N/W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA apresentado na Seção 2.2

4: Classificação dos Dados:

- a) Forme os subconjuntos dos dados clusterizados $\{\mathcal{C}_q\}_{q=1}^M$ da seguinte forma:
$$\mathcal{C}_q = \left\{ \bigcup_{j=1}^{N/W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M.$$
- b) Estime os instantes de chaveamento do sistema procurando no conjunto de dados amostrados (u_k, y_k) pontos consecutivos que pertencem a diferentes modelos locais.

5: Identificação Final dos Modelos Locais:

- a) Identifique os modelos locais com os dados de $\{\mathcal{C}_q\}_{q=1}^M$, usando as metodologias descritas nas seções 2.1.1, 2.1.2 e 2.1.3
-

dois importantes trabalhos da literatura [39] e [25]. O sistema utilizado é composto de quatro modelos locais SISO de segunda ordem representados por:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 0,8 \\ -0,8 & 0,5 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0,4 \\ 0 \end{bmatrix}, & C_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & D_1 &= 0, \\ A_2 &= \begin{bmatrix} 0 & 0,5 \\ -0,5 & 0 \end{bmatrix}, & B_2 &= \begin{bmatrix} 1 \\ 0,5 \end{bmatrix}, & C_2 &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & D_2 &= 0, \\ A_3 &= \begin{bmatrix} 0,8 & 0 \\ 0 & -0,3 \end{bmatrix}, & B_3 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}, & C_3 &= \begin{bmatrix} 1 & 1 \end{bmatrix}, & D_3 &= 0, \\ A_4 &= \begin{bmatrix} 0 & 0,4 \\ -0,4 & 0 \end{bmatrix}, & B_4 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & C_4 &= \begin{bmatrix} 1 & 0,5 \end{bmatrix}, & D_4 &= 0. \end{aligned}$$

Este exemplo, embora simples, por tratar de sistemas SISO, apresenta desafios do ponto

de vista de identificação, uma vez que o sistema híbrido apresenta modelos locais com dinâmicas que são relativamente próximas, especialmente os modelos 2 e 4, que apresentam autovalores em $(\pm 0, 5j)$ e $(\pm 0, 4j)$, respectivamente.

Em todas as simulações realizadas o sistema foi excitado por um sinal PRBS (do inglês, *Pseudo Random Binary Sequence*) de média zero e variância unitária. Adicionou-se ao sinal de saída um ruído branco na proporção de $\text{SNR} = 35$ dB. O termo SNR (do inglês, *Signal to Noise Ratio*), é a relação sinal-ruído definida como:

$$\text{SNR} = 20 \log_{10} \frac{\text{var}(y_k)}{\text{var}(v_k)}, \quad (3.9)$$

em que $\text{var}(\bullet)$ indica a variância do sinal \bullet .

Nas simulações geraram-se 12000 pontos de dados de u_k e y_k , que são utilizados na identificação dos sistemas. Esses sinais estão mostrados nas Figuras 3.3b e 3.3c. No entanto, note que apenas um pequeno trecho do sinal de excitação é apresentado para facilitar a visualização. O sinal de chaveamento utilizado está mostrado na Figura 3.3a. É importante observar que, no sinal de chaveamento considerou-se a hipótese que os instantes das mudanças de modo ocorrem em instantes múltiplos da janela W utilizada. Essa restrição será retirada posteriormente.

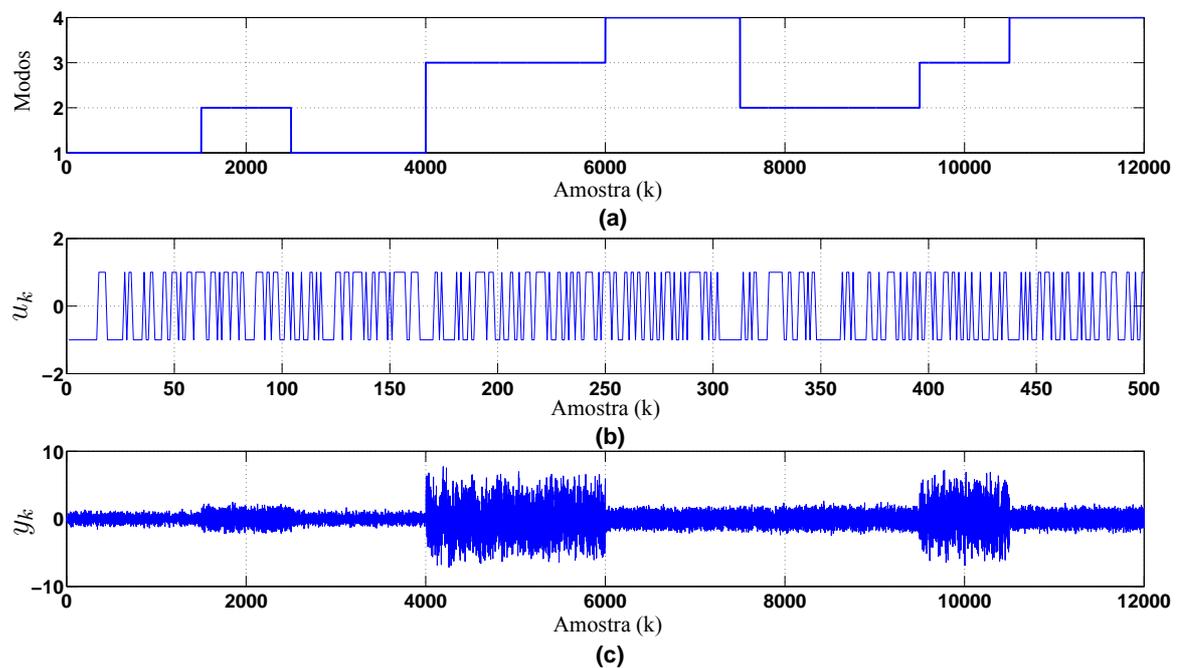


Figura 3.3: Sinais utilizados e obtidos com o sistema (3.1). (a) Sequencia de chaveamento utilizada na simulação. (b) Pequeno trecho do sinal PRBS aplicado ao sistema. (c) Sinal de saída utilizado para a identificação dos modelos.

Neste primeiro momento, a intenção é avaliar a eficiência da metodologia proposta para a identificação de sistemas dinâmicos híbridos. Assim, inicialmente será considerado que a

ordem dos modelos locais é conhecida a priori. Posteriormente, será avaliada a eficiência do algoritmo na estimação deste valor.

Os parâmetros necessários no algoritmo foram ajustados como segue: No algoritmo de identificação em subespaços utilizou-se o tamanho dos blocos linhas (i) igual a 10, o tamanho da janela (W) igual a 100 e a ordem real do sistema igual a 2. Desta forma, com base na Equação (3.6) determinou-se $p = 3$, ou seja, obteve-se uma representação 3D para o vetor dos parâmetros de Markov, isso facilitará a apresentação e visualização dos resultados. No RCA os valores da maior parte dos parâmetros foi mantido constante nas diferentes simulações realizadas neste trabalho. O único parâmetro alterado ao longo das simulações foi o parâmetro ϵ_0 . Assim, os valores utilizados foram $c_0 = 12$, $c_{\min} = 4$, $\Delta_c = 1$, $H = 10$, $\alpha = 0$, $\tau = 10$, $\eta_0 = 10^{-4}$, $k_0 = 5$. Nesta primeira simulação, foi utilizado para o parâmetro ϵ_0 uma seleção dinâmica onde a cada iteração seu valor era igual 5% do valor da cardinalidade robusta N_s (ver Seção 2.2 para definição desses parâmetros).

Para avaliar a eficiência do algoritmo observou-se a sua capacidade em detectar corretamente o número de modos, os instantes de chaveamento e a habilidade do modelo estimado em reconstruir a saída do sistema a partir do sinal de entrada real. Para isso utilizou-se o valor de VAF definido na equação (2.69) em um conjunto de dados diferente do utilizado para a identificação. Esse conjunto de dados será chamado de conjunto de dados de validação e é composto por 1200 pontos de u_k e y_k . O processo de validação dos modelos foi introduzido na seção 2.4.

Os resultados obtidos estão apresentados nas Figuras 3.4 a 3.6. A Figura 3.4 mostra o resultado obtido com o algoritmo de clustering. Observe que, nesta figura o algoritmo foi capaz de identificar corretamente os quatro modos do sistema. Obviamente, a separação clara entre os diferentes modos foi possível devido a quantidade moderada de ruído adicionada no sinal de saída do sistema.

A Figura 3.5 mostra que os instantes de chaveamento foram corretamente estimados. No entanto, é importante destacar que essa estimação perfeita dos instantes de chaveamentos só foi possível porque os chaveamentos ocorreram em instantes de tempo múltiplos da janela móvel utilizada. Se esses chaveamentos ocorrerem em instantes de tempos não múltiplos da janela móvel resultará na criação de conjuntos LD_j mistos. Como resultado, alguns pontos desses conjuntos serão atribuídos a um modelo local incorreto, caracterizando uma estimação incorreta dos instantes de chaveamento. Esses erros podem se tornar muito grandes quando a dinâmica de chaveamento for muito rápida e/ou a dimensão da janela móvel utilizada for muito grande. De fato, essa é uma das principais desvantagens do algoritmo 1 e será melhor discutida na Seção 3.5.

A Figura 3.6 apresenta a comparação entre o sinal de saída real e o sinal de saída simulado com os sistemas identificados. Nesta figura, apenas um pequeno trecho dos sinais está sendo apresentado para facilitar a visualização. No caso apresentado, o valor de VAF obtido foi de

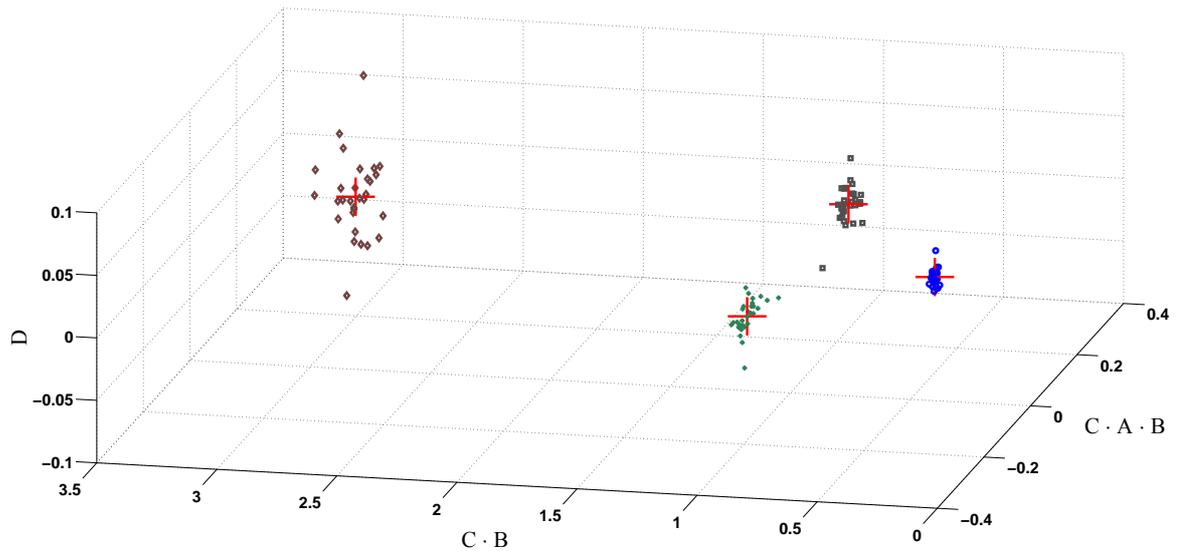


Figura 3.4: Resultado do algoritmo de clustering mostrando que quatro modelos locais foram identificados. As cruces em vermelho indicam os centros de cada cluster.

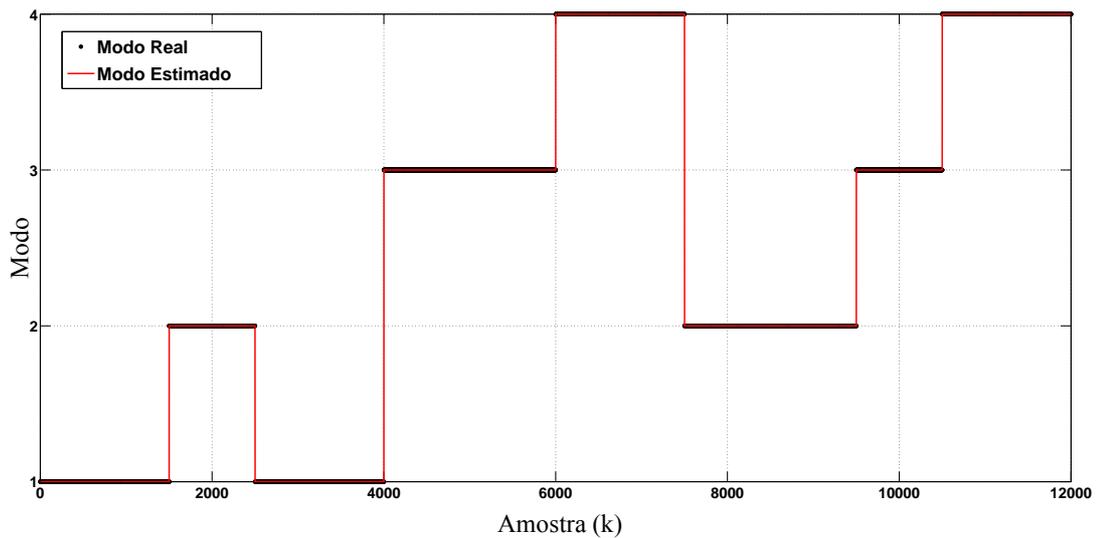


Figura 3.5: Resultado da estimação dos modos do sistema com um SNR de 35 dB.

98,49%. Isso sugere que o modelo estimado foi capaz de reconstruir de maneira eficiente o sinal de saída real. De fato, os valores dos pólos dos sistemas estimados ficaram muito próximos aos valores do sistema real, conforme é mostrado na Tabela 3.1.

Para avaliar a sensibilidade do algoritmo com relação ao ruído foram realizadas simulações de Monte Carlo com 100 diferentes realizações de entrada e saída e quatro diferentes níveis de ruído. A Tabela 3.2 apresenta os valores médios de VAF obtidos nas 100 simulações realizadas.

A Figura 3.7 apresenta o histograma com as distribuições destes valores ao longo das

Tabela 3.1: Comparação entre os pólos reais e estimados com SNR = 35dB.

Modelo Local	Pólos Reais	Pólos Estimados
1	$(0, 25 \pm 0, 7599j)$	$(0, 2498 \pm 0, 7563j)$
2	$(\pm 0, 5j)$	$(-0, 0068 \pm 0, 4959j)$
3	$(-0, 3, 0, 8)$	$(-0, 2905, 0, 8013)$
4	$(\pm 0, 4j)$	$(0, 0138 \pm 0, 3881j)$

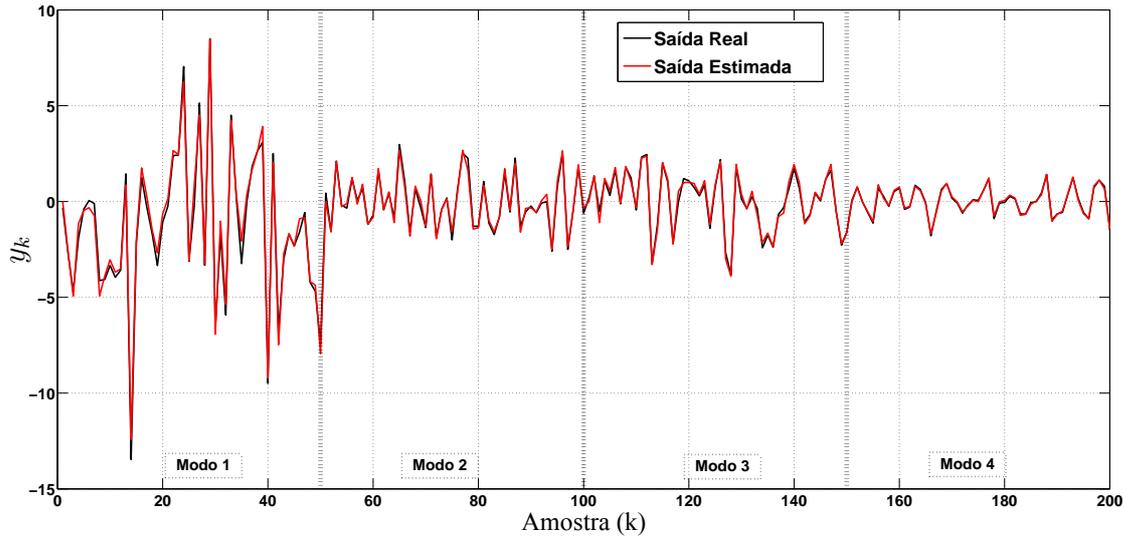


Figura 3.6: Validação dos dados com SNR de 35 dB. Resultado VAF = 98,49%. As linhas verticais tracejadas indicam os instantes de chaveamento.

Tabela 3.2: Desempenho do Algoritmo 1 com diferentes proporções de ruído. Aqui, o termo Sucessos indica o número de vezes que o algoritmo identificou o número de modelos locais corretamente nas 100 simulações.

SNR (dB)	10	20	30	35
VAF (%)	76,88	90,97	95,62	98,54
Sucessos (%)	97	99	99	100

100 simulações e a Figura 3.8 apresenta a comparação dos pólos dos modelos estimados com os pólos dos modelos reais para os diferentes níveis de ruído considerados. A análise conjunta destas figuras e da Tabela 3.2 permite concluir que o algoritmo foi capaz de fornecer estimativas satisfatórias em quase todas as simulações. No entanto, conforme esperado, a qualidade dos modelos estimados diminuiu com o aumento do nível de ruído no sistema. De fato, observe como a variância dos valores de VAF e a dispersão dos pólos no plano complexo aumenta com o aumento do nível de ruído. Os piores desempenhos foram obtidos com nível

de ruído de SNR = 10 dB. Neste caso, a média dos valores de VAF ficou próxima a 80%, mas a variância das estimativas foi muito grande e em 4% das simulações o algoritmo não foi capaz de fornecer modelos consistentes (VAF < 60%). Contudo, nos demais cenários o algoritmo apresentou excelentes resultados.

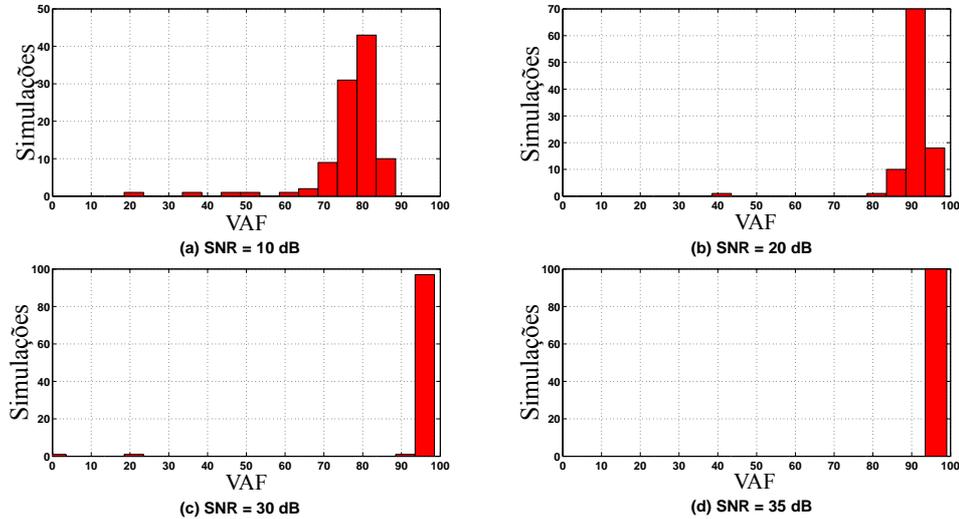


Figura 3.7: Histogramas dos valores de VAF (%) obtidos com a validação dos dados com os modelos estimados nos diferentes níveis de ruído considerados. A distribuição de valores de VAF de 0% a 100% foram divididos em conjuntos (bins) de 5%. Para cada bin é mostrado quantas das 100 simulações ficaram dentro daquele bin.

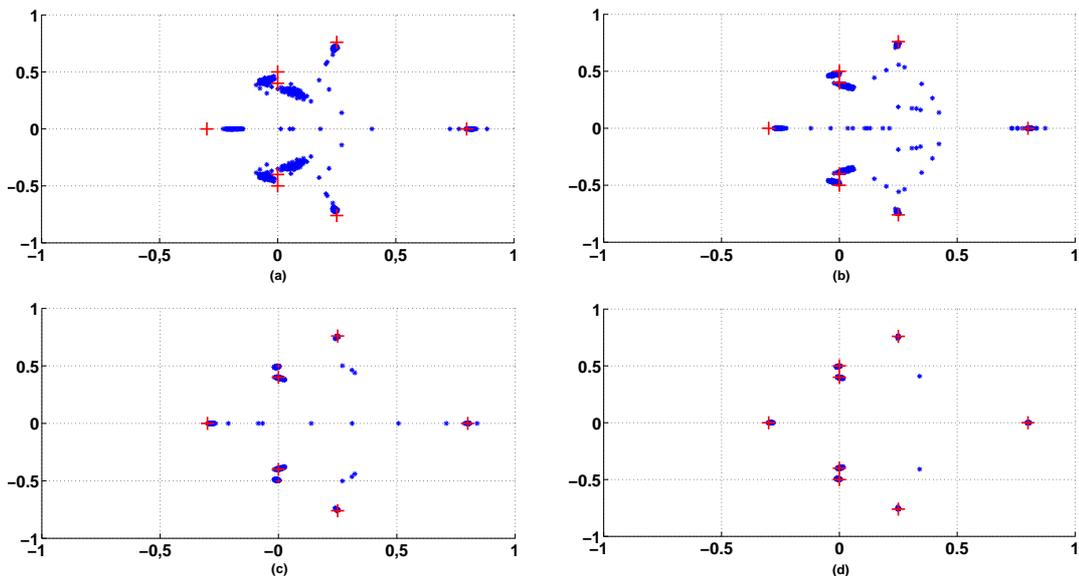


Figura 3.8: Autovalores das matrizes estimadas A_{m_k} no plano complexo para 100 experimentos com (a) SNR = 10 dB (b) SNR = 20 dB (c) SNR = 30 dB (d) SNR = 35 dB. As cruces em vermelho correspondem aos pólos reais dos sistemas.

3.3.1.1 Simulação sem chaveamentos entre os modos

Esta seção visa comparar o algoritmo proposto com uma técnica clássica de identificação em subespaços para sistemas dinâmicos com apenas um modelo. Para isso, utilizou-se o algoritmo ACR (apresentado na Seção 2.1.1) e considerou-se o cenário onde somente o modelo 1 está ativo durante toda a simulação.

As simulações foram realizadas utilizando-se o mesmo sinal de excitação descrito anteriormente e adicionando-se um ruído branco de média nula ao sinal de saída na proporção de $\text{SNR} = 20$ dB. Os parâmetros utilizados na simulação são iguais aos descritos anteriormente. Foi realizada uma simulação de Monte Carlo com 100 realizações. A média dos valores de VAF obtidas ao longo das 100 simulações independentes é apresentada na Tabela 3.3 e comparada com a média dos valores de VAF obtidas utilizando-se o ACR com todos os dados simulados.

Tabela 3.3: Comparação de desempenho do Algoritmo 1 com o ACR para o cenário onde apenas um modelo local está ativo e $\text{SNR} = 20$ dB.

	VAF (%)
Algoritmo 1	90,55
Método clássico de identificação em subespaços	90,72

Os resultados são ligeiramente diferentes porque o algoritmo detectou incorretamente em 87% das simulações a existência de dois modos no sistema e em 13% a existência de três modos. No entanto, nessas simulações todos os modelos estimados eram muito semelhantes e próximos do sistema real.

Alterando-se o valor do parâmetro ϵ_0 no RCA para 50% do valor da cardinalidade robusta, o algoritmo estima corretamente o número de modos em todas as simulações e, neste caso, o valor de VAF se torna igual ao obtido com o algoritmo clássico. Isso já era esperado, pois ao detectar apenas um modo ativo no sistema, o algoritmo classifica todos os dados originais no mesmo modo e, consequentemente, os dois métodos tornam-se iguais.

3.3.2 Ajuste dos Parâmetros do Algoritmo

Pode-se argumentar que uma desvantagem da metodologia proposta é a necessidade de ajuste de um grande número de parâmetros. Contudo, o problema de ajuste desses parâmetros é relativamente simples.

O número de blocos linhas (i) e o tamanho dos conjuntos locais (W) relacionados ao algoritmo de identificação em subespaços são comuns a todas abordagens de identificação baseadas em subespaços. O parâmetro i é facilmente determinado e deve ser maior ou igual

a ordem do sistema a ser identificado. Conforme já discutido, o principal problema é a determinação de W que permanece em aberto na literatura.

A maior quantidade de parâmetros refere-se ao algoritmo de clustering. No entanto, como mostrado em [116], o algoritmo RCA é relativamente robusto com relação a dependência de inicialização dos parâmetros. De fato, nas diversas simulações realizadas ao longo do desenvolvimento deste trabalho e considerando diferentes sistemas, todos os parâmetros do RCA foram mantidos constantes. A única exceção foi o parâmetro ϵ_0 . Com todos os outros parâmetros fixos, ϵ_0 tem influência direta na estimação do número de modos do sistema. A definição de valores elevados para ϵ_0 provoca a identificação de um número menor de modos. Por outro lado, para valores pequenos de ϵ_0 o número de modos é superestimado. No entanto, neste caso, ocorre a identificação de vários modelos semelhantes.

Assim, para a determinação do melhor valor para o parâmetro ϵ_0 , recomenda-se realizar o processo de identificação várias vezes. No início deve-se ajustar um valor pequeno para esse parâmetro. Esse valor deve ser aumentado gradativamente até se observar que entre os resultados obtidos não existem modelos semelhantes. A escolha do valor inicial e o passo de aumento dependerá de cada aplicação. Apenas como referência, nos diferentes exemplos considerados no desenvolvimento deste trabalho o melhor valor para o parâmetro ϵ_0 ficou entre 2,5% e 5,5% do valor da cardinalidade robusta N_s .

3.4 ALGORITMO 2

Conforme mencionado anteriormente, a nova metodologia de identificação apresentada na seção 3.3 permite duas diferentes abordagens para a estimação final dos modelos locais. A primeira, discutida e explorada no Algoritmo 1, utiliza um procedimento de classificação dos dados após a etapa de clustering. Esta classificação permite que um novo processo de identificação seja realizado utilizando uma quantidade maior de dados.

A segunda abordagem, que será proposta no Algoritmo 2, obtém uma representação das matrizes dos modelos locais do sistema a partir das estimativas dos centros dos clusters identificados pelo algoritmo de clustering. Ao melhor conhecimento do autor, essa natural solução do procedimento baseado em clustering ainda não foi utilizada em nenhum algoritmo existente na literatura.

Essa importante diferença torna o algoritmo mais rápido, conforme será ilustrado nos resultados simulados. Esse ganho de velocidade deve-se a eliminação das etapas 4 e 5, uma vez que as estimativas dos modelos locais são obtidas a partir dos centros dos clusters identificados. Assim, a estrutura do Algoritmo 2 é composta apenas pelas etapas de 1 a 3 do procedimento apresentado na Figura 3.2. As duas primeiras etapas não sofrem nenhuma alteração com relação ao Algoritmo 1. Na etapa 3, após a aplicação do algoritmo de clustering é necessário extrair os modelos locais a partir dos centros dos clusters. As

etapas do Algoritmo 2 são descritas a seguir:

Etapas 1 e 2:

Essas etapas permanecem iguais às descritas no Algoritmo 1.

Etapa 3: Clustering

Assim como no Algoritmo 1, nesta etapa aplica-se um algoritmo de clustering para agrupar os vetores dos parâmetros de Markov ($\{\theta_j\}_{j=1}^{N/W}$) estimados na etapa anterior em vários clusters. Os centros de cada cluster identificado representam estimativas dos parâmetros de Markov dos modelos locais do sistema híbrido. Assim, é possível extrair uma representação desses sub-modelos a partir dessas estimativas.

Em [128], é apresentado o procedimento para a extração destes modelos. Por simplicidade, a análise apresentada neste trabalho será restrita a sistemas com uma entrada e uma saída (SISO, do inglês, *Single Input Single Output*). A metodologia para sistemas MIMO está disponível em [128].

Assim, segundo [128] as matrizes dos sistemas podem ser obtidas a partir dos parâmetros de Markov do sistema da seguinte forma:

$$\hat{A}_q = \tilde{T}_q(n, n)T_q^{-1}(n, n) \quad (3.10)$$

$$\hat{B}_q = \begin{bmatrix} h_q(1) & h_q(2) & \cdots & h_q(n-1) & h_q(n) \end{bmatrix}^T, \quad (3.11)$$

$$\hat{C}_q = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad (3.12)$$

$$\hat{D}_q = h_q(0), \quad (3.13)$$

em que $h_q(\cdot)$ são os componentes do vetor que caracterizam o centro do cluster q (ξ_q), ou seja, $\xi_q = \begin{bmatrix} h_q(n+2) & h_q(n+1) & \cdots & h_q(0) \end{bmatrix}^T$, e $\tilde{T}_q(n, n)$ e $T_q(n, n)$ são matrizes definidas como:

$$T_q(n, n) = \begin{bmatrix} h_q(1) & h_q(2) & \cdots & h_q(n) \\ h_q(2) & h_q(3) & \cdots & h_q(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ h_q(n) & h_q(n+1) & \cdots & h_q(2n-1) \end{bmatrix}, \quad (3.14)$$

$$\tilde{T}_q(n, n) = \begin{bmatrix} h_q(2) & h_q(3) & \cdots & h_q(n+1) \\ h_q(3) & h_q(4) & \cdots & h_q(n+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_q(n+1) & h_q(n+2) & \cdots & h_q(2n) \end{bmatrix}. \quad (3.15)$$

Analisando as equações (3.14) e (3.15) é possível concluir que são necessários $2n + 1$ parâmetros de Markov para se recuperar as matrizes do sistema. Isso resulta que o valor teórico mínimo do parâmetro p na equação (3.5) passa a ser:

$$p = 2n + 1. \quad (3.16)$$

Desta forma, o preço pago para que seja possível extrair as matrizes dos modelos a partir das estimativas dos centros dos clusters é se trabalhar com uma dimensão maior do vetor de parâmetros de Markov θ_j em relação ao Algoritmo 1.

A metodologia proposta está resumida no Algoritmo 2.

Algoritmo 2:

1: Segmentação dos Dados:

- a) Crie pequenos conjuntos de dados $\{LD_j\}_{j=1}^{N/W}$, de forma que $LD_j = \{(u_q, y_q) \in \mathcal{D} : q \in \mathcal{T}_j\}$, e \mathcal{T}_j é formado pela divisão do horizonte de tempo \mathcal{T} em conjuntos menores tais que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$

2: Identificação dos Modelos Locais:

- a) Determine θ_j para cada conjunto de dados em $\{LD_j\}_{j=1}^{N/W}$ usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.

3: Clustering:

- a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N/W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA apresentado na Seção 2.2
- b) Estime as matrizes dos modelos locais através das Equações (3.10) - (3.13).
-

3.4.1 Resultados Numéricos

Nesta seção os Algoritmos 1 e 2 serão comparados utilizando um experimento simulado. O sistema chaveado considerado é o mesmo descrito na seção 3.3.1.

Dois diferentes cenários de simulação foram considerados. No primeiro, utilizou-se uma dinâmica de chaveamento lenta no sistema, ou seja, poucas transições entre os diferentes modos. A Figura 3.9a ilustra o sinal utilizado. No segundo cenário, utilizou-se uma dinâmica de chaveamento rápida com vários chaveamentos entre os modos. Este sinal está mostrado na Figura 3.9b. Destaca-se que, na geração destes sinais foi removida a hipótese que os chaveamentos ocorrem em instantes múltiplos do parâmetro W .

Para cada cenário foram realizadas simulações de Monte Carlo com 100 diferentes realizações de entrada e saída. Em todas as simulações o sistema foi excitado por um sinal PRBS de média zero e variância unitária. As simulações foram realizadas adicionando-se um ruído branco de média zero ao sinal de saída. Consideraram-se quatro diferentes níveis de ruído (SNR = 10, 20, 30 e 35 dB). Em cada simulação foram gerados 12000 pontos de dados de $u(k)$ e $y(k)$ que foram utilizados para identificar o sistema. Para o processo de validação dos

modelos geraram-se 1200 pontos.

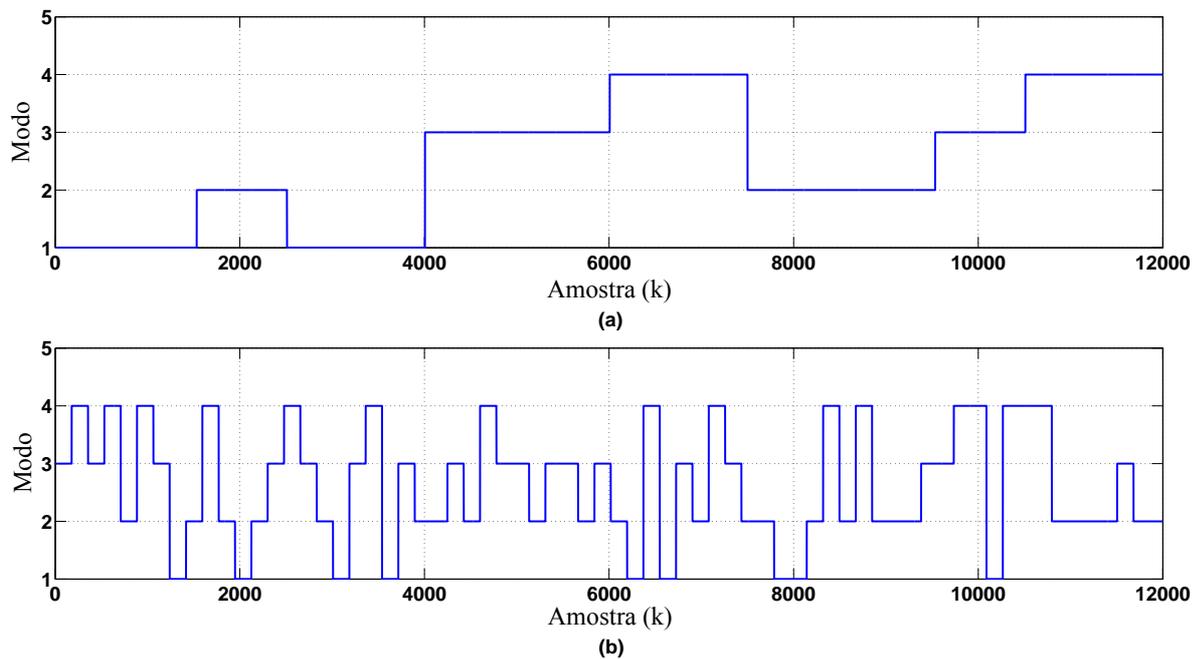


Figura 3.9: Sinais de chaveamentos utilizados nas simulações. (a) Dinâmica de chaveamento lenta. (b) Dinâmica de chaveamento rápida.

Novamente assumiu-se que a ordem dos modelos locais são conhecidas a priori. Assim, de acordo com a equação (3.6) determinou-se $p = 3$ para o Algoritmo 1 e de acordo com a equação (3.16) definiu-se $p = 5$ para o Algoritmo 2. Os demais parâmetros dos algoritmos foram ajustados conforme descrito na seção 3.3.1.

Para avaliar a eficiência dos algoritmos observou-se o tempo total de processamento necessário para a identificação (T_{proc}), a capacidade em detectar corretamente o número de modos, a capacidade em classificar corretamente os pontos amostrados e a habilidade do modelo estimado em reconstruir a saída do sistema a partir do sinal de entrada real. Para isso novamente utilizou-se o valor de VAF entre os sinais medido e simulado.

Os resultados obtidos nos dois cenários considerados estão resumidos nas Tabelas 3.4 e 3.5. Observe nestas tabelas que os dois algoritmos são capazes de determinar o número correto de modos do sistema na maior parte das simulações. Com relação a qualidade dos modelos estimados, observa-se que no cenário onde a dinâmica de chaveamento do sistema é lenta e o nível de ruído é moderado que os dois algoritmos apresentam desempenho semelhante, pois apresentam valores de VAF e erro de classificação próximos.

Contudo, o Algoritmo 1 apresenta melhor desempenho nos cenários com elevado nível de ruído e/ou dinâmica de chaveamento rápida. Essa perda de desempenho deve-se a dois fatores. Primeiro, conforme discutido no Algoritmo 1, os dados enviados ao algoritmo de

clustering são provenientes da identificação em uma pequena janela de tempo. Desta forma, a influência dos ruídos de processo/medida pode ser significativa sobre os modelos identificados. Assim, o processo de classificação dos dados realizado no Algoritmo 1 permite que um novo processo de identificação seja realizado. Esta nova identificação utiliza uma quantidade maior de dados o que reduz a influência dos efeitos dos ruídos.

Segundo, quando a dinâmica de chaveamento é rápida e os instantes de chaveamento não são múltiplos do parâmetro W , a segmentação dos dados pode resultar na criação de muitos LDs mistos. Com efeito, há um aumento da quantidade de outliers enviados ao algoritmo de clustering. Na Figura 3.10 é mostrada a comparação dos pontos utilizados na etapa de clustering nos dois cenários. Observe como no cenário onde a dinâmica de chaveamento é rápida (Figura 3.10b) como os pontos ficam muito mais dispersos. Mesmo o algoritmo RCA sendo robusto a esses outliers, o aumento da quantidade de outliers influencia a determinação dos centros dos clusters, ou seja, esses outliers deslocam os centros dos clusters. Como consequência, ocorre uma redução na qualidade dos modelos locais quando esses são estimados a partir dos centros dos clusters.

É importante destacar que os efeitos dos LDs mistos influenciam negativamente na qualidade dos modelos estimados nos dois algoritmos. De fato, observe na Tabela 3.5 como a porcentagem de erros de classificação aumenta significativamente em comparação com o cenário de dinâmica de chaveamento lenta. A Figura 3.11 apresenta a comparação entre os sinais de chaveamento estimado pelos dois algoritmos. O aumento do erro de classificação reduz a acurácia dos modelos estimados. Contudo, no Algoritmo 1, a etapa de classificação dos dados reduz a influência desses conjuntos. O desenvolvimento do Algoritmo 3, que será apresentado na próxima seção, visa atacar esse problema e melhorar o desempenho da identificação em cenários mais críticos.

Tabela 3.4: Comparação de desempenho dos algoritmos com dinâmica lenta de chaveamento. Aqui, o termo Erro representa a média dos erros na estimação dos modos.

SNR (dB)	Algoritmo	T_{proc} médio (s)	VAF (%)	Erro (%)	Sucessos (%)
10	1	3,71	75,49	1,74	100
	2	2,01	71,39	5,43	95
20	1	3,50	90,54	1,06	100
	2	1,88	89,09	0,97	100
30	1	4,01	96,86	0,96	100
	2	1,83	96,13	1,41	100
35	1	4,08	98,21	0,92	100
	2	1,96	97,82	0,93	100

No entanto, apesar do desempenho inferior na estimação dos modelos, o Algoritmo 2 apresenta um custo computacional muito menor. De fato, observou-se uma redução média de aproximadamente 51% no tempo de processamento. Desta forma, o Algoritmo 2 é mais recomendado nos casos onde o sistema híbrido a ser identificado apresenta dinâmica de chaveamento lenta e níveis moderados de ruído.

Tabela 3.5: Comparação de desempenho dos algoritmos com dinâmica de chaveamento rápida.

SNR (dB)	Algoritmo	T_{proc} médio (s)	VAF (%)	Erro (%)	Sucessos (%)
10	1	4,62	69,53	24,92	99
	2	1,78	65,47	34,17	99
20	1	3,81	88,32	17,12	98
	2	2,21	86,79	21,62	100
30	1	4,42	95,20	16,40	99
	2	1,87	94,61	18,63	100
35	1	4,27	96,23	17,27	100
	2	2,13	95,13	22,05	99

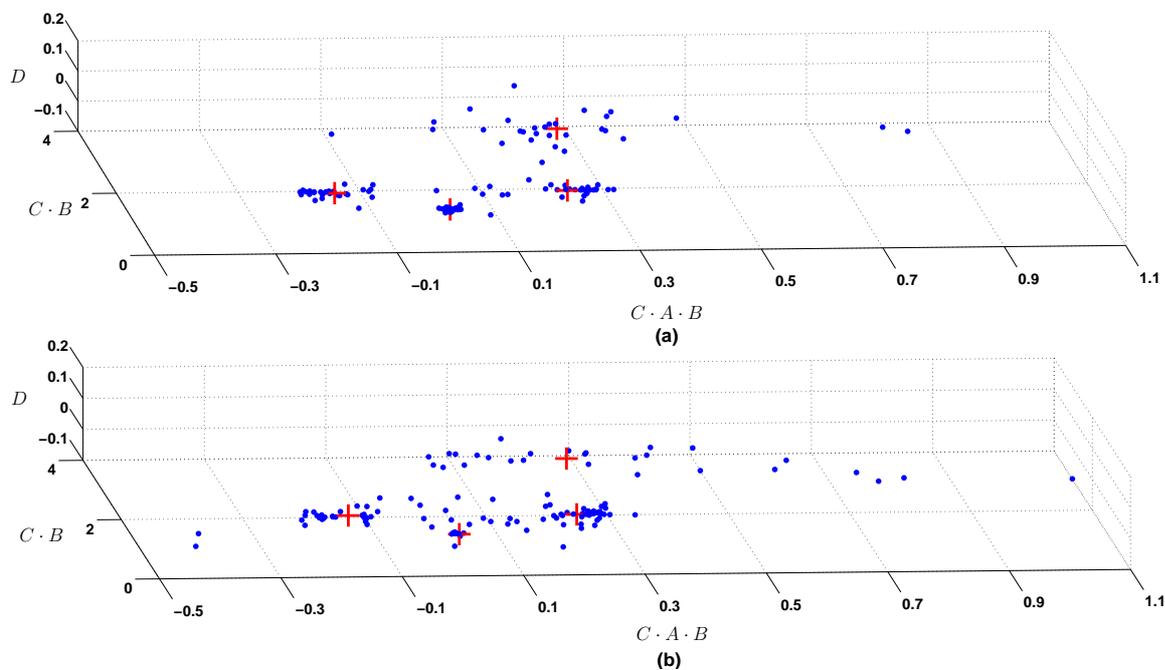


Figura 3.10: Resultado típico do algoritmo de clustering para o Algoritmo 1. (a) Dinâmica de chaveamento lenta. (b) Dinâmica de chaveamento rápida.

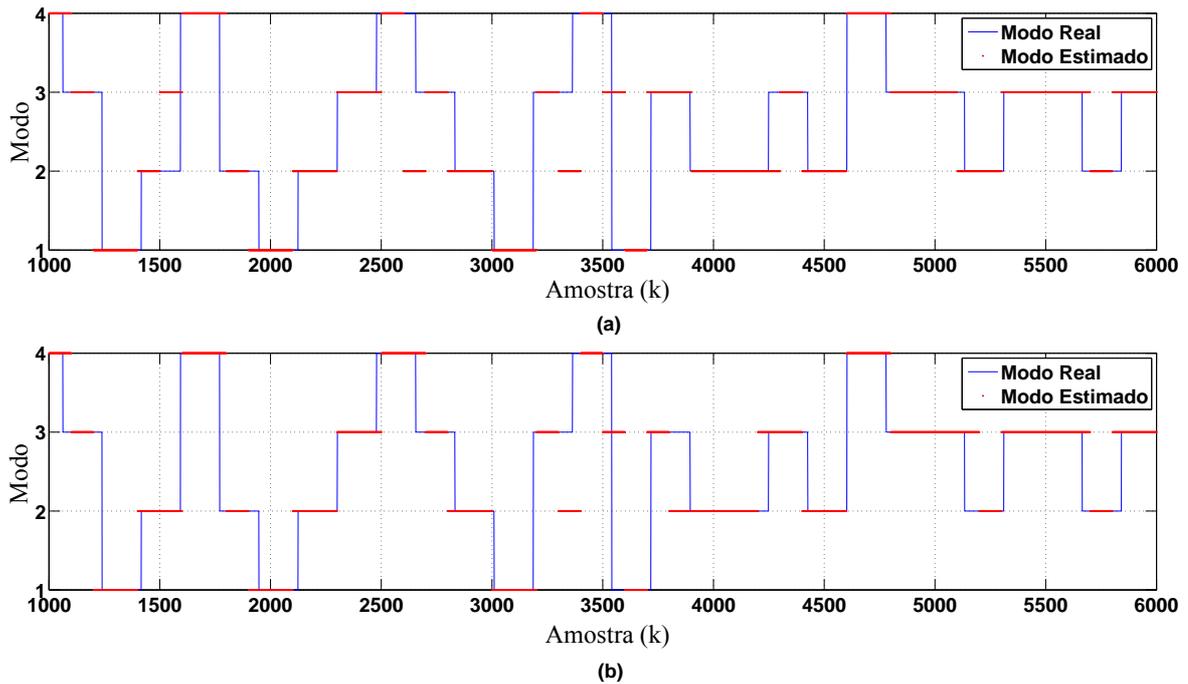


Figura 3.11: Resultado típico para estimação de modos do sistema com SNR = 35dB. (a) Algoritmo 1. (b) Algoritmo 2. Para melhor visualização é apresentado apenas um pequeno trecho dos sinais de chaveamento estimados.

3.5 ALGORITMO 3

O principal objetivo no desenvolvimento do terceiro algoritmo é minimizar a influência dos LDs mistos na qualidade dos modelos estimados. Esta influência foi discutida na seção 3.4.1.

Desta forma, o Algoritmo 3, é uma evolução do primeiro algoritmo no sentido que incorpora modificações na etapa de classificação dos dados e uma nova etapa nomeada de atualização da segmentação dos dados. A ideia explorada no algoritmo foi dividir a identificação dos modelos locais em duas partes.

Na primeira parte, chamada de atualização dos modelos dos locais é realizada a identificação parcial dos modelos locais. Para isso, elimina-se do processo de identificação os pontos duvidosos, ou seja, os pontos que não se tem certeza sobre a qual modo eles pertencem. O objetivo é reduzir a influência desses outliers na estimação dos modelos. Esses outliers são identificados na etapa de classificação dos dados utilizando-se as informações fornecidas pelo RCA.

De posse das estimativas parciais dos modelos locais, realiza-se uma etapa de atualização da segmentação dos dados. Para isso, utiliza-se o algoritmo IMM, descrito na Seção 2.3, que permite determinar o conjunto de equações de medições mais adequado para cada um dos pontos (u_k, y_k) amostrado. Ou seja, o IMM determina individualmente o modo de cada um

dos pontos existentes no conjunto de dados amostrados \mathcal{D} . Essa nova classificação permite a formação dos conjuntos dos dados clusterizados $\{C_q^c\}_{q=1}^M$. A função desta etapa é minimizar os erros de classificações dos dados obtidos com o algoritmo de clustering e classificar todos os dados amostrados nos diferentes modos identificados, inclusive os pontos duvidosos que não foram utilizados para a identificação parcial dos modelos locais.

A segunda parte da identificação, chamada de identificação final dos modelos locais, é realizada após o processo de atualização da segmentação dos dados e, portanto, utiliza uma quantidade maior de dados resultando em estimativas mais precisas dos modelos locais. A Figura 3.12 ilustra os passos deste algoritmo e os detalhes dessas modificações são descritas a seguir.

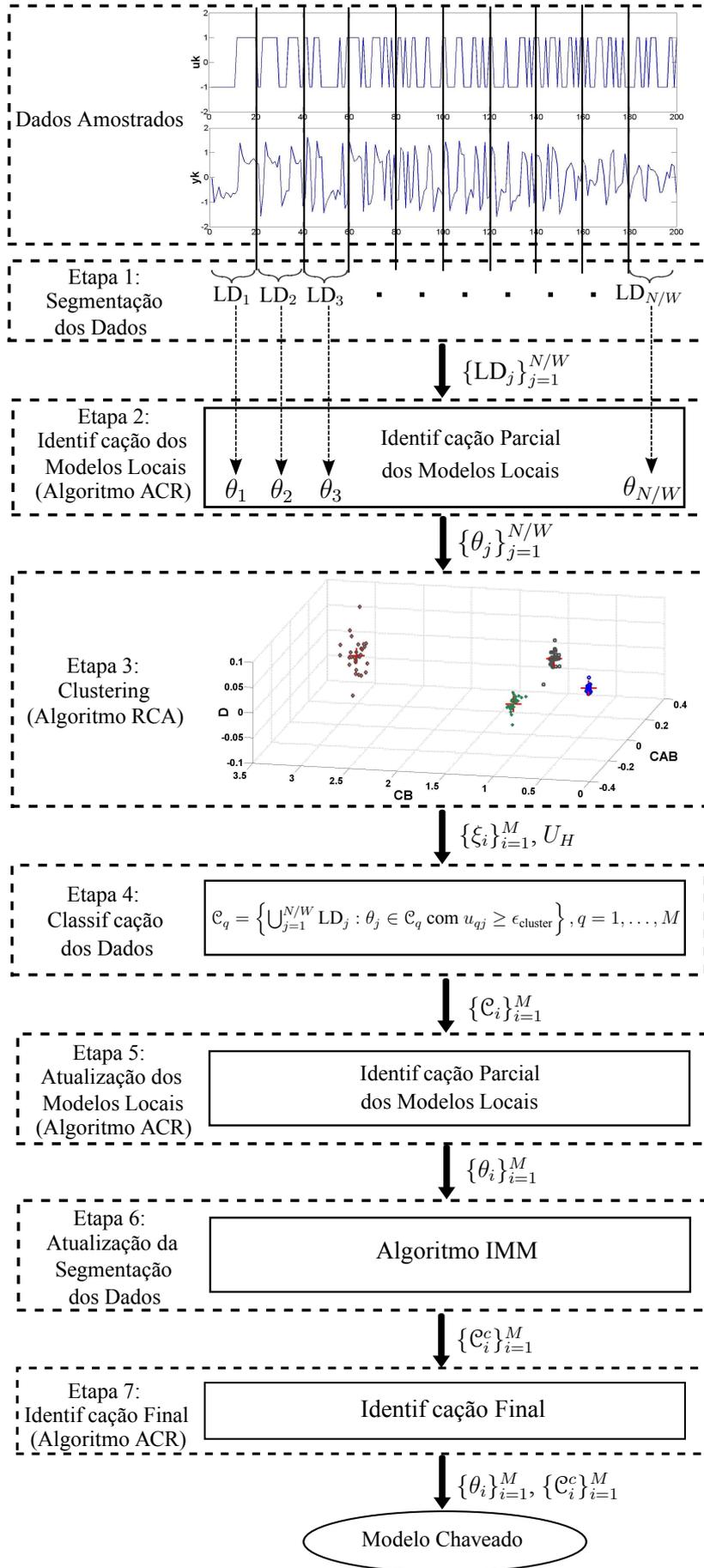


Figura 3.12: Diagrama do Algoritmo 3.

Etapas 1 a 3:

Essas etapas permanecem inalteradas do Algoritmo 1.

Etapa 4: Classificação dos Dados:

Para minimizar os efeitos dos outliers causados pelos conjuntos mistos, os pontos θ_j que são duvidosos são descartados nesta etapa de classificação dos dados. São considerados pontos duvidosos aqueles pontos que possuem alta probabilidade de pertencer a mais de um cluster ao mesmo tempo. Essa informação pode ser obtida do RCA que fornece a matriz $U_H = [u_{qj}]$ que indica o grau no qual o ponto θ_j pertence ao cluster ξ_q .

Essa matriz é usada para selecionar somente aqueles pontos que pertencem ao cluster ξ_q com um grau de pertinência acima de um determinado limiar $\epsilon_{\text{cluster}}$. Os pontos não classificados não são utilizados no processo de identificação realizado na próxima fase. Assim, os dados são classificados nos clusters $\{\mathcal{C}_q\}_{q=1}^M$ de acordo com a regra:

$$\mathcal{C}_q = \left\{ \bigcup \text{LD}_j : \theta_j \in \mathcal{C}_q \text{ com } u_{qj} \geq \epsilon_{\text{cluster}}, j = 1, 2, \dots, N/W \right\}. \quad (3.17)$$

Etapa 5: Atualização dos Modelos Locais:

Nesta etapa os dados agrupados nos clusters $\{\mathcal{C}_q\}_{q=1}^M$ são utilizados para estimação dos modelos locais. Esta etapa é semelhante a etapa 5 do Algoritmo 1. No entanto, observe que agora a quantidade de dados utilizados no processo de identificação é menor, pois os pontos duvidosos foram eliminados. Desta forma, os modelos locais estimados ainda não são as estimativas finais. Por isso o nome da etapa foi alterado.

Etapa 6: Atualização da Segmentação dos Dados:

As etapas anteriores permitem a obtenção de estimativas consistentes para os modelos locais, porém através da eliminação de alguns dados do processo de identificação. Esse processo de eliminação apresenta como desvantagem a possibilidade de não identificação de alguma dinâmica ou característica do sistema. Desta forma, um processo de atualização da segmentação dos dados é importante para permitir que o conjunto completo dos dados amostrados \mathcal{D} seja atribuído a um modo particular do sistema e usados para refinar as estimativas.

Para realizar esta etapa pode-se utilizar qualquer algoritmo de estimação dos modos do sistema ou qualquer algoritmo de filtragem híbrida que proporcione também a estimação dos modos. A seção 2.3 apresentou uma pequena revisão dos principais métodos existentes na literatura. Neste trabalho, foi utilizado o algoritmo IMM. Esta escolha foi motivada pelo fato deste ser o algoritmo de estimação híbrida mais utilizado e o que apresenta melhor relação custo-benefício na estimação de estados em sistemas híbridos [1].

Ressalta-se que os algoritmos de filtragem híbrida fornecem estimativas dos estados contínuos e discretos do sistema híbrido. No entanto, neste trabalho o interesse é apenas na de-

terminação do modo a cada instante de tempo. Desta forma, o IMM é utilizado para realizar uma correção ou aproximação corrigida das janelas de tempo em que cada subsistema esteve ativo, ou seja, é utilizado para formar os novos conjuntos de dados clusterizados $\{c_q^c\}_{q=1}^M$. Nesta notação, foi utilizado o sobre-escrito c para indicar a correção na classificação dos dados.

Por fim, é importante observar que a classificação dos dados realizada pelo IMM, constitui uma estimação dos instantes de chaveamento do sistema.

Etapa 7: Identificação Final dos Modelos Locais:

Uma vez que todos os dados amostrados estão classificados, os modelos locais finais podem ser identificados. Novamente, utilizou-se a combinação das técnicas apresentadas nas seções 2.1.1 e 2.1.2 para obter as estimativas dos modelos no espaço de estados. Note que esta etapa produz melhores estimativas em comparação com a estimação do passo 5, pois a quantidade de dados usadas para estimar os modelos é maior devido a atualização da segmentação dos dados.

O algoritmo proposto está resumido no Algoritmo 3.

3.5.1 Resultados Numéricos

Nesta seção, será realizada a comparação das abordagens propostas para ilustrar a importância das modificações realizadas no desempenho final do algoritmo. Uma vez que o Algoritmo 1 possui uma estrutura mais completa e apresentou melhores resultados quando comparado ao Algoritmo 2, ele foi o escolhido para a comparação com o Algoritmo 3.

Para realizar as comparações, foram realizadas simulações numéricas utilizando-se novamente o sistema híbrido descrito na Seção 3.3.1. No entanto, agora foi considerado que este sistema tem uma dinâmica de chaveamento mais rápida, ou seja, com várias transições entre os modos.

Para isso, foi adotado uma função para o sinal de chaveamento que é constante por partes no tempo e que chaveia periodicamente a cada 150 amostras ($\tau_{\text{dwell}} = 150$) do seu valor atual para um valor alvo escolhido aleatoriamente entre $\{1, \dots, 4\}$. A Figura 3.14 ilustra o sinal utilizado. De forma a comparar os dois algoritmos nas mesmas condições, o mesmo sinal de chaveamento foi utilizado em todas as simulações.

Novamente o sistema foi excitado por um sinal PRBS de média zero e variância unitária e uma sequência com 12000 pontos de u_k e y_k foi utilizada para identificar o sistema. Para a validação dos modelos estimados foram gerados mais 1200 pontos. O sinal de saída foi simulado adicionando-se um sinal de ruído branco Gaussiano de média zero. Considerou-se quatro diferentes valores de magnitude para esse ruído, de forma que a relação sinal ruído (SNR) fosse igual a 10, 20, 30 e 35 dB com relação a saída do sistema.

Algoritmo 3:

1: Segmentação dos Dados:

- a) Crie pequenos conjuntos de dados $\{\text{LD}_j\}_{j=1}^{N/W}$, de forma que $\text{LD}_j = \{(u_q, y_q) \in \mathcal{D} : q \in \mathcal{T}_j\}$, e \mathcal{T}_j é formado pela divisão do horizonte de tempo \mathcal{T} em conjuntos menores tais que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$

2: Identificação dos Modelos Locais:

- a) Determine θ_j para cada conjunto de dados em $\{\text{LD}_j\}_{j=1}^{N/W}$ usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.

3: Clustering:

- a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N/W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA apresentado na Seção 2.2

4: Classificação dos Dados:

- a) Defina o limiar $\epsilon_{\text{cluster}}$
- b) Forme os conjuntos dos dados $\{\mathcal{C}_q\}_{q=1}^M$ de acordo com a seguinte regra:

$$\mathcal{C}_q = \left\{ \bigcup \text{LD}_j : \theta_j \in \mathcal{C}_q \text{ com } u_{qj} \geq \epsilon_{\text{cluster}}, j = 1, 2, \dots, N/W \right\}.$$

5: Atualização dos Modelos Locais:

- a) Estime os modelos locais usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.

6: Atualização da Segmentação dos Dados:

- a) Com as estimativas dos modelos locais obtidas na Etapa 5, use o algoritmo IMM apresentado na Seção 2.3 para corrigir os conjuntos dos dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$

7: Identificação Final dos Modelos Locais:

- a) Identifique os modelos locais finais com os dados de $\{\mathcal{C}_q^c\}_{q=1}^M$, usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
 - b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.
-

Os parâmetros iniciais dos algoritmos foram definidos como descrito na seção 3.3.1. A exceção foi o parâmetro ϵ_0 do algoritmo RCA que foi modificado para 5,5% do valor da cardinalidade robusta N_s nos dois algoritmos. O limiar usado para eliminar os pontos duvidosos no Algoritmo 3 foi $\epsilon_{\text{cluster}} = 0,9$. A seguinte matriz de transição de probabilidade foi usada no algoritmo IMM:

$$\Pi = \begin{bmatrix} 0,99 & \frac{0,01}{(M-1)} & \cdots & \frac{0,01}{(M-1)} \\ \frac{0,01}{(M-1)} & 0,99 & \cdots & \frac{0,01}{(M-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{0,01}{(M-1)} & \frac{0,01}{(M-1)} & \cdots & 0,99 \end{bmatrix}, \quad (3.18)$$

em que M é o número de modelos locais identificados em cada simulação realizada.

A primeira vista pode parecer que a probabilidade do sistema permanecer no mesmo modo foi ajustada com um valor muito elevado (0,99). Contudo, deve ser observado que a imposição do $\tau_{\text{dwell}} = 150$ implica necessariamente em uma alta probabilidade do sistema permanecer no modo atual. Adicionalmente, justifica-se que o IMM se mostrou pouco sensível às variações na MPT.

Para comparar os dois algoritmos, foram realizadas simulações de Monte Carlo com 100 realizações para os quatro níveis de ruídos considerados. Em cada uma das simulações, uma diferente realização do sinal de entrada e ruído foi considerada. O desempenho do sistema identificado foi avaliado examinando-se o valor de VAF no conjunto de dados de validação. A habilidade em determinar corretamente o número de modos, classificar os dados nos diferentes modos identificados e o tempo de processamento também foram examinados.

Os resultados estão resumidos na Tabela 3.6. Observe nesta tabela que os dois algoritmos são capazes de determinar o número correto de modos na maior parte das simulações. Nota-se que o desempenho dos dois algoritmos são, essencialmente, semelhantes. Isso já era esperado, uma vez que os algoritmos são iguais até a etapa de clustering. A Figura 3.13 ilustra o processo de segmentação apresentando um resultado típico do algoritmo de clustering utilizado no Algoritmo 3. Observe nesta figura como a eliminação dos pontos duvidosos torna a separação entre os diferentes clusters mais clara.

Tabela 3.6: Comparação de desempenho dos algoritmos com diferentes níveis de ruído.

	Algoritmo 1				Algoritmo 3			
	10	20	30	35	10	20	30	35
SNR (dB)								
VAF (%)	76,92	89,56	94,28	95,83	73,88	91,66	97,08	98,22
Erro (%)	19,71	18,80	17,95	17,24	14,05	6,22	4,11	3,83
Successos (%)	98	96	99	100	97	100	100	100
T_{proc} médio (s)	4,20	4,06	3,92	3,91	15,47	12,58	18,20	12,80

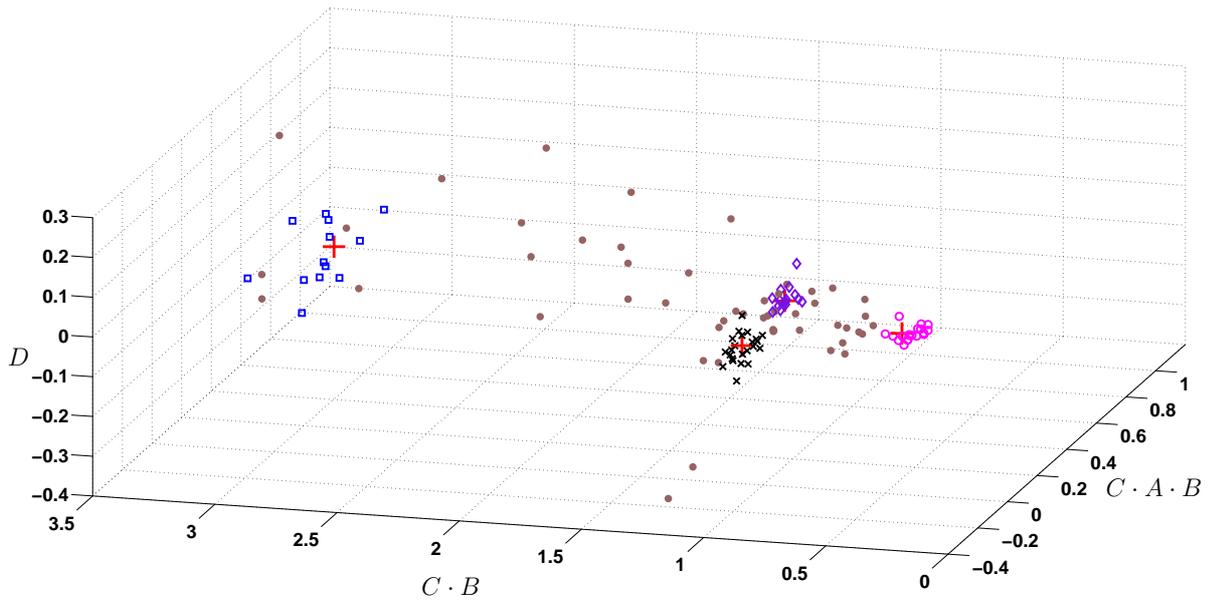


Figura 3.13: O resultado de clustering obtido com o Algoritmo 3, demonstrando que os quatro modelos locais foram identificados. As cruces em vermelho são os centros de cada cluster e os pontos em marrom representam os pontos dúbios e não utilizados na identificação parcial dos modelos locais.

Observe na Tabela 3.6 como a porcentagem de erro de estimação dos modos é sensivelmente menor no Algoritmo 3. Em média, o Algoritmo 3 reduz em 62% os erros de classificação quando comparado com o Algoritmo 1. No melhor caso, a redução foi de 77,78%.

Essa redução é atribuída, principalmente, ao passo de atualização da segmentação dos dados. Para entender essa redução é preciso lembrar do método utilizado para segmentação dos dados amostrados. O particionamento divide o conjunto de dados amostrados de forma que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$. Assim, se um chaveamento ocorrer no período de tempo considerado em uma determinada janela, parte dos dados constituintes desta janela serão atribuídos a um modo incorreto. Desta forma, a fase de atualização da segmentação dos dados permite avaliar individualmente todos os dados amostrados (conjunto de dados \mathcal{D}) e determinar o modo mais provável. A Figura 3.14 apresenta resultados típicos de estimação dos modos para os dois algoritmos. A Figura 3.15 mostra os detalhes dessa estimação considerando apenas um pequeno trecho dos sinais de chaveamento estimados.

A definição do tamanho W na segmentação dos dados tem uma influência direta na porcentagem de erro de estimação dos modos no Algoritmo 1. Quanto maior for a janela de tempo considerada, maior poderá ser o erro de estimação dos modos, uma vez que mais pontos poderão ser atribuídos a modos incorretos. Por outro lado, a utilização de uma pequena janela, implica em poucos dados no processo de identificação e pode levar a obtenção de modelos imprecisos. Assim, conforme já mencionado, a definição do tamanho W tem um

papel fundamental no algoritmo de identificação.

É importante ressaltar que, a utilização do algoritmo IMM no Algoritmo 3 reduz a influência do tamanho desta janela na quantidade de erros de estimação dos modos e, consequentemente na qualidade dos modelos estimados, uma vez que os dados amostrados são avaliados individualmente. Contudo, permanece a relação inicial de compromisso entre a janela conter uma quantidade de dados suficiente para que seja possível realizar a identificação dos modos e, ao mesmo tempo, ter uma duração pequena o suficiente para que possa ser possível detectar as variações nas dinâmicas do sistema.

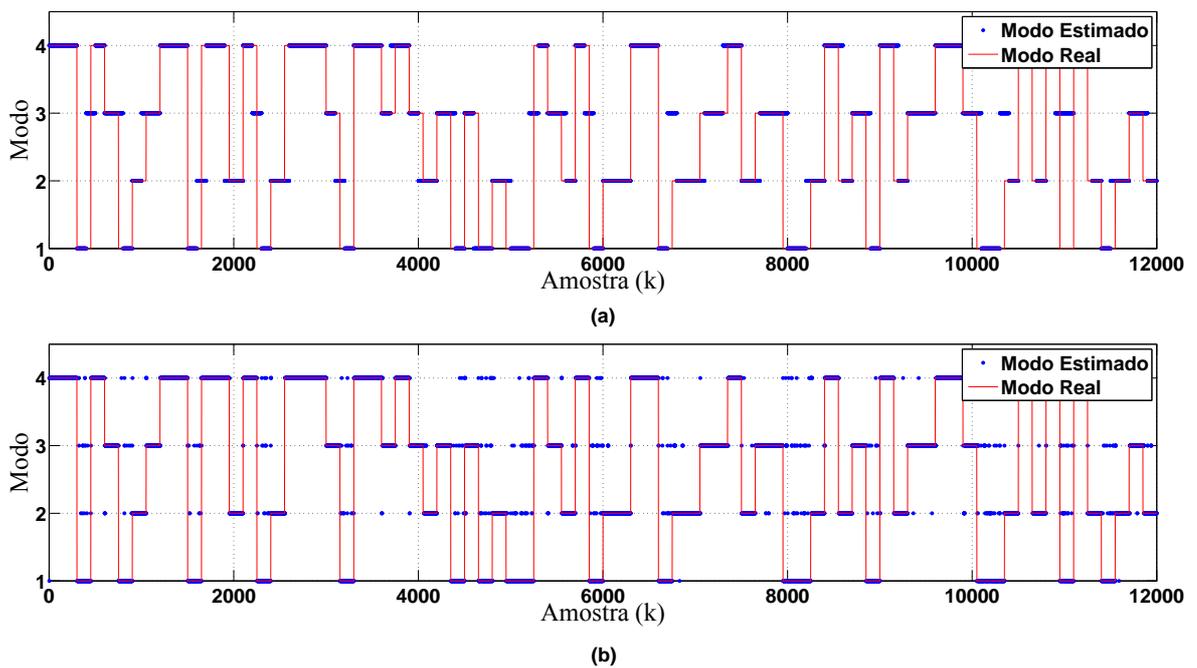


Figura 3.14: Resultado típico para a estimação de modos com SNR de 20 dB. (a) Resultado obtido com o Algoritmo 1 (17,92% de erros de classificação). (b) Resultado obtido com o Algoritmo 3 (5,1% de erros de classificação).

A consequência da redução do erro médio na estimação dos modos é a obtenção de modelos mais precisos e consistentes. De fato, observe na Tabela 3.6 que o Algoritmo 3 apresenta valores maiores de VAF quando comparado com o Algoritmo 1.

A Figura 3.16 reforça o melhor desempenho do Algoritmo 3, mostrando a distribuição dos valores de VAF obtidos nas 100 simulações independentes realizadas com SNR= 20 dB. Essa figura revela que na maioria das simulações os dois algoritmos produzem valores de VAF maiores que 85%. No entanto, no terceiro algoritmo, 91% das simulações produzem valores de VAF maior que 90%, enquanto que esse valor se reduz para 46% quando o primeiro algoritmo é utilizado.

A Figura 3.17 ilustra o melhor desempenho do Algoritmo 3 comparando resultados tí-

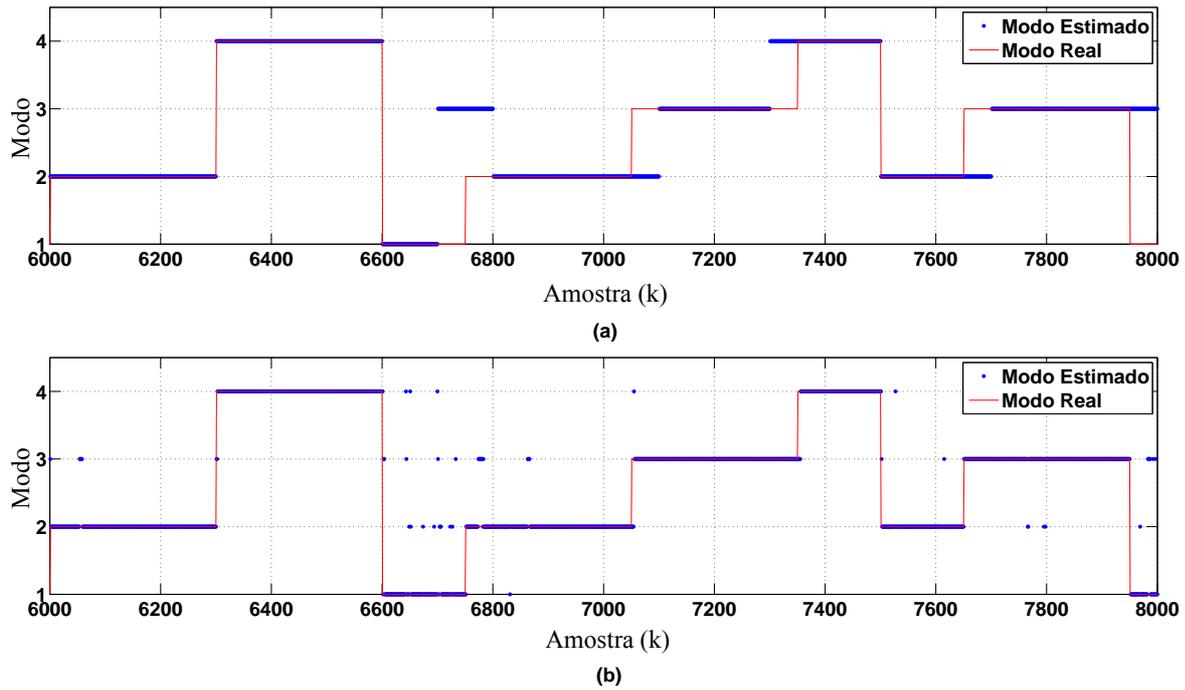


Figura 3.15: Re-dimensionamento dos eixos da Figura 3.14 para ilustrar os detalhes da estimação dos modos. (a) Resultado obtido com o Algoritmo 1. (b) Resultado obtido com o Algoritmo 3.

picos de validação utilizando os dois algoritmos. Note como o Algoritmo 3 é capaz de reconstruir de forma mais eficiente o sinal de saída.

Conforme os resultados demonstraram, a abordagem proposta no Algoritmo 3 proporcionou melhorias significativas no desempenho final do algoritmo. Contudo, apresentou como desvantagem aumento do custo computacional. De fato, no Algoritmo 3, levou-se em média 10,74s a mais para se obter uma estimativa dos modelos locais. Esse número representa um aumento médio de 267,5% no tempo de processamento em relação ao primeiro algoritmo. No entanto, isso não é considerado um problema, visto que os métodos de identificação propostos são off-line.

3.5.2 Ajuste dos Parâmetros do Algoritmo

Com as modificações propostas no Algoritmo 3 aumentou a quantidade de parâmetros para ajustar antes do processo de identificação. O problema de ajuste dos parâmetros do algoritmo IMM é similar ao problema de sintonia de um filtro de Kalman tradicional. A principal diferença está relacionada ao ajuste da MPT. Duas diferentes abordagens podem ser utilizadas de acordo com o grau de conhecimento disponível do sistema. Se houver algum conhecimento prévio do sistema, ele pode ser utilizado para definir esta matriz. Se não houver nenhum conhecimento do sistema, pode-se lançar mão de algoritmos que estimam

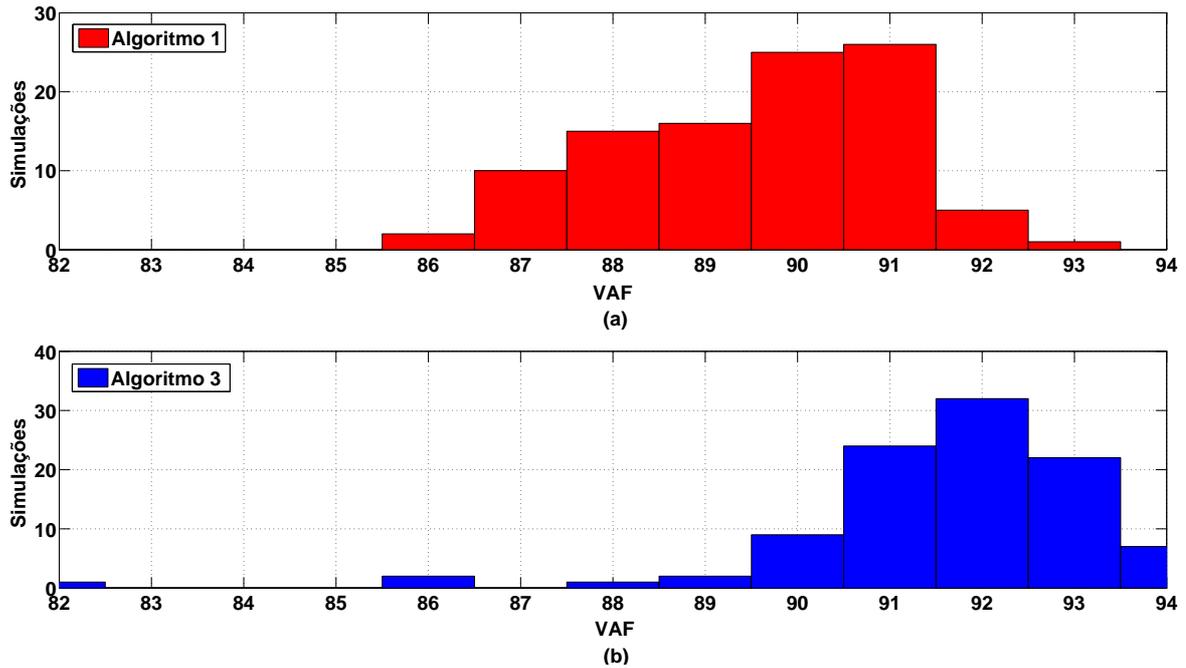


Figura 3.16: Distribuição das medidas de VAF nas 100 simulações independentes dos algoritmos com SNR de 20 dB. (a) Distribuição obtida utilizando-se o Algoritmo 1. (b) Distribuição obtida usando-se o Algoritmo 3. Nestas figuras os valores foram divididos em bins de 1%.

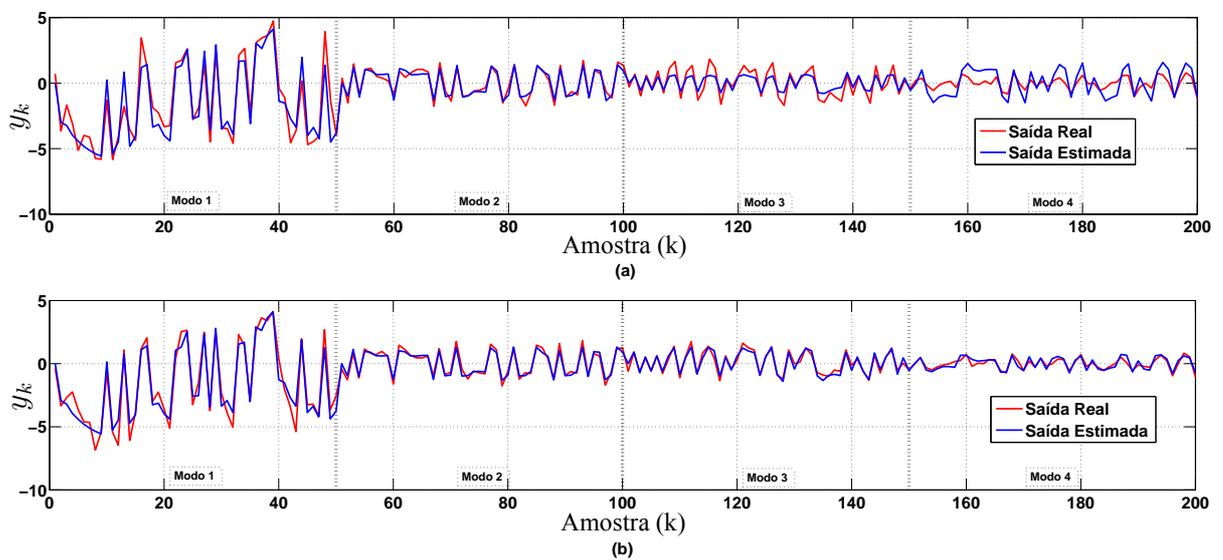


Figura 3.17: Validação de dados com SNR de 20 dB. (a) Resultado obtido com o Algoritmo 1 (VAF = 83,99%). (b) Resultado obtido com o Algoritmo 3 (VAF = 93,38%).

esta matriz com base nos dados de entrada e saída do sistema [1]. No entanto, verificou-se que o algoritmo é pouco sensível às variações nesta matriz, ou seja, a determinação desta

matriz tem pouca influência no resultado de identificação.

O principal problema enfrentado foi o ajuste do novo parâmetro $\epsilon_{\text{cluster}}$. Verificou-se que o desempenho do algoritmo é diretamente influenciado por esse parâmetro. Para ilustrar essa dependência realizou-se novas simulações de Monte Carlo com 100 simulações independentes, considerando o cenário onde a magnitude do ruído de medição proporcionava um valor de SNR igual a 20 dB. Consideraram-se cinco diferentes valores de $\epsilon_{\text{cluster}}$ e os resultados obtidos estão apresentados na Tabela 3.7.

Tabela 3.7: Desempenho do Algoritmo 3 para diferentes valores do parâmetro $\epsilon_{\text{cluster}}$.

	Algoritmo 3				
$\epsilon_{\text{cluster}}$	0,9	0,7	0,5	0,3	0
VAF (%)	91,66	90,87	73,68	22,26	24,87
# Erro (%)	6,22	8,99	21,71	48,34	48,10
Successos (%)	100	95	100	99	99

Observe na Tabela 3.7 como o desempenho do algoritmo se degrada com a diminuição do valor de $\epsilon_{\text{cluster}}$. Esse resultado já era esperado, uma vez que diminuindo-se o valor de $\epsilon_{\text{cluster}}$, mais pontos duvidosos são utilizados no processo de identificação. Isso aumenta a possibilidade de utilização de dados gerados por diferentes modelos locais no processo de identificação de um determinado modelo local. De fato, isto é comprovado pelo aumento do erro médio na classificação dos pontos mostrado na Tabela 3.7.

O principal inconveniente desta dependência é o aumento da dificuldade na inicialização dos parâmetros do algoritmo, em especial, dos parâmetros ϵ_0 , $\epsilon_{\text{cluster}}$ e o tamanho da janela W . Conforme já mencionado, o tamanho W define a qualidade das estimativas, quanto mais dados utilizados na identificação, na teoria, melhor será a qualidade dos modelos estimados, em contrapartida aumenta a probabilidade de formação de conjuntos mistos.

Consequentemente, o tamanho desta janela influenciará a dispersão dos pontos (vetores de parâmetros de Markov) que serão considerados no algoritmo de clustering. Quanto melhor forem as estimativas dos modelos, mais condensados ficarão os pontos e mais fácil será para o algoritmo de clustering determinar os diferentes modos. Por outro lado, estimativas pobres resultarão em grande dispersão dos pontos, aumentando a dificuldade na determinação do número correto de clusters. Portanto, o parâmetro ϵ_0 do RCA, que é o principal responsável pela determinação do número de modos, é muito dependente da qualidade das estimativas e, consequentemente, de W .

Adicionalmente, no cenário em que há grande dispersão dos pontos é necessário aumentar o valor de $\epsilon_{\text{cluster}}$ para reduzir a influência dos conjuntos mistos e produzir estimativas de qualidade. No entanto, esse processo não é tão trivial quanto parece. Quando muitos

pontos duvidosos são eliminados, diminui-se a quantidade de dados utilizados no processo de identificação dos modelos locais. Com efeito, pode-se aumentar a influência dos ruídos de processo/medição na estimação desses modelos o que afetará o desempenho do algoritmo IMM, aumentando os erros de classificação dos pontos e resultando em estimativas ruins para os modelos locais finais. O melhor cenário de identificação é quando há pouca dispersão dos pontos estimados. Assim, pode-se reduzir o valor de $\epsilon_{\text{cluster}}$ e ainda manter elevada a quantidade de dados no processo de identificação dos modelos locais. Neste contexto, o parâmetro $\epsilon_{\text{cluster}}$ também é muito dependente da qualidade das estimativas.

Desta forma, recomenda-se que vários valores de $\epsilon_{\text{cluster}}$ sejam testados. O melhor valor para este parâmetro será aquele que resultar na estimação do sistema híbrido com o maior valor de VAF.

3.6 ALGORITMO 4

Os conhecimentos adquiridos no desenvolvimento dos algoritmos anteriores motivaram a proposição de um novo algoritmo inspirado nas ideias do algoritmo de *Expectation Maximization* [123]. Nesta abordagem o objetivo é maximizar a FDP a posteriori conjunta dos parâmetros e da segmentação. Essa maximização é realizada através de um procedimento iterativo que consiste em realizar maximizações alternadas da FDP a posteriori dos parâmetros contínuos seguida da maximização da FDP a posteriori dos parâmetros discretos. O processo termina quando os parâmetros do sistema não sofrem mais alterações significativas entre duas iterações consecutivas. A estrutura desta metodologia foi apresentada na Figura 3.1.

A nova metodologia que será apresentada no Algoritmo 4 também procurou reduzir a influência dos parâmetros iniciais no desempenho do algoritmo, em especial do parâmetro W . Para isso, o método utilizado para segmentação inicial dos dados foi alterado. O novo método utiliza uma janela móvel no tempo. Esta janela é aplicada ao conjunto de dados amostrados e se desloca em apenas uma unidade a cada conjunto criado.

Esta nova regra também ataca um problema adicional dos algoritmos anteriores que é a necessidade de grande quantidade de dados no processo de identificação. O algoritmo de clustering precisa de uma quantidade razoável de pontos para fornecer resultados confiáveis, do contrário os outliers podem ter grande influência no resultado. Desta forma, nos algoritmos anteriores era necessário uma enorme quantidade de dados no processo de identificação devido ao método utilizado para segmentação dos dados (conjuntos formados de forma que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$). A nova regra de segmentação desenvolvida permite a redução na quantidade de dados amostrados mas mantém elevada a quantidade de pontos enviados ao algoritmo de clustering.

Para entender a ideia da nova metodologia proposta é necessário retornar ao Problema 3.1.

Este problema pode ser interpretado de uma maneira diferente se os parâmetros desconhecidos forem considerados como variáveis aleatórias e descritos através de sua FDP. Assim, o Problema 3.1 se torna um problema de se determinar a FDP condicional conjunta dos parâmetros e da segmentação. Esta FDP é definida como:

$$p(\Theta_1, \dots, \Theta_M, S_1, \dots, S_N | u_1, \dots, u_N, y_1, \dots, y_N) = p(\Theta, \mathbf{S} | \mathbf{U}, \mathbf{Y}), \quad (3.19)$$

em que M é a quantidade de modelos locais, Θ_i são os parâmetros dos modelos e S_i é chamado de mapa de segmentação e indica para cada par de dados (u_i, y_i) qual é o seu modelo correspondente, ou seja, $S_i : \mathcal{D} \rightarrow \{1, \dots, M\}$, onde $\mathcal{D} = \{(u_k, y_k)\}_{k=1}^N$. No lado direito desta equação, utilizou-se $\Theta = \{\Theta_1, \dots, \Theta_M\}$, $\mathbf{S} = \{S_1, \dots, S_N\}$, $\mathbf{U} = \{u_1, \dots, u_N\}$ e $\mathbf{Y} = \{y_1, \dots, y_N\}$ para obter uma representação mais compacta.

A FDP (3.19) contém a informação estatística completa sobre os parâmetros e a segmentação que podem ser inferidas a partir dos dados observados e de algumas informações disponíveis a priori [29]. Em conjunto com o sistema (3.1), a FDP (3.19) pode ser utilizada para prever o valor da saída y_k no sentido que pode-se obter a função densidade de probabilidade p_{y_k} . Neste sentido, (3.1) e (3.19) formam um modelo completo do sistema. Estimativas pontuais dos parâmetros e da segmentação podem ser obtidas, por exemplo, através da estimativa de máxima verossimilhança:

$$\{\Theta^*, \mathbf{S}^*\} = \arg \max p(\Theta, \mathbf{S} | \mathbf{U}, \mathbf{Y}). \quad (3.20)$$

Este problema envolve a otimização conjunta dos parâmetros e da segmentação, tornando-o um difícil problema de otimização não-convexa, com vários pontos de máximo locais cuja a solução é numericamente inviável [29]. Portanto, será realizada uma abordagem alternativa.

Utilizando-se regras de probabilidade básica, a FDP (3.19) pode ser fatorada da seguinte forma:

$$p(\Theta, \mathbf{S} | \mathbf{U}, \mathbf{Y}) = p(\mathbf{S} | \Theta, \mathbf{U}, \mathbf{Y}) \cdot p(\Theta | \mathbf{U}, \mathbf{Y}). \quad (3.21)$$

Neste caso, o problema de estimação dos parâmetros e segmentação seria formulado da seguinte forma:

$$\{\Theta^*, \mathbf{S}^*\} = \arg \max \left\{ \underbrace{p(\mathbf{S} | \Theta, \mathbf{U}, \mathbf{Y})}_{\text{termo 1}} \cdot \underbrace{p(\Theta | \mathbf{U}, \mathbf{Y})}_{\text{termo 2}} \right\}. \quad (3.22)$$

Portanto, essa estimativa de verossimilhança consiste na maximização dos dois termos destacados na equação (3.22). Analisando cuidadosamente esses dois termos, pode-se obter importantes conclusões. No primeiro termo, $p(\mathbf{S} | \Theta, \mathbf{U}, \mathbf{Y})$, deseja-se obter o mapa de segmentação do sistema baseado nos dados amostrados e nos modelos locais do sistema. No entanto, isso é exatamente o que é realizado pelo algoritmo IMM descrito na Seção 2.3. Este

algoritmo, baseado nos dados amostrados e nos modelos fornecidos do sistema, utiliza um conjunto de filtros de Kalman operando em paralelo para estimar o modo mais provável para cada um dos dados amostrados (etapa de correção das probabilidades dos modos).

No segundo termo, $p(\Theta|U, Y)$ o interesse é na estimação dos modelos locais baseado apenas nos dados amostrados. Isso é realizado pelo Algoritmo 3 nas etapas 1 - 5, ou seja, sem nenhuma informação sobre o mapa de segmentação obtém-se estimativas dos modelos locais. Assim, essas etapas se caracterizam como um estimador para o máximo (ou moda) de $p(\Theta|U, Y)$.

Desta forma, o Algoritmo 3 pode ser visto como um algoritmo que visa obter o máximo (ou moda) de (3.22) através da maximização dos dois termos destacados de forma separada. Contudo, como o problema de estimação é híbrido e existe forte acoplamento entre o mapa de segmentação e estimação dos parâmetros, não é possível obter o máximo em apenas uma execução do algoritmo.

Desta forma, a ideia explorada no Algoritmo 4 é realizar a maximização de (3.22) por meio de iterações alternadas, buscando maximizar $p(\Theta|U, Y)$ seguido da maximização de $p(S|\Theta, U, Y)$. Em outras palavras, a cada iteração utiliza-se a estimativa corrente dos modelos locais do sistema (Θ) para obter um novo mapa de segmentação (S) através do algoritmo IMM. O novo mapa de segmentação é utilizado na formação de novos conjuntos de dados para encontrar um novo máximo de $p(\Theta|U, Y)$. Esse procedimento é realizado repetidamente até que as mudanças nos modelos locais sejam pouco significativas. Essa ideia é semelhante a utilizada no algoritmo EM [123].

É importante destacar que, o conhecimento de uma estimativa do mapa de segmentação proporciona uma melhora significativa na identificação dos modelos locais, pois pode-se eliminar ou reduzir a criação dos conjuntos mistos aplicando-se a janela móvel apenas nos conjuntos de dados pertencentes ao mesmo modo. Com uma quantidade menor de outliers, aumenta-se o desempenho do algoritmo de clustering e, conseqüentemente, melhoram-se as estimativas dos modelos locais. Com efeito, consegue-se eliminar a influência do parâmetro $\epsilon_{cluster}$, uma vez que os pontos duvidosos vão sendo eliminados naturalmente do processo de identificação a cada iteração. Isso será ilustrado através de exemplos numéricos na Seção 3.6.1.

As diferentes fases deste algoritmo estão ilustradas na Figura 3.18 e serão detalhadas a seguir. No entanto, ao contrário dos casos anteriores, as diferentes fases do algoritmo serão repetidas até que se obtenha convergência dos parâmetros dos modelos.

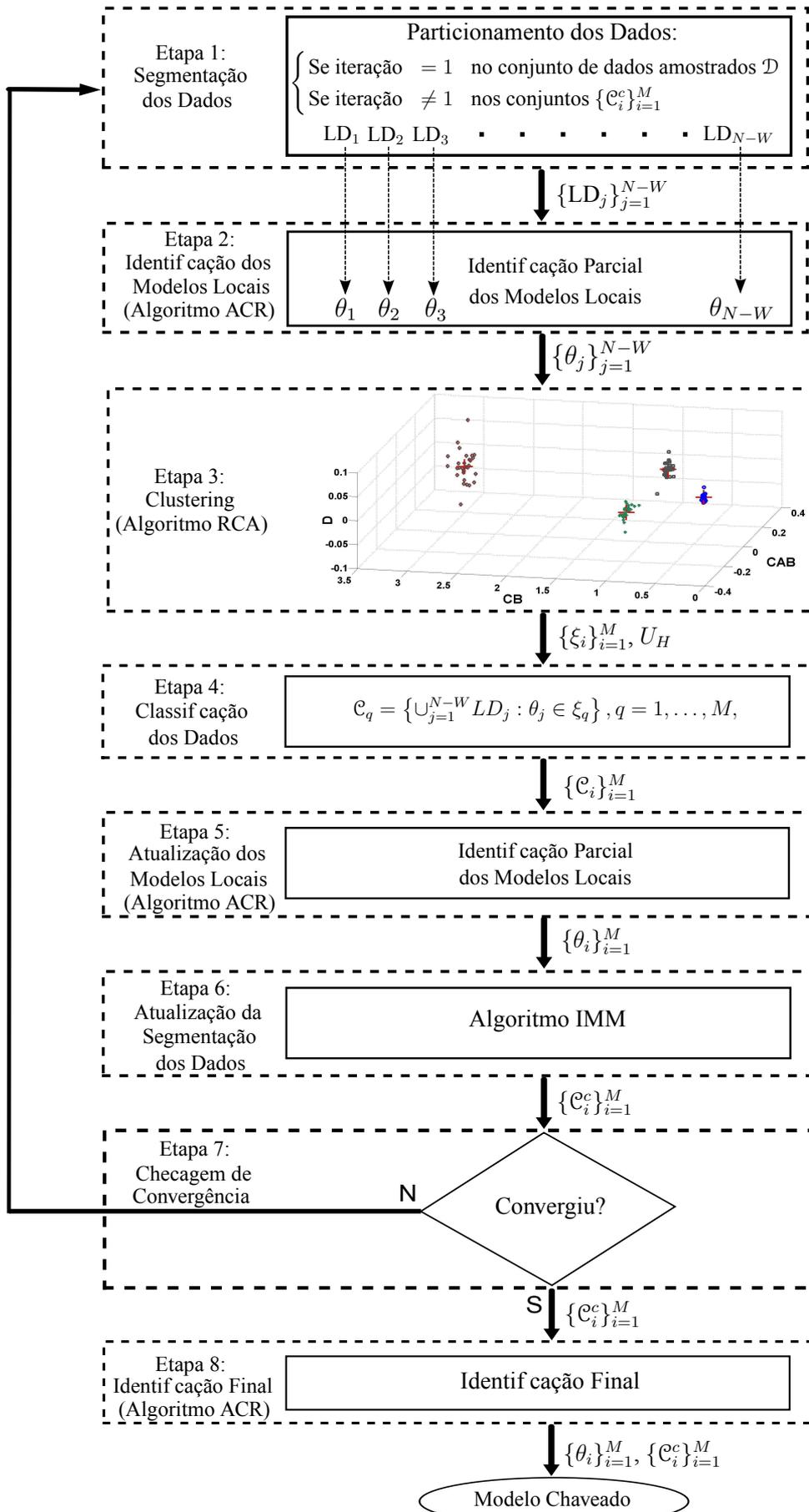


Figura 3.18: Diagrama do Algoritmo 4.

Etapa 1: Segmentação dos Dados:

Assim como nos algoritmos anteriores a primeira etapa do processo de identificação é a segmentação dos dados amostrados \mathcal{D} em pequenos subconjuntos. Para isso, novamente utiliza-se um particionamento que separa os dados em pequenos conjuntos com W elementos. No entanto, agora é utilizado uma janela móvel no tempo que cria esses conjuntos de acordo com a seguinte regra:

$$\mathcal{T}_j - \mathcal{T}_q = \{j\}, \quad \forall j = q + 1. \quad (3.23)$$

Isso significa que a janela é aplicada ao conjunto de dados amostrados mas se deslocando de amostra em amostra. Os índices de tempo em \mathcal{T} são utilizados para a formação dos conjuntos $\{\text{LD}_q\}_{q=1}^{N-W}$. Note que, com a nova regra a quantidade de subconjuntos formados é igual a $N - W$. Conforme mencionado, essa alteração visa reduzir a influência do parâmetro W no desempenho do algoritmo e reduzir a quantidade de dados necessários para o processo de identificação. No entanto, é importante observar que, mesmo reduzindo a quantidade de dados amostrados, a alteração proposta mantém elevada a quantidade de pontos enviados ao algoritmo de clustering. Como resultado, é possível manter a eficiência do RCA.

Uma outra diferença significativa desta etapa em relação aos algoritmos anteriores é que o conjunto de dados no qual a janela móvel é aplicada muda de acordo com a iteração. Na primeira iteração, quando não há nenhuma estimativa do mapa de segmentação (\mathbf{S}) disponível, a janela móvel é aplicada em todo o conjunto de dados amostrados, como nos algoritmos anteriores.

No entanto, após a primeira iteração, já existe uma estimativa do mapa de segmentação (fornecido pelo algoritmo IMM). Assim, a janela móvel é aplicada de forma separada nos diferentes conjuntos de dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$.

O objetivo desta alteração é minimizar a criação dos conjuntos mistos, melhorando, conseqüentemente, a identificação posterior dos modelos locais. Obviamente, haverá ainda erros na classificação dos pontos que provocará a existência de outliers na segmentação. Contudo, espera-se que esses outliers diminuam a cada iteração realizada.

Etapas 2 e 3:

As etapas de Identificação dos Modelos Locais (Etapa 2) e Clustering (Etapa 3) permanecem inalteradas do Algoritmo 3.

Etapa 4: Classificação dos Dados

No Algoritmo 3, a influência dos outliers no processo de identificação foi minimizada através da inclusão do parâmetro $\epsilon_{\text{cluster}}$ na etapa de classificação dos dados. Esse parâmetro era utilizado para selecionar somente os pontos que pertenciam a um determinado cluster com alto grau de confiança. Os resultados obtidos foram satisfatórios. Contudo, observou-se

que a eficiência do algoritmo era diretamente influenciada pelo ajuste inicial deste parâmetro. E, conforme discutido, o problema de ajuste inicial deste parâmetro é uma tarefa difícil e tediosa.

Assim, para eliminar esse problema, no Algoritmo 4, o parâmetro $\epsilon_{\text{cluster}}$ foi retirado do processo de classificação dos dados. Neste algoritmo, a influência dos outliers é reduzida utilizando-se a estrutura iterativa e o novo processo de segmentação descrito na Etapa 1. Espera-se que a cada iteração a quantidade de outliers seja reduzida.

Desta forma, o processo de atualização da segmentação dos dados é realizado como no Algoritmo 1, ou seja, é utilizada a matriz U_H fornecida pelo RCA e cada ponto θ_j é atribuído ao cluster no qual ele apresenta maior grau de pertinência. Os conjuntos de dados, denotados por $\{\mathcal{C}_i\}_{i=1}^M$, são criados de acordo com a seguinte regra:

$$\mathcal{C}_q = \left\{ \bigcup_{j=1}^{N-W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M, \quad (3.24)$$

lembra-se que ξ_q representam os clusters identificados pelo RCA.

Etapa 5: Atualização dos Modelos Locais

Esta etapa permanece inalterada do Algoritmo 3, ou seja, os dados agrupados nos clusters $\{\mathcal{C}_q\}_{q=1}^M$ são utilizados para estimação parcial dos modelos locais.

Etapa 6: Atualização da Segmentação dos Dados

Esta etapa também permanece inalterada do Algoritmo 3. No entanto, ressalta-se que esta etapa fornece uma estimativa para o mapa de segmentação, uma vez que os conjuntos dos dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$ são formados utilizando-se as informações fornecidas pelo algoritmo IMM. Assim, o mapa de segmentação é formado como $\mathbf{S} = \mathcal{C}_1^c \cup \mathcal{C}_2^c \cup \dots \cup \mathcal{C}_M^c$.

Etapa 7: Checagem de Convergência

Esta etapa é a responsável por verificar a convergência do algoritmo. As etapas de 1 a 6 devem ser realizadas repetidamente até que seja observada a convergência do algoritmo. Diferentes critérios podem ser utilizados para se avaliar essa convergência, por exemplo, alterações pouco significativas no mapa de segmentação ou a distância entre os modelos estimados em duas iterações sucessivas for menor que um determinado valor.

No entanto, neste trabalho utilizou-se um critério mais simples que é avaliar, em iterações sucessivas, o valor de VAF obtido através da simulação dos modelos estimados. O algoritmo IMM fornece uma estimativa do mapa de segmentação e este é utilizado para simular o sistema. Esta simulação é realizada utilizando-se as estimativas dos modelos locais e o vetor de entrada usado no processo de identificação. De posse da saída simulada, calcula-se o valor de VAF com a saída real do sistema. Se a diferença do valor de VAF entre duas iterações sucessivas for menor que um limiar ξ_{conv} considera-se que o algoritmo convergiu.

Note que, se houver alterações significantes no mapa de segmentação e/ou nos parâmetros dos modelos estimados entre duas iterações sucessivas, o valor de VAF será bem diferente. Por outro lado, se os valores são próximos entre duas iterações significa que não houve alterações no mapa de segmentação e nos parâmetros dos modelos. Portanto, esse critério engloba a avaliação do mapa de segmentação e dos parâmetros dos modelos.

Contudo, a abordagem desenvolvida não fornece garantias de convergência do algoritmo, de forma que o número de iterações pode se tornar muito grande. Assim, para evitar uma computação excessiva inseriu-se o critério onde o número máximo de iterações não pode exceder um determinado valor $\xi_{\text{iterações}}$.

Se os critérios de convergência são atingidos o algoritmo avança para a Etapa 8, caso contrário retorna a Etapa 1 e todo o procedimento é repetido.

Etapa 8: Identificação Final dos Modelos Locais:

Quando os critérios de convergência são atingidos o algoritmo utiliza o último mapa de segmentação obtido para realizar a identificação final dos modelos locais.

Esta metodologia está resumida no Algoritmo 4. Neste algoritmo a variável k representa o número da iteração do algoritmo.

Algoritmo 4:

- 2 **enquanto** $VAF_k - VAF_{k-1} > \xi_{conv}$ **ou** $k < \xi_{iterações}$ **faça**
- 3 **Etapa 1 - Segmentação dos Dados:**
- a) Crie pequenos subconjuntos de dados $\{LD_j\}_{j=1}^{N-W}$ contendo W elementos, aplicando o particionamento nos seguintes conjuntos:
- se** $k=1$ **então**
- | No conjunto total de dados amostrados \mathcal{D} ;
- senão**
- └ Nos conjuntos de dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$;
- 4 **Etapa 2 - Identificação dos Modelos Locais:**
- a) Determine θ_j para cada conjunto de dados em $\{LD_j\}_{j=1}^{N-W}$ usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- 5 **Etapa 3 - Clustering:**
- a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N-W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA.
- 6 **Etapa 4 - Classificação dos Dados:**
- a) Forme os subconjuntos dos dados $\{\mathcal{C}_q\}_{q=1}^M$ da seguinte forma:
- $$\mathcal{C}_q = \left\{ \bigcup_{j=1}^{N-W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M.$$
- 7 **Etapa 5 - Atualização dos Modelos Locais:**
- a) Para cada conjunto de dados $\{\mathcal{C}_q\}_{q=1}^M$ estime os modelos locais usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
- 8 **Etapa 6 - Atualização da Segmentação dos Dados:**
- a) Com as estimativas dos modelos locais obtidas na Etapa 5, use o algoritmo IMM apresentado na Seção 2.3 para corrigir os conjuntos dos dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$
- 9 **Etapa 7: Checagem de Convergência:**
- a) Simule o sistema híbrido identificado utilizando o mapa de segmentação obtido na Etapa 6 e calcule o valor de VAF_k ;
- b) Calcule a diferença $VAF_k - VAF_{k-1}$;
- 10 **Etapa 8: Identificação Final dos Modelos Locais:**
- a) Identifique os modelos locais finais com os dados de $\{\mathcal{C}_q^c\}_{q=1}^M$, usando as metodologias descritas nas seções 2.1.1 e 2.1.2.
-

3.6.1 Resultados Numéricos

Para ilustrar a eficácia das modificações propostas, o Algoritmo 4 foi comparado com o Algoritmo 3. Para isso, foi considerado o mesmo sistema híbrido utilizado anteriormente e os mesmos cenários de simulação descritos na Seção 3.5.1. Contudo, para o Algoritmo 4, a quantidade de dados utilizado para a identificação foi reduzida para 1200 pontos. Esse número representa apenas 10% da quantidade de dados usada no algoritmo 3. É importante destacar que, não é possível utilizar o Algoritmo 3 com essa quantidade reduzida de dados, devido a pequena quantidade de pontos enviada ao algoritmo de clustering.

Os sinais de chaveamento, excitação e saída utilizados no processo de identificação estão mostrados na Figura 3.19. Os parâmetros utilizados no algoritmo de identificação em subespaços e no algoritmo IMM não foram alterados. No RCA utilizou-se dois valores diferentes para o parâmetro ϵ_0 . Na primeira iteração do algoritmo esse valor era igual a 5,0% do valor da cardinalidade robusta. Após a primeira iteração esse valor era reduzido para 2,5%.

Os parâmetros de avaliação de convergência do algoritmo foram definidos como $\xi_{\text{conv}} = 0,1$ e $\xi_{\text{iterações}} = 6$. O valor $\xi_{\text{iterações}} = 6$ foi estabelecido empiricamente observando-se que, na maioria das simulações, a convergência era atingida em até seis iterações.

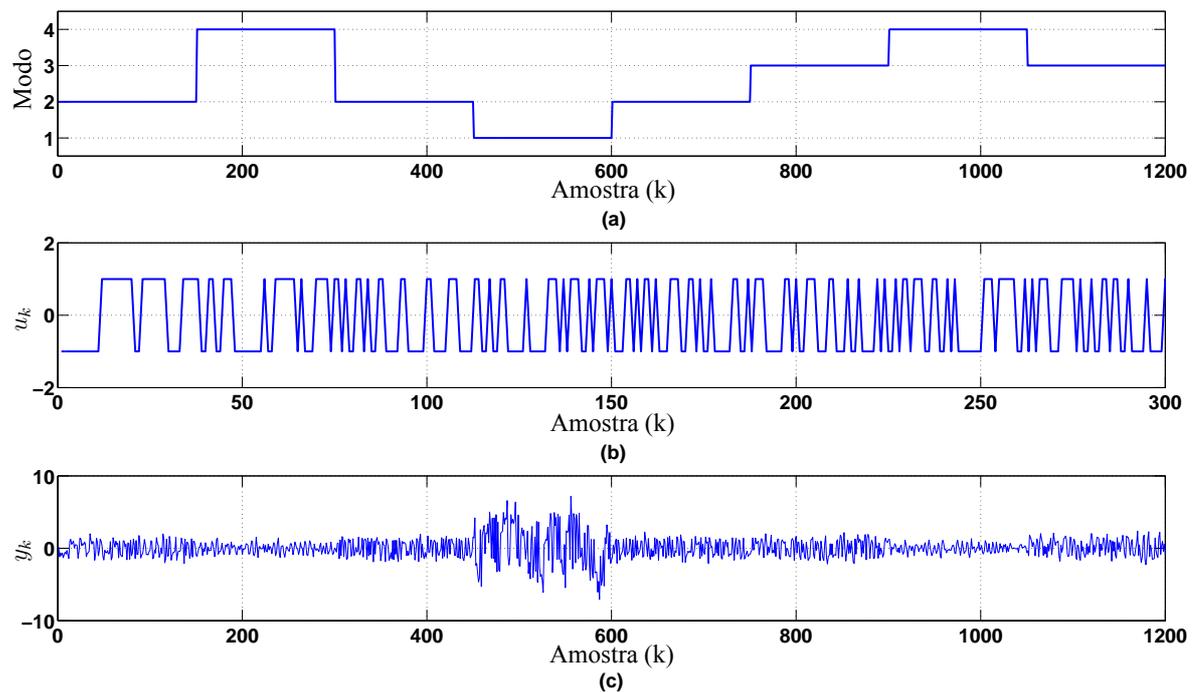


Figura 3.19: Sinais utilizados no processo de identificação com o Algoritmo 4. a) Sinal de chaveamento. b) Pequeno trecho do sinal de excitação. c) Saída do sistema.

Os resultados obtidos estão resumidos na Tabela 3.8. Nesta tabela estão apresentados novamente os resultados obtidos com o Algoritmo 3, descritos na Seção 3.5.1, para facilitar a comparação. Observa-se que o Algoritmo 4 apresenta um desempenho superior com relação

aos erros de identificação dos modos. De fato, o Algoritmo 4 apresentou uma redução média de 12,63% nos erros de classificação. Com efeito, os modelos estimados por esse algoritmo são mais acurados (maiores valores de VAF).

Adicionalmente, observa-se que as estimativas obtidas com o Algoritmo 4 apresentam uma dispersão menor. A Figura 3.20 comprova essa afirmação apresentando a distribuição dos valores de VAF ao longo das 100 simulações de Monte Carlo realizadas para os quatro níveis de ruído considerados. Observe que os histogramas das distribuições obtidas com o Algoritmo 4 possuem variância menor quando comparadas com o Algoritmo 3.

A Tabela 3.8 mostra ainda que ambos algoritmos são capazes de identificar corretamente o número de modos do sistema na maioria das simulações. Os resultados são essencialmente semelhantes.

Tabela 3.8: Comparação de performance dos Algoritmos 3 e 4 com diferentes níveis de ruído.

	Algoritmo 3				Algoritmo 4			
SNR (dB)	10	20	30	35	10	20	30	35
VAF (%)	73,88	91,66	97,08	98,22	74,63	91,33	97,10	98,39
Erro (%)	14,05	6,22	4,11	3,83	12,13	5,92	3,48	3,19
Successos (%)	97	100	100	100	96	99	100	100
T_{proc} médio (s)	15,47	12,58	18,20	12,80	58,06	41,46	41,67	39,72

Ressalta-se que o melhor desempenho do Algoritmo 4 foi obtido sem a utilização do parâmetro $\epsilon_{\text{cluster}}$. No caso do Algoritmo 3, esse desempenho só foi obtido quando muitos pontos duvidosos eram descartados do processo de identificação. Agora, nenhum ponto é descartado e, mesmo assim, a eficiência de identificação foi superior.

Isso ocorre porque a cada iteração o conhecimento a respeito do mapa de segmentação é utilizado para eliminar ou pelo menos reduzir a criação dos conjuntos mistos. A Figura 3.21 ilustra esse processo. Observe como a quantidade de outliers é reduzido a cada iteração, principalmente da primeira para a segunda iteração, evidenciando como o conhecimento do mapa de segmentação proporciona uma melhora significativa no processo de identificação. Na primeira iteração a quantidade excessiva de outliers deve-se ao desconhecimento, a priori, do mapa de segmentação.

A eliminação da dependência do parâmetro $\epsilon_{\text{cluster}}$ é um aspecto importante do Algoritmo 4, pois torna a tarefa de ajuste dos parâmetros iniciais menos árdua. De fato, observou-se uma facilidade maior no ajuste dos parâmetros do algoritmo. Adicionalmente, observou-se uma leve redução na influência do parâmetro ϵ_0 na determinação do número de modos do sistema. Em várias simulações, verificou-se que nas primeiras iterações do algoritmo o número de modos era identificado incorretamente (3 ou 5 sistemas identificados), mas após

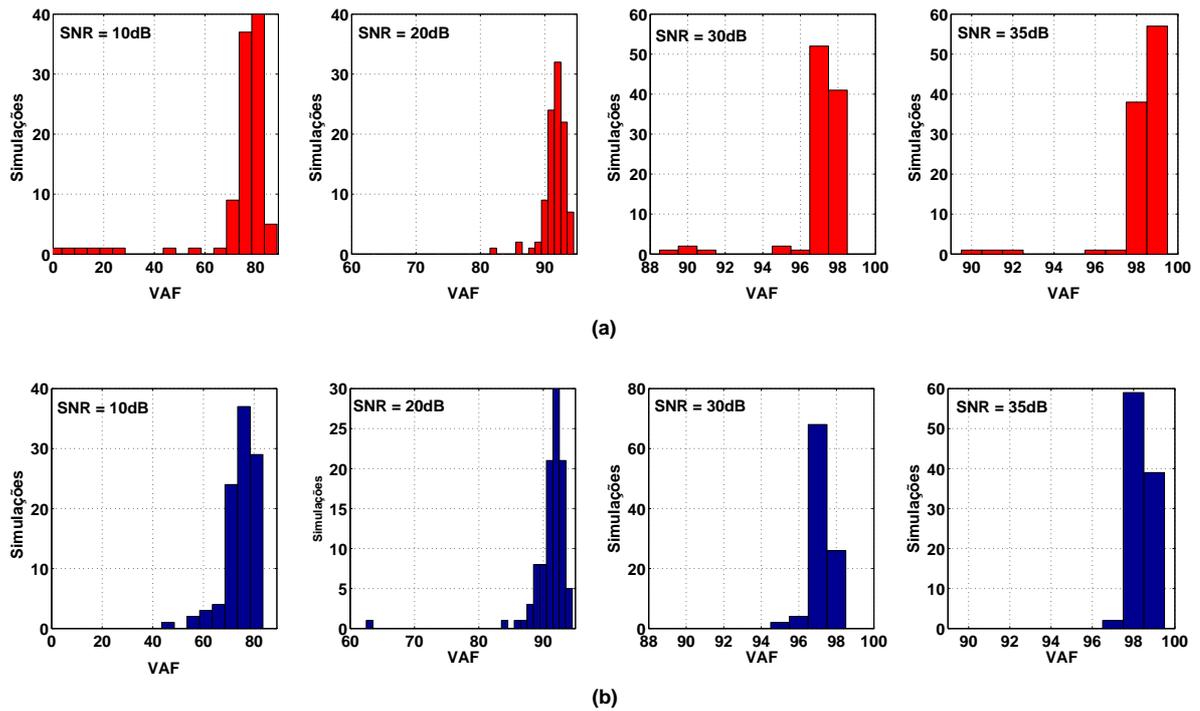


Figura 3.20: Comparação das distribuições dos valores de VAF nas 100 simulações realizadas nos diferentes níveis de ruído considerados. (a) Histogramas obtidos com o Algoritmo 3. (b) Histogramas obtidos com o Algoritmo 4.

algumas iterações esse valor convergia para o valor correto.

A principal desvantagem do Algoritmo 4 em relação aos seus antecessores é o aumento da carga computacional. No Algoritmo 1, o tempo médio gasto em cada identificação foi de 4,02 segundos. No Algoritmo 3, esse tempo sobe para 14,76 segundos. O aumento da carga computacional no terceiro algoritmo deve-se a utilização do algoritmo IMM. Já no Algoritmo 4 esse tempo é, em média, de 45,23 segundos¹. Esse valor representa um aumento de 206% na carga computacional com relação ao Algoritmo 3. Isso já era esperado, uma vez que o Algoritmo 4, basicamente, executa o Algoritmo 3 a cada iteração.

No entanto, esse aumento da carga computacional não é considerado um problema, pois a abordagem desenvolvida foca apenas no problema de identificação off-line onde todos os dados necessários já foram armazenados. Assim, não existe uma exigência de fornecer uma estimativa do sistema dentro de um determinado período de tempo. Desta forma, apesar do aumento do tempo de processamento, justifica-se a utilização do Algoritmo 4 em razão do ganho de desempenho na estimação dos modelos locais e maior facilidade no ajuste dos parâmetros iniciais do algoritmo.

Um problema adicional do Algoritmo 4 é a ausência da prova de convergência. Uma

¹Todas as simulações foram realizadas em um computador com processador Intel Core i7 com 2GHz de clock e 8GB de memória RAM.

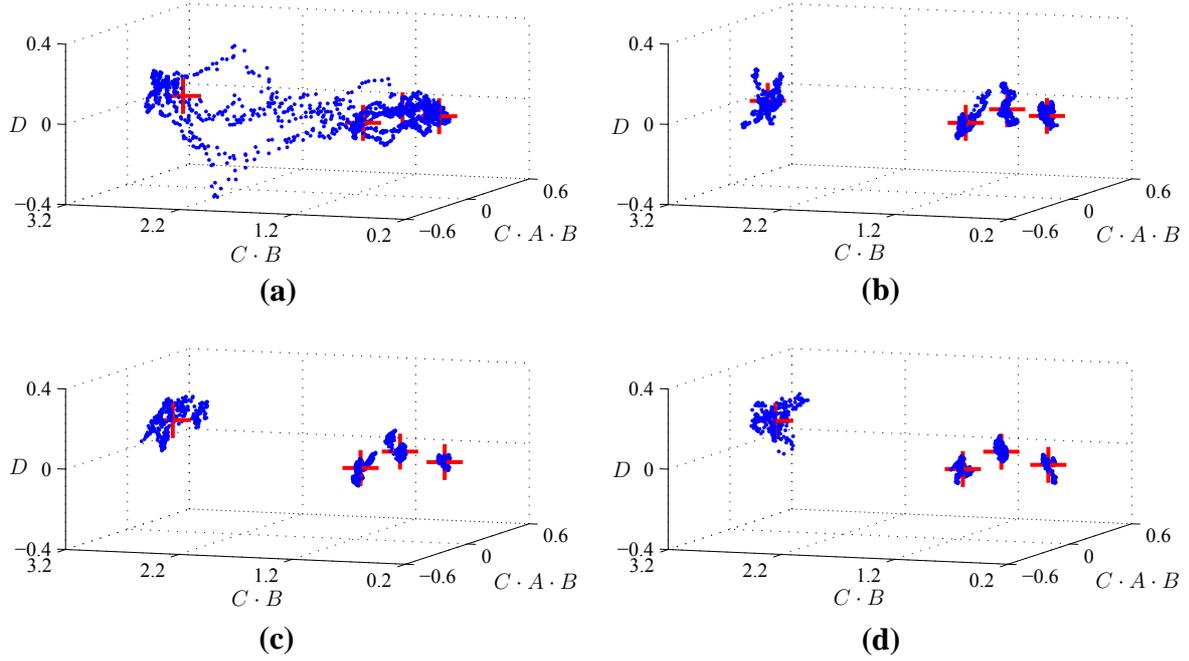


Figura 3.21: Resultado do algoritmo de clustering em diferentes iterações do algoritmo. (a) Primeira iteração. (b) Segunda iteração. (c) Terceira iteração. (d) Quarta iteração. No caso apresentado, o algoritmo convergiu após 4 iterações.

discussão mais detalhada sobre os problemas de convergência do algoritmo será postergada até a seção 3.8.1.2. Aqui é suficiente dizer que na maioria das simulações os parâmetros dos sistemas convergiam para uma solução válida. Adicionalmente, ressalta-se que nas simulações onde não observou-se a convergência dos parâmetros, após um determinado número de iterações o processo de maximização era encerrado e o último mapa de segmentação utilizado para a identificação final dos modelos locais. Assim, em todos os casos simulados foi possível obter uma estimativa consistente para os modelos do sistema.

3.7 ALGORITMO 5

A metodologia proposta na Figura 3.1 foi inicialmente desenvolvida para a identificação de sistemas dinâmicos híbridos chaveados. Nesta seção, serão descritas modificações nesta metodologia para que seja possível identificar sistemas dinâmicos híbridos chaveados afim por partes. Segundo [7], esses modelos podem ser descritos como:

$$\begin{cases} x_{k+1} = A_{m_k} x_k + B_{m_k} u_k + f_{m_k} + w_k, \\ y_k = C_{m_k} x_k + D_{m_k} u_k + g_{m_k} + v_k, \end{cases} \quad (3.25)$$

em que $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ refere-se ao estado discreto, e M é o número de submodelos. Os vetores $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$ e $y_k \in \mathbb{R}^{n_y}$ são, respectivamente, os estados contínuos, as entradas e saídas do sistema e n é a ordem dos modelos. As matrizes/vetores

$A_{m_k}, B_{m_k}, C_{m_k}, D_{m_k}, f_{m_k}, g_{m_k}$ tem dimensões apropriadas e descrevem as dinâmicas af ns dos sistemas locais, $w_k \in \mathbb{R}^n$, $w_k \sim N(0, Q_k)$ e $v_k \in \mathbb{R}^{n_y}$, com $v_k \sim N(0, R_k)$, são termos de ruído de processo e medida, respectivamente, considerados como sequencias de ruído branco ou colorido. com média zero.

O sistema (3.25) pode ser reescrito de forma que os modelos locais são combinados utilizando um vetor de pesos, ou um sinal de chaveamento, que determina qual modelo local está ativo no instante de tempo k , com a restrição que somente um modo está ativo a cada instante de tempo:

$$\begin{cases} x_{k+1} &= \sum_{i=1}^M p_{i(k)} (A_i x_k + B_i u_k + f_i) + w_k, \\ y_k &= \sum_{i=1}^M p_{i(k)} (C_i x_k + D_i u_k + g_i) + v_k, \end{cases} \quad (3.26)$$

em que $i \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$ e o sinal de chaveamento, $p_{i(k)}$, é def nido como,

$$p_{i(k)} \in \{0, 1\} \quad \text{e} \quad \sum_{i=1}^M p_{i(k)} = 1. \quad (3.27)$$

Desta forma, o objetivo do Algoritmo 5, que será apresentado nesta seção, é a identif cação de sistemas chaveados af m por partes descritos pela formulação (3.26). A importância da identif cação destes sistemas na estrutura proposta em (3.26) é que essa formulação, segundo [75], é equivalente à formulação de sistemas híbridos do tipo PWA.

Sistemas PWA constituem uma importante sub-classe de sistemas híbridos e são formulados de forma similar a (3.25) mas o estado discreto $m_k \in \mathbb{M} \triangleq \{1, 2, \dots, M\}$, é usado para relacionar a dinâmica do k -ésimo modo local a respectiva região poliédrica no espaço de entrada-estado \mathcal{P}_k . Ou seja, m_k é função da entrada u_k e do estado x_k do sistema no instante de tempo k , e é def nida como,

$$m_k(u_k, x_k) := \begin{cases} 1 & \text{se } (u_k, x_k) \in \mathcal{P}_1 \\ 2 & \text{se } (u_k, x_k) \in \mathcal{P}_2 \\ \vdots & \\ M & \text{se } (u_k, x_k) \in \mathcal{P}_M \end{cases}. \quad (3.28)$$

As partições $\mathcal{P}_i \subset \mathbb{R}^{n_u} \times \mathbb{R}^n$ do espaço de entrada-estado são def nidas da seguinte forma,

$$\bigcup_{i=1}^M \mathcal{P}_i = \mathbb{R}^{n_u} \times \mathbb{R}^n, \quad \text{e} \quad \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \text{para } i \neq j. \quad (3.29)$$

Conforme descrito na Seção 1.3, os principais trabalhos existentes na área de identif cação de sistemas dinâmicos híbridos focam neste tipo de sistema. Contudo, as abordagens são limitadas à modelos descritos na forma de entrada-saída SISO, ou, em alguns casos, MISO. A principal dificuldade na identif cação de sistemas híbridos do tipo PWA no espaço de estados é que a formulação (3.28) é baseada no particionamento do espaço de entrada-estado

e requer explicitamente a estimação do vetor de estados x_k a todo instante de tempo k . No entanto, quando os parâmetros dos modelos locais são desconhecidos a priori a estimação do vetor de estados é uma tarefa muito difícil.

Contudo, em [75] o autor demonstra que a formulação dos modelos PWA definida em (3.25) e (3.28) é equivalente ao sistema (3.26) quando as partições do espaço de entrada-estado $\{\mathcal{P}_i\}_{i=1}^M$ e os pesos $\{p_i(k)\}_{i=1}^M$ estão disponíveis a todo instante de tempo k . No entanto, do ponto de vista de identificação de sistemas, pode-se argumentar que a formulação (3.26) proporciona uma classe de sistemas mais efetiva, uma vez que não requer a estimação explícita do vetor de estados e a definição do particionamento como função do espaço de entrada-estado [75].

Desta forma, a metodologia que será apresentada nesta seção para a identificação de sistemas chaveados lineares afm por partes também pode ser utilizada para a identificação de sistemas híbridos do tipo PWA. Ao melhor conhecimento do autor, o único trabalho a abordar a identificação de sistemas PWA MIMO no espaço de estados foi publicado em [75]. Neste trabalho, a abordagem proposta pelos autores é baseada na detecção das variações dos subespaços lineares nos momentos em que ocorrem transições entre os modos do sistema. Contudo, esta abordagem requer um número suficientemente grande de transições entre os diferentes modos para que seja possível representar todas as matrizes dos submodelos na mesma base de estados. Desta forma, a abordagem proposta no presente trabalho resolve este problema e permite que seja aplicada para a identificação de sistemas onde ocorrem poucos chaveamentos entre os modos do sistema.

Para o desenvolvimento desta nova metodologia foi considerado a estrutura de identificação mais simples apresentada no Algoritmo 1. Esta escolha foi feita por questões didáticas de forma a facilitar o entendimento do novo método. Ressalta-se que o principal objetivo é apenas demonstrar a viabilidade da metodologia para a identificação de sistemas chaveados afm por partes. Os demais algoritmos apresentados anteriormente também podem ser adaptados de forma a melhorar os resultados apresentados nesta seção.

Para a identificação de sistemas chaveados afm por partes, as únicas modificações necessárias no Algoritmo 1 são relacionadas a etapa de Identificação dos Modelos Locais (etapas 2 e 5). A estrutura de identificação e as modificações propostas serão descritas abaixo. Para diferenciar as duas propostas o novo algoritmo será denominado de Algoritmo 5. As principais etapas do algoritmo proposto estão descritas abaixo.

Etapa 1: Segmentação dos Dados:

Esta etapa permanece igual a descrita no Algoritmo 1.

Etapa 2 - Identificação dos Modelos Locais

Esta etapa apresenta as principais modificações com relação ao Algoritmo 1. O objetivo

aqui continua sendo a identificação parcial dos modelos locais. Para isso, para cada conjunto de dados $\{LD_j\}_{j=1}^{N/W}$ criado na etapa anterior é identificado um modelo j através de uma técnica de identificação de sistemas.

Cada modelo identificado constituirá uma observação que será enviada ao algoritmo de cluster para segmentação. Contudo, nesta abordagem é proposto um vetor de parâmetros de Markov modificado. A principal motivação é incluir os efeitos dos vetores f_j e g_j . Este novo vetor é definido como:

$$\tilde{\theta}_j = \text{vec} \left(\begin{bmatrix} C_j A_j^{p-2} B_j & \cdots & C_j B_j & D_j & C_j f_j f_j^T B_j & g_j \end{bmatrix} \right) \quad (3.30)$$

em que $\tilde{\theta}_j \in \mathbb{R}^{n_y n_u (p+1) + n_y \times 1}$.

Observe que, a estrutura do vetor proposto em (3.30) possibilita a estimação de sistemas de diferentes ordens mantendo a mesma dimensão do vetor θ_j . Isso é fundamental para a estrutura de identificação proposta, uma vez que não existem garantias que o algoritmo de estimação de ordem irá estimar sistemas com ordens iguais.

No contexto de identificação de sistemas, não é possível utilizar a metodologia de identificação em subespaços apresentada na seção 2.1.1 para a identificação dos modelos locais, pois ela não permite a estimação dos termos afins f_i e g_i . Assim, nesta metodologia será utilizado um algoritmo proposto por [75].

Este algoritmo é uma adaptação do algoritmo PO-MOESP proposto por [129] e permite a identificação dos termos afins dos modelos locais. Uma vantagem adicional deste algoritmo é a possibilidade de identificação de sistemas corrompidos com ruído de processo e medida colorido. A seguir será realizada uma breve descrição, sem provas, desta metodologia. Para maiores detalhes deve-se consultar o trabalho original.

3.7.1 Algoritmo PO-MOESP adaptado

O sistema (3.26) pode ser reescrito utilizando a representação na forma de inovações,

$$\begin{aligned} x(k+1) &= \sum_{i=1}^M p_i(k) (A_i x(k) + B_i u(k) + f_i) + K_i e(k), \\ y(k) &= \sum_{i=1}^M p_i(k) (C_i x_k + D_i u_k + g_i) + e(k), \end{aligned} \quad (3.31)$$

em que K_i é o ganho de Kalman para o i -ésimo modo do sistema e $e(k)$ é o termo de inovação.

Considerando um conjunto de N_w saídas produzidas por um único modelo i (ou, seja, nenhum chaveamento ocorreu no sistema), as saídas do sistema em (3.31) podem ser combinadas em uma forma matricial mais compacta:

$$Y_{k_0, s, N_w} = \Gamma_{k_0, s} X_{k_0, N_w} + \tilde{\Phi}_{k_0, s} \tilde{U}_{k_0, s, N_w} + \Xi_{k_0, s} E_{k_0, s, N_w}, \quad (3.32)$$

em que, as matrizes que utilizam a notação $(\cdot)_{k_0, s, N_w}$ tem uma estrutura comum e são normalmente chamadas de matrizes de *block Hankel*. O sub-escrito k_0 denota o instante de tempo da entrada superior esquerda da matriz, N_w é o número de colunas e $s \ll N_w$ o número de linhas. As matrizes de *Hankel* são definidas da seguinte maneira:

$$Y_{k_0, s, N_w} := \begin{bmatrix} y(k_0) & y(k_0 + 1) & \cdots & y(k_0 + N_w - 1) \\ y(k_0 + 1) & y(k_0 + 2) & \cdots & y(k_0 + N_w) \\ \vdots & \vdots & \ddots & \vdots \\ y(k_0 + s - 1) & y(k_0 + s) & \cdots & y(k_0 + N_w + s - 2) \end{bmatrix}, \quad (3.33)$$

com U_{k_0, s, N_w} e E_{k_0, s, N_w} construídas de forma semelhante mas utilizando os dados de entrada $u(k)$ e os termos de inovação $e(k)$, respectivamente, e

$$\tilde{U}_{k_0, s, N_w} := \begin{bmatrix} \mathbf{1} \\ U_{k_0, s, N_w} \end{bmatrix}, \quad (3.34)$$

em que o termo unitário é definido como $\mathbf{1} := [1 \ 1 \ 1 \ \cdots \ 1]$.

A matriz com o vetor de estados é definida como:

$$X_{k_0, N_w} := \text{diag}(x(k_0), (k_0 + 1), \dots, x(k_0 + N_w - 1)). \quad (3.35)$$

$$\Xi_{k_0, s} := \begin{bmatrix} I_{n_y} & 0 & \cdots & 0 \\ \xi(k_0, k_0 + 1) & I_{n_y} & \cdots & 0 \\ \xi(k_0, k_0 + 2) & \xi(k_0 + 1, k_0 + 2) & \ddots & \vdots \\ \vdots & \vdots & \ddots & I_{n_y} \\ \xi(k_0, k_0 + s - 1) & \xi(k_0 + 1, k_0 + s) & \cdots & I_{n_y} \end{bmatrix}, \quad (3.36)$$

em que,

$$\xi(t, k) := \sum_{i=1}^M p_i(k) C_i \left(\prod_{\tau=t+1}^{k-1} \sum_{j=1}^M p_j(\tau) A_j \right) p_i(t) K_i, \quad (3.37)$$

A matriz de observabilidade $\Gamma_{k_0, s}$ é definida da seguinte forma:

$$\Gamma_{k_0, s} := \begin{bmatrix} \Gamma_{k_0, k_0}^T & \Gamma_{k_0, k_0+1}^T & \cdots & \Gamma_{k_0, k_0+s-1}^T \end{bmatrix}^T, \quad (3.38)$$

em que

$$\Gamma(k_0, k) := \sum_{i=1}^M p_i(k) C_i \left(\prod_{\sigma=k_0}^{k-1} \sum_{\tau=1}^M p_\tau(\sigma) A_\tau \right), \quad (3.39)$$

Por fim, o termo $\tilde{\Phi}_{k_0, s}$ é definido como,

$$\tilde{\Phi}_{k_0, s} := \begin{bmatrix} \mathcal{O}_{k_0, s} & \Phi_{k_0, s} \end{bmatrix}, \quad (3.40)$$

onde,

$$\Phi(t, k) := \sum_{i=1}^M p_i(k) C_i \left(\prod_{\tau=t+1}^{k-1} \sum_{j=1}^M p_j(\tau) A_j \right) p_i(t) B_i, \quad (3.41)$$

e,

$$\mathcal{O}_{k_0,s} = \begin{bmatrix} g_i \\ C_i f_i + g_i \\ C_i(A_i + I)f_i + g_i \\ \vdots \\ C_i \left(\sum_{j=k_0}^{s-2} (A_i)^j + I \right) f_i + g_i \end{bmatrix} \mathbf{1}. \quad (3.42)$$

A equação (3.32) permite que as matrizes do sistema possam ser estimadas utilizando-se o método PO-MOESP de identificação em subespaços. Este método inicia-se com o cálculo da seguinte fatoração RQ:

$$\begin{bmatrix} \tilde{U}_{s,s,N_w} \\ Z_{k,s,N_w} \\ Y_{s,s,N_w} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}, \quad (3.43)$$

em que Z_{k,s,N_w} é a matriz de variáveis instrumentais definida como:

$$Z_{k,s,N_w} := \begin{bmatrix} U_{k,s,N_w} \\ Y_{k,s,N_w} \end{bmatrix}. \quad (3.44)$$

As matrizes A_i e C_i podem ser recuperadas através do espaço coluna de R_{32} . Esse espaço coluna é determinado através da seguinte decomposição em valores singulares,

$$R_{32} = \begin{bmatrix} \mathcal{U} & \mathcal{U}_\perp \end{bmatrix} \begin{bmatrix} \mathcal{S} & 0 \\ 0 & \mathcal{S}_\perp \end{bmatrix} \begin{bmatrix} \mathcal{V} \\ \mathcal{V}_\perp \end{bmatrix}, \quad (3.45)$$

em que \mathcal{S} é uma matriz diagonal contendo os n valores singulares dominantes. A determinação da ordem do sistema consiste em identificar esses n valores singulares dominantes. Contudo, conforme já discutido anteriormente, esta não é uma tarefa trivial e para a estimação da ordem dos modelos locais será utilizado o algoritmo de estimação de ordem apresentado na seção 2.1.3.

Desta forma, uma estimativa das matrizes A_i e C_i pode ser obtida da seguinte forma,

$$\hat{C}_i = \mathcal{U}(1 : n_y, :), \quad (3.46)$$

$$\hat{A}_i = (\mathcal{U}(1 : (s-1)n_y, :))^\dagger \mathcal{U}(n_y + 1 : sn_y, :), \quad (3.47)$$

em que é utilizada a notação como no Matlab ($1 : n_y, :$) para se referir as primeiras n_y linhas e $(\cdot)^\dagger$ denota a pseudo-inversa de Moore-Penrose de (\cdot) .

As estimativas das matrizes \hat{B}_i , \hat{D}_i , \hat{f}_i e \hat{g}_i são obtidas através da resolução de um conjunto de equações lineares,

$$\hat{\Phi}_{k,s} = \begin{bmatrix} \hat{O}_{k,s} & \hat{\Phi}_{k,s} \end{bmatrix} = R_{21} R_{11}^{-1}. \quad (3.48)$$

Por fim, destaca-se que o algoritmo de identificação descrito foi combinado com a metodologia descrita na seção 2.1.2 para manter o controle da base de estados nas quais as

matrizes do sistema são calculadas. Adicionalmente, é utilizado o algoritmo de estimação de ordem descrito na seção 2.1.3 para estimar a ordem dos modelos identificados.

É importante destacar que, esta metodologia permite a identificação de sistemas dinâmicos chaveados lineares. Neste caso, a estimação $\mathcal{O}_{k,s}$ em (3.48) não é necessária. Desta forma, a metodologia de identificação em subespaços aqui apresentada pode ser utilizada nos Algoritmos 1 a 4 descritos anteriormente.

Etapas 3 a 4:

As etapas de clustering (etapa 3) e de classificação dos dados (etapa 4) permanecem inalteradas do Algoritmo 1.

Etapa 5: Identificação Final dos Modelos Locais

O último passo do algoritmo é a identificação final do sistema. Esta etapa é semelhante à do Algoritmo 1, com a diferença que é utilizada a metodologia descrita na seção 3.7.1 combinada com as metodologias descritas nas seções 2.1.2 e 2.1.3 para a identificação dos modelos locais.

As etapas descritas estão resumidas no Algoritmo 5.

Algoritmo 5:

1: Segmentação dos Dados:

- a) Crie pequenos conjuntos de dados $\{LD_j\}_{j=1}^{N/W}$, de forma que $LD_j = \{(u_q, y_q) \in \mathcal{D} : q \in \mathcal{T}_j\}$, e \mathcal{T}_j é formado pela divisão do horizonte de tempo \mathcal{T} em conjuntos menores tais que $\mathcal{T}_j \cap \mathcal{T}_q = \emptyset, \forall j \neq q$

2: Identificação dos Modelos Locais:

- a) Determine $\tilde{\theta}_j$ para cada conjunto de dados em $\{LD_j\}_{j=1}^{N/W}$ usando as metodologias descritas nas seções 3.7.1 e 2.1.2. Sendo que $\tilde{\theta}_j$ definido em (3.30).
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.

3: Clustering:

- a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N/W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA apresentado na Seção 2.2

4: Classificação dos Dados:

- a) Forme os subconjuntos dos dados clusterizados $\{\mathcal{C}_q\}_{q=1}^M$ da seguinte forma:
$$\mathcal{C}_q = \left\{ \cup_{j=1}^{N/W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M.$$
- b) Estime os instantes de chaveamento do sistema procurando no conjunto de dados amostrados (u_k, y_k) pontos consecutivos que pertencem a diferentes modelos locais.

5: Identificação Final dos Modelos Locais:

- a) Identifique os modelos locais finais com os dados de $\{\mathcal{C}_q\}_{q=1}^M$, usando as metodologias descritas nas seções 3.7.1, 2.1.2.
- b) Utilize a metodologia apresentada na seção 2.1.3 para determinar a ordem dos modelos locais.
-

3.7.2 Resultados Numéricos

Nesta seção, o algoritmo de identificação proposto será testado empregando um exemplo de simulação. O sistema híbrido considerado é composto de três modelos MIMO de terceira ordem representados por:

$$A_1 = \begin{bmatrix} 0,32 & 0,31 & 0 \\ -0,32 & 0,31 & 0 \\ 0 & 0 & -0,18 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0,9 & -0,7 \\ 0,71 & -0,5 \\ 0,8 & 0,43 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0,5 \\ 1 \\ 0 \end{bmatrix},$$
$$C_1 = \begin{bmatrix} -0,55 & 0,2 & 0,8 \\ 0,45 & 0,3 & 0,58 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad g_1 = \begin{bmatrix} 0,3 \\ 1 \end{bmatrix}.$$

$$\begin{aligned}
A_2 &= \begin{bmatrix} -0,1 & -0,4 & 0 \\ 0,5 & -0,4 & 0 \\ 0 & 0 & 0,26 \end{bmatrix}, & B_2 &= \begin{bmatrix} 0,1 & -0,6 \\ 0,32 & -0,66 \\ 0,3 & 0,82 \end{bmatrix}, & f_2 &= \begin{bmatrix} 1 \\ 0,1 \\ 0 \end{bmatrix}, \\
C_2 &= \begin{bmatrix} -0,8 & -0,1 & 0,7 \\ 0,3 & 0,48 & 0,9 \end{bmatrix}, & D_2 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, & g_2 &= \begin{bmatrix} 0,1 \\ -0,5 \end{bmatrix}, \\
A_3 &= A_1, & B_3 &= B_1, & C_3 &= C_1, & D_3 &= D_1, \\
f_3 &= \begin{bmatrix} 0 \\ 0,1 \\ 1 \end{bmatrix}, & g_3 &= \begin{bmatrix} 2 \\ 0,2 \end{bmatrix}.
\end{aligned}$$

Nas simulações geraram-se 15000 pontos de dados de $u(k)$ e $y(k)$ que são utilizados no processo de identificação dos sistemas. O sinal de excitação utilizado é um sinal PRBS de média zero e variância unitária. A sequência de chaveamento foi definida de forma que o sistema 1 está ativo nos intervalos de $[1, 499]$, $[1501, 4005]$ e $[10511, 13000]$, o sistema 2 entre $[500, 1500]$, $[6508, 7300]$, $[9734, 10510]$ e $[13001, 15000]$ e o sistema 3 nos intervalos $[4006, 6507]$ e $[7301, 9733]$. A Figura 3.22 ilustra o sinal utilizado. Observe que, este sinal de chaveamento foi definido respeitando as restrições impostas em (3.27) e o mínimo tempo de permanência em cada um dos modos (τ_{dwell}).

Dois diferentes cenários de simulação foram considerados. No primeiro, os sistemas estão sujeitos somente a ação de ruído branco. Neste cenário, considerou-se que o ruído de processo dos sistemas tem a proporção sinal ruído igual a $\text{SNR} = 40$ dB. Para o ruído de medição foram avaliados quatro diferentes níveis de ruído ($\text{SNR} = 10, 20, 30$ e 40 dB). No segundo cenário, os níveis de ruídos adicionados foram os mesmos descritos anteriormente, mas considerou-se que tanto o ruído de processo $w(k)$ quanto o de medição $v(k)$ adicionados aos sistemas são sequências de ruído colorido. A geração deste tipo de ruído foi realizada filtrando uma sequência de ruído branco. Para isto, utilizou-se o seguinte filtro:

$$v(k) = \frac{q^{-1} + 0,5q^{-2}}{1 - 1,69q^{-1} + 0,96q^{-2}}e(k), \quad (3.49)$$

em que, $e(k)$ é uma sequência de ruído branco com média nula e uma variância especificada.

Diferentemente dos exemplos anteriores, nestas simulações foi considerado que a ordem dos modelos locais é desconhecida a priori. Portanto, deve ser estimada junto com os modelos. Para isso, será utilizado o algoritmo de estimação de ordem descrito na Seção 2.1.3. É importante destacar que, esse algoritmo é chamado em dois momentos da metodologia. Primeiro na Etapa 2 durante a estimação parcial dos modelos locais. Contudo, a ordem final dos modelos locais é estimada somente na Etapa 5.

Os principais parâmetros necessários no algoritmo foram ajustados como descrito na Seção 3.3.1. A exceção foi o parâmetro ϵ do algoritmo RCA que foi definido igual a 12% da cardinalidade robusta no primeiro cenário e 11% no segundo cenário.

A avaliação de desempenho do algoritmo foi realizada através de simulações de Monte Carlo com 100 realizações para cada nível de ruído mencionado. Em cada simulação era utilizado uma realização diferente dos sinais de entrada, saída e ruído. Para avaliar a eficiência do algoritmo de identificação observou-se a sua capacidade em detectar corretamente o número de modos, classificar corretamente os dados amostrados nos diferentes modos do sistema, estimar corretamente a ordem dos modelos e a habilidade do modelo estimado em reconstruir a saída do sistema a partir do sinal de entrada real. Para isso utilizou-se o valor de VAF em um conjunto de dados diferente do utilizado para a identificação. Para essa validação, utilizou-se 1500 pontos de $u(k)$ e $y(k)$.

Os resultados obtidos estão resumidos nas Tabelas 3.9 e 3.10. Esses resultados indicam que o algoritmo apresentado é capaz de identificar os sistemas sujeitos a ação tanto de ruído branco quanto de ruído colorido. Adicionalmente, é possível observar que o algoritmo apresenta resultados semelhantes nos dois cenários simulados. A Figura 3.23 reforça esses resultados comparando os pólos dos modelos estimados com os pólos dos modelos reais para dois diferentes níveis e tipos de ruído. Observe como os valores estimados são próximos dos valores reais do sistema.

Tabela 3.9: Desempenho do algoritmo quando os sistemas estão sujeitos somente a ação de ruído branco. Aqui, o termo VAF1 (%) e VAF2 (%) indicam o cálculo do valor de VAF para as duas saídas do sistema. Ordem X (%) indica o número de vezes que o algoritmo estimou corretamente a ordem do sistema X.

SNR (dB)	10	20	30	40
VAF1 (%)	87,73	96,61	98,90	99,54
VAF2 (%)	80,67	92,85	96,76	98,04
Erro (%)	1,59	0,37	0,37	0,37
Sucessos (%)	95	100	100	100
Ordem 1 (%)	85	100	100	100
Ordem 2 (%)	88	99	100	100
Ordem 3 (%)	71	99	100	100

O algoritmo apresentou excelentes resultados com relação a classificação dos dados nos dois cenários simulados. De fato, observa-se nas Tabelas 3.9 e 3.10 que os erros médios de estimação ficaram abaixo de 5%. A Figura 3.22 ilustra esse bom desempenho apresentando um resultado típico para a estimação dos instantes de chaveamento do sistema. A consequência da classificação correta dos dados é a estimação de modelos acurados que são capazes de reconstruir de maneira eficiente o sinal de saída real. Isso pode ser confirmado pelos valores de VAF elevados mostrados nas Tabelas 3.9 e 3.10 e pelo gráfico mostrado na Figura 3.24.

Tabela 3.10: Desempenho do algoritmo para cenário onde os sistemas estão sujeitos a ação de ruído colorido.

SNR (dB)	10	20	30	40
VAF1 (%)	87,76	96,91	98,89	99,54
VAF2 (%)	79,14	92,85	96,72	98,03
Erro (%)	4,13	0,37	0,38	0,37
Sucessos (%)	80	100	100	100
Ordem 1 (%)	53	94	100	100
Ordem 2 (%)	55	95	100	100
Ordem 3 (%)	49	81	97	100

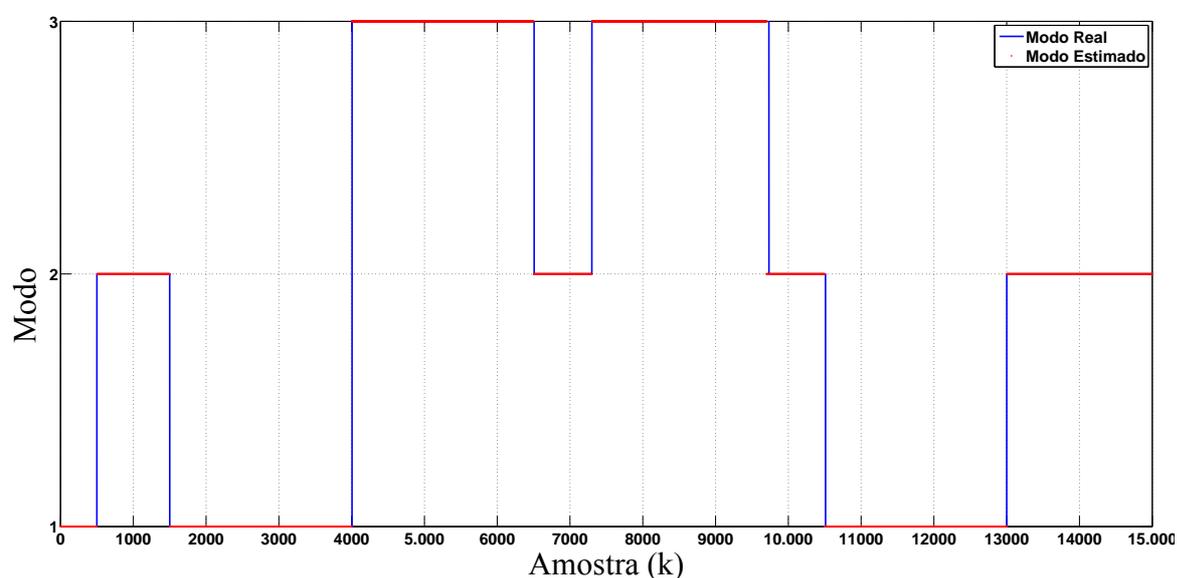


Figura 3.22: Típico resultado para a estimação dos modos. Neste caso, o erro na estimação foi de 0,37%.

Com relação a estimação da ordem dos sistemas, observa-se bons resultados para níveis moderados de ruído. Observe nas Tabelas 3.9 e 3.10 que para $SNR = 30$ e 40 dB o algoritmo identifica a ordem correta na maioria das simulações. No entanto, com níveis mais altos de ruído a eficiência do algoritmo é comprometida. Esse resultado já era esperado, uma vez que a estimação da ordem é realizada com base nos valores singulares das matrizes de *block-Hankel* do sistema. Para níveis moderados de ruído, é fácil separar os valores singulares relacionados à dinâmica do sistema dos relacionados ao ruído. Por outro lado, quando o nível de ruído é elevado é muito mais difícil fazer essa distinção. Contudo, ressalta-se que, no presente exemplo, mesmo nos casos onde a ordem não foi corretamente estimada, os sistemas identificados conseguem reconstruir adequadamente o sinal de saída real do sistema.

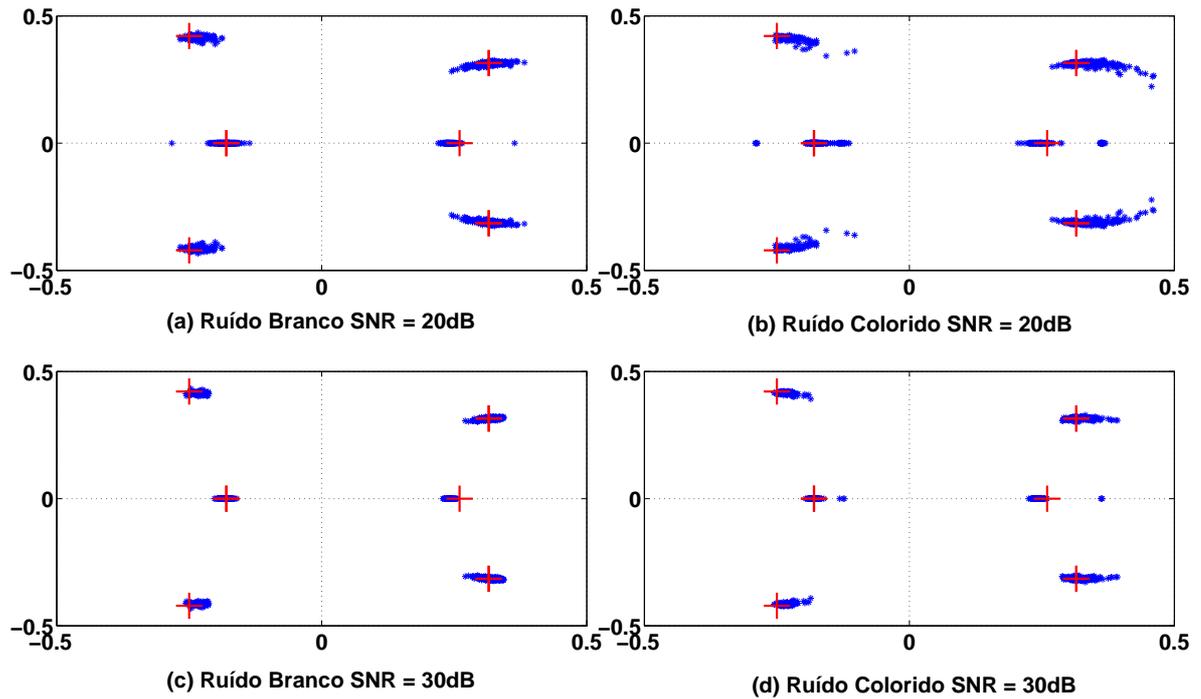


Figura 3.23: Pólos estimados dos sistemas no plano complexo para 100 simulações.

Nos resultados apresentados observa-se que o algoritmo consegue identificar o número correto de modos na maior parte das simulações. A exceção ocorre quando o nível de ruído adicionado ao sistema é muito elevado (SNR = 10 dB). Neste caso, o problema na identificação ocorre devido à estimação de ordem dos sistemas. Na Etapa 2 do algoritmo, como a quantidade de dados utilizados na identificação é pequena e o nível de ruído é muito elevado o algoritmo de estimação de ordem não apresenta bons resultados. Com efeito, observa-se uma maior dispersão nos vetores $\tilde{\theta}_j$ e isso prejudica a performance do algoritmo de clustering.

Desta forma, quando o nível de ruído do sistema é muito elevado, recomenda-se que a ordem dos modelos seja fixada na Etapa 2 do algoritmo de identificação. O objetivo é reduzir a dispersão dos vetores $\tilde{\theta}_j$. Contudo, como o valor da ordem é, a priori, desconhecido, deve-se especificar um valor que seja maior que a máxima ordem esperada do sistema. Na Etapa 5 do algoritmo, como uma quantidade maior de dados é utilizada para a identificação final dos modelos locais pode-se utilizar o algoritmo para estimar a ordem real dos sistemas.

Para comprovar a eficácia deste processo, as simulações com ruído elevado (SNR = 10 dB) foram repetidas fazendo-se (apenas na Etapa 2) a ordem dos sistemas em 5. Os resultados apresentados na Tabela 3.11 mostram a melhora no desempenho do algoritmo.

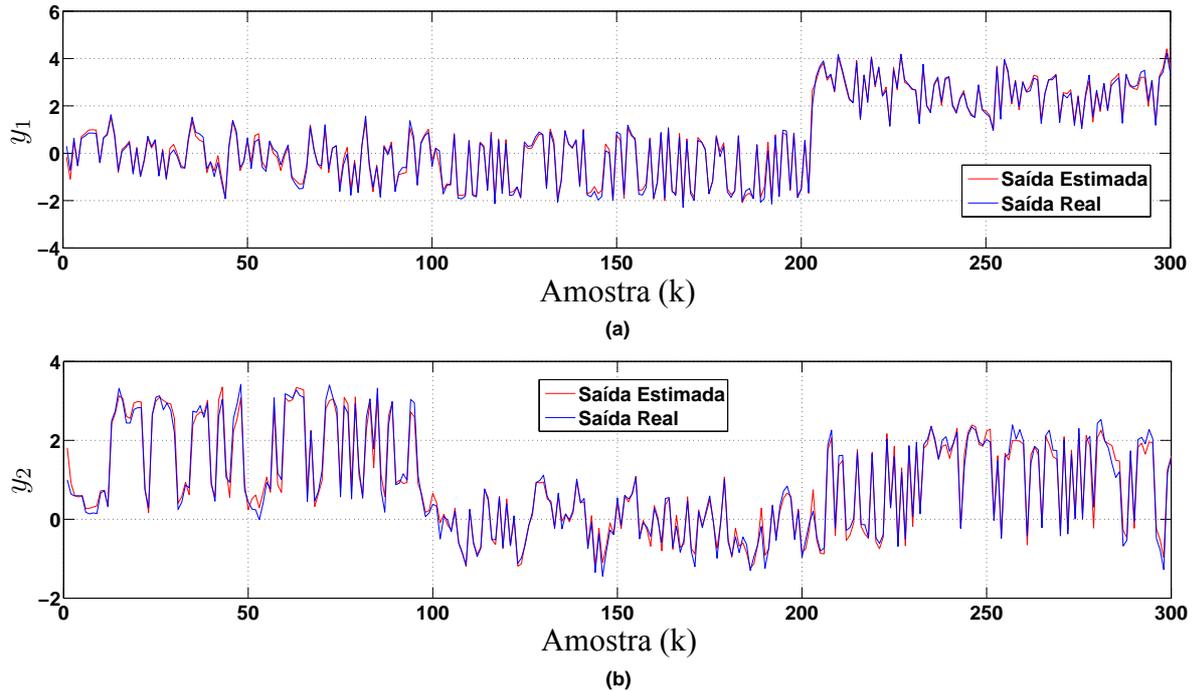


Figura 3.24: Validação dos dados considerando ruído colorido com SNR = 30 dB. (a) Primeira saída do sistema e o valor de VAF obtido foi de 99,09%. (b) Segunda saída do sistema com VAF = 97,02%. Nesta figura, para melhor visualização, apenas um pequeno trecho do sinal utilizado para validação está sendo apresentado.

Tabela 3.11: Resultados do algoritmo com ruído SNR = 10 dB e ordem dos modelos fixada em 5 na Etapa 2 do algoritmo.

SNR (dB)	Ruído Branco	Ruído Colorido
VAF1 (%)	91,09	91,26
VAF2 (%)	82,27	82,29
Erro (%)	0,37	0,37
Sucessos (%)	100	100
Ordem 1 (%)	89	61
Ordem 2 (%)	87	59
Ordem 3 (%)	75	50

3.8 ALGORITMO 6

Nesta seção serão descritas modificações no Algoritmo 4 de forma a reduzir a sua carga computacional. Conforme discutido na seção 3.6.1 a principal desvantagem deste algoritmo com relação aos seus antecessores é o custo computacional mais elevado. Desta forma, as modificações que serão apresentadas tem o objetivo de deixar a abordagem iterativa mais vantajosa, não apenas em termos de desempenho na estimação dos modelos locais mas também em termos de custo computacional.

A estrutura de identificação utilizada é exatamente igual a descrita no Algoritmo 4. São apenas duas modificações propostas. A primeira refere-se à metodologia para a identificação dos modelos locais. Com o desenvolvimento do Algoritmo 5 observou-se que a metodologia utilizada para a identificação dos modelos locais proposta por [75] (e descrita na seção 3.7.1) é mais eficiente computacionalmente e apresenta menores dispersões nas estimativas quando comparada com o algoritmo ACR utilizado anteriormente. Desta forma, em todas as etapas de identificação dos modelos locais, o algoritmo ACR foi substituído pela metodologia apresentada na seção 3.7.1.

Assim como nos algoritmos anteriores, esta metodologia é combinada com a metodologia descrita na seção 2.1.2 para representar todos os modelos locais na mesma base de estados. Adicionalmente, é utilizado o algoritmo de estimação de ordem, descrito na seção 2.1.3, para a estimação da ordem de cada modelo local. No entanto, uma importante observação deve ser feita. No procedimento iterativo proposto, é utilizado o algoritmo IMM para realizar a atualização da segmentação dos dados (Etapa 6). Esse algoritmo requer que as ordens dos diferentes modelos locais sejam iguais. Contudo, não é possível garantir a estimação de valores iguais para todos os sub-modelos.

Portanto, as ordens devem ser estimadas apenas na etapa de identificação final dos modelos locais (Etapa 8), ou seja, após a aplicação do algoritmo IMM. Isso significa que deve-se definir um valor fixo para a ordem dos modelos estimados durante as etapas 2 e 5 da metodologia. Contudo, como as ordens são desconhecidas a priori, recomenda-se definir um valor que seja igual ou maior que a máxima ordem esperada. Justifica-se essa sugestão com base no trabalho [97], onde os autores defendem que as perdas na caracterização do sistema são menores quando a ordem é superestimada em comparação com valores de ordem subestimadas.

A utilização desse procedimento elimina um outro problema potencial. Na etapa 2, a estimação de modelos de diferentes ordens provoca uma maior dispersão nos pontos enviados ao algoritmo de clustering. Com efeito, no cenário onde o nível de ruído é muito elevado ocorre a redução do desempenho do algoritmo. Esse problema foi apresentado e discutido na seção 3.7.2.

A segunda modificação proposta refere-se ao critério de convergência do algoritmo (Etapa

7). O critério utilizado no Algoritmo 4 apresenta como desvantagem a necessidade de simulação do sistema híbrido a cada iteração de modo a obter o valor de VAF. Esse procedimento não é computacionalmente eficiente e pode ser crítico quando a quantidade de dados amostrado for grande.

Assim, será utilizado um processo mais simples que consiste em comparar os centros dos clusters identificados em uma determinada iteração com os centros dos clusters identificados na iteração anterior. Quando esses centros forem próximos será considerado que os parâmetros dos modelos locais convergiram. Essa comparação é feita utilizando a distância euclidiana. Portanto, quando a distância entre os centros dos clusters de duas iterações sucessivas for menor que um determinado valor ξ_{conv} assume-se a convergência do algoritmo.

Note que, embora mais simples, esse novo critério é igualmente eficaz na determinação de mudanças no mapa de segmentação e/ou parâmetros dos modelos locais, uma vez que, qualquer mudança significativa nesses conjuntos significará variações nos centros dos clusters identificados. Adicionalmente, esse novo critério é computacionalmente mais eficiente.

As mudanças descritas e o procedimento completo para o processo de identificação estão resumidas no Algoritmo 6. No entanto, destaca-se que não se trata de uma nova metodologia de identificação, uma vez que utilizou-se a mesma estrutura apresentada no Algoritmo 4. A nova denominação Algoritmo 6 foi utilizada apenas por questões didáticas e facilitar a apresentação dos resultados.

Algoritmo 6:

2 **enquanto** $Dist(\xi_k - \xi_{k-1})^a > \xi_{conv}$ **ou** $k < \xi_{iterações}$ **faça**

3 **Etapa 1 - Segmentação dos Dados:**

a) Crie pequenos subconjuntos de dados $\{LD_j\}_{j=1}^{N-W}$ contendo W elementos, aplicando o particionamento nos seguintes conjuntos:

se $k=1$ **então**

 | No conjunto total de dados amostrados \mathcal{D} ;

senão

 | Nos conjuntos de dados clusterizados $\{\mathcal{C}_q^c\}_{q=1}^M$;

4 **Etapa 2 - Identificação dos Modelos Locais:**

a) Defina um valor de ordem igual para todos os modelos locais.

b) Determine θ_j para cada conjunto de dados em $\{LD_j\}_{j=1}^{N-W}$ usando as metodologias descritas nas seções 3.7.1 e 2.1.2.

5 **Etapa 3 - Clustering:**

a) Agrupe todos os vetores $\{\theta_j\}_{j=1}^{N-W}$ em M clusters $\{\xi_i\}_{i=1}^M$ usando o RCA.

6 **Etapa 4 - Classificação dos Dados:**

a) Forme os subconjuntos dos dados $\{\mathcal{C}_q\}_{q=1}^M$ da seguinte forma:

$$\mathcal{C}_q = \left\{ \cup_{j=1}^{N-W} LD_j : \theta_j \in \xi_q \right\}, q = 1, \dots, M.$$

7 **Etapa 5 - Atualização dos Modelos Locais:**

a) Defina um valor de ordem igual para todos os modelos locais.

b) Para cada conjunto de dados $\{\mathcal{C}_q\}_{q=1}^M$ estime os modelos locais usando as metodologias descritas nas seções 3.7.1 e 2.1.2.

8 **Etapa 6 - Atualização da Segmentação dos Dados:**

a) Com as estimativas dos modelos locais obtidas na Etapa 5, use o algoritmo IMM apresentado na Seção 2.3 para corrigir os conjuntos dos dados clusterizados

$$\{\mathcal{C}_q^c\}_{q=1}^M$$

9 **Etapa 7: Checagem de Convergência:**

a) Calcule a distância entre os clusters identificados nas iterações k e $k - 1$.

10 **Etapa 8: Identificação Final dos Modelos Locais:**

a) Utilize a metodologia descrita na seção 2.1.3 para a estimação da ordem dos modelos locais com base nos dados dos conjuntos $\{\mathcal{C}_q^c\}_{q=1}^M$.

b) Identifique os modelos locais finais com os dados de $\{\mathcal{C}_q^c\}_{q=1}^M$, usando as metodologias descritas nas seções 3.7.1 e 2.1.2.

^a $Dist(\xi_k - \xi_{k-1})$ denota a distância euclidiana entre os centros dos clusters identificados na 1ª iteração k e $k - 1$

3.8.1 Resultados Numéricos

Para comprovar a eficiência das modificações propostas o Algoritmo 6 foi comparado com o Algoritmo 4 em um exemplo simulado. O sistema híbrido considerado e os procedimentos utilizados são exatamente iguais aos descritos na seção 3.6.1. Inicialmente as ordens dos modelos locais são consideradas conhecidas a priori.

Os resultados obtidos estão resumidos na Tabela 3.12. Nesta tabela é possível observar que a carga computacional na execução do Algoritmo 6 é muito menor. De fato, observou-se uma redução média de 80,9% no tempo de processamento quando o Algoritmo 6 é utilizado. Os tempos de processamento obtidos com o Algoritmo 6 são melhores inclusive do que os tempos de processamento obtidos com o Algoritmo 3 que não utiliza a metodologia iterativa.

Adicionalmente, o desempenho do Algoritmo 6 também foi muito superior ao do Algoritmo 4 em relação à classificação dos dados. Para níveis moderado de ruído (20, 30 e 35 dB), obteve-se uma redução média de 60,45% nos erros de classificação. Para um ruído na proporção de 10 dB, os erros de classificação foram semelhantes. A redução nos erros de classificação implica no identificação de modelos locais mais acurados.

Tabela 3.12: Comparação de performance dos Algoritmos 4 e 6 com diferentes níveis de ruído.

SNR (dB)	Algoritmo 4				Algoritmo 6			
	10	20	30	35	10	20	30	35
VAF (%)	74,63	91,33	97,10	98,39	75,46	92,09	97,32	98,49
Erro (%)	12,13	5,92	3,48	3,19	12,02	2,20	1,46	1,27
Successos (%)	96	99	100	100	95	100	100	100
$T_{\text{proc}} \text{ médio (s)}$	58,06	41,46	41,67	39,72	14,85	7,12	6,98	6,71

3.8.1.1 Avaliação da eficiência na estimação da ordem dos modelos locais

Nesta seção, o sistema híbrido utilizado anteriormente foi novamente simulado, mas retirando-se a restrição que a ordem dos modelos locais é conhecida a priori. O objetivo é avaliar a eficiência do algoritmo na estimação das ordens dos modelos locais.

Nestas simulações, a sequencia de chaveamento foi gerada utilizando uma função constante por partes no tempo e que chaveia periodicamente a cada 120 amostras ($\tau_{\text{dwell}} = 120$) do seu valor atual para um valor alvo escolhido aleatoriamente entre $\{1, \dots, 4\}$. Este sinal é mostrado na Figura 3.25. O sistema foi excitado por um sinal PRBS de média zero e variância unitária. Geraram-se 2000 pontos de u_k e y_k para a identificação do sistema. O sinal de saída foi simulado adicionando-se um sinal de ruído branco Gaussiano de média zero.

Considerou-se três diferentes valores de magnitude para esse ruído, de forma que a relação sinal ruído fosse igual a 20, 30 e 40 dB. Para demonstrar a eficiência do algoritmo foi realizado simulações de Monte Carlo com 100 realizações para cada nível de ruído mencionado. Em cada simulação, era considerado uma diferente realização para o sinal de entrada, saída e ruído. Para validação dos modelos estimados, considerou-se um conjunto de dados com 1200 pontos de entrada e saída diferente do utilizado para a identificação.

Foi definido um valor de ordem igual a 4 para a estimação dos parcial dos modelos locais (nas etapas 2 e 5), uma vez que, nestas simulações, a ordem dos modelos locais foi considerada desconhecida a priori. Lembra-se que, a ordem dos modelos é estimada apenas na última etapa do algoritmo (Etapa 8 - Identificação Final dos Modelos Locais).

Os resultados obtidos estão resumidos na Tabela 3.13. Como pode ser observado, o algoritmo apresenta bons resultados para a estimação da ordem dos modelos locais nos cenários de ruído moderado. De fato, a porcentagem de acerto na estimação das ordens é superior a 90%. Conforme esperado, o desempenho é reduzido à medida que o nível de ruído do sistema é aumentado.

No entanto, é importante destacar que, no exemplo considerado, a estimação incorreta da ordem não causou grande influência no desempenho dos modelos estimados, como pode ser comprovado comparando-se as Tabelas 3.12 e 3.13. O aumento na porcentagem de erro de classificação dos dados observado na Tabela 3.13 é atribuído ao aumento da quantidade de chaveamentos do sistema e não a erros na estimação das ordens. A Figura 3.25 ilustra um resultado típico de estimação de modos para dois diferentes níveis de ruído.

Tabela 3.13: Desempenho do Algoritmo 6 na estimação de ordem dos modelos locais com diferentes níveis de ruído. Aqui, o termo Ordem_x representa o número de vezes que o algoritmo estimou corretamente a ordem do modelo local x nas 100 simulações.

SNR	VAF (%)	Erro (%)	Sucessos (%)	Ordem_1	Ordem_2	Ordem_3	Ordem_4
20 dB	90,32	5,52	97	97	81	83	97
30 dB	96,59	2,57	100	98	93	91	98
40 dB	98,94	2,31	100	100	92	95	100

3.8.1.2 Discussão sobre os problemas de convergência

Conforme já mencionado, um dos principais problemas da metodologia proposta é a ausência de prova de convergência do algoritmo. No entanto, esse é um dos muitos problemas desafadores que ainda estão em aberto no campo de identificação de sistemas dinâmicos híbridos [64]. A principal dificuldade em se obter garantias de convergência dos algoritmos

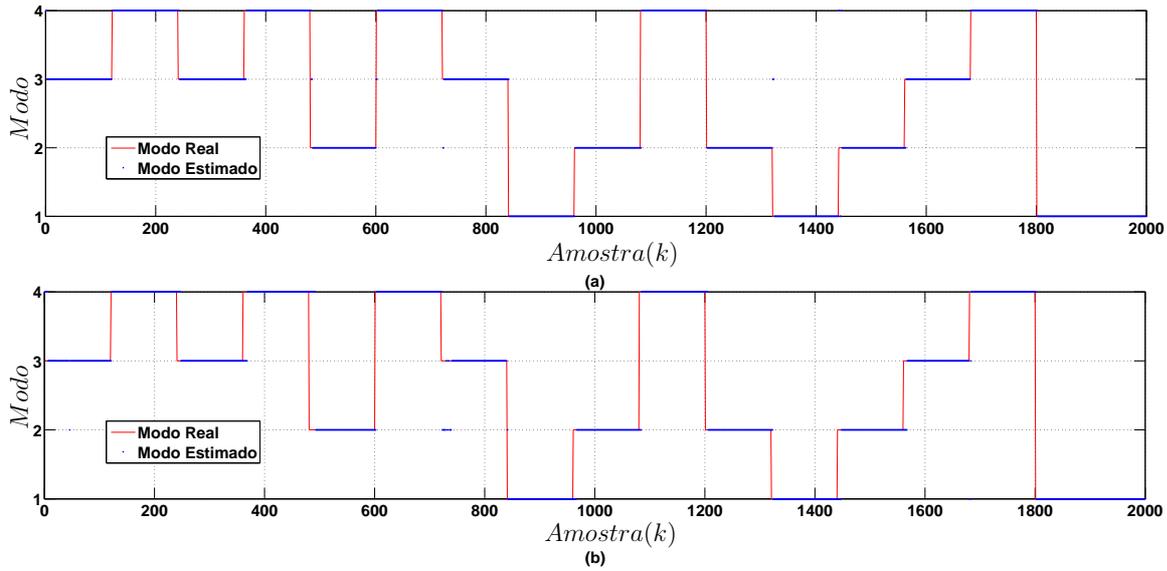


Figura 3.25: Resultado para a estimação dos modos com: (a) SNR of 40 dB (Erro de classificação = 1, 80%); (b) SNR of 20 dB (Erro de classificação = 4, 45%).

mos está relacionada ao forte acoplamento entre a identificação dos estados discretos e a estimação dos parâmetros dos modelos contínuos [130].

Na metodologia apresentada, a convergência do algoritmo, embora não demonstrada teoricamente, parece ser facilmente obtida com a escolha adequada do parâmetro ξ_{conv} . De fato, nos resultados apresentados na Tabela 3.13, o algoritmo convergiu em todas as simulações. No entanto, o parâmetro ξ_{conv} tem um papel fundamental na convergência, velocidade de convergência e desempenho do algoritmo na identificação dos modelos. A Tabela 3.14 apresenta um estudo sobre o desempenho do algoritmo em função do parâmetro ξ_{conv} .

Tabela 3.14: Análise de convergência do algoritmo com diferentes valores do parâmetro ξ_{conv} . As simulações foram realizadas com SNR = 20 dB e $\xi_{\text{iterações}} = 7$.

φ_c	0,01	0,05	0,1	0,2
VAF (%)	90,37	90,35	90,32	90,30
Erro (%)	5,31	5,48	5,52	5,71
Convergência (%)	3	99	100	100
$T_{\text{proc}} \text{ médio(s)}$	40,56	24,72	15,13	14,78

Na Tabela 3.14 nota-se uma tendência de melhora no desempenho do algoritmo à medida que diminui-se o valor do parâmetro ξ_{conv} . Mas, por outro lado, observa-se um aumento significativo no tempo de processamento do algoritmo. Adicionalmente, a redução de ξ_{conv} também causa uma redução no número de vezes que o algoritmo converge.

Desta forma, o ajuste de ξ_{conv} é um compromisso entre o tempo de processamento e desempenho do algoritmo na identificação dos modelos locais. Por exemplo, para os resultados obtidos nas Tabelas 3.12 e 3.13, foi utilizado o valor $\xi_{\text{conv}} = 0,1$, pois é o que apresenta a melhor relação entre o desempenho de estimação e tempo de processamento, como pode ser observado na Tabela 3.14.

É importante ressaltar que, mesmo nos casos em que o algoritmo não convergiu, os modelos estimados foram consistentes. De fato, nas simulações com $\xi_{\text{conv}} = 0,01$, o algoritmo convergiu em apenas 3% das simulações e mesmo assim os modelos identificados apresentaram o melhor desempenho, como mostrado na Tabela 3.14.

Por fim, destaca-se que a escolha adequada do parâmetro ξ_{conv} permite estabelecer empiricamente o valor de $\xi_{\text{iterações}}$. Este valor deve ser definido de forma a permitir a convergência do algoritmo no maior número de simulações possíveis.

4 RESULTADOS COMPARATIVOS

4.1 INTRODUÇÃO

A revisão bibliográfica apresentada na Seção 1.3 revelou uma importante lacuna na literatura, qual seja, a ausência de comparação entre as principais metodologias desenvolvidas para a identificação de sistemas dinâmicos híbridos no espaço de estados. Isso torna difícil a definição da melhor técnica ou da técnica de referência para os demais trabalhos. Desta forma, este capítulo visa dar uma pequena contribuição comparando a abordagem mais completa desenvolvida neste trabalho (Algoritmo 6) com uma abordagem existente na literatura.

A escolha da técnica a ser comparada foi baseada na Tabela 1.2, apresentada na Seção 1.3. Nesta tabela é realizada uma comparação entre as principais considerações realizadas nas diferentes abordagens off-line existentes para a identificação de sistemas dinâmicos chaveados com modelos representados no espaço de estados. Esta tabela deixa claro que os trabalhos [71], [72], [73], [50], [74] e [75] representam as melhores opções para comparação com a abordagem apresentada nesta tese, visto que assumem as mesmas considerações. Destes, os trabalhos [50] e [74] apresentam algoritmos que são versões preliminares da metodologia proposta em [75]. Da mesma forma, o trabalho proposto em [72] é uma versão mais completa da metodologia proposta em [71]. Assim, essencialmente, existem três abordagens, [72], [73] e [75], que são as mais apropriadas para comparação com a metodologia apresentada neste trabalho.

Destas três abordagens, a abordagem proposta em [73] possui a limitação de requerer uma dinâmica de chaveamento lenta no sistema híbrido. A proposta apresentada em [75] tem como principal desvantagem a elevada sensibilidade aos ruídos de processo e medida, uma vez que utiliza uma metodologia de detecção dos instantes de chaveamento baseada nas variações dos subespaços projetados. Como resultado, o algoritmo não apresenta bons resultados para níveis elevado de ruído. Adicionalmente, a metodologia não aborda problemas de atraso na detecção e requer um número suficientemente grande de chaveamentos entre os diferentes modos do sistema para que seja possível representar todos os modelos locais na mesma base de estados.

Desta forma, o trabalho proposto em [72] é o mais completo entre as três metodologias destacadas, no sentido que não apresenta as desvantagens mencionadas acima e aborda os problemas de atraso na detecção. Portanto, foi a abordagem escolhida para comparação com o presente trabalho. O algoritmo apresentado em [72] foi desenvolvido pelos autores K. M Pekpe, G. Mourot, K. Gasso e J. Ragot. Contudo, para facilitar a referência a este algoritmo no restante do texto ele será denominado apenas por Algoritmo do Pekpe. A metodologia utilizada neste algoritmo é descrita a seguir.

4.2 ALGORITMO DO PEKPE

O algoritmo proposto em [72] é baseado em uma técnica de identificação em subespaços. A ideia básica consiste em aplicar um método de subespaços em uma matriz de projeção e utilizar testes estatísticos para estimar os instantes de chaveamento do sistema. O método em subespaço é aplicado a pequenos conjuntos de dados. Para formação destes conjuntos é utilizada uma janela móvel no tempo. Contudo, a utilização de testes estatísticos apresenta como desvantagem o atraso na detecção dos instantes de chaveamento.

Assim, os dados imediatamente antes dos instantes de chaveamento são considerados como pontos duvidosos e são classificados somente após a estimação dos parâmetros de Markov. Um algoritmo de fusão, com base na comparação dos parâmetros de Markov, é aplicado para combinar os modelos locais semelhantes. Em seguida, os parâmetros dos modelos que foram combinados são estimados novamente. Este procedimento é repetido até se observar que o número de modelos do sistema não sofre mais alterações. Neste momento, é considerado que o algoritmo convergiu. Esta metodologia está resumida no diagrama de blocos mostrado na Figura 4.1 e será brevemente descrita nesta seção. Para maiores detalhes e demonstrações matemáticas deve-se consultar o trabalho original.

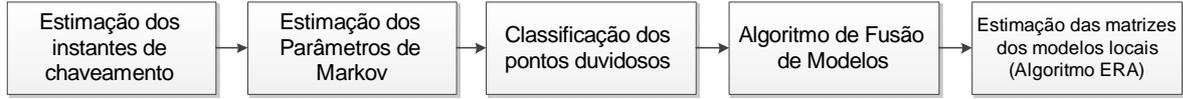


Figura 4.1: Diagrama de blocos ilustrando a metodologia utilizada no Algoritmo do Pekpe.

Os autores utilizam o sistema híbrido chaveado como descrito na equação (3.1). Contudo, a saída do sistema é representada como uma soma ponderada das saídas dos modelos locais:

$$y_k = \sum_{j=1}^M p_{j,k} y_{j,k}, \quad (4.1)$$

em que k representa o índice de tempo, $y_k \in \mathbb{R}^{n_y}$ é a saída do sistema, $y_{j,k} \in \mathbb{R}^{n_y}$ é a saída do modelo local j e $p_{j,k}$ é o peso associado a $y_{j,k}$. Os pesos dos modelos locais obedecem a seguinte restrição:

$$p_{j,k} \in \{0, 1\} \quad \text{e} \quad \sum_{j=1}^M p_{j,k} = 1, \quad \forall k \in [1, N], \quad (4.2)$$

em que N denota o número total de dados amostrados.

Assim, o primeiro passo do procedimento é re-escrever a saída do sistema (3.1) como uma função dos parâmetros de Markov e das matrizes de *block-Hankel* do sistema:

$$\bar{y}_{j,k,L} \simeq M_{j,s}^d U_L + M_{j,s}^v V_{j,L} + \bar{w}_{j,k,L}, \quad (4.3)$$

em que o sub-escrito j é utilizado para indicar que o j -ésimo modelo local está ativo no instante de tempo k . O parâmetro L representa o tamanho da janela móvel utilizada. As

matrizes U_L e $V_{j,L}$ são as matrizes de *block-Hankel* definidas de maneira similar a (2.9), mas considerando que o número de colunas é igual a L . A matriz $\bar{y}_{j,k,L}$ é uma matriz com as saídas do sistema definida como:

$$\bar{y}_{j,k,L} = \begin{bmatrix} y_k & y_{k+1} & \cdots & y_{k+L} \end{bmatrix} \in \mathbb{R}^{n_y \times L}, \quad (4.4)$$

$\bar{w}_{j,k,L}$ é a matriz de ruído de medidas e é definida de maneira similar.

As matrizes $M_{j,s}^d \in \mathbb{R}^{n_y \times n_{us}}$ e $M_{j,s}^v \in \mathbb{R}^{n \times n_{us}}$ são os parâmetros de Markov da parte determinística e estocástica, respectivamente, do j -ésimo modelo local. Essas matrizes são definidas como:

$$M_{j,s}^d = \begin{bmatrix} C_j A_j^{s-2} B_j & C_j A_j^{s-3} B_j & \cdots & C_j B_j & D_j \end{bmatrix}. \quad (4.5)$$

$$M_{j,s}^v = \begin{bmatrix} C_j A_j^{s-2} & C_j A_j^{s-3} & \cdots & C_j & 0 \end{bmatrix}. \quad (4.6)$$

A influência dos termos de entrada é removida da equação (4.3) através da projeção do espaço linha da matriz $\bar{y}_{j,k,L}$ sobre o espaço ortogonal ao espaço linha da matriz de *block-Hankel* de entrada:

$$\bar{y}_{j,k,L} \Pi_{(U_L)^\perp} \simeq M_{j,s}^v V_{j,L} \Pi_{(U_L)^\perp} + \bar{w}_{j,k,L} \Pi_{(U_L)^\perp}, \quad (4.7)$$

em que a matriz $\Pi_{(U_L)^\perp}$ é um operador que realiza a projeção ortogonal e é definido como $\Pi_{(U_L)^\perp} = I - U_L^\dagger U_L$.

Os autores em [72] mostram que a relação em (4.7) fornece um vetor residual contendo apenas os termos de ruído. Assim, sob a consideração que os ruídos que afetam o sistema são sequências de ruído branco Gaussiano, mostra-se que o vetor residual obtido é um vetor Gaussiano de média zero na ausência de chaveamentos no sistema.

Se um chaveamento ocorre, o valor médio deste vetor deixa de ser zero. Portanto, um teste χ^2 pode ser aplicado para detectar as variações no valor médio deste vetor. O vetor de resíduos é dado pela seguinte relação:

$$\epsilon_{j,k} = \bar{y}_{j,k,L} \Pi_{(U_L)^\perp} Z \quad (4.8)$$

em que $Z = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T \in \mathbb{R}^{L \times 1}$ é apenas um vetor seletor.

Considere que μ represente a média e R_ϵ a matriz de variância do vetor residual. Assim, o termo $\varpi_{j,k} = \epsilon_{j,k}^T R_\epsilon^{-1} \epsilon_{j,k}$ segue uma distribuição χ^2 com n_y graus de liberdade. Portanto, enquanto $\varpi_{j,k}$ for menor que um determinado limiar β é considerado que nenhum chaveamento ocorreu no sistema.

Após a obtenção dos instantes de chaveamento, são estimados os pesos $p_{j,k}$ dos modelos locais. Somente um modelo local (modelo j) está ativo entre dois instantes de chaveamentos consecutivos k_1 e k_2 . A classificação dos dados amostrados nos diferentes modos de operação do sistema durante o intervalo $[k_1, k_2]$ é delicada por causa do atraso na detecção. O

atraso na detecção é o tempo que os testes estatísticos levam para reconhecer as variações no valor médio do vetor de resíduos. Se t_{\max} representa o máximo tempo de atraso na detecção, então os dados pertencentes ao intervalo $]k_2 - t_{\max}, k_2]$ não podem ser atribuídos a nenhum modo de operação. Assim, a classificação destes pontos duvidosos é realizada somente após a primeira estimação dos parâmetros de Markov dos modelos locais.

O próximo passo do procedimento é a identificação dos parâmetros de Markov dos modelos locais. Esses parâmetros são obtidos por mínimos quadrados como segue:

$$\bar{y}_{j,k,z} P_j U_j^T (U_z P_j U_z^T)^{-1} \rightarrow M_{j,s}^d \quad \text{para } z \rightarrow \infty, \quad (4.9)$$

em que a matriz de pesos $P_j \in \mathbb{R}^{L \times L}$ do modelo local j é definida como:

$$P_j = \begin{bmatrix} p_{j,k} & 0 & \cdots & 0 \\ 0 & p_{j,k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{j,k+L-1} \end{bmatrix}. \quad (4.10)$$

Após a estimação dos parâmetros de Markov, os dados duvidosos são classificados. Isso é realizado utilizando a seguinte equação de erro:

$$e_{j,k} = |y_k - \hat{M}_{j,s}^d u_k|. \quad (4.11)$$

Os pontos duvidosos são associados ao modelo local que apresenta o menor erro. Em seguida, um algoritmo de fusão de modelos é utilizado para combinar os modelos locais semelhantes. Esse procedimento é baseado na comparação dos parâmetros de Markov. A similaridade entre dois modelos locais é medida pela distância entre seus parâmetros de Markov. Um teste χ^2 com n_y graus de liberdade pode ser utilizado para realizar o teste de similaridade. Considerando β com um limiar do teste, se a distância entre dois parâmetros de Markov é menor do que β então os modelos locais são semelhantes.

Após os teste de similaridade, os modelos locais semelhantes são combinados. Os parâmetros de Markov dos modelos locais combinados são re-estimados e uma nova busca por modelos semelhantes é realizada. Os autores destacam que essa nova estimação proporciona a estimação de modelos locais mais acurados porque o número de dados usado para estimar os parâmetros é aumentado após a combinação dos modelos. Adicionalmente, a re-aplicação do teste de similaridade após a combinação de modelos semelhantes é importante. Esse procedimento torna possível reconhecer modelos semelhantes que não tinham sido identificados na etapa anterior devido a falta de exatidão dos seus parâmetros causada pela identificação com uma quantidade insuficientes de dados.

O último passo do algoritmo é a estimação de uma realização (A_j, B_j, C_j, D_j) de cada modelo local. Isso é realizado a partir do conhecimento dos parâmetros de Markov de cada modelo local. De forma a obter uma realização mínima e balanceada os autores em [72] utilizam o Algoritmo ERA (do inglês, *Eigenvalue Realization Algorithm*) proposto em [131].

4.3 RESULTADOS NUMÉRICOS

Nesta seção o Algoritmo 6 proposto na seção 3.8 será comparado com o Algoritmo do Pekpe através de um exemplo simulado. Para isso, será considerado um sistema híbrido chaveado composto de três modelos locais MIMO de terceira ordem. Esses modelos estão representados abaixo:

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 0,32 & 0,31 & 0 \\ -0,32 & 0,31 & 0 \\ 0 & 0 & -0,18 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0,9 & -0,7 \\ 0,71 & -0,5 \\ 0,8 & 0,47 \end{bmatrix}, \\
 C_1 &= \begin{bmatrix} -0,55 & 0,2 & 0,8 \\ 0,45 & 0,3 & 0,58 \end{bmatrix}, & D_1 &= \begin{bmatrix} 0,97 & 0,63 \\ -0,32 & 0,95 \end{bmatrix}; \\
 A_2 &= \begin{bmatrix} 0,2397 & -0,522 & -0,4535 \\ -0,522 & -0,0653 & -0,4436 \\ -0,4535 & -0,4436 & 0,3772 \end{bmatrix}, & B_2 &= \begin{bmatrix} 0 & 0,5096 \\ 2,143 & 0 \\ -0,8066 & -0,6178 \end{bmatrix}, \\
 C_2 &= \begin{bmatrix} -0,3859 & 1,495 & 0 \\ 1,202 & 0,4929 & 0,2108 \end{bmatrix}, & D_2 &= \begin{bmatrix} 0,5 & 0,3 \\ -0,2 & 0,5 \end{bmatrix}; \\
 A_3 &= \begin{bmatrix} 0 & 0,8 & 0 \\ -0,8 & 0,5 & 0,3 \\ 0 & 0 & 0,1 \end{bmatrix}, & B_3 &= \begin{bmatrix} 0,4 & 1 \\ 0 & 1 \\ 1 & -0,2 \end{bmatrix}, \\
 C_3 &= \begin{bmatrix} 1 & 0 & 1 \\ 0,1 & -0,5 & 1 \end{bmatrix}, & D_3 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.
 \end{aligned}$$

Nas simulações geraram-se 1500 pontos de dados de u_k e y_k que são utilizados no processo de identificação do sistema. A sequência de chaveamento utilizada está mostrada na Figura 4.2. O sistema foi excitado por um sinal PRBS de média zero e variância unitária. O sistema foi simulado e adicionado ao sinal de saída um sinal de ruído branco de média zero. Foram consideradas três diferentes amplitudes para esse ruído 10, 20 e 30 dB. De maneira similar aos exemplos anteriores, para cada nível de ruído mencionado foram realizadas simulações de Monte-Carlo de tamanho 100 com diferentes realizações de entrada, saída e ruído. A validação dos modelos estimados foi realizada utilizando-se um conjunto de dados diferente do utilizado para a identificação dos modelos. Este conjunto de dados de validação continha 900 amostras.

Uma vez que o principal objetivo é comparar os dois procedimentos, a ordem dos modelos locais foi considerada conhecida a priori. A motivação para isso foi evitar variações provenientes do algoritmo de estimação de ordem. Exaustivas simulações foram realizadas para determinar o melhor ajuste inicial dos parâmetros dos dois algoritmos. Para o Algoritmo do Pekpe os parâmetros selecionados foram $\beta = 18,467$, $t_{max} = 10$. Os demais parâmetros foram $(i = 18, L_i = 100)$, $(i = 9, L_i = 86)$ e $(i = 8, L_i = 40)$ para os níveis de ruído de 10, 20 e 30 dB, respectivamente. Para o Algoritmo 6 os parâmetros foram ajustados como

segue: $W = 60$, o número de blocos linhas foi igual a 4, $\xi_{\text{conv}} = 0,1$ para os níveis de ruído 20 e 30 dB, e $\xi_{\text{conv}} = 0,3$ para 10 dB. Os demais parâmetros foram ajustados como descrito nos exemplos anteriores.

Os algoritmos foram comparados em termos da acurácia dos modelos identificados, tolerância ao ruído, capacidade em estimar corretamente o número de modos do sistema, capacidade em classificar corretamente os dados nos diferentes modos e tempo de computação. Para quantificar a acurácia dos modelos estimados, além do valor de VAF, foi utilizado como medida de performance adicional o valor de Erro Médio Quadrático (MSE), definido na equação (2.70).

Os resultados obtidos estão resumidos na Tabela 4.1. É importante destacar que os valores relativamente elevados para o índice MSE observado em ambos algoritmos resultam da elevada amplitude do ruído adicionado ao sinal de saída. Como pode ser visto nesta tabela, os dois algoritmos obtiveram resultados semelhantes para o nível de ruído de SNR = 30 dB. No entanto, o Algoritmo do Pekpe tem um custo computacional muito menor. De fato, o Algoritmo 6 necessita, em média, de 4,44s a mais de tempo de processamento para fornecer uma estimativa do sistema.

No entanto, o Algoritmo 6 apresenta resultados muito melhores a medida que o nível de ruído é aumentado. Por exemplo, para o ruído de 20 dB, o Algoritmo 6 reduziu em 91% os erros de classificação. Como consequência, os modelos estimados conseguem reconstruir a saída real do sistema com muito mais precisão. Isso é comprovado analisando o valor de MSE onde o Algoritmo 6 apresentou uma redução de 26% em relação ao Algoritmo do Pekpe. Já para o nível de ruído de 10 dB, o Algoritmo 6 apresentou uma redução de 84,7% nos erros de classificação e 30,1% no valor de MSE.

Uma vantagem adicional do Algoritmo 6 em relação ao Algoritmo do Pekpe está relacionada com a dispersão das estimativas obtidas. Observou-se que o Algoritmo 6 produz menor variação nas estimativas quando comparado com o Algoritmo do Pekpe, como pode ser observado nos valores de desvio padrão apresentados na Tabela 4.1 e na distribuição dos valores de VAF apresentada Figura 4.3. Isso implica em uma maior robustez do Algoritmo 6 com relação ao ruído que afeta o sistema. De fato, o Algoritmo 6 consegue fornecer estimativas mais consistentes e com pouca variação mesmo quando o sistema está sujeito a elevados níveis de ruído. Por outro lado, o Algoritmo do Pekpe é mais afetado pelo ruído do sistema, resultando em estimativas com maior desvio padrão e, em alguns casos, na identificação de modelos locais pouco acurados (valores de VAF menores que 60%).

Tabela 4.1: Comparação de desempenho dos algoritmos com diferentes níveis de ruído. Aqui, os termos VAF_x e MSE_x indicam o cálculo do valor de VAF e MSE, respectivamente, para a saída x do sistema e o termo σ_Z indica o desvio padrão da medida Z .

	Algoritmo do Pekpe			Algoritmo 6		
	30	20	10	30	20	10
SNR (dB)						
VAF_1 (%)	96,57	84,72	63,80	96,63	90,36	73,22
σ_{VAF_1}	1,14	2,50	6,50	0,21	0,54	3,39
VAF_2 (%)	96,43	84,06	59,70	96,50	90,15	72,61
σ_{VAF_2}	1,38	3,46	7,22	0,22	0,52	4,63
MSE_1	0,29	0,97	3,33	0,24	0,72	2,38
σ_{MSE_1}	0,06	0,15	0,38	0,02	0,04	0,30
MSE_2	0,10	0,42	1,46	0,10	0,31	1,02
σ_{MSE_2}	0,03	0,08	0,21	0,01	0,02	0,18
Erro (%)	1,52	21,23	47,21	1,49	1,89	7,22
Sucessos (%)	95	8	11	100	100	99
T_{proc} médio(s)	0,77	0,98	1,15	5,21	7,41	7,82

Os resultados apresentados na Tabela 4.1 também indicam que o Algoritmo 6 consegue estimar corretamente o número de modos do sistema para todos os níveis de ruídos considerados, enquanto que o Algoritmo do Pekpe apresentou bons resultados apenas para SNR = 30 dB. Na Figura 4.2 é apresentado um resultado típico para estimação dos modos nos dois algoritmos.

O Algoritmo 6 apresenta duas principais desvantagens com relação ao Algoritmo do Pekpe. A primeira é relacionada ao número mais elevado de parâmetros para ajuste no início do processo de identificação. Contudo, conforme já discutido, apesar da quantidade de parâmetros o problema de ajuste é relativamente simples. A segunda desvantagem está relacionada ao custo computacional. Nas simulações realizadas obteve-se, no pior caso, um aumento de 656% no tempo de processamento. No entanto, uma vez que os métodos apresentados abordam o problema de identificação de modo off-line, esse aumento no tempo de processamento não é considerado um problema. Portanto, apesar destas duas desvantagens, acredita-se que o ganho de desempenho obtido com o Algoritmo 6 justifica a sua utilização.

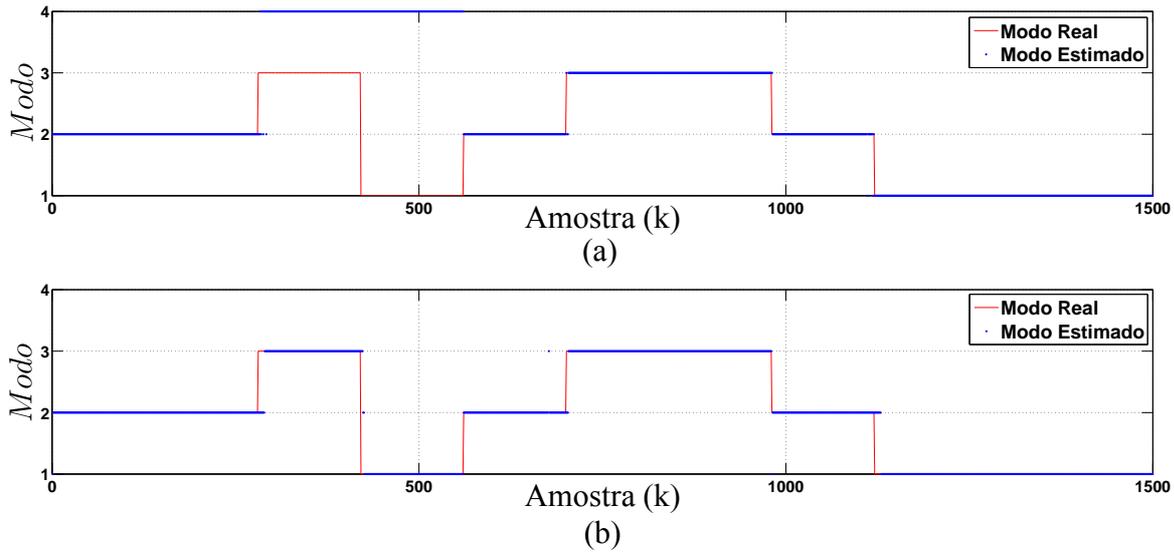


Figura 4.2: Resultado típico para a estimação de modos com SNR = 20 dB: (a) Algoritmo do Pekpe. (b) Algoritmo 6.

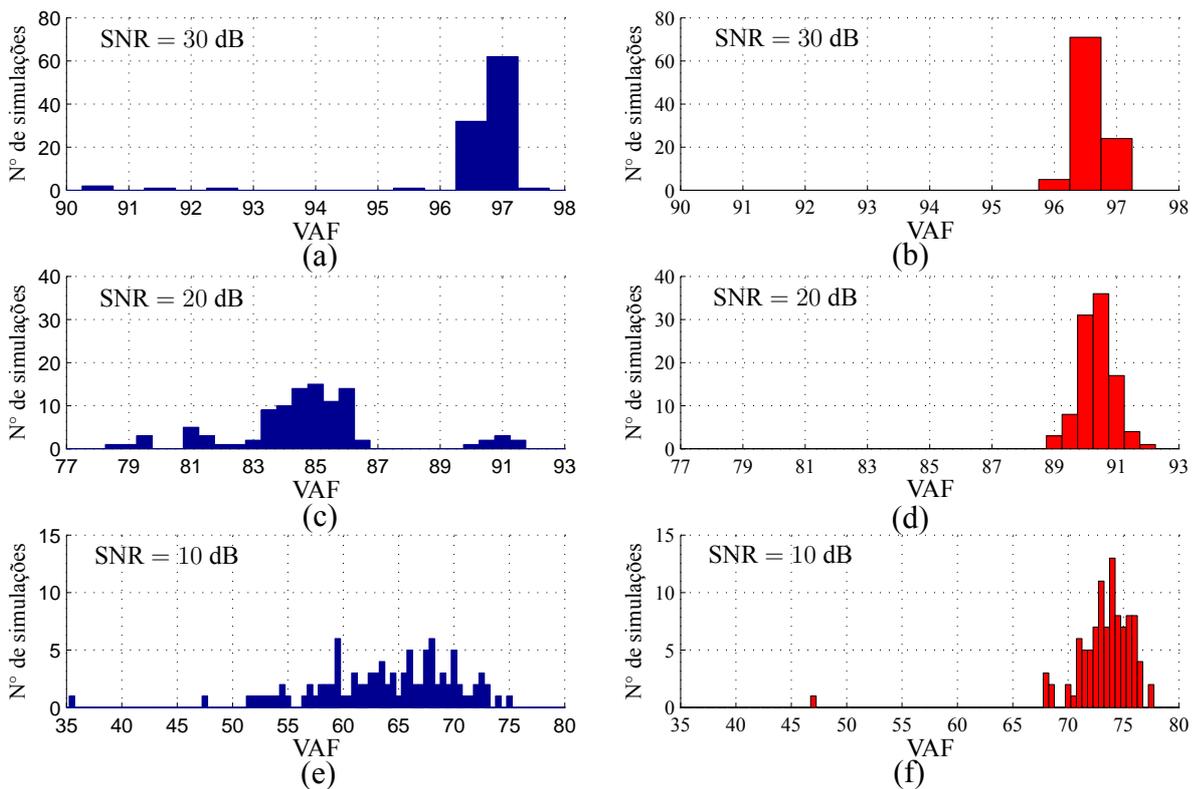


Figura 4.3: Distribuições dos valores de VAF nas 100 simulações independentes realizadas com os dois algoritmos. (a) Algoritmo do Pekpe com SNR = 30 dB. (b) Algoritmo 6 com SNR = 30 dB. (c) Algoritmo do Pekpe com SNR = 20 dB. (d) Algoritmo 6 com SNR = 20 dB. (e) Algoritmo do Pekpe com SNR = 10 dB. (f) Algoritmo 6 com SNR = 10 dB.

5 CONCLUSÕES

Em anos recentes, o problema de identificação de sistemas dinâmicos híbridos tem atraído crescente interesse dos pesquisadores, motivados por suas potenciais aplicações práticas e pelo fato do paradigma híbrido proporcionar a obtenção de modelos matemáticos mais flexíveis e representativos do sistema.

Neste trabalho, foi proposta e discutida uma nova estrutura para a identificação de sistemas híbridos no espaço de estados, mais especificamente o caso particular de sistemas chaveados lineares de tempo discreto, em que a variável discreta denota o modelo matemático que rege a evolução dos estados contínuos. A escolha pela identificação de modelos representados no espaço de estados foi motivada por dois fatores. Primeiro, estes modelos são mais convenientes e permitem uma representação mais elegante e compacta para sistemas MIMO. Segundo, a maioria dos métodos de identificação existentes na literatura abordam o problema de identificação de modelos na forma de entrada e saída, geralmente do tipo PWA e se limitam a sistemas SISO ou, em poucos casos, MISO. De fato, ainda existem poucos métodos desenvolvidos para identificação dos sistemas híbridos na forma de espaço de estados.

A abordagem desenvolvida considerou pressupostos muito gerais de que os instantes de chaveamento, o número de modelos locais, a ordem e os parâmetros destes modelos podem ser simultaneamente desconhecidos. A principal contribuição deste trabalho foi a proposição de uma nova estrutura (*framework*) de identificação onde os modelos locais do sistema híbrido são identificados antes da estimação dos instantes de chaveamento. Desta forma, o conhecimento dos parâmetros dos modelos locais pode ser explorado para classificar corretamente os dados amostrados nos diferentes modos do sistema. Para realizar estas tarefas foi explorado o uso de técnicas de identificação em subespaços, clustering, filtragem híbrida e classificação de dados.

Esta nova estrutura de identificação resultou no desenvolvimento de seis novos algoritmos. Os dois primeiros algoritmos desenvolvidos introduziram as ideias mencionadas. Nestes, a identificação final dos modelos locais foi baseada nas informações obtidas pelo algoritmo de clustering. No Algoritmo 1, são utilizadas informações fornecidas pelo algoritmo de clustering para classificar os dados nos diferentes modos de operação do sistema. Após essa classificação é aplicada uma técnica de identificação em subespaços para identificar os modelos locais. Já no Algoritmo 2, os modelos locais são extraídos a partir dos centros dos clusters identificados pelo algoritmo de clustering. Os resultados obtidos demonstraram a viabilidade e a eficiência das propostas, uma vez que os algoritmos apresentaram bom desempenho na detecção do número de modos do sistema, na estimação dos instantes de chaveamento e na reconstrução dos modelos locais. Contudo, os resultados obtidos com o Algoritmo 1 foram melhores quando comparados aos do Algoritmo 2, especialmente, no

cenário onde a dinâmica de chaveamento do sistema era rápida. A principal vantagem do Algoritmo 2 está relacionada ao custo computacional. Observou-se uma redução média de aproximadamente 51% no tempo de processamento.

No entanto, verificou-se que o desempenho de ambos algoritmos era reduzido quando a dinâmica de chaveamento do sistema era rápida e os instantes de chaveamento não múltiplos do tamanho da janela utilizada na segmentação dos dados. Este problema foi contornado com o desenvolvimento do terceiro algoritmo. Este algoritmo foi uma evolução dos dois primeiros no sentido que incorporou uma etapa de atualização da segmentação dos dados. Esta etapa é realizada pelo algoritmo IMM utilizando-se as estimativas parciais dos modelos locais. Esta abordagem apresentou também a ideia de descartar os pontos duvidosos do processo de identificação utilizando, para isso, as informações fornecidas pelo algoritmo de clustering. Essas modificações representaram ganhos expressivos de desempenho do algoritmo. Contudo, tornou o processo de inicialização dos parâmetros do algoritmo uma tarefa árdua e tediosa.

O quarto algoritmo desenvolvido introduziu uma abordagem iterativa no processo de identificação. Neste algoritmo o problema de identificação foi interpretado como um problema de maximização da função de máxima probabilidade a posteriori do sistema. A proposta então foi realizar esta maximização através de um processo iterativo, onde a cada iteração utiliza-se a estimativa corrente dos modelos locais do sistema para obter um novo mapa de segmentação e, este novo mapa de segmentação é utilizado para encontrar um novo máximo para os parâmetros do modelo. Essa abordagem se mostrou muito eficiente, uma vez que foram obtidos resultados semelhantes aos do Algoritmo 3 sem a necessidade de descartar pontos no processo de identificação. De fato, observou-se que o número de outliers era naturalmente reduzido ao longo das iterações. A principal desvantagem deste algoritmo para os seus antecessores é o aumento da carga computacional.

O Algoritmo 6 foi desenvolvido para tornar a metodologia mais eficiente computacionalmente. Para isso, este algoritmo herdou a mesma estrutura de identificação proposta no Algoritmo 4, mas utilizou um método de identificação em subespaços mais eficiente do ponto de vista computacional e um novo critério de convergência. O resultado foi uma redução média de 81% no tempo de processamento. As modificações propostas também proporcionaram uma melhora significativa na qualidade dos modelos estimados e na classificação dos dados nos diferentes modos.

A metodologia de identificação proposta apresenta dois principais problemas. O primeiro é o número elevado de parâmetros para ajuste no início do processo de identificação. No entanto, apesar da quantidade de parâmetros, o problema de ajuste é relativamente simples. O segundo problema está relacionado à ausência de provas de convergência dos algoritmos. Contudo, este é um problema comum às principais abordagens de identificação existentes na literatura. A principal dificuldade em se obter garantias de convergência dos algoritmos está relacionada ao forte acoplamento entre a identificação dos estados discretos e a estimação

dos parâmetros dos modelos contínuos. Neste trabalho, verificou-se, empiricamente, que a escolha adequada do parâmetro ξ_{conv} permite que a convergência seja obtida na maioria dos casos.

Uma outra importante contribuição deste trabalho foi a comparação da metodologia proposta com uma abordagem já existente na literatura. A revisão bibliográfica realizada para o desenvolvimento deste trabalho revelou que a ausência de comparação entre as principais metodologias desenvolvidas é uma importante lacuna existente na literatura, uma vez que esta ausência torna difícil a definição da melhor técnica ou da técnica referência para os demais trabalhos. Desta forma, o Algoritmo 6 desenvolvido foi comparado com o trabalho proposto em [72].

Os resultados obtidos mostraram que, para níveis moderados de ruído, as duas abordagens apresentam desempenho semelhante. Contudo, para níveis elevados de ruído, o Algoritmo 6 estima modelos locais muito mais precisos. De fato, no melhor caso, obteve-se uma redução média de 30% no erro médio quadrático e 91% de redução nos erros de classificação dos dados.

A principal desvantagem do Algoritmo 6 em relação ao trabalho proposto em [72] é o custo computacional. No pior caso, o Algoritmo 6 precisou de 6,67s a mais de tempo de processamento para fornecer uma estimativa do sistema híbrido. No entanto, uma vez que os métodos apresentados abordam o problema de identificação de modo off-line, esse aumento no tempo de processamento não é considerado um problema.

Por fim, a metodologia inicialmente desenvolvida para identificação de sistemas dinâmicos chaveados foi estendida para a realizar a identificação de sistemas chaveados afm por partes. Esta extensão permite ainda a identificação de sistemas híbridos do tipo PWA no espaço de estados. Esta é uma importante característica da metodologia, uma vez que, ao melhor conhecimento do autor, apenas um trabalho existente na literatura abordou este tipo de problema. As principais abordagens de identificação de sistemas PWA focam na identificação de modelos descritos na forma de entrada-saída.

5.1 TRABALHOS FUTUROS

Ainda nesta linha de pesquisa sugere-se os seguintes trabalhos:

- Nesta tese foi desenvolvido uma nova estrutura de identificação de sistemas dinâmicos híbridos que permite a identificação das matrizes dos modelos locais e a estimação dos instantes de chaveamento. No entanto, a estrutura proposta não focou na identificação da dinâmica discreta, ou seja, na identificação de modelos matemáticos que descrevem a evolução do estado discreto. A extensão do trabalho para identificação da dinâmica discreta tornaria a estrutura de identificação proposta mais completa e

o modelo híbrido final mais adequado para diversas aplicações como, por exemplo, desenvolvimento de estratégias de controle.

- Desenvolvimento de bases teóricas formais para determinação do tamanho da janela móvel utilizada no processo de identificação.
- Análise matemática do algoritmo para determinação das condições necessárias e suficientes para se obter a convergência do algoritmo.
- Na etapa de identificação final dos modelos locais, desenvolver um método de identificação em subespaços ponderado. As ponderações utilizadas seriam as probabilidades dos modos fornecidas pelo algoritmo IMM. Esse procedimento poderia proporcionar uma maior robustez ao algoritmo, uma vez que os pontos que são classificados com baixo grau de confiança receberiam peso menor no processo de identificação.
- Aplicação da metodologia desenvolvida para a identificação de um sistema híbrido real. Todos os algoritmos desenvolvidos foram validados considerando exemplos simulados. Assim, é importante verificar o desempenho da metodologia proposta quando utilizada em dados reais.
- Investigar a possibilidade de se utilizar um indicador de dispersão dos vetores dos parâmetros de Markov obtidos na fase de clustering para criar um procedimento iterativo para auxiliar no ajuste dos parâmetros iniciais do algoritmo. Um possível indicador de dispersão seria a média das distâncias de cada ponto ao centro de cada cluster ($d_{q,k}$) ponderadas pelos pesos $u_{q,k}$. Assim, a ideia seria usar este indicador para redefinir os parâmetros iniciais do algoritmo e reiniciar o algoritmo. O principal objetivo seria reduzir a influência do usuário no ajuste dos parâmetros.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] P. H. R. Q. A. Santana, “Filtragem estocástica para sistemas híbridos e suas aplicações em robótica aérea,” Dissertação de Mestrado em Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 152p., Fevereiro 2011.
- [2] L. A. Aguirre, *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 2000.
- [3] S. E. P. C. Bean, “Abordagens LMI para análise de uma classe de sistemas híbridos,” Tese de doutorado, Universidade Federal de Santa Catarina, Setembro 2002.
- [4] S. Gil, “Robust learning of probabilistic hybrid models,” PhD Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA, USA, September 2008.
- [5] L. Blackmore, S. Gil, S. Chung, and B. Williams, “Model learning for switching linear systems with autonomous mode transitions,” in *In Proc. of 46th IEEE Conference on Decision and Control*, December 2007, pp. 4648 – 4655.
- [6] R. Goebel, R. G. Sanfelice, and A. R. Teel, “Hybrid dynamical systems - robust stability and control for systems that combine continuous-time and discrete-time dynamics,” *IEEE Control Systems*, vol. 29, no. 2, pp. 28 – 93, April 2009.
- [7] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal, “Identification of hybrid systems: a tutorial,” *European Journal of Control*, vol. 13, no. 2-3, pp. 242 – 260, May 2007.
- [8] A. B. Leal, “Controle supervisorio modular de sistemas híbridos,” Tese de doutorado, Universidade Federal de Santa Catarina, Setembro 2005.
- [9] R. de Sousa Baptista, “Identificação de sistemas híbridos e controle H_∞ para o desenvolvimento de VANT autônomo,” Dissertação de Mestrado, 106p, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 2011.
- [10] R. Vidal, A. Chiuso, and S. Soatto, “Applications of hybrid system identification in computer vision,” in *Proceedings of the European Control Conference*, 2007.
- [11] R. Murray-Smith, “Modeling human control behavior with context-dependent Markov switching multiple-models,” in *IFAC Man-Machine Systems Conference*, Kyoto, Japan, 1998.

- [12] R. Möbus, M. Baotic, and M. Morari, “Multi-object adaptive cruise control,” *In Hybrid systems: Computation and Control, volume 2623 of Lecture Notes in Computer Science*, pp. 359 – 374. Springer Verlag., 2003.
- [13] I. A. Hiskens and M. Pai, “Hybrid systems view of power system modeling,” in *Proceedings of the IEEE International Symposium on circuits and Systems*, Geneva, Switzerland, May 2000.
- [14] S. Pettersson, “Analysis and design of hybrid systems,” PhD Thesis, Department of signals and systems, Chalmers university of technology, Goteborg, Sweden, 1999.
- [15] C. Tomlin, G. Pappas, and S. Sastry, “Conflict resolution for air traffic management: a study in multiagent hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 509 – 521, 1998.
- [16] A. S. D. Mestre, “Hybrid subspace identification: An application to HIV infection,” Phd Thesis, Universidade Técnica de Lisboa - Instituto Superior Técnico, June 2010.
- [17] E. K. Boukas, P. Shi, and K. Benjelloun, “Robust stochastic stabilization of discrete-time linear systems with markovian jumping parameters,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 121, pp. 331 – 334, 1999.
- [18] A. N. Vargas, W. Furloni, and J. B. R. do Val, “Constrained MPC of jump linear systems with noise and non-observed Markov state,” in *American Control Conference*, Minneapolis, Minnesota USA, 2006.
- [19] P. H. R. Q. A. Santana, G. A. Borges, and J. Y. Ishihara, “Hybrid data fusion for 3D localization under heavy disturbances,” in *In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Taipei, Taiwan, Oct 2010.
- [20] E. F. Camacho, D. R. Ramirez, D. Limon, D. M. de la Peña, and T. Alamo, “Model predictive control techniques for hybrid systems,” *Annual Reviews in Control*, vol. 34, pp. 21 – 31, 2010.
- [21] W. P. M. H. Heemels, B. D. Schutter, J. Lunze, and M. Lazar, “Stability analysis and controller synthesis for hybrid dynamical systems.” *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, vol. 368, pp. 4937 – 4960, 2010.
- [22] J. Thomas, “Analytical non-linear model predictive control for hybrid systems with discrete inputs only,” *Control Theory & Applications, IET*, vol. 6, no. 8, pp. 1080 – 1088, 2013.
- [23] J. J. B. Biemond, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, “Tracking control for hybrid systems with state-triggered jumps,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 876 – 890, 2013.

- [24] N. Canty, T. OMahony, and M. Cychowski, “An output error algorithm for piecewise affine system identification,” *Control Engineering Practice*, vol. 20, pp. 444–452, January 2012.
- [25] L. Bako, G. Mercère, and S. Lecoeuche, “On-line structured subspace identification with application to switched linear systems,” *International Journal of Control*, vol. 82, no. 8, pp. 1496 – 1515, August 2009.
- [26] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, “An algebraic geometric approach to the identification of a class of linear hybrid systems,” in *Proc. of IEEE Conference on Decision and Control*, 2003.
- [27] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, “A greedy approach to identification of piecewise affine models,” *Lecture Notes in Computer Science*, vol. 2623, pp. 97 – 112, 2003.
- [28] J. Roll, A. Bemporad, and L. Ljung, “Identification of piecewise affine systems via mixed-integer programming,” *Automatica*, vol. 40, no. 1, pp. 37 – 50, 2004.
- [29] A. Juloski, S. Wieland, and W. P. M. H. Heemels, “A bayesian approach to identification of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 50, pp. 1520 – 1533, 2005.
- [30] N. H. Hudson, “Inference in hybrid systems with applications in neural prosthetics,” Phd Thesis, California Institute of Technology, Pasadena, California, September 2008.
- [31] A. Bemporad, J. Roll, and L. Ljung, “Identification of hybrid systems via mixed-integer programming,” in *In Proc. of IEEE Conference on Decision and Control*, Orlando, Florida, USA, 2001, pp. 786 – 792.
- [32] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, “A clustering technique for the identification of piecewise affine systems,” *Automatica*, vol. 39, no. 2, pp. 205 – 217, 2003.
- [33] J. Ragot, G. Mourot, and D. Maquin, “Parameter estimation of switching piecewise linear systems,” in *42nd IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 5783 – 5788.
- [34] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, “A bounded-error approach to piecewise affine systems identification,” *IEEE Transactions on Automatic Control*, vol. 50, pp. 1567 – 1580, 2005.
- [35] H. Nakada and K. T. and T. Katayama, “Identification of piecewise affine systems based on statistical clustering technique,” *Automatica*, vol. 41, no. 5, pp. 905 – 913, 2005.

- [36] L. Bako and R. Vidal, “Algebraic identification of switched mimo arx models,” in *Hybrid Systems: Control and Computation*, St Louis MO, USA, 2008.
- [37] R. S. Baptista, J. Y. Ishihara, and G. A. Borges, “A split and merge algorithm for identification of piecewise affine systems,” in *2011 American Control Conference*, San Francisco, CA, USA, June 29 - July 01 2011.
- [38] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche, “Identification of switched linear state space models without minimum dwell time,” in *15th IFAC Symposium on System Identification*, vol. 15, Brussels, Belgium, 2009, pp. 569 – 574.
- [39] V. Verdult and M. Verhaegen, “Subspace identification of piecewise linear systems,” in *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004, pp. 3838 – 3843.
- [40] H. S. Witsenhausen, “A class of hybrid-state continuous-time dynamic systems,” *IEEE Transactions on Automatic Control*, vol. 11, no. 2, pp. 161 – 167, April 1966.
- [41] M. S. Branicky, “Studies in hybrid systems: Modeling, analysis, and control,” PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, Electrical Engineering and Computer Science Dept., June 2005.
- [42] W. P. M. H. Heemels, B. D. Schutter, and A. Bemporad, “Equivalence of hybrid dynamical models,” *Automatica*, vol. 37, pp. 1085 – 1091, 2001.
- [43] M. J. P. García, “Uma contribuição ao controle de sistemas comutados,” Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, Outubro 2009.
- [44] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, pp. 407 – 427, 1999.
- [45] B. D. Schutter and T. V. den Boom, “On model predictive control for max-min-plus-scaling discrete event systems,” Technical Report bds :00-04, Control Lab, Fac. ITS, Delft Univ. Techn., Delft, The Netherlands,, Tech. Rep., 2000.
- [46] W. P. M. H. Heemels, J. M. Schumacher, and S. Weiland, “Linear complementarity systems,” *SIAM Journal of Applied Mathematics*, vol. 60, pp. 1234 – 1269, 2000.
- [47] A. J. V. der Schaft and J. M. Schumacher, “Complementarity modelling of hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 483 – 490, 1998.
- [48] B. D. Schutter, “Optimal control of a class of linear hybrid systems with saturation,” *SIAM Journal on Control and Optimization*, vol. 39, pp. 835 – 851, 2000.
- [49] E. D. Sontag, “Nonlinear regulation: The piecewise linear approach,” *IEEE Transaction on Automatic Control*, vol. 26, pp. 346 – 357, 1981.

- [50] J. Borges, V. Verdult, M. Verhaegen, and M. A. Botto, “A switching detection method based on projected subspace classification,” in *44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, December 2005, pp. 344 – 349.
- [51] J. Wang, “Identification of switched linear systems,” PhD Thesis, University of Alberta, Department of Electrical and Computer Engineering, 2013.
- [52] B. H. G. Barbosa, “Computação evolucionária e máquinas de comitê na identificação de sistemas não-lineares,” Tese de doutorado, Universidade Federal de Minas Gerais, Departamento de Engenharia Eletrônica, Outubro 2009.
- [53] L. Bako, “Contribution à l’identification de systèmes dynamiques hybrides,” PhD Thesis, l’Université des Sciences et Technologies de Lille, Nov 2008.
- [54] J. Wang and T. Chen, “Online identification of switched linear output error models,” in *IEEE International Symposium on Computer-Aided Control System Design (CACSD) Part of 2011 IEEE Multi-Conference on Systems and Control*, Denver, CO, USA, September 2011, pp. 1379 – 1384.
- [55] Y. Ma, R. Vidal, M. Morari, L. Thiele, and F. Rossi, “Identification of deterministic switched arx systems via identification of algebraic varieties,” *Hybrid Systems: Computation and Control*, vol. 3414 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/Heidelberg, pp. 449 – 465, 2005.
- [56] R. Vidal and B. Anderson, “Recursive identification of switched arx hybrid models: exponential convergence and persistence of excitation.” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004, pp. 32 – 37.
- [57] J. Stevek, A. Szucs, M. Kvasnica, S. Kozak, and M. Fikar, “Smart technique for identifying hybrid systems,” in *10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*, Herl’any, Slovakia, January 2012, pp. 383 – 388.
- [58] B. H. G. Barbosa, L. A. Aguirre, and A. P. Braga, “Algoritmo evolucionários na identificação de sistemas híbridos PWA,” in *X SBAI - Simpósio Brasileiro de Automação Inteligente*, São João del-Rei - MG - Brasil, Setembro 2011, pp. 111 – 116.
- [59] S. Ghosh and S. Maka, “A fuzzy clustering based technique for piecewise affine approximation of a class of nonlinear systems,” *Commun Nonlinear Sci Numer Simulat*, vol. 15, pp. 2235 – 2244, 2010.
- [60] X. Jin and B. Huang, “Identification of switched markov autoregressive exogenous systems with hidden switching state,” *Automatica*, vol. 48, pp. 436 – 441, 2012.

- [61] I. Maruta and T. Sugie, "Identification of pwa models via data compression based on l_1 optimization," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, December 2011, pp. 2800 – 2805.
- [62] C. Y. Lai, C. Xiang, and T. H. Lee, "Data-based identification and control of nonlinear systems via piecewise affine approximation," *IEEE Transaction on Neural Networks*, vol. 22, no. 12, pp. 2189 – 2200, December 2011.
- [63] N. Ozay, M. Sznajder, C. M. Lagoa, and O. I. Camps, "A sparsification approach to set membership identification of switched affine systems," *IEEE Transaction on Automatic Control*, vol. 57, no. 3, pp. 634 – 648, March 2012.
- [64] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, pp. 688 – 677, February 2011.
- [65] R. Vidal, "Identification of pwarx hybrid models with unknown and possibly different orders." in *Proceeding of the IEEE American Control Conference*, Boston, MA., 2004, pp. 547 – 552.
- [66] S. Paoletti, "Identification of piecewise affine models," PhD Thesis, Department of Information Engineering, University of Siena, Siena, Italy, 2004.
- [67] M. E. Gegúndez, J. Aroba, and J. M. Bravo, "Identification of piecewise affine systems by means of fuzzy clustering and competitive learning," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 1321– 1329, June 2008.
- [68] A. H. Shirdel, K.-M. Björk, and H. T. Toivonen, "Identification of linear switching system with unknown dimensions," in *47th International Conference on System Science*, Hawaii, USA, 2014.
- [69] A. Nakabayashi, S. Ukai, H. Wada, and T. Ohtani, "A bayesian robust identification method for piecewise affine autoregressive exogenous model from outlier-contaminated data," in *SICE Annual Conference 2013*, Nagoya University, Nagoya, Japan, September 2013.
- [70] Z. Lassoued and K. Abderrahim, "A new clustering technique for the identification of pwarx hybrid models," in *9th Asian Control Conference*, Univ. of Gabes, Gabes, Tunisia, 2013.
- [71] K. M. Pekpe, K. Gasso, G. Mourot, and J. Ragot, "Subspace identification of switching model," in *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

- [72] K. M. Pekpe, G. Mourot, K. Gasso, and J. Ragot, “Identification of switching systems using change detection technique in the subspace framework,” in *43rd IEEE Conference on Decision and Control*, vol. 4, Paradise Island, Bahamas, December 2004, pp. 3720 – 3725.
- [73] K. Huang, A. Wagner, and Y. Ma, “Identification of hybrid linear time-invariant systems via subspace embedding and segmentation (SES).” in *43rd IEEE Conference on Decision and Control*, vol. 3, Paradise Island, Bahamas, December 2004, pp. 3227 – 3234.
- [74] J. B. Borges, V. Verdult, and M. Verhaegen, “Iterative subspace identification of piecewise linear systems,” in *14th IFAC Symposium on System Identification*, Newcastle, Australia, 2006, pp. 368 – 373.
- [75] J. Borges, “State-space system identification new developments and applications,” Phd Thesis, Instituto Superior Técnico, Lisboa, 2007.
- [76] S. Lecoeuche, G. Mercère, and H. Amadou-Boubacar, “Modelling on non stationary systems based on a dynamical decision space,” in *14th IFAC Symposium on System Identification*, vol. 14, no. 1, Newcastle, Australia, 2006, pp. 1222 – 1227.
- [77] K. M. Pekpe and S. Lecouche, “Online clustering of switching models based on a subspace framework,” *Nonlinear Analysis: Hybrid Systems*, vol. 2, no. 3, pp. 735 – 749, August 2008.
- [78] L. Bako, V. L. Le, F. Lauer, and G. Bloch, “Identification of MIMO switched state-space models,” in *American Control Conference*, Washington, DC, USA, 2013.
- [79] P. Egbunonu and M. Guay, “Identification of switched linear systems using subspace and integer programming techniques,” *Nonlinear Analysis: Hybrid Systems*, vol. 1, no. 4, pp. 577 – 592, December 2007.
- [80] H. Blom, “An efficient filter for abruptly changing systems,” in *Proc. of 23rd Conference on Decision and Control*, vol. 4, no. 30, December 1984, pp. 656 – 658.
- [81] H. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with markovian switching coefficients,” *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780 – 783, August 1988.
- [82] C. J. M. Delgado, “Identificação no subespaço de estados de sistemas lineares - novas abordagens e recursividade,” Tese de Doutoramento, Faculdade de Engenharia FEUP, Universidade do Porto, Maio 2004.
- [83] L. Ljung, *System Identification - Theory for the User*, second edition ed. Linköping University, Sweden: Prentice Hall PTR, 2009.

- [84] B. L. Ho and R. E. Kalman, “Effective construction of linear state-variable models from input/output functions.” *Regelungstechnik*, vol. 14, no. 12, pp. 545 – 592, 1966.
- [85] H. Zeiger and A. McEwen, “Approximate linear realization of given dimension via ho’s algorithm,” *IEEE Transaction on Automatic Control*, vol. 19, no. 2, p. 1953, April 1974.
- [86] R. A. Ricco, A. Verly, B. O. S. Teixeira, and L. A. Aguirre, “Identificação usando técnicas de subespaços de uma planta de bombeamento de água - um estudo de caso,” in *X SBAI - Simpósio Brasileiro de Automação Inteligente*, São João del-Rei - MG - Brasil, Setembro 2011, pp. 296 – 301.
- [87] P. van Overschee and B. L. Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications*, 1st ed. Dordrecht, The Netherlands: Kluwer Academic Publishers, May 1996.
- [88] J. P. Matura, “Detecção de falhas em sistemas dinâmicos com redes bayesianas aprendidas a partir de estimação de estados,” Tese de Doutorado, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, 2006.
- [89] B. Huang and R. Kadali, *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-driven Subspace Approach*. London: Springer-Verlag, 2008.
- [90] T. Katayama, *Subspace Methods for System Identification*. London: Springer-Verlag, 2005.
- [91] G. Mercèrea and L. Bako, “Parameterization and identification of multivariable state-space systems: A canonical approach,” *Automatica*, vol. 47, no. 8, pp. 1547 – 1555, August 2011.
- [92] S. D. M. Borjas and C. Garcia, “Subspace identification using the integration of moesp and n4sid methods applied to the shell benchmark of a distillation column,” in *Brazilian Conference on Dynamics Control and their Applications*, June 2010.
- [93] D. Bauer, “Order estimation for subspace methods,” *Automatica*, vol. 37, pp. 1561 – 1573, 2001.
- [94] H.-J. Kima and J. E. Cavanaughb, “Model selection criteria based on kullback information measures for nonlinear regression,” *Journal of Statistical Planning and Inference*, vol. 134, no. 2, pp. 332 – 349, October 2005.
- [95] M. D. Cruz, “O problema de clusterização automática,” Tese de Doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Julho 2010.

- [96] T. Bengtsson^a and J. E. Cavanaugh^b, “An improved akaike information criterion for state-space model selection,” *Computational Statistics & Data Analysis*, vol. 50, no. 10, pp. 2635 – 2654, June 2006.
- [97] A. García-Hiernaux, M. Jerez, and J. Casals, “Estimating the system order by subspace methods,” *Computational Statistics*, vol. 3, no. 27, pp. 411 – 425, 2012.
- [98] T. Cassar, K. P. Camilleri, and S. G. Fabri, “Order estimation of multivariate arma models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 494 – 503, June 2010.
- [99] W. Urtubey, “Identification of farm models by bayesian methods applied to stream flow data,” in *Proc. 9th Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2006.
- [100] H. Akaike, “Fitting autoregressive models for prediction,” *Annals of the Institute of Statistical Mathematics*, vol. 21, pp. 243 – 247, 1969.
- [101] L. K. Sen and M. Shitan, “The performance of AICC as an order selection criterion in arma time series models,” *Pertanika Journal of Science & Technology*, vol. 10, no. 1, pp. 25 – 33, 2002.
- [102] G. Schwarz, “Estimating the dimension of a model,” *Annals of Statistics*, vol. 6, pp. 461 – 464, 1978.
- [103] E. J. Hannan and B. G. Quinn, “The determination of the order of an autoregression,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 2, pp. 190 – 195, 1979.
- [104] G. Liang, D. M. Wilkes, and J. A. Cadzow, “ARMA model order estimation based on the eigenvalues of the covariance matrix,” *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3003 – 3009, October 1993.
- [105] C. E. Davila and H. L. Chian, “An algorithm for pole-zero system model order estimation,” *IEEE Trans. Signal Process.*, vol. 43, no. 4, pp. 1013 – 1017, April 1995.
- [106] K. Peternell, “Identification of linear dynamical systems by subspace and realization-based algorithms.” PhD thesis, TU Wien, 1995.
- [107] R. Xu and D. C. W. II, *Clustering*, L. Hanzo, Ed. A John Wiley & Sons, Inc., 2009.
- [108] R. Xu, , and D. C. W. II, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645 – 678, May 2005.
- [109] G. A. Borges and M.-J. Aldon, “A split-and-merge segmentation algorithm for line extraction in 2D range images,” in *5th International Conference on Pattern Recognition*, vol. 1, 2000, pp. 441 – 444.

- [110] M. Halkid, Y. Batistakis, and M. Vazirgiannis, “Clustering algorithms and validity measures,” in *International Conference on Scientific and Statistical Database Management. SSDBM*, 2001, pp. 2 – 22.
- [111] J. R. Kettenring, “The practice of cluster analysis,” *Journal of Classification*, vol. 23, no. 1, pp. 3 – 30, Jun 2006.
- [112] G. Ferrari-Trecate and M. Muselli, “Single-linkage clustering for optimal classification in piecewise affine regression,” in *IFAC Conference on Analysis and Design of Hybrid Systems*, Saint-Malo, 2003.
- [113] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium*, vol. 1, 1967, pp. 281 – 297.
- [114] E. Forgy, “Cluster analysis of multivariate data: efficiency vs. interpretability of classifications,” *Biometrics*, vol. 21, pp. 768 – 780, 1965.
- [115] R. Krishnapuram and J. Keller, “The possibilistic c-means algorithm: Insights and recommendations,” *IEEE transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385 – 393, 1996.
- [116] H. Frigui and R. Krishnapuram, “A robust competitive clustering algorithm with applications in computer vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 450 – 465, May 1999.
- [117] G. Ackerson and K. Fu, “On state estimation in switching environments,” *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 10 – 17, Feb. 1970.
- [118] J. Tugnait, “Detection and estimation for abruptly changing systems,” *Automatica*, vol. 18, pp. 607 – 615, 1982.
- [119] P. D. Hanlon and P. S. Maybeck, “Multiple-model adaptive estimation using a residual correlation kalman filter bank,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 393 – 406, Apr 2000.
- [120] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, “Interacting multiple model methods in target tracking: a survey,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103 – 123, January 1998.
- [121] R. V. Lopes, G. A. Borges, and J. Y. Ishihara, “New algorithm for identification of switched linear systems,” in *American Control Conference (ACC2013)*, Washington, DC, June 2013.
- [122] R. V. Lopes, J. Y. Ishihara, and G. A. Borges, “Procedimento baseado em clustering para identificação de sistemas dinâmicos híbridos no espaço de estados,” in *XI Simpósio Brasileiro de Automação Inteligente (SBAI)*, Fortaleza, Brasil, Outubro 2013.

- [123] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood for incomplete data via the em algorithm," *Journal of Royal Statistical Society B*, vol. 39, pp. 1 – 38, 1977.
- [124] L. H. Rajaoarisoa, N. K. M'sirdi, E.-K. Boukas, and A. Naamane, "Subspace technique for identification of hybrid complex systems." in *16th Mediterranean Conference on Control and Automation*, Ajaccio, Jun 2008, pp. 89 – 94.
- [125] A. K. Hajdasinski and A. A. H. Damen, "Realization of the markov parameter sequences using the singular value decomposition of the hankel matrix," Eindhoven University of Technology - Department of Electrical Engineering, Eindhoven, The Netherlands, Tech. Rep., May 1979.
- [126] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2006 – 2021, Nov. 2010.
- [127] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "Identification of piecewise affine and hybrid systems," in *Proceedings of the American Control Conference*, vol. 5, Arlington, VA, June 2001, pp. 3521 – 3526.
- [128] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. The Oxford series in electrical and computer engineering, 1999.
- [129] M. Verhaegen, "Identification of the deterministic part of mimo state space models given in innovation form from input-output data," *Automatica (Special issue on Statistical Signal Processing and Control)*, vol. 30, no. 1, pp. 61 – 74, 1994.
- [130] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, "A recursive identification algorithm for switched linear/affine models," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 242 – 253, May 2011.
- [131] C. Juang, *Applied system identification*, E. C. N. Prentice-Hall, Ed., 1994.

APÊNDICES

A. LISTA DE PUBLICAÇÕES

Anteriormente ao aprofundamento desta tese no tema de identificação de sistemas híbridos, a revisão da literatura acerca de sistemas híbridos levou ao estudo sobre desenvolvimento de sistemas de controle para veículos aéreos não-tripulados e estimação híbrida, culminando com as seguintes publicações:

- **R. J. M. Afonso, R. V. Lopes, R. K. H. Galvão. Controle Preditivo com garantia de estabilidade nominal aplicado a um helicóptero com três graus de liberdade empregando relaxamento das restrições de saída. In: XVIII Congresso Brasileiro de Automática (CBA 2010), Bonito - MS, 2010:** Este trabalho abordou as vantagens do emprego de uma política de relaxamento de restrições sobre as variáveis de saída em uma formulação de Controle Preditivo quando identificada a ocorrência de não-factibilidade do problema de otimização com restrições, cuja solução é requerida por este controlador. A formulação de Controle Preditivo adotada neste estudo possui garantia de estabilidade nominal assintótica, desde que o problema de otimização a ser resolvido seja factível. Para ilustração, apresenta-se um estudo de caso envolvendo um modelo de simulação não-linear de um mini-helicóptero com três graus de liberdade.
- **R. V. Lopes, P. H. R. Q. E. A. Santana, G. A. Borges, J. Y. Ishihara. Model Predictive Control applied to tracking and attitude stabilization of a VTOL quadrotor aircraft. In: 21st International Congress of Mechanical Engineering (COBEM 2011), Natal - RN, 2011:** Este trabalho apresentou uma nova abordagem de controle de um mini-helicóptero do tipo quadricóptero usando uma técnica de controle preditivo. A formulação empregada é baseada em um modelo não-linear obtido pela linearização da dinâmica da planta em torno de um ponto de equilíbrio. A sequência de controle ótimo é implementada seguindo uma estratégia de controle retrocedente. A otimização é repetida a cada instante de amostragem levando em consideração as novas leituras dos sensores. A habilidade do MPC em respeitar restrições operacionais nas variáveis de entrada e/ou saída é explorada para garantir o controle de posição e a estabilização de todo o sistema com um único controlador na malha de controle. Resultados de simulação são comparados com os controladores convencionais PID e backstepping não-linear.
- **Z. R. M. Junior, R. V. Lopes. Modelagem de um helicóptero elétrico em uma plataforma 3DOF. In: XIX Congresso Nacional de Estudantes de Engenharia Mecânica (CREEM 2012), São Carlos, 2012.:** Este trabalho apresentou a modelagem matemática de um helicóptero elétrico fixo a uma plataforma com 3 graus de liberdade que está sendo desenvolvida na Universidade de Brasília. A validação do modelo é realizada através de análises das simulações numéricas em ambiente Matlab/Simulink.

Uma vez delimitado o tema da tese ao campo identificação de sistemas dinâmicos híbridos, a continuação da pesquisa deu origem aos seguintes trabalhos científicos:

- **R. V. Lopes, G. A. Borges, J. Y. Ishihara. New Algorithm for Identification of Discrete-Time Switched Linear Systems. American Control Conference (ACC 2013), Jun. 2013:** Este trabalho apresentou o primeiro algoritmo desenvolvido nesta tese para identificação de sistemas dinâmicos híbridos. A metodologia off-line proposta explora o uso de técnicas de identificação em subespaços, clustering e classificação de dados para estimar a quantidade, a ordem e as matrizes dos sub-modelos. Adicionalmente, permite obter uma estimativa dos instantes de chaveamento do sistema. Uma característica importante do algoritmo é que as matrizes dos diferentes sub-modelos podem ser combinadas diretamente pois são obtidas na mesma base de estados. A eficiência do método desenvolvido é ilustrada através de um exemplo de simulação.
- **R. V. Lopes, G. A. Borges, J. Y. Ishihara. Procedimento Baseado em Clustering para Identificação de Sistemas Dinâmicos Híbridos no Espaço de Estados. Simpósio Brasileiro de Automação Inteligente (SBAI 2013), Out. 2013:** Este trabalho apresentou a versão modificada do primeiro algoritmo. A principal contribuição do algoritmo é a estimação dos modelos locais do sistema a partir das estimativas fornecidas pelo algoritmo de clustering. A metodologia proposta foi comparada com o Algoritmo 1 através de simulações numéricas.
- **R. V. Lopes, G. A. Borges, J. Y. Ishihara. Algoritmo para Identificação de Sistemas Híbridos PWA no Espaço de Estados. Congresso Brasileiro de Automática (CBA 2014), Set. 2014:** Este artigo endereça o problema de identificação de uma classe de sistemas dinâmicos híbridos conhecida como sistemas híbridos com modelos afins por partes descritos no espaço de estado. A abordagem proposta estendeu uma metodologia baseada em clustering, inicialmente restrita a identificação de sistemas chaveados lineares, para sistemas híbridos com modelos afins por partes. As principais contribuições do algoritmo são a não necessidade de um número elevado de chaveamentos entre os diferentes modos do sistema e a possibilidade de considerar sistemas sujeitos a ação de ruído colorido. A eficiência do algoritmo proposto foi ilustrada através de exemplos de simulação. O trabalho apresentado neste artigo é o Algoritmo 5 apresentado nesta tese.
- **P. H. R. Q. E. A. Santana, R. V. Lopes, B. G. R. Amui, G. A. Borges, J. Y. Ishihara and Brian Williams. A New Filter for Hybrid Systems and its Applications to Robust Attitude Estimation. IEEE Conference on Decision and Control (CDC 2014) (Submetido):** Este trabalho aborda o problema de filtragem estocástica para sistemas híbridos. Inspirado por uma aplicação de estimação de atitude de um quadricóptero com

sensores defeituosos, este artigo apresentou um novo filtro híbrido chamado de IP-MHMF. Este filtro generaliza o bem conhecido IMM e introduz um novo passo de eliminação de hipóteses considerando uma maior quantidade de informações quando comparado com os algoritmos existentes na literatura. Os resultados teóricos são validados por meio de experimentos reais e comparações com algoritmos existentes na literatura. Os resultados obtidos apresentam o IP-MHMF como a melhor alternativa em termos de robustez das estimativas a perturbações nas medidas dos sensores, estabilidade, erro médio quadrático e carga computacional.

- **R. V. Lopes, G. A. Borges, J. Y. Ishihara. Identification of state-space switched linear systems with data classification (A ser submetido):** Este artigo aborda o problema de identificação de sistemas dinâmicos híbridos chaveados MIMO no espaço de estados e apresenta a metodologia de identificação proposta no Algoritmo 3 desta tese. Simulações numéricas são utilizadas para validação da metodologia e demonstração do ganho de desempenho obtido com as modificações propostas com relação ao Algoritmo 1 desta tese.
- **R. V. Lopes, G. A. Borges, J. Y. Ishihara. New Framework for Identification of Discrete-Time Switched Linear Systems. IEEE Transactions on Automatic Control, Vol. X, No. X, 2014 (Submetido):** Este artigo apresenta a estrutura de identificação mais completa (Algoritmo 6) apresentada nesta tese para a identificação de sistemas dinâmicos híbridos chaveados MIMO no espaço de estados. Assim, neste trabalho a metodologia apresentada consiste em calcular iterativamente a função densidade de probabilidade a posteriori dos modelos discretos e contínuos. Isto é realizado explorando o uso combinado de métodos de identificação em subespaços, clustering e filtragem híbrida para segmentar os dados em diferentes sub-modelos e realizar a estimação de modelos locais. O desempenho do algoritmo proposto é demonstrado através de exemplos numéricos e comparação dos resultados com uma abordagem existente na literatura.