

Filtragem *via* Métodos de Monte Carlo para Processos Lineares com Saltos Markovianos

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do
título de Mestre em Engenharia Elétrica

por

Gustavo Vinicius Lourenço Moisés

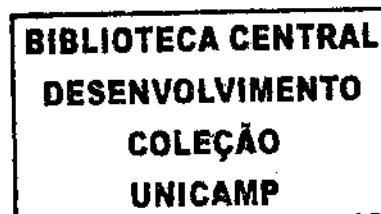
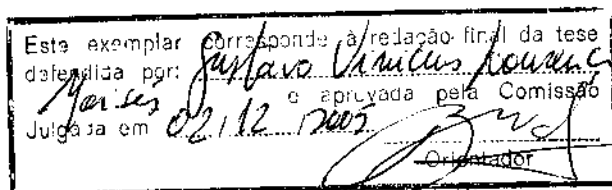
Engenheiro de Controle e Automação – UnB

02 de Dezembro de 2005.

Orientador: **Prof. Dr. João Bosco Ribeiro do Val** FEEC/UNICAMP

Banca Examinadora

Prof. Dr. Eduardo Fontoura Costa	ICMC/USP
Prof. Dr. Wagner Caradori do Amaral	FEEC/UNICAMP
Prof. Dr. João Bosco Ribeiro do Val	FEEC/UNICAMP (orientador)



UNIDADE	36
Nº CHAMADA	TIUMCAMP
	M7284
V	EX
TOMBO BC	68782
PROC.	16.123-06
C	☐
D	☒
PREÇO	11,00
DATA	02/06/06

Bib. id 382742

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M7284	Moisés, Gustavo Vinicius Lourenço
M7291	Filtragem via métodos de Monte Carlo para processos lineares com saltos Markovianos / Gustavo Vinicius Lourenço Moisés. –Campinas, SP: [s.n.], 2005.
	Orientador: João Bosco Ribeiro do Val
	Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.
	1. Filtros adaptativos. 2. Sistemas estocásticos. 3. Sistemas lineares. 4. Markov, Processos de. I. Val, João Bosco Ribeiro do. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Titulo em Inglês: Filtering via Monte Carlo methods for linear processes with Markovian jumps

Palavras-chave em Inglês: Adaptive filters, Stochastic systems, Linear systems, Markov chains

Área de concentração: Automação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Eduardo Fontoura Costa, Wagner Caradori do Amaral

Data da defesa: 02/12/2005

Resumo

Esta dissertação possui como tema a filtragem via Métodos de Monte Carlo para Cadeia de Markov. Através do estudo e da análise dos algoritmos de amostragem estocástica aliados às implementações numéricas, foi desenvolvida uma metodologia para avaliar e comparar as diversas técnicas de filtragem encontradas na literatura. Aplicações associando a filtragem recursiva ao controle via horizonte retrocedente também foram utilizadas para verificar o desempenho e a estabilidade do conjunto filtro/controle.

Abstract

The dissertation's theme is the filtering problem via Markov Chain Monte Carlo methods. Combining the study and the analysis of the stochastic sampling algorithms with numerical implementations, we developed a methodology to evaluate and compare several filters in literature. Applications of recursive filtering in association with receding horizon control techniques were used to verify the quality and stability of the filter/control combination.

2006111017

ÀS MULHERES DA MINHA VIDA
BEATRIZ, CYNTHIA E DAISE.

Agradecimentos

Para homenagear os meus bravos e incansáveis incentivadores que de alguma forma contribuíram para o desenvolvimento desta dissertação, venho através de uma estória declamar a minha gratidão. O viajor que vai pelos caminhos da ciência nunca caminha sozinho. Pensam muitos que o estudo é poço de solidão compenetrada onde a dedicação e a persistência são as únicas ferramentas de trabalho, mas esquecem, esses tolos, que sempre acompanha o viajante a trupe do Ripilica, grupo de extensa monta que a todo momento faz a extensa jornada parecer simples caminhada. Fazem parte deste grupo exemplares, um pouco excêntricos, a quem o singelo cientista faz questão de agradecer. Obrigado:

- Prof. João Bosco, pela maestria com que soube afinar tão destoantes timbres orquestrais (meus pensamentos e questionamentos) com paciência e competência;
- Querida esposa, pela sua persistência, apoio e, principalmente, pelo seu amor que completa e torna perfeitos e felizes todos os meus dias;
- Filha, pelos seus cucos tão carinhosos e reconfortantes que transformaram os dias estressantes em momentos raros e fugidios;
- Mãe, pelas suas preocupações e conselhos ao telefone que iluminaram e guiaram o meu trabalho;
- Pai, pelo seu exemplo de ética e hombridade que permeia meus preceitos;
- Família, por estar sempre me apoiando e incentivando;
- Amigos do Departamento de Telemática, pelos sorrisos do dia-a-dia;
- Professores Eduardo e Wagner, pelo empenho dedicado à correção deste trabalho;

Agradeço ainda à FAPESP pelo apoio financeiro do primeiro ano de pesquisa e à PETROBRAS por incentivar o término desta dissertação.

Finalmente, agradeço a DEUS por me acompanhar a todo instante e por me carregar nos braços nos momentos mais difíceis do meu caminhar.

Notação

A seguir, apresentamos uma breve lista de símbolos que serão usados com maior frequência neste trabalho:

$E[\cdot]$	Esperança matemática
$p(\mathcal{C})$	Probabilidade de ocorrência do evento $\mathcal{C}$
$\bar{p}(\mathcal{C})$	Verossimilhança de ocorrência do evento $\mathcal{C}$
S	Espaço de estados da Cadeia de Markov
θ_k	Estado da Cadeia de Markov no instante k
x_k	Estado contínuo no instante k
y_k	Vetor de observação no instante k
p_{ij}	Probabilidade de transição do estado i para j
$\mathbb{P} = [p_{ij}], \forall i, j \in S$	Matriz de transição de probabilidade
μ_{k+t}	Distribuição de probabilidade no instante $k + t$ dado o conhecimento até instante k
π	Vetor de probabilidades do estado inicial
$y_{1:k}$	Conjunto de vetores $\{y_1, y_2, \dots, y_k\}$
O_k	Observação no instante k
U'	Transposta da matriz U
U^{-1}	Inversa da matriz U
$\ \cdot\ $ ($\ U\ $)	Norma vetorial padrão em $\mathbb{R}^n$ (correspondente norma induzida da matriz U)
I_n	Matriz identidade de ordem $n \times n$
$0_{n \times m}$	Matriz de zeros de ordem $n \times m$
$\mathbb{I}_{\{\cdot\}}$	Função delta de Dirac
$\text{tr}\{\cdot\}$	Traço de Matriz
$\mathbf{U}$	Conjunto de matrizes $\{U_1, U_2, \dots, U_s\}$
$\mathcal{E}_i(\mathbf{U}) = \sum_{j \in S} p_{ij} U_j$	Operador de transição sobre o conjunto $\mathbf{U}$

$\langle \mathbf{U}, \mathbf{V} \rangle = \sum_{i \in S} \text{tr}\{U_i' V_i\}$	Produto interno de $\mathbf{U}$ e $\mathbf{V}$
$\mathcal{N}(m, \Phi)$	Distribuição normal com média m e variância Φ
λ	Parâmetros do modelo
$[A]$	Menor inteiro mais próximo do número real A

Sumário

Resumo / Abstract	i
Agradecimentos	v
Notação	vii
1 Introdução	1
1.1 Descrição Geral dos Sistemas Lineares Sujeitos a Saltos Markovianos	1
1.2 Filtragem	3
1.3 Estrutura da Dissertação	5
2 Conceitos básicos	7
2.1 Processos Discretos de Markov	7
2.2 Modelo de Markov de Cadeia Escondida	9
2.3 Associando Filtragem ao Modelo	11
2.4 Formulação do Problema	12
2.5 Cenários	13
3 Filtragem em <i>batch</i>	15
3.1 Kalman <i>Smoother</i>	16
3.2 <i>Smoother</i> do Modelo de Cadeia Escondida	17
3.2.1 Versão Clássica	17
3.2.2 <i>Smoother</i> Adaptado	23
3.3 Filtragem sem Observação de Estados	24
3.3.1 Algoritmo de Maximização do Valor Esperado	26
3.3.2 <i>Data Augmentation</i>	28
3.3.3 Algoritmo Metropolis-Hastings/ <i>Data Augmentation</i> com <i>Simulated Annealing</i>	32
3.3.4 Algoritmo do Mínimo Erro Quadrático Médio	34
3.3.5 Estimativa de $\theta_{1:T}$ no Sentido MAP Marginal	36

4	Filtragem Recursiva	39
4.1	Filtro de Partículas de Monte Carlo	39
4.2	Redução da Variância	42
4.3	Amostragem Sequencial por Importância	43
4.3.1	Escolha da Distribuição por Importância	44
4.3.2	Degeneração do Algoritmo	44
4.3.3	Reamostragem	45
4.4	Algoritmo do Filtro de Partículas de Monte Carlo	47
4.5	Questões de Implementação	49
5	Aplicações	51
5.1	Identificação de Fácies Litológicas	51
5.2	Rastreamento de Alvo Móvel	58
5.3	Sensibilidade Paramétrica	62
5.3.1	Ponto Inicial	63
5.3.2	Ruído Associado	66
5.3.3	Matriz de Transição	67
5.4	Trajetórias para Diferentes Matrizes de Transição	68
6	Controle via Horizonte Retrocedente	71
6.1	Formulação do Problema	72
6.2	Cálculo da Ação de Controle	73
6.2.1	Cenários com Observação do estado de Markov	73
6.2.2	Cenários sem Observação do estado de Markov	74
6.2.3	Método Variacional	78
6.3	Critério de Comparação	79
6.4	Aplicações de Controle	80
6.4.1	O Braço do Robô	80
6.4.2	Sensibilidade do Conjunto Filtro-Controle a Alteração da P	84
7	Conclusões	87
A	Algoritmo de Metropolis	91
	Bibliografia	93

Lista de Figuras

1.1	Representação de um possível diagrama de estados para o exemplo econômico.	2
2.1	Diagrama do modelo de previsão com 3 estados de Markov.	8
3.1	Estrutura geral com combinação de dois estimadores.	25
4.1	Evolução do erro quadrático médio com o aumento do número de partículas com confiabilidade de 95%.	50
4.2	Comportamento do tempo computacional por iteração com o aumento do número de partículas.	50
5.1	Perfil radioativo associado à estimativa via (A) <i>Data Augmentation</i> e (B) <i>Data Augmentation</i> com <i>Simulated Annealing</i>	55
5.2	Perfil radioativo associado à estimativa via (A) MMAP2 versão estocástica e (B) MMAP1 versão determinística.	56
5.3	Perfil radioativo associado à estimativa via (A) Mínimo Erro Quadrático e (B) Filtro de Partículas.	57
5.4	Trajetória do alvo móvel e sua observação com $s_{x,0} = -500$ e $s_{y,0} = 500$	59
5.5	Módulo da velocidade e sua observação em função da amostra k com $\mathcal{V}_{x,0} = -20$ e $\mathcal{V}_{y,0} = 10$	60
5.6	(A) Evolução do erro quadrático para λ no cálculo das estimativas. (B) Detalhe de (A).	62
5.7	(A) Evolução do erro quadrático para $\lambda_1^{(a)}$ no cálculo das estimativas. (B) Detalhe de (A).	64
5.8	Evolução do erro quadrático para $\lambda_1^{(b)}$ no cálculo das estimativas.	65
5.9	(A) Evolução do erro quadrático para λ_7 no cálculo das estimativas. (B) Detalhe de (A).	69
5.10	Evolução do erro quadrático para λ_8 no cálculo das estimativas.	70
5.11	Evolução do erro quadrático para λ_9 no cálculo das estimativas.	70

6.1	(A) Evolução do ângulo médio da junta inercial Ω_k ; (B) Evolução da ação de controle média $\bar{u}_k$ para $T_{sim} = 50$	82
6.2	(A) Evolução da estimativa média do custo funcional J_k^N ; (B) Evolução da média do custo real realizado $Custo_k^N$ para $T_{sim} = 50$	83
6.3	(A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_1$	84
6.4	(A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_2$	85
6.5	(A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_3$	86

Lista de Tabelas

2.1	Probabilidade de ocorrência das cores por urna.	10
5.1	Erro quadrático médio, desvio padrão do erro quadrático e tempo computacional dos algoritmos de filtragem sem observação.	55
5.2	Erro quadrático médio, desvio padrão médio e tempo computacional dos algoritmos de filtragem sem observação para o conjunto λ	61
5.3	Erro quadrático médio dos algoritmos de filtragem sem observação para $\lambda_1^{(a)}$ e λ	63
5.4	Erro quadrático médio dos algoritmos de filtragem sem observação para $\lambda_1^{(b)}$ e λ	65
5.5	Erro quadrático médio dos algoritmos de filtragem sem observação para λ e λ_3 até λ_6	67
5.6	Erro quadrático médio dos algoritmos de filtragem sem observação para λ e λ_7 até λ_9	68
5.7	Erro quadrático médio dos algoritmos de filtragem para diferentes trajetórias para λ e λ_7 até λ_9	69
6.1	Valores dos parâmetros sujeitos a saltos markovianos para robô ERA.	81

Capítulo 1

Introdução

Neste primeiro capítulo descreveremos as principais motivações do estudo das técnicas de filtragem aplicadas a sistemas lineares sujeitos a saltos markovianos, assim como a evolução dos algoritmos via métodos de Monte Carlo para cadeia de Markov.

Na última seção, descreveremos como os capítulos seguintes estão organizados, enfatizando os principais objetivos de nossa investigação.

1.1 Descrição Geral dos Sistemas Lineares Sujeitos a Saltos Markovianos

Suponha que num dado país, o desempenho da economia possa assumir um dos três modos de operação: *ruim*, *neutro* e *bom* e que a autoridade monetária responsável pela economia possua um modelo matemático que represente com certo grau de confiabilidade cada um destes estados. Considere que de um instante para o outro fatores externos obriguem o modo de operação a sofrer mudanças abruptas, alterando repentinamente, por exemplo, o desempenho da economia de *bom* para *ruim*. Para complicar um pouco mais, imagine que não temos acesso ao modo de operação do sistema e que a única informação do sistema advém de um vetor de observações com distúrbio (ruído) associado ao modo atual. Se nenhuma ação eficiente de controle for implementada, o comportamento do sistema se desviará do desejado. Na figura 1.1 podemos visualizar as transições entre os modos de operação da economia.

Posto isso, deseja-se modelar processos que possuam incertezas em seus parâmetros, de tal modo que as mudanças inesperadas sejam representadas no modelo. Em alguns casos, modelos desta natureza podem ser caracterizados pelos Sistemas Lineares sujeitos a Saltos Markovianos, que formam uma classe de processos usada para modelar sistemas dinâmicos sujeitos a mudanças abruptas em suas estruturas. Adiante, usaremos as siglas “sistemas SLSM” para designar esta classe de sistemas.

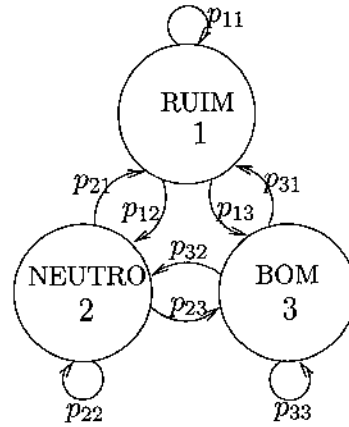


Figura 1.1: Representação de um possível diagrama de estados para o exemplo econômico.

Sistemas SLSM formam uma importante classe de sistemas lineares estocásticos, e um considerável interesse tem sido focado nestes sistemas nos últimos anos, particularmente no que concerne à análise, condições de estabilidade e problemas de controle ótimo desses sistemas. Podemos citar (Ji e Chizeck, 1990a; Ji e Chizeck, 1990b; Costa e Fragoso, 1993; Costa e Fragoso, 1995; Costa et al., 2005) como importantes contribuições no desenvolvimento teórico de SLSM. Os processos SLSM podem ser utilizados para modelar sistemas dinâmicos sujeitos a fenômenos aleatórios que apresentam mudanças abruptas em sua estrutura ou parâmetros, tais como troca ou falha de componentes, mudanças ambientais súbitas, modificação do ponto de operação de um sistema linear ou não linear, alterações repentinas em parâmetros de processos econômicos, entre outras. Aplicações podem ser encontradas, como exemplo, em sistemas aeronáuticos (Athans et al., 1977), sistemas robóticos (Saridis, 1983), receptores térmicos solares (Sworder e Rogers, 1983), modelos macroeconômicos (do Val e Basar, 1999) e indústria manufatureira de papel (Khanbaghi et al., 2002).

De forma geral, os SLSM são sistemas estocásticos cuja dinâmica altera-se de forma abrupta em certos instantes aleatórios e se comportam como sistemas lineares entre estes instantes. As alterações na dinâmica se referem a mudanças repentinas em parâmetros do sistema cujo valor evolui de acordo com uma Cadeia de Markov, isto é, a mudança para outro modo de operação ocorre de acordo com certa probabilidade (conhecida *a priori*) dependente apenas do modo de operação no qual o sistema se encontre em cada instante.

Falhas em atuadores ou sensores, ou grandes alterações na dinâmica básica do sistema, além da possível presença de ruídos atuando ininterruptamente, são ocorrências que precisam ser levadas em conta tanto nas técnicas de filtragem quanto na síntese do controlador. Assim, devido à característica essencialmente aleatória das mudanças abruptas, a abordagem determinística torna-se inadequada. Por exemplo, no caso do controle, quando as alterações nos

parâmetros são pequenas, a abordagem via controle robusto da teoria de sistemas lineares, na qual se considera intervalos ou regiões dentro dos quais os parâmetros devam permanecer, ou variações paramétricas limitadas através de normas das matrizes do sistema, são em geral adequadas. Entretanto, se extensas forem as alterações nos parâmetros, o conservadorismo inerente a estas abordagens pode tornar inadequada a sua aplicação. Nestes casos, a utilização de um modelo estocástico baseado em pontos representativos que descrevam vários possíveis cenários ou combinações de parâmetros, como é o caso dos sistemas SLSM, é claramente preferível.

Os sistemas SLSM podem ser modelados por um certo número de possíveis formas ou modos de operação, em que cada uma expressa uma possível combinação de eventos. Em particular, se cada uma dessas formas for linear e a transição entre elas possuir estrutura estocástica Markoviana, estas transições podem ser representadas através de uma Cadeia de Markov com um número de estados finito (ou eventualmente enumerável infinito), dando origem aos “Sistemas Lineares sujeitos Saltos Markovianos”. Denominamos por *saltos* os instantes em que ocorrem transições nos modos de operação do sistema.

1.2 Filtragem

As técnicas de filtragem têm como objetivo estimar o vetor de estados de um sistema dinâmico, vetor este que contém toda a informação relevante requerida para descrever o sistema sobre investigação.

A filtragem se faz necessária seja devido ao acesso parcial dos estados, seja pela presença de ruído que impossibilita a observação direta deles. Com a recuperação do vetor de estados, poderemos desenvolver uma estratégia de controle que venha a atender os critérios de desempenho desejados, assim como obter uma estimativa dos estados, problema de interesse em muitas aplicações.

A filtragem dos sistemas SLSM é utilizada em vários campos do processamento de sinais, incluindo processamento de sinais sísmicos (Mendel, 1990), comunicação digital como a supressão da interferência em sistemas CDMA de espectros dispersos (Logothetis e Krishnamurthy, 1999), rastreamento de alvo móvel (Bar-Shalom e Li, 1995), (Mazor et al., 1998), (Doucet et al., 2001) e identificação de fácies litológicas (Ruanaidh e Fitzgerald, 1996; Salomão, 2006).

Uma grande revolução nas técnicas de filtragem foi o desenvolvimento dos algoritmos de amostragem estocástica denominados coletivamente de métodos de Monte Carlo para cadeia de Markov (MCMC). Esses métodos são algoritmos baseados em simulações que geram

poderosos métodos numéricos para o cálculo de verossimilhança, das distribuições *a posteriori* e das estimativas derivadas delas.

O desenvolvimento desses métodos teve início com o algoritmo determinístico baseado na maximização do valor esperado proposto por (Logothetis e Krishnamurthy, 1999). A partir de (Doucet et al., 2000), os métodos de amostragem estocásticas começaram a ser utilizados, culminando numa eficiente decomposição da distribuição *a posteriori* apresentada em (Doucet e Andrieu, 2001). Esta abordagem permitiu o desenvolvimento de um algoritmo *on line* desenvolvido em (Doucet et al., 2001).

Nosso intuito é desenvolver uma análise comparativa entre os algoritmos, levando em consideração o custo computacional e o erro associado ao cálculo das estimativas. Tanto o rastreamento de alvo móvel quanto a identificação de fácies litológicas foram utilizados como exemplos de aplicações.

Para ratificar a qualidade das estimativas obtidas, resolvemos utilizar o controle via horizonte retrocedente estudado na literatura, particularmente por (Bitmead et al., 1990) e (Kwon et al., 1983) para sistemas lineares determinísticos, e por (Mayne e Michalska, 1990) para sistemas não-lineares. Resultados iniciais aplicados em sistemas SLSM são mais recentes (do Val e Basar, 1999; Costa e do Val, 2000).

Também conhecido como controle de horizonte deslizante, ou ainda como *Model Predictive Control* (MPC) (vide, por exemplo (Camacho e Bordons, 1999; Mosca, 1995)), o controle por horizonte retrocedente tem-se mostrado como uma estratégia de sucesso, tanto no meio acadêmico como na indústria, com diversas aplicações em processos industriais.

O princípio de controle retrocedente nos permite assumir que o sistema tem um modelo preciso somente dentro do curto e médio prazos, ou seja, os possíveis valores dos parâmetros do modelo e a probabilidade de transição entre seus valores são conhecidos somente dentro destes prazos. Depois de certo período, entretanto, não há um modelo apropriado, acarretando sérias dificuldades para a abordagem do problema de longo prazo.

Para que o controle via horizonte retrocedente possa ser aplicado, as estimativas dos estados devem ser calculadas de maneira recursiva e o filtro de partículas desenvolvido por Doucet et al. (2001) foi utilizado. A aplicação do braço do robô modificado por (Vargas, 2004) e o exemplo 6.4 de (Ji e Chizeck, 1990b) foram considerados para comparar o desempenho e para verificar a estabilidade do conjunto filtro/controle.

1.3 Estrutura da Dissertação

No próximo capítulo, algumas definições e conceitos básicos serão apresentados, assim como mostraremos em termos formais o sistema SLSM com ruído e observação. Dividimos as técnicas de filtragem em dois grupos, desenvolvendo no Capítulo 3 as técnicas de filtragem em *batch*, e no capítulo 4, a filtragem *on line* representada pelo filtro de partículas de Monte Carlo.

No Capítulo 5, apresentaremos duas aplicações sujeitas a mudanças abruptas, sendo que as mesmas podem ser modeladas de acordo com os sistemas SLSM. Adotaremos como solução para estas aplicações os métodos desenvolvidos nos Capítulos 3 e 4.

No Capítulo 6, desenvolveremos a solução para problema de controle retrocedente associado ao problema de filtragem.

Finalmente, o Capítulo 7 é dedicado a conclusões, e uma breve discussão será feita sobre os possíveis trabalhos futuros.

Capítulo 2

Conceitos básicos

No capítulo anterior, introduzimos de maneira geral os sistemas SLSM e o problema de filtragem associado a estimação de estados, citando motivações para o estudo desta classe de problemas.

Neste capítulo, apresentamos alguns conceitos básicos que serão extensivamente utilizados ao longo da tese, tais como processos discretos de Markov, modelo de cadeia escondida e filtragem. Para facilitar o entendimento do sistema SLSM, fomos associando os conceitos um a um, encadeando-os de maneira didática com exemplos ilustrativos. Ao final do capítulo, caracterizaremos formalmente o sistema SLSM, apresentando o problema de filtragem associado.

2.1 Processos Discretos de Markov

O estado discreto θ é descrito num dado instante k por um elemento do conjunto $S = \{1, 2, \dots, s\}$. Em espaços regulares de tempo, θ passa por uma mudança de acordo com certas probabilidades, assumindo valores dentro deste conjunto. Em geral, a forma probabilística de θ_k é dada pela especificação de todos os estados que o antecederam. No caso especial dos processos discretos de Markov, apenas o estado predecessor é necessário para descrever probabilisticamente o estado atual, isto é,

$$p(\theta_k \mid \theta_{k-1}, \theta_{k-2}, \dots) = p(\theta_k \mid \theta_{k-1}) \quad (2.1)$$

em que θ_k representa o estado de Markov no instante k .

Considerando apenas os processos em que o lado direito de (2.1) é independente do tempo, denominados de cadeia de Markov, o conjunto de probabilidades de transição de estados é dado por

$$p_{ij} = p(\theta_{k+1} = j \mid \theta_k = i) \geq 0, \forall i, j \in S = \{1, 2, \dots, s\} \quad (2.2)$$

com $\sum_{j=1}^s p_{ij} = 1$ para cada $i \in S$. Por notação, temos que a matriz de transição de probabilidade é

$$\mathbb{P} = [p_{ij}; i, j \in S]. \quad (2.3)$$

Para completar o modelo deste processo, temos que definir o vetor de probabilidade inicial $\pi = \{\pi_i\}$, em que $\pi_i = p(\theta_1 = i)$ para $i \in S$, de tal forma que $\pi_i \geq 0$ para $\forall i \in S$ e $\sum_{i=1}^s \pi_i = 1$.

Como o vetor de observação deste processo é dado pelo estado de Markov $O_k = \theta_k$, o processo estocástico acima é chamado de modelo observável da cadeia de Markov. A partir desta seção, teremos θ_k para $\forall k \in \{1, 2, 3, \dots\}$ como notação para o estado de Markov.

Para entender melhor este processo discreto de Markov, considere um simples modelo de previsão meteorológica com três estados de Markov cujo o diagrama pode ser visto na figura 2.1.

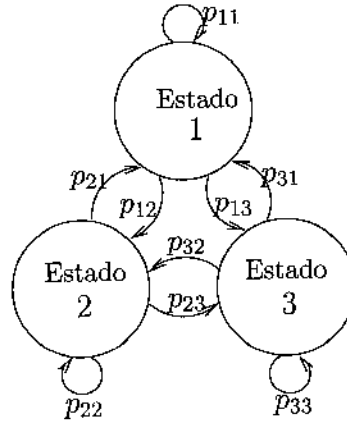


Figura 2.1: Diagrama do modelo de previsão com 3 estados de Markov.

Nós assumimos que uma vez por dia, a condição meteorológica é observada e dada por

Estado 1: Chuvoso;
 Estado 2: Nublado;
 Estado 3: Ensolarado.

Considerando o histórico da região em que se deseja realizar a previsão, temos que as transições de uma condição meteorológica para outra podem ser caracterizadas pela matriz

$$\mathbb{P} = \begin{bmatrix} 0,4 & 0,3 & 0,3 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}. \quad (2.4)$$

Como exemplo ilustrativo, vamos calcular a probabilidade de que as condições de tempo para os próximos 3 dias sejam “ensolarado-nublado-chuvoso”, dado que no dia de hoje ($k = 1$) a condição de tempo é ensolarado (estado 3).

Com o vetor de observações representado por $O_{1:4} = \{3, 3, 2, 1\}$, temos que esta probabilidade é dada por:

$$\begin{aligned} p(O_{2:4}|O_1) &= p(\theta_4 = 1, \theta_3 = 2, \theta_2 = 3 \mid \theta_1 = 3) \\ &= p(\theta_4 = 1 \mid \theta_3 = 2, \theta_2 = 3, \theta_1 = 3) \cdot p(\theta_3 = 2 \mid \theta_2 = 3, \theta_1 = 3) \cdot \\ &\quad \cdot p(\theta_2 = 3 \mid \theta_1 = 3). \end{aligned} \quad (2.5)$$

Como θ_k evolui de acordo com a cadeia de Markov com $\mathbb{P}$ descrito em (2.4), (2.5) pode ser descrita por:

$$\begin{aligned} p(O_{2:4}|O_1) &= p(\theta_4 = 1 \mid \theta_3 = 2) \cdot p(\theta_3 = 2 \mid \theta_2 = 3) \cdot p(\theta_2 = 3 \mid \theta_1 = 3) \\ &= p_{21} \cdot p_{32} \cdot p_{33} \\ &= 0,2 \cdot 0,1 \cdot 0,8 \\ &= 0,016. \end{aligned} \quad (2.6)$$

Dado que a condição de tempo no dia de hoje ($k = 1$) é ensolarado (estado 3), a probabilidade de que as condições de tempo para os próximos 3 dias sejam “ensolarado-nublado-chuvoso” é de 1,6%.

2.2 Modelo de Markov de Cadeia Escondida

Nesta seção, iremos desenvolver o modelo de Markov de cadeia escondida em que a informação sobre os estados de Markov é dada por uma certa função de probabilidade associada a eles. Este modelo é imerso em dois processos estocásticos, sendo o primeiro, os estados de Markov, os quais não são observados diretamente, e o segundo, uma sequência de observações obtidas de uma função probabilística do primeiro (processo subjacente). Portanto, os estados de Markov podem ser estimados a partir das observações produzidas pelo segundo processo.

Para compreender melhor o modelo de cadeia escondida vamos utilizar como exemplo, o modelo de bolas e urnas (Rabiner, 1989). Vamos assumir que existem s grandes urnas transparentes em uma sala e que dentro de cada uma delas podemos encontrar um grande número de bolas coloridas. Nós assumimos que existem m cores distintas de bolas e que a quantidade de bolas de cada cor que cada urna contém é conhecida. O processo físico para a obtenção das observações é dado por:

- 1) A primeira urna é sorteada de acordo com um dado processo aleatório;
- 2) Da urna escolhida, uma bola é sorteada aleatoriamente e sua cor é registrada como observação. Somente a cor é anotada, não se conhecendo qual foi a urna escolhida;

- 3) A bola sorteada é devolvida para a mesma urna da qual foi retirada;
- 4) Uma nova urna é selecionada de acordo com o processo de transição aleatório associado à urna atual, sem que se saiba qual foi a urna escolhida.

A partir deste ponto, os procedimentos de 2 até 4 devem ser repetidos até o número de observações desejadas. O processo gera uma sequência finita de cores, que nós podemos modelar como o vetor de observação do modelo de Markov de cadeia escondida. Fica claro no modelo de bolas e urnas que os estados de Markov são definidos pela sequência de urnas e a função probabilística, pela probabilidade de ocorrência das cores que depende da urna (estado de Markov) escolhida. Vale ressaltar que a escolha da urna é feita através da matriz de transição do modelo e que esta escolha depende unicamente da urna escolhida anteriormente.

Considere um exemplo com 3 urnas e 4 cores distintas (laranja, amarelo, roxo e verde). Nós assumimos que a probabilidade de escolha da primeira urna é dada pelo vetor $\pi = (0,2 \ 0,3 \ 0,5)'$ e que a probabilidade de ocorrer uma dada cor em uma dada urna é representada na tabela 2.1.

Tabela 2.1: Probabilidade de ocorrência das cores por urna.

	Urna 1	Urna 2	Urna 3
Probabilidade da cor Laranja (L)	0,1	0	0,5
Probabilidade da cor Amarela (A)	0,4	0,3	0
Probabilidade da cor Roxa (R)	0,3	0,6	0
Probabilidade da cor Verde (V)	0,2	0,1	0,5

Nós postulamos também que a matriz $\mathbb{P}$ de transição de probabilidades dos estados (urnas), que representa o processo estocástico de escolha da próxima urna baseado na urna atual, é conhecida e dada por

$$\mathbb{P} = \begin{bmatrix} 0,8 & 0,2 & 0 \\ 0,2 & 0,7 & 0,1 \\ 0,3 & 0,3 & 0,4 \end{bmatrix}. \quad (2.7)$$

Como exemplo ilustrativo, obtivemos uma sequência de cinco cores de acordo com o procedimento acima descrito, gerando o vetor de observação

$$O_{1:5} = \{V, L, A, R, V\}.$$

Como podemos, através deste vetor e dos parâmetros conhecidos do modelo, recuperar a informação de quais urnas foram retiradas a sequência de bolas sorteadas? A resposta para

esta pergunta será fornecida na seção 3.2 quando o *smoother* do modelo de cadeia escondida será apresentado.

Agora que o modelo de cadeia escondida ficou claro, vamos entender como este modelo é a base para a formulação do problema de filtragem de sistemas SLSM sem observação do estado de Markov.

2.3 Associando Filtragem ao Modelo

As técnicas de filtragem têm como objetivo estimar o vetor de estados de um sistema dinâmico a partir do vetor de observação. Nós adotamos a abordagem via representação no espaço de estados para modelar os sistemas dinâmicos a tempo discreto, devido a sua conveniência e praticidade.

Esta abordagem é baseada no vetor de estados do sistema, que contém toda a informação relevante requerida para descrever o sistema sobre investigação. Dividiremos este vetor em duas parcelas, a contínua e a discreta.

Consideremos que um sistema possua vários modos, dependendo do seu ponto de operação, e que além do mais, estes modos evoluam no tempo de acordo com uma cadeia de Markov. Neste caso, a parcela discreta do vetor de estados é determinada pelo estado de Markov, representando os modos do sistema. A parcela contínua deste vetor, por sua vez, representa os estados contínuos que evoluem baseados no modo atual de operação. Por exemplo, nos problemas de rastreamento esta parcela contém todas as características cinemáticas do alvo, alternativamente, nos problemas de econometria, pode estar relacionada ao fluxo monetário, a taxa de atratividade, inflação ou outros índices econômicos, que dependem dos modos de operação da economia de um dado país. O modelo de cadeia escondida poderá ser aplicado se não tivermos acesso direto à parcela discreta do vetor de estados.

O vetor de observação, por sua vez, é uma função da parcela contínua do vetor de estados com ou sem ruído de observação que depende do modo de operação do sistema. Normalmente, este vetor de observação é de menor dimensão se comparado à parcela contínua do vetor de estados. Com o intuito de simplificar a nomenclatura, a partir desta seção, denominaremos a parcela contínua do vetor de estados de *estados contínuos* e a discreta, de *estado de Markov*.

Para analisar e fazer inferências sobre o modo de operação do sistema definido pelo estado de Markov θ_k , dois modelos são necessários: o modelo do sistema, que descreve a evolução dos estados contínuos no instante k , e o modelo de medição, que relaciona o vetor de observação aos estados contínuos. Nós assumimos que os modelos estão disponíveis numa forma estocástica.

Genericamente, tanto o modelo do sistema quanto o de medição podem representar sistemas lineares e não lineares, além disso, ruído pode ser adicionado a ambos os modelos. No modelo do sistema, o ruído representa as imprecisões dadas pela identificação do sistema real quando comparada ao modelo teórico escolhido. No modelo de medição, o ruído representa quão distante de uma combinação dos estados contínuos está a medição. Os ruídos são modelados/representados por processos estocásticos que podem ser ou não Gaussianos. Nós fixamos nosso estudo em sistemas lineares imersos em ruído Gaussiano. Em (Ristic et al., 2004), informações mais detalhadas sobre filtragem de modelos não lineares podem ser encontradas.

2.4 Formulação do Problema

Sistemas lineares sujeito a saltos Markovianos são sistemas lineares cujos parâmetros evoluem no tempo de acordo com uma cadeia de Markov de estados finitos. Deixe $k \in \{1, 2, \dots\}$ representar o tempo discreto e θ_k , a parcela discreta do vetor de estados ou estado de Markov com um total de s estados possíveis, cujas probabilidades de transição são dadas por

$$p_{ij} = p(\theta_{k+1} = j \mid \theta_k = i) \geq 0, \forall i, j \in S = \{1, 2, \dots, s\}$$

com $\sum_{j=1}^s p_{ij} = 1$ para cada $i \in S$, em que a matriz de transição de probabilidade é dada por $\mathbb{P} = [p_{ij}; i, j \in S]$. Representamos também a distribuição de probabilidade inicial por $\pi = \{\pi_i\}$, em que $\pi_i = p(\theta_1 = i)$ para $i \in S$, de tal forma que $\pi_i \geq 0$ para $\forall i \in S$ e $\sum_{i=1}^s \pi_i = 1$.

Desta forma, o sistema linear sujeito a saltos markovianos, a tempo discreto, $\mathcal{G}$ pode ser descrito por

$$\mathcal{G} : \begin{cases} x_{k+1} = A(\theta_{k+1})x_k + B(\theta_{k+1})v_{k+1} + F(\theta_{k+1})u_{k+1}, \\ y_k = C(\theta_k)x_k + D(\theta_k)z_k + G(\theta_k)u_k, \end{cases}$$

sendo que $x_k \in \mathbb{R}^{n_x}$ é a parcela contínua do vetor de estados ou estado contínuo do sistema, $y_k \in \mathbb{R}^{n_y}$ é o vetor de observação, $u_k \in \mathbb{R}^{n_u}$ é uma entrada conhecida determinística, $v_k \in \mathbb{R}^{n_v}$ é uma sequência de ruído branco gaussiano com média nula cuja variância é identidade I_{n_v} e $z_k \in \mathbb{R}^{n_z}$ é uma sequência de ruído branco gaussiano com média nula cuja variância é identidade I_{n_z} .

A, B, C, D, F e G representam coleções de matrizes que são funções do estado da cadeia de Markov θ_k e evoluem de acordo com a realização da cadeia de Markov finita. Se $\theta_k = i$,

temos $A(\theta_k) = A_i$, $B(\theta_k) = B_i$, $C(\theta_k) = C_i$, $D(\theta_k) = D_i$, $F(\theta_k) = F_i$ e $G(\theta_k) = G_i$.

Assumimos que $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$ e consideramos que x_0 , v_k e z_k são mutuamente independentes para todo k . Por conveniência de notação, temos $y_{1:t} \triangleq \{y_1, \dots, y_t\}$, $x_{0:t} = \{x_0, x_1, \dots, x_t\}$ e $\theta_{1:t} = \{\theta_1, \dots, \theta_t\}$, representando o conjunto de vetores de observação, os estados contínuos e os estados de Markov, respectivamente.

2.5 Cenários

Para facilitar didaticamente o entendimento das técnicas de filtragem de sistemas SLSM, quatro grandes cenários baseados na observação ou não dos estados de Markov θ_k e dos estados contínuos x_k foram definidos. Estes cenários são:

Cenário 1 - Ambos os estados são observáveis: o estado contínuo x_k e o estado de Markov θ_k são observáveis a todo instante k . Neste cenário, as técnicas de filtragem não são necessárias, pois temos acesso direto a todos os estados do sistema.

Cenário 2 - Estado de Markov observável e estado contínuo não observável: o estado de Markov θ_k e o vetor de observação y_k são observáveis para todo instante k . Temos que recuperar a estimativa do estado contínuo x_k para todo instante k .

Cenário 3 - Estado contínuo observável e estado de Markov não observável: apenas o estado contínuo x_k é observável para todo instante k . Temos que recuperar a estimativa do estado de Markov θ_k para todo instante k .

Cenário 4 - Sem observação de ambos os estados: não temos acesso nem ao estado contínuo x_k e nem ao estado de Markov θ_k . Temos que recuperar as estimativas destes estados baseadas no vetor de observação y_k .

Nós dividimos as técnicas de filtragem em dois grandes sub-grupos dados pelos algoritmos em *batch* e pelos filtros recursivos. O primeiro grupo calcula as estimativas dos estados condicionadas a um conjunto fixo de observações e o segundo, por sua vez, considera os filtros recursivos que recuperam os estados através da sequência de observações obtida até o instante em que se deseja calcular a estimativa. O estudo e análise destes grupos podem ser vistos nos capítulos 3 e 4.

Capítulo 3

Filtragem em *batch*

A palavra *batch* em inglês significa batelada, conjunto de bens produzidos num dado intervalo de tempo com uma determinada especificação e configuração da máquina ou processo que o produziu. Com base nesta definição temos que filtragem em *batch* representa as técnicas de filtragem que geram estimativas dos estados não observáveis a partir de um conjunto de observações $O_{1:T} = \{O_1, O_2, \dots, O_T\}$. Estes algoritmos tendem a considerar que se temos acesso a duas observações, podemos obter estimativas mais precisas, se compararmos ao caso em que temos acesso apenas a uma observação.

Na situação em que o estado de Markov θ_k é conhecido para todo k , cenário 2, a filtragem é dada pela extração do estado contínuo x_k de um processo linear de Gauss-Markov. Sabe-se que a estimativa ótima da sequência de estados contínuos $x_{1:T}$ é obtida pelo Kalman *smoother* (Anderson e Moore, 1979) apresentado na seção 3.1.

Por outro lado, se o estado contínuo x_k é perfeitamente conhecido para todo k , cenário 3, a tarefa do cálculo das estimativas da sequência $\theta_{1:T}$ se reduz a um problema de extração do estado de Markov θ_k em ruído branco de observação. Neste caso, o vetor de observação é $O_k = x_k$ e as estimativas do estado de Markov $\theta_{1:T}$ podem ser obtidas pela adaptação do *smoother* do modelo de cadeia escondida (Rabiner, 1989) que será apresentado na seção 3.2.

Quando não se tem acesso aos estados, cenário 4, o único dado consistente é a sequência de observações $y_{1:T}$ e as estimativas dos estados de Markov $\theta_{1:T}$ e dos estados contínuos $x_{1:T}$ são obtidas, em princípio, a partir da distribuição conjunta a *posteriori* $p(\theta_{1:T}, x_{1:T} | y_{1:T})$.

Vamos entender melhor a dificuldade de encontrarmos as estimativas do estado contínuo x_k e do estado de Markov θ_k , tendo acesso apenas as observações imersas em ruído y_k . Inicialmente, temos acesso à função de distribuição de probabilidade do estado contínuo no instante $k = 0$, que é representada por uma gaussiana cuja média é $\hat{x}_0$ e a variância P_0 . Como não temos acesso ao estado de Markov no tempo $k = 1$, temos s possíveis sistemas lineares,

representando todos os estados possíveis da cadeia de Markov. Para sistemas lineares imersos em ruído branco e gaussiano com vetor de estados de entrada também gaussiano, temos que a observação de saída também é representada por uma função de distribuição de probabilidade gaussiana.

Com isso, ao final do instante $k = 1$ teremos s gaussianas, representando a observação de saída, uma para cada estado de Markov. Como não temos acesso ao verdadeiro estado de Markov, teremos s^k distribuições gaussianas no instante k . Infelizmente, podemos perceber que o custo computacional do cálculo exato destas estimativas cresce exponencialmente com o número de observações, o que acarreta um custo computacional proibitivo (Doucet et al., 2000). Por este motivo, algoritmos que aproximam a distribuição conjunta a *posteriori* $p(\theta_{1:T}, x_{1:T} | y_{1:T})$ e que possuem custo computacional linearmente crescente com o número de observações serão levados em consideração na seção 3.3.

3.1 Kalman Smoother

Se o estado de Markov θ_k é conhecido para todo instante k , cenário 2, o modelo linear com saltos markovianos se reduz a um modelo linear gaussiano e a filtragem, a um problema de extração do estado contínuo x_k do processo linear Gauss-Markov. É conhecido que as estimativas ótimas do estado contínuo $x_{1:T}$ são obtidas pelo Kalman *smoother* (Anderson e Moore, 1979). Como *Smoothers* são estimadores baseados em um conjunto de observações, esperamos obter estimativas mais precisas do que os filtros.

Normalmente os *smoothers* podem ser divididos em dois procedimentos: o procedimento em avanço, geralmente representado por um filtro, e o retrocedente, em que a informação obtida pelo procedimento em avanço retorna, agregando e adaptando as informações obtidas nos instantes anteriores. Para aplicar o Kalman *smoother*, supomos que $B_i B_i' > 0$ e $D_i D_i' > 0$ para $\forall i \in S$.

No Kalman *smoother*, o procedimento em avanço é um banco de filtros de Kalman, pois a cada instante k o modelo linear de espaço de estados definido por θ_k é conhecido. O procedimento em avanço pode ser descrito por:

- Deixe $m_{0|0} = \hat{x}_0$ e $P_{0|0} = P_0$, então compute as equações do filtro de Kalman para $k =$

$1, \dots, T$

$$m_{k|k-1} = A(\theta_k)m_{k-1|k-1} + F(\theta_k)u_k \quad (3.1)$$

$$P_{k|k-1} = A(\theta_k)P_{k-1|k-1}A'(\theta_k) + B(\theta_k)B'(\theta_k) \quad (3.2)$$

$$z_{k|k-1} = C(\theta_k)m_{k|k-1} + G(\theta_k)u_k \quad (3.3)$$

$$\tilde{y}_{k|k-1} = y_k - z_{k|k-1} \quad (3.4)$$

$$S_k = C(\theta_k)P_{k|k-1}C'(\theta_k) + D(\theta_k)D'(\theta_k) \quad (3.5)$$

$$m_{k|k} = m_{k|k-1} + P_{k|k-1}C'(\theta_k)S_k^{-1}\tilde{y}_{k|k-1} \quad (3.6)$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}C'(\theta_k)S_k^{-1}C(\theta_k)P_{k|k-1} \quad (3.7)$$

e armazene para $k = 1, \dots, T$: $m_{k|k-1} = E[x_k|y_{1:k-1}, \theta_{1:k}]$, $m_{k|k} = E[x_k|y_{1:k}, \theta_{1:k}]$, $P_{k|k-1} = E[(x_k - m_{k|k-1})(x_k - m_{k|k-1})'|y_{1:k-1}, \theta_{1:k}]$ e $P_{k|k} = E[(x_k - m_{k|k})(x_k - m_{k|k})'|y_{1:k}, \theta_{1:k}]$.

O procedimento retrocedente faz o caminho inverso, trazendo a informação de trás para frente. O procedimento retrocedente é dado por:

- Para $k = T, \dots, 1$, calcule

$$m_{k-1|T} = m_{k-1|k-1} + \mathcal{S}_{k-1}(m_{k|T} - m_{k|k-1}) \quad (3.8)$$

$$P_{k-1|T} = P_{k-1|k-1} + \mathcal{S}_{k-1}(P_{k|T} - P_{k|k-1})\mathcal{S}'_{k-1}, \quad (3.9)$$

sendo $\mathcal{S}_{k-1} = P_{k-1|k-1}A'(\theta_{k-1})P_{k|k-1}^{-1}$.

3.2 Smoother do Modelo de Cadeia Escondida

No caso em que apenas o estado contínuo x_k é observável, cenário 3, as estimativas dos estados de Markov $\theta_{1:T}$ podem ser obtidas através do *smoother* do modelo de cadeia escondida desenvolvido por (Rabiner, 1989). Sua versão clássica será detalhada na subseção 3.2.1 e sua adaptação para o sistema $\mathcal{G}$ na seção 3.2.2.

3.2.1 Versão Clássica

Primeiramente, temos a variável em avanço definida por

$$\alpha_k(i) = p(O_{1:k}, \theta_k = i), \quad (3.10)$$

que representa a probabilidade da sequência de observações $O_{1:k} = \{O_1, \dots, O_k\}$ e do estado de Markov θ_k ser igual a i no instante k . Gostaríamos de ressaltar que O_k é uma variável aleatória discreta.

Nós podemos calcular $\alpha_k(i)$ recursivamente de acordo com o procedimento:

1) Inicialização: Para $i = 1, \dots, s$, calcule

$$\alpha_1(i) = \pi_i p(O_1 | \theta_1 = i); \quad (3.11)$$

2) Indução: Para $j = 1, \dots, s$ e $1 \leq k \leq T$, calcule

$$\begin{aligned} \alpha_{k+1}(j) &= \left[\sum_{i=1}^s \alpha_k(i) p(\theta_{k+1} = j | \theta_k = i) \right] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= \left[\sum_{i=1}^s \alpha_k(i) p_{ij} \right] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j), \end{aligned} \quad (3.12)$$

sendo que O_k representa o vetor de observações no instante k .

Se tivermos o valor de $\alpha_k(i)$ para $\forall i \in S$ em um dado instante k , podemos calcular a probabilidade do estado de Markov θ_k ser igual a i condicionada à seqüência parcial de observações $O_{1:k} = \{O_1, \dots, O_k\}$, isto é,

$$p(\theta_k = i | O_{1:k}) = \frac{p(O_{1:k}, \theta_k = i)}{p(O_{1:k})} = \frac{p(O_{1:k}, \theta_k = i)}{\sum_{j=1}^s p(O_{1:k}, \theta_k = j)} = \frac{\alpha_k(i)}{\sum_{j=1}^s \alpha_k(j)}. \quad (3.13)$$

Segundo Rabiner (1989) é necessário utilizar técnicas de escalonamento para implementar os procedimentos do modelo de cadeia escondida porque para um k suficientemente grande, o range dinâmico computacional de $\alpha_k(i)$ irá exceder o range de precisão do processador utilizado. Considerando o fator de escala c_k e o coeficiente escalonado $\hat{\alpha}_k(i)$, temos que

$$\hat{\alpha}_k(i) = \frac{\alpha_k(i)}{c_k}. \quad (3.14)$$

O coeficiente escalonado para $k = 1$ é dado por

$$\hat{\alpha}_1(i) = \frac{\pi_i p(O_1 | \theta_1 = i)}{c_1} \quad (3.15)$$

com $c_1 = \sum_{\ell=1}^s \pi_\ell p(O_1 | \theta_1 = \ell)$.

Para $k > 1$, o coeficiente escalonado pode ser calculado por

$$\hat{\alpha}_k(i) = \frac{\left[\sum_{j=1}^s \hat{\alpha}_{k-1}(j) p_{ji} \right] p(O_k | O_{1:k-1}, \theta_k = i)}{d_k}. \quad (3.16)$$

com

$$d_k = \sum_{\ell=1}^s \left[\sum_{j=1}^s \hat{\alpha}_{k-1}(j) p_{j\ell} \right] p(O_k | O_{1:k-1}, \theta_k = \ell). \quad (3.17)$$

Ao reescrever (3.12) com (3.14), (3.16) e (3.17), temos

$$\begin{aligned} \alpha_{k+1}(j) &= \left[\sum_{i=1}^s \alpha_k(i) p_{ij} \right] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= \left[\sum_{i=1}^s c_k \hat{\alpha}_k(i) p_{ij} \right] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= c_k \left[\sum_{i=1}^s \hat{\alpha}_k(i) p_{ij} \right] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= c_k d_{k+1} \hat{\alpha}_{k+1}(j). \end{aligned} \quad (3.18)$$

De acordo com (3.18) podemos obter o fator de escala c_k para todo instante k recursivamente através de

$$c_k = c_{k-1} d_k. \quad (3.19)$$

De maneira análoga podemos definir a variável retrocedente

$$\beta_k(i) = p(O_{k+1:T} | \theta_k = i) \quad (3.20)$$

que representa a probabilidade da sequência parcial de observações a partir de $k+1$ até T , considerando que o estado de Markov assume o estado i no tempo k . De novo podemos sequencialmente calcular $\beta_k(i)$ através de

1) Inicialização: Para $1 \leq i \leq s$, considere

$$\beta_T(i) = \text{const}, \quad (3.21)$$

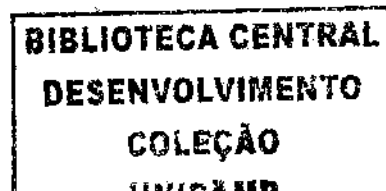
em que const representa uma constante arbitrária diferente de zero;

2) Indução: Para $1 \leq i \leq S$ e $k = T-1, \dots, 1$, calcule

$$\begin{aligned} \beta_k(i) &= \sum_{j=1}^s p(\theta_{k+1} = j | \theta_k = i) \beta_{k+1}(j) p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= \sum_{j=1}^s p_{ij} \beta_{k+1}(j) p(O_{k+1} | O_{1:k}, \theta_{k+1} = j). \end{aligned} \quad (3.22)$$

Considerando o fator de escala g_k e o coeficiente escalonado $\hat{\beta}_k(i)$, temos que

$$\hat{\beta}_k(i) = \frac{\beta_k(i)}{g_k}. \quad (3.23)$$



O escalonamento aplicado à variável retrocedente gera

$$\hat{\beta}_T(i) = \frac{\beta_T(i)}{\sum_{\ell=1}^s \beta_T(\ell)} = \frac{\text{const}}{\sum_{\ell=1}^s \text{const}} = \frac{1}{s}, \quad (3.24)$$

$$\hat{\beta}_k(i) = \frac{\sum_{j=1}^s p_{ij} \beta_{k+1}(j) p(O_{k+1} | O_{1:k}, \theta_{k+1} = j)}{e_k} \quad (3.25)$$

com

$$e_k = \sum_{\ell=1}^s \sum_{j=1}^s p_{\ell j} \beta_{k+1}(j) p(O_{k+1} | O_{1:k}, \theta_{k+1} = j). \quad (3.26)$$

Seguindo o mesmo procedimento descrito no escalonamento da variável em avanço e considerando o fator de escala g_T no instante T igual a s , temos que para $k < T$

$$g_k = g_{k+1} e_k. \quad (3.27)$$

Para finalizar o *smoother* de cadeia escondida precisamos definir a variável

$$\begin{aligned} \gamma_k(i) &= \frac{p(\theta_k = i | O_{1:T})}{p(O_{1:T})} \\ &= \frac{p(O_{1:T}, \theta_k = i)}{p(O_{1:T})} \\ &= \frac{p(O_{1:k}, \theta_k = i) p(O_{k+1:T} | \theta_k = i)}{p(O_{1:T})} \\ &= \frac{p(O_{1:k}, \theta_k = i) p(O_{k+1:T} | \theta_k = i)}{\sum_{\ell=1}^s p(O_{1:k}, \theta_k = \ell) p(O_{k+1:T} | \theta_k = \ell)} \end{aligned} \quad (3.28)$$

que representa a probabilidade da cadeia de Markov θ_k estar no estado i no instante k condicionada à sequência de observações $O_{1:T}$. A equação (3.28) pode ser expressa em termo das variáveis α_k e β_k

$$\gamma_k(i) = \frac{\alpha_k(i) \beta_k(i)}{\sum_{\ell=1}^s \alpha_k(\ell) \beta_k(\ell)}. \quad (3.29)$$

Ainda, podemos representar (3.28) através dos coeficientes escalonados

$$\gamma_k(i) = \frac{c_k \hat{\alpha}_k(i) g_k \hat{\beta}_k(i)}{\sum_{\ell=1}^s c_k \hat{\alpha}_k(\ell) g_k \hat{\beta}_k(\ell)} = \frac{\hat{\alpha}_k(i) \hat{\beta}_k(i)}{\sum_{\ell=1}^s \hat{\alpha}_k(\ell) \hat{\beta}_k(\ell)}. \quad (3.30)$$

Em cada instante k nós temos pelo *smoother* da cadeia escondida um vetor de probabilidade, associando uma probabilidade de ocorrência a cada estado de Markov. Na tentativa

de obter uma estimativa da sequência de estados de Markov $\bar{\theta}_{1:T}$, alguns critérios podem ser utilizados. Um critério seria escolher os estados de Markov que são individualmente mais prováveis, o que leva à

$$\bar{\theta}_k = \underset{1 \leq i \leq s}{\operatorname{argmax}} [\gamma_k(i)], \quad 1 \leq k \leq T. \quad (3.31)$$

Embora este critério maximize o número esperado de estados individualmente, alguns problemas podem ocorrer na sequência estimada de estados de Markov. Por exemplo, quando o modelo de cadeia escondida tiver transições de estados com probabilidade nula, a sequência obtida pode não representar uma sequência válida. Isso ocorre porque a solução deste critério determina o estado mais provável a cada instante, sem se preocupar com a probabilidade de transição entre os estados.

Outro critério muito utilizado é encontrar a melhor sequência estimada de estados de Markov (único caminho) que maximiza $p(\theta_{1:T}|O_{1:T})$, equivalente a maximizar $p(\theta_{1:T}, O_{1:T})$, pois

$$p(\theta_{1:T}|O_{1:T}) = \frac{p(\theta_{1:T}, O_{1:T})}{P(O_{1:T})} = \frac{p(\theta_{1:T}, O_{1:T})}{\sum_{m=\{1, \dots, 1\}}^{\{s, \dots, s\}} p(\theta_{1:T} = m, O_{1:T})} \quad (3.32)$$

em que $\sum_{m=\{1, \dots, 1\}}^{\{s, \dots, s\}} p(\theta_{1:T} = m, O_{1:T})$ representa uma constante de normalização dada pelo somatório de todas as possíveis combinações da sequência $\theta_{1:T}$ para $\theta_k \in \{1, \dots, s\}$.

O algoritmo de Viterbi, baseado nos métodos de programação dinâmica, é a técnica formal utilizada para encontrar este caminho. Para isso é necessário definir a quantidade

$$\delta_k(i) = \max_{\theta_1, \theta_2, \dots, \theta_T} p(\theta_{1:k-1}, \theta_k = i, O_{1:k}), \quad (3.33)$$

que representa a maior probabilidade ao longo de um único caminho, no instante k , que leva em conta as k primeiras observações e termina no estado de Markov i . Como o processo é Markoviano, temos recursivamente que

$$\begin{aligned} \delta_{k+1}(j) &= [\max_i \delta_k(i) p(\theta_{k+1} = j | \theta_k = i)] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j) \\ &= [\max_i \delta_k(i) p_{ij}] p(O_{k+1} | O_{1:k}, \theta_{k+1} = j). \end{aligned} \quad (3.34)$$

Para realmente rastrear a sequência de estados, nós precisamos guardar os argumentos que maximizam a função para cada k e j . Isso é feito através do vetor $\psi_k(j)$. O procedimento completo do algoritmo de Viterbi para achar a melhor sequência de estados é dado por:

1) Inicialização: Para $1 \leq i \leq s$, calcule

$$\delta_1(i) = \pi_i p(O_1 | \theta_1 = i) \quad (3.35)$$

$$\psi_1(i) = 0; \quad (3.36)$$

2) Indução: Para $1 \leq j \leq s$ e $2 \leq k \leq T$, calcule

$$\delta_k(j) = \max_{1 \leq i \leq s} [\delta_{k-1}(i) p_{ij} p(O_k | O_{1:k-1}, \theta_k = j)] \quad (3.37)$$

$$\psi_k(j) = \underset{1 \leq i \leq s}{\operatorname{argmax}} [\delta_{k-1}(i) p_{ij}]; \quad (3.38)$$

3) Término:

$$\bar{\theta}_T = \underset{1 \leq i \leq s}{\operatorname{argmax}} [\delta_T(i)]; \quad (3.39)$$

4) Caminho retrocedente (seqüência de estados): Para $k = T-1, T-2, \dots, 1$

$$\bar{\theta}_k = \psi_{k+1}(\bar{\theta}_{k+1}). \quad (3.40)$$

Para ilustrar o funcionamento do *smoother* de cadeia escondida, vamos retornar ao modelo de bolas e urnas do final da seção 2.2. Como a probabilidade de ocorrer uma cor depende unicamente da urna sorteada, temos que

$$p(O_k | O_{1:k-1}, \theta_k = i) = p(O_k | \theta_k = i), \quad (3.41)$$

que corresponde à probabilidade de uma bola ter uma determinada cor dado que esta bola foi sorteada da i -ésima urna.

Podemos, a partir do vetor de observações $O_{1:5} = \{V, L, A, R, V\}$, calcular os coeficientes escalonados das variáveis em avanço e retrocedente

$$\begin{aligned} \hat{\alpha}_1 &= \begin{bmatrix} 0,13 \\ 0,09 \\ 0,78 \end{bmatrix}, \hat{\alpha}_2 = \begin{bmatrix} 0,18 \\ 0 \\ 0,82 \end{bmatrix}, \hat{\alpha}_3 = \begin{bmatrix} 0,65 \\ 0,35 \\ 0 \end{bmatrix}, \hat{\alpha}_4 = \begin{bmatrix} 0,44 \\ 0,56 \\ 0 \end{bmatrix}, \hat{\alpha}_5 = \begin{bmatrix} 0,55 \\ 0,28 \\ 0,17 \end{bmatrix} \\ \hat{\beta}_1 &= \begin{bmatrix} 0,29 \\ 0,18 \\ 0,53 \end{bmatrix}, \hat{\beta}_2 = \begin{bmatrix} 0,41 \\ 0,35 \\ 0,24 \end{bmatrix}, \hat{\beta}_3 = \begin{bmatrix} 0,34 \\ 0,42 \\ 0,24 \end{bmatrix}, \hat{\beta}_4 = \begin{bmatrix} 0,29 \\ 0,25 \\ 0,46 \end{bmatrix}, \hat{\beta}_5 = \begin{bmatrix} 0,33 \\ 0,33 \\ 0,33 \end{bmatrix}. \end{aligned}$$

Com estas duas variáveis é possível calcular $\gamma_k(i)$ que representa a probabilidade no instante k do estado de Markov θ_k ser igual a i condicionada ao vetor observação $O_{1:T}$

$$\gamma_1 = \begin{bmatrix} 0,08 \\ 0,03 \\ 0,89 \end{bmatrix}, \gamma_2 = \begin{bmatrix} 0,27 \\ 0 \\ 0,73 \end{bmatrix}, \gamma_3 = \begin{bmatrix} 0,60 \\ 0,40 \\ 0 \end{bmatrix}, \gamma_4 = \begin{bmatrix} 0,47 \\ 0,53 \\ 0 \end{bmatrix}, \gamma_5 = \begin{bmatrix} 0,55 \\ 0,28 \\ 0,17 \end{bmatrix}.$$

Na tentativa de recuperar o estado de Markov, podemos utilizar o critério que maximize o número de estados esperados individualmente ou o algoritmo de Viterbi. Aplicando estes procedimentos ao modelo das bolas e urnas, temos que no primeiro caso apenas 4 urnas foram identificadas corretamente, enquanto no segundo, todas as urnas foram identificadas corretamente. Devemos ter em mente que ao definirmos um dado critério, não necessariamente estaremos recuperando corretamente os estados de Markov.

Se a aplicação exigir um procedimento *on line*, o procedimento em avanço do *smoother* do modelo de cadeia escondida α_k poderá ser utilizado para calcular $p(\theta_k = i | O_{1:k})$, gerando uma estimativa do estado de Markov no instante k . Este procedimento *on line* será utilizado na seção 6.2.2, recebendo o nome de filtro do modelo de cadeia escondida.

3.2.2 Smoother Adaptado

Para entendermos o porquê da adaptação do *smoother* do modelo de cadeia escondida, alguns conceitos básicos de probabilidade serão apresentados como função de distribuição acumulada, função densidade de probabilidade (pdf) e verossimilhança.

A função de distribuição acumulada $F_{\mathbf{a}_k}$ de uma variável aleatória contínua $\mathbf{a}_k$ no instante k pode ser representada por

$$F_{\mathbf{a}_k}(a) = p(\mathbf{a}_k \leq a), \quad (3.42)$$

definida para todo a de $-\infty$ até ∞ . Reescrevendo (3.42) em função da densidade de probabilidade f_k obtemos

$$F_{\mathbf{a}_k}(a) = \int_{-\infty}^a f_k(y) dy, \quad (3.43)$$

para todo $a \in (-\infty, \infty)$.

Podemos dizer que a probabilidade de ocorrência de um resultado particular a em uma distribuição acumulada de uma variável contínua $\mathbf{a}_k$ é nula, pois

$$p(\mathbf{a}_k = a) = p(a \leq \mathbf{a}_k \leq a) = F_{\mathbf{a}_k}(a) - F_{\mathbf{a}_k}(a) = 0. \quad (3.44)$$

Tendo como base o modelo $\mathcal{G}$ aplicado ao cenário 3 com $O_k = x_k$, temos

$$p(O_{k+1} | O_{1:k}, \theta_{k+1} = i) = p(x_{k+1} | x_{1:k}, \theta_{k+1} = i) = 0. \quad (3.45)$$

Infelizmente, o resultado de (3.45) inviabiliza o uso da versão clássica do *smoother* do modelo de cadeia escondida apresentado na seção 3.2.1, entretanto, uma versão adaptada que substitui $p(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ por uma função de mérito $b(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ foi proposta. Esta função associa um custo à ocorrência da observação O_{k+1} condicionada à

seqüência de observações $O_{1:k}$ e à $\theta_k = i$ para $\forall i \in S$, o que gera uma medida comparativa entre os diversos estados da cadeia de Markov.

Como perdemos a noção de probabilidade, cabe ao pesquisador escolher a função de mérito que mais lhe apraz. Na literatura, o uso da verossimilhança como função de mérito é muito recorrente, pois seu custo é dado pela função de densidade de probabilidade f_k aplicada ao resultado particular O_k , gerando pesos

$$w_k^{(i)} = f_k(O_k | O_{1:k-1}, \theta_k = i), \forall i \in S, \quad (3.46)$$

que normalizados levam à verossimilhança

$$\bar{p}(O_k | O_{1:k-1}, \theta_k = i) = \frac{w_k^{(i)}}{\sum_{j=1}^s w_k^{(j)}}, \forall i \in S. \quad (3.47)$$

A verossimilhança baseada no modelo $\mathcal{G}$ com $O_k = x_k$ pode ser calculada através de (3.46) e (3.47), considerando

$$f_k(x_k | x_{1:k-1}, \theta_k = i) = \frac{\exp^{-\frac{1}{2}(x_k - A_i x_{k-1} - F_i u_k)'(B_i B_i')^{-1}(x_k - A_i x_{k-1} - F_i u_k)}}{\sqrt{(2\pi)^{n_x} |B_i B_i'|}}, \quad (3.48)$$

com $B_i B_i'$ definida positiva para $\forall i \in S$.

O *smoother* adaptado do modelo de cadeia escondida é dado pela substituição de $p(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ por $\bar{p}(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ nas equações de (3.10) até (3.40) da versão clássica, calculada por (3.46), (3.47) e (3.48). Gostaríamos de lembrar que o *smoother* adaptado gera uma estimativa da distribuição de probabilidade do estado de Markov θ_k para todo instante k .

3.3 Filtragem sem Observação de Estados

Quando não se tem acesso aos estados de Markov $\theta_{1:T}$ e aos estados contínuos $x_{1:T}$, algoritmos que aproximam a distribuição de probabilidade a *posteriori* $p(\theta_{1:T}, x_{1:T} | y_{1:T})$ são utilizados, pois o custo da estimativa exata é proibitivo como mostrado intuitivamente no início deste capítulo. Os primeiros algoritmos utilizados para o cálculo das estimativas dos estados tinham como estrutura padrão a combinação de dois estimadores, um relacionado ao estado de Markov, estado finito, e outro ao estado contínuo como pode ser visto na figura 3.1.

Com a evolução do estado da arte, foi se dando mais importância aos estimadores dos estados de Markov, pois percebeu-se que o estado contínuo poderia ser recuperado facilmente

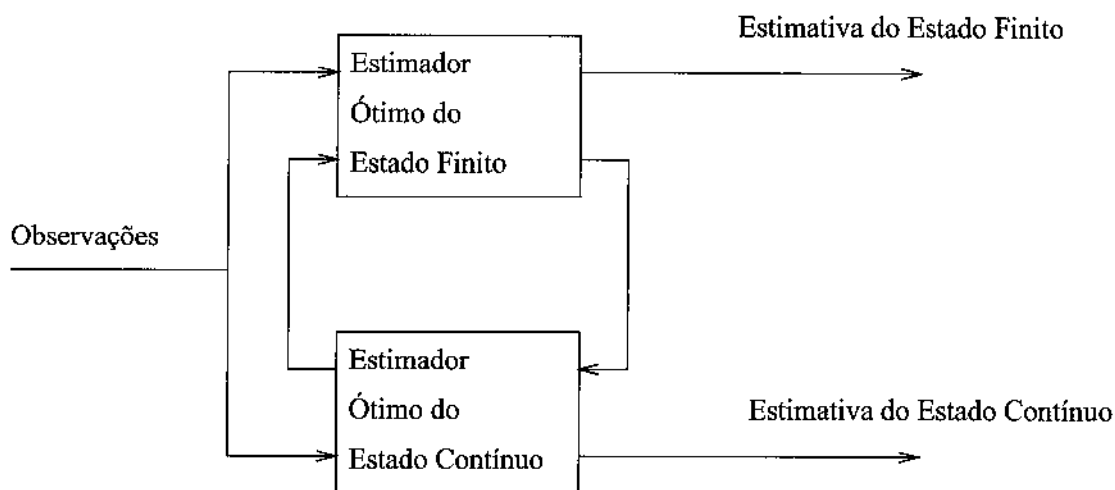


Figura 3.1: Estrutura geral com combinação de dois estimadores.

a partir do estado de Markov estimado. A busca pela melhor estimativa da distribuição de probabilidade *a posteriori* $p(\theta_{1:T}, x_{1:T} | y_{1:T})$ permeará o desenvolvimento desta seção, pois mostraremos como a partir dos algoritmos básicos quase inoperantes chegou-se a uma eficiente decomposição da distribuição de probabilidade marginal de $p(\theta_{1:T} | y_{1:T})$.

Um conceito por nós utilizado e que facilitará o entendimento das próximas seções é a distinção entre algoritmos determinísticos e estocásticos. O primeiro tipo de algoritmo leva em consideração a melhor sequência de estados no sentido de maximizar uma dada distribuição de probabilidade *a posteriori*. O segundo, por sua vez, simula aleatoriamente sequências prováveis de estados, utilizando a melhor distribuição encontrada até aquele instante.

Saber quando usar um ou outro algoritmo é uma tarefa muito complexa, pois além da formulação ter sempre uma lógica interna, eles são apresentados na literatura sempre do ponto de vista de suas vantagens. Nosso intuito é desenvolver uma análise comparativa entre os algoritmos, tendo como referência o custo computacional e o erro associado. Destacamos o fato de que esta abordagem é bastante recente e a literatura atual não apresenta nenhum estudo de natureza comparativa até onde pudemos investigar.

Em cada uma das seções seguintes, vamos apresentar o conteúdo e a estrutura dos algoritmos utilizados na filtragem sem observação de estados. O estudo dos algoritmos é iniciado pelos algoritmos determinísticos baseados na maximização do valor esperado proposto em (Logothetis e Krishnamurthy, 1999) e desenvolvidos na seção 3.3.1. A passagem dos algoritmos determinísticos para os estocásticos fica a cargo do método de Monte Carlo para cadeia de Markov, no qual algoritmos de amostragem estocástica relativos ao *Data Augmentation* (Doucet et al., 2000) são utilizados para a obtenção das estimativas do estado contínuo $x_{1:T}$. Estes algoritmos estão descritos nas seções 3.3.2 e 3.3.3. Outra evolução são os algoritmos

iterativos como o erro mínimo quadrático (MMSE) e o *maximum a posteriori* no sentido marginal (MMAP) (Doucet e Andrieu, 2001), que serão apresentados nas seções 3.3.4 e 3.3.5, finalizando o estudo dos algoritmos em *batch*

3.3.1 Algoritmo de Maximização do Valor Esperado

O algoritmo de Maximização do Valor Esperado (EM) é um algoritmo iterativo determinístico utilizado para o cômputo da sequência estimada via *maximum a posteriori* (MAP) dos estados através da maximização da função de densidade conjunta $p(x_{1:T}, \theta_{1:T} | y_{1:T})$. As estimativas de $\theta_{1:T}$ e $x_{1:T}$ são dadas por

$$(\hat{\theta}_{1:T}, \hat{x}_{1:T}) = \underset{x_{1:T}, \theta_{1:T}}{\operatorname{argmax}} p(x_{1:T}, \theta_{1:T} | y_{1:T}). \quad (3.49)$$

A estrutura do algoritmo EM é caracterizada pela combinação de dois estimadores de estado: o *smoother* para o modelo de Markov de cadeia escondida, apresentado na seção 3.2, e o Kalman *smoother*, apresentado na seção 3.1. O primeiro *smoother* é utilizado para obter as estimativas no sentido MAP do estado de Markov e o segundo, do estado contínuo.

Para maximizar a função de densidade conjunta *a posteriori* de $x_{1:T}$ e $\theta_{1:T}$ dado conjunto de observação $y_{1:T}$, um método iterativo chamado método de coordenada descendente foi utilizado. Dadas as estimativas iniciais dos estados $x_{1:T}^{(0)}$ e $\theta_{1:T}^{(0)}$, cada iteração consiste no seguinte passo duplo:

- Passo 1: Com $y_{1:T}$ e $\theta_{1:T}^{(\ell)}$ obtido da iteração anterior, obtemos $x_{1:T}^{(\ell+1)}$, maximizando $p(x_{1:T}, \theta_{1:T}^{(\ell)} | y_{1:T})$ em função de $x_{1:T}$

$$x_{1:T}^{(\ell+1)} = \underset{x_{1:T}}{\operatorname{argmax}} p(x_{1:T}, \theta_{1:T}^{(\ell)} | y_{1:T}). \quad (3.50)$$

$$(3.51)$$

Este passo é implementado através do Kalman *smoother* de intervalo fixo apresentado na seção 3.1.

- Passo 2: Com $y_{1:T}$ e $x_{1:T}^{(\ell+1)}$ de (3.50) nós obtemos $\theta_{1:T}^{(\ell+1)}$ através de

$$\theta_{1:T}^{(\ell+1)} = \underset{\theta_{1:T}}{\operatorname{argmax}} p(x_{1:T}^{(\ell+1)}, \theta_{1:T} | y_{1:T}). \quad (3.52)$$

Este passo é implementado através da adaptação do algoritmo de Viterbi com a substituição de $p(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ por $\bar{p}(O_{k+1} | O_{1:k}, \theta_{k+1} = i)$ nas equações de (3.32) até (3.40) da versão clássica.

Supondo $B_i B_i' > 0$ e $D_i D_i' > 0$, a verossimilhança $\bar{p}$ baseada no modelo $\mathcal{G}$ com $O_k = (y_k, x_k^{(\ell+1)})$ pode ser calculada através de (3.46) e (3.47), considerando

$$f(y_k, x_k^{(\ell+1)} | x_{1:k-1}^{(\ell+1)}, \theta_k) = f(y_k | x_k^{(\ell+1)}, \theta_k) f(x_k^{(\ell+1)} | x_{1:k-1}^{(\ell+1)}, \theta_k), \quad (3.53)$$

lembrando que

$$f(y_k | x_k^{(\ell+1)}, \theta_k = i) = \frac{\exp^{-\frac{1}{2}(y_k - C_i x_k^{(\ell+1)} - G_i u_k)'(D_i D_i')^{-1}(y_k - C_i x_k^{(\ell+1)} - G_i u_k)}}{\sqrt{(2\pi)^{n_y} |D_i D_i'|}} \quad (3.54)$$

$$f(x_k^{(\ell+1)} | x_{1:k-1}^{(\ell+1)}, \theta_k = i) = \frac{\exp^{-\frac{1}{2}(x_k^{(\ell+1)} - A_i x_{1:k-1}^{(\ell+1)} - F_i u_k)'(B_i B_i')^{-1}(x_k^{(\ell+1)} - A_i x_{1:k-1}^{(\ell+1)} - F_i u_k)}}{\sqrt{(2\pi)^{n_x} |B_i B_i'|}}. \quad (3.55)$$

Em (Logothetis e Krishnamurthy, 1999), $\theta_{1:T}^{(\ell+1)}$ é calculado através do logaritmo neperiano de $\bar{p}(y_{1:T}, x_{1:T}^{(\ell+1)}, \theta_{1:T})$, dado por

$$\theta_{1:T}^{(\ell+1)} = \underset{\theta_{1:T}}{\operatorname{argmax}} \ln \left[\bar{p}(y_{1:T}, x_{1:T}^{(\ell+1)}, \theta_{1:T}) \right], \quad (3.56)$$

que pode ser substituído por

$$\theta_{1:T}^{(\ell+1)} = \underset{\theta_{1:T}}{\operatorname{argmax}} \left[\sum_{k=1}^T \ln[\bar{p}(y_k, x_k^{(\ell+1)} | x_{1:k-1}^{(\ell+1)}, \theta_k)] + \sum_{k=2}^T \ln(p_{\theta_{k-1}, \theta_k}) + \ln(\pi_{\theta_1}) \right]. \quad (3.57)$$

Daí, o estado estimado de Markov $\theta_{1:T}^{(\ell+1)}$ na $\ell + 1$ -ésima iteração, dado $x_{1:T}^{\ell+1}$ é calculado por

1. Inicialização: Para $i = 1, \dots, s$, calcule

$$\delta_1(i) = \ln(\pi_i) + \ln[\bar{p}(y_1, x_1^{(\ell+1)} | \theta_1 = i)] \quad (3.58)$$

$$\psi_1(i) = 0; \quad (3.59)$$

2. Recursividade: Para $2 \leq k \leq T$ e $1 \leq j \leq N$, calcule

$$\delta_k(j) = \ln[\bar{p}(y_k, x_k^{(\ell+1)} | \theta_k = j)] + \max_i [\delta_{k-1}(i) + \ln(p_{ij})] \quad (3.60)$$

$$\psi_k(j) = \underset{i}{\operatorname{argmax}} [\delta_{k-1}(i) + \ln(p_{ij})]; \quad (3.61)$$

3. Término:

$$\theta_T^{(\ell+1)} = \underset{i}{\operatorname{argmax}} [\delta_T(i)]; \quad (3.62)$$

4. Caminho único (seqüência de estados) *backtracking*: Para $k = T-1, T-2, \dots, 1$, temos

$$\theta_k^{(\ell+1)} = \psi_{k+1}(\theta_{k+1}^{(\ell+1)}). \quad (3.63)$$

Logothetis e Krishnamurthy (1999) garantem que o EM aumenta a verossimilhança conjunta dos estados $x_{1:T}$ e $\theta_{1:T}$ na $\ell + 1$ -ésima iteração pela escolha de $x_{1:T}^{(\ell+1)}$ e $\theta_{1:T}^{(\ell+1)}$, mas não conseguem provar que se $\ell \rightarrow \infty$, então $(\theta_{1:T}^{(\ell+1)}, x_{1:T}^{(\ell+1)})$ tenderá para $(\hat{\theta}_{1:T}, \hat{x}_{1:T})$ dado por (3.49). Além disso, afirmam apenas que o algoritmo tende a um ponto estacionário dado por um máximo local ou ponto de inflexão.

Após implementarmos este algoritmo, tendo como base os exemplos numéricos apresentados na seção 5, chegamos à conclusão que o algoritmo EM não consegue transpor a sua própria condição inicial. Isso ocorre porque quando o primeiro passo é executado, o Kalman *smoother* calcula a estimativa do estado contínuo $x_{1:T}^{(\ell+1)}$ baseado no conjunto de estados de Markov $\theta_{1:T}^{(\ell)}$ e ao realizar o segundo passo, percebemos que o *smoother* de cadeia escondida baseado em $x_{1:T}^{(\ell+1)}$ restaura a mesma seqüência de estados de Markov $\theta_{1:T}^{(\ell+1)} = \theta_{1:T}^{(\ell)}$.

A idéia de combinar dois estimadores é a principal contribuição deste algoritmo na evolução dos algoritmos de filtragem sem observação de estados, mas baseados no resultado obtido pelas simulações, não podemos corroborar a eficácia deste algoritmo. Por isso, um estudo futuro mais detalhado, abrangendo um número maior de exemplos, se faz necessário. Este resultado não será apresentado, devido a sua ineficácia.

3.3.2 Data Augmentation

O *Data Augmentation* (DA) desenvolvido por (Doucet et al., 2000) inicia a série de algoritmos estocásticos baseados nos métodos de Monte Carlo que permite, a partir de simulações aleatórias, gerar uma estimativa do estado contínuo $E[x_{1:T}|y_{1:T}]$. Se nós pudéssemos obter N i.i.d. amostras aleatórias $\{\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)}; \ell = 1, \dots, N\}$ de $p(\theta_{1:k}, x_{1:k}|y_{1:k})$, então o estimador empírico desta distribuição seria dado por

$$\bar{p}_N(\theta_{1:k}, x_{1:k}|y_{1:k}) = \frac{1}{N} \sum_{\ell=1}^N \delta_{(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)})}(\theta_{1:k}, x_{1:k}) \quad (3.64)$$

e, como corolário a estimativa de $p(\theta_k, x_k|y_{1:k})$ seria dada por

$$\bar{p}_N(\theta_k, x_k|y_{1:k}) = \frac{1}{N} \sum_{\ell=1}^N \delta_{(\theta_k^{(\ell)}, x_k^{(\ell)})}(\theta_k, x_k) \quad (3.65)$$

com

$$\int_c^d f(x) \delta_A(x) dx = \begin{cases} f(A), & c \leq A \leq d \\ 0, & A < c \text{ e } A > d \end{cases}.$$

Como o valor esperado de uma função $\varphi_{k|k}(\theta_k, x_k)$ é dado por

$$E(\varphi_{k|k}) = \int \varphi_{k|k}(\theta_k, x_k) p(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_k dx_k, \quad (3.66)$$

temos que uma estimativa deste valor esperado através do estimador empírico (3.65) pode ser dada por

$$\begin{aligned} \bar{E}_N(\varphi_{k|k}) &= \int \varphi_{k|k}(\theta_k, x_k) \bar{p}_N(\theta_k, x_k | y_{1:k}) d\theta_k dx_k \\ &= \frac{1}{N} \sum_{\ell=1}^N \varphi_{k|k}(\theta_k^{(\ell)}, x_k^{(\ell)}). \end{aligned} \quad (3.67)$$

Esta estimativa é não tendenciosa e, pela lei forte dos grandes números (LFGN), $\bar{E}_N(\varphi_{k|k})$ converge quase certamente para $E(\varphi_{k|k})$ quando $N \rightarrow +\infty$. Se $\sigma_{\varphi_{k|k}}^2 \triangleq \text{var}_{p(\theta_k, x_k | y_{1:k})}(\varphi_{k|k}(\theta_k, x_k)) < +\infty$, então o teorema do limite central (TLC) garante que

$$\sqrt{N} [\bar{E}_N(\varphi_{k|k}) - E(\varphi_{k|k})] \xrightarrow[N \rightarrow +\infty]{} \mathcal{N}(0, \sigma_{\varphi_{k|k}}^2) \quad (3.68)$$

em que “ $\Rightarrow$ ” denota convergência na distribuição. Além da facilidade de estimar $E(\varphi_{k|k})$ para qualquer $\varphi_{k|k}$, os métodos de Monte Carlo possuem taxa de convergência independente de k . Como infelizmente, a obtenção de tais amostras não é direta, um esquema de amostragem utilizando métodos de Monte Carlo para cadeia de Markov (MCMC) foi adotado. A principal idéia dos métodos MCMC é simular uma cadeia de Markov ergódica cuja distribuição invariante é a própria distribuição de interesse.

Com esta poderosa ferramenta estocástica, capaz de amostrar distribuições multivariáveis complexas, dois problemas de estimação Bayesianos podem ser aplicados ao DA:

1. Cálculo da estimativa da média condicional no sentido médio quadrático de $x_{1:T}$ representada por

$$\hat{x}_{1:T} = E[x_{1:T} | y_{1:T}]. \quad (3.69)$$

2. Cálculo da estimativa MAP de $\theta_{1:T}$ através da maximização da distribuição posterior marginal

$$\hat{\theta}_{1:T} = \underset{\theta_{1:T}}{\operatorname{argmax}} p(\theta_{1:T} | y_{1:T}). \quad (3.70)$$

Tendo, como base, os dois problemas de estimação Bayesianos considerados, dois algoritmos baseados em métodos de amostragem estocástica da distribuição $p(\theta_{1:T}, x_{1:T} | y_{1:T})$ com convergência global e custo por iteração linear em relação ao número de observações são estudados. O primeiro algoritmo, apresentado nesta seção, é o *Data Augmentation* (DA) que

calcula a média condicional do estado contínuo x_k , e o segundo, que será detalhado na seção 3.3.3, é uma versão do DA com *Simulated Annealing* (SA) que calcula a estimativa no sentido MAP do estado de Markov com amostragem a partir do algoritmo de Metropolis.

Como explicado na seção 3.2.2, ao invés da probabilidade p , utilizaremos a verossimilhança $\bar{p}$ quando necessário. O algoritmo *Data Augmentation* pode ser resumido no procedimento descrito a seguir.

1. Inicialização: Seleção aleatória $(\theta_{1:T}^{(0)}, x_{1:T}^{(0)})$;
2. Iteração: Dado $(\theta_{1:T}^{(\ell-1)}, x_{1:T}^{(\ell-1)})$, calcule $(\theta_{1:T}^{(\ell)}, x_{1:T}^{(\ell)})$ para ℓ -ésima iteração ($\ell = 1, 2, \dots$) segundo:

Simule $x_{1:T}^{(\ell)}$ a partir de

$$x_{1:T}^{(\ell)} \sim \bar{p}\left(x_{1:T}|y_{1:T}, \theta_{1:T}^{(\ell-1)}\right); \quad (3.71)$$

Simule $\theta_{1:T}^{(\ell)}$ a partir de

$$\theta_{1:T}^{(\ell)} \sim p\left(\theta_{1:T}|y_{1:T}, x_{1:T}^{(\ell)}\right); \quad (3.72)$$

3. Estimação do Estado: Cálculo da estimativa da média condicional, usando o estimador empírico (3.65) é dado por

$$\bar{x}_{1:T}(N) \triangleq \frac{1}{N} \sum_{\ell=0}^{N-1} x_{1:T}^{(\ell)}. \quad (3.73)$$

A amostragem representada, no segundo passo, pode ser dividida em duas partes, a filtragem em avanço (*Forward Filtering*) e a amostragem retrocedente (*Backward Sampling*). Para facilitar o entendimento de ambas, o desenvolvimento da amostragem a partir das distribuições condicionais de $\bar{p}(x_{1:T}|y_{1:T}, \theta_{1:T}^{(\ell-1)})$ e $p(\theta_{1:T}|y_{1:T}, x_{1:T}^{(\ell)})$ será explicitado para os dois casos.

Em (3.71), $\bar{p}(x_{1:T}|y_{1:T}, \theta_{1:T}^{(\ell-1)})$ pode ser decomposta por

$$\bar{p}(x_{1:T}|y_{1:T}, \theta_{1:T}^{(\ell-1)}) = \bar{p}(x_T|y_{1:T}, \theta_{1:T}^{(\ell-1)}) \prod_{k=0}^{T-1} \bar{p}(x_k|y_{1:k}, \theta_{1:T}^{(\ell-1)}, x_{k+1:T}). \quad (3.74)$$

Esta decomposição associada a distribuição de probabilidade gaussiana de x_{k+1} , $\bar{p}(x_k|y_{1:k}, \theta_{1:T}^{(\ell-1)}, x_{k+1:T})$, sugere o seguinte algoritmo para amostrar $\bar{p}(x_{1:T}|y_{1:T}, \theta_{1:T}^{(\ell-1)})$:

1. Filtragem em avanço: Filtro de Kalman apresentado na seção 3.1;
2. Amostragem reversa: Para $k = T, \dots, 0$, amostre $x_k^{(\ell)}$ de $\mathcal{N}(m_{k|T}^*, P_{k|T}^*)$, em que $m_{T|T}^* = m_{T|T}$, $P_{T|T}^* = P_{T|T}$, e para $0 \leq k < T$

$$m_{k|T}^* = m_{k|k} + P_{k|k} A'(\theta_{k+1}) P_{k+1|k}^{-1} (x_{k+1}^{(\ell)} - m_{k+1|k}) \quad (3.75)$$

$$P_{k|T}^* = P_{k|k} - P_{k|k} A'(\theta_{k+1}) P_{k+1|k}^{-1} A(\theta_{k+1}) P_{k|k}. \quad (3.76)$$

Em (3.72), temos a seguinte decomposição para a distribuição $p(\theta_{1:T} | y_{1:T}, x_{1:T}^{(\ell)})$

$$p(\theta_{1:T} | y_{1:T}, x_{1:T}^{(\ell)}) = p(\theta_T | y_{1:T}, x_{1:T}^{(\ell)}) \prod_{k=1}^{T-1} p(\theta_k | y_{1:k}, x_{1:k}^{(\ell)}, \theta_{k+1:T}). \quad (3.77)$$

Esta decomposição associada a distribuição discreta θ_{k+1} , $p(\theta_k | y_{1:k}, x_{1:k}^{(\ell)}, \theta_{k+1:T})$, sugere o seguinte algoritmo para amostragem de $p(\theta_{1:T} | y_{1:T}, x_{1:T}^{(\ell)})$:

1. Filtragem em avanço: Para $k = 2, \dots, T$ e $i = 1, \dots, s$, calcule

$$p(\theta_k = i | y_{1:k-1}, x_{1:k-1}^{(\ell)}) = \sum_{j=1}^s p_{ji} p(\theta_{k-1} = j | y_{1:k-1}, x_{1:k-1}^{(\ell)}) \quad (3.78)$$

$$p(\theta_k = i | y_{1:k}, x_{1:k}^{(\ell)}) = \frac{\bar{p}(y_k, x_k^{(\ell)} | y_{k-1}, x_{k-1}^{(\ell)}, \theta_k = i) p(\theta_k = i | y_{1:k-1}, x_{1:k-1}^{(\ell)})}{\sum_{j=1}^s \bar{p}(y_k, x_k^{(\ell)} | y_{k-1}, x_{k-1}^{(\ell)}, \theta_k = j) p(\theta_k = j | y_{1:k-1}, x_{1:k-1}^{(\ell)})} \quad (3.79)$$

e guarde $p(\theta_k = i | y_{1:k-1}, x_{1:k-1}^{(\ell)})$ e $p(\theta_k = i | y_{1:k}, x_{1:k}^{(\ell)})$ para $k = 1, \dots, T$ e $i = 1, \dots, s$.

2. Amostragem reversa: Amostre $\theta_T^{(\ell)}$ de $p(\theta_T | y_{1:T}, x_{1:T}^{(\ell)})$ e $\theta_k^{(\ell)}$ de $p(\theta_k | y_{1:k}, x_{1:k}^{(\ell)}, \theta_{k+1:T})$ para $k = T-1, \dots, 1$ com

$$p(\theta_k = i | y_{1:k}, x_{1:k}^{(\ell)}, \theta_{k+1:T}) = \frac{p(\theta_{k+1} | \theta_k = i) p(\theta_k = i | y_{1:k}, x_{1:k}^{(\ell)})}{\sum_{j=1}^s p(\theta_{k+1} | \theta_k = j) p(\theta_k = j | y_{1:k}, x_{1:k}^{(\ell)})} \quad (3.80)$$

para $\forall i \in S$.

Normalmente, as primeiras iterações não devem ser utilizadas no cômputo das estimativas do estado contínuo $\bar{x}_{1:T}$, pois tendem a estar correlacionadas à condição inicial proposta, podendo gerar estimativas tendenciosas. A este conjunto de iterações desconsideradas é dado o nome de tempo de queima.

Como não sabemos a partir de qual iteração ℓ , a sequência de estados contínuos $x_{1:T}^{(\ell)}$ deixa de estar correlacionada à condição inicial $x_{1:T}^{(0)}$, assim como não conseguimos determinar a partir de quantas iterações N as estimativas $\bar{x}_{1:T}(N)$ se aproximam $\hat{x}_{1:T}(N)$ dado por (3.69), dois parâmetros arbitrários, o tempo de queima e o número total de iterações, são utilizados no DA. Não encontramos nenhuma metodologia que possa nos auxiliar na escolha destes parâmetros.

O *Data Augmentation* é o primeiro algoritmo da vertente estocástica que utiliza a seleção aleatória. Com a evolução dos algoritmos o sorteio dos estados de Markov baseado na distribuição discreta de probabilidade passa a ser a chave primordial da filtragem de sistemas lineares com saltos. Isso difere do algoritmo de maximização do valor esperado, que deterministicamente escolhe o estado de Markov mais provável. Uma outra melhora detectada está relacionada à decomposição desenvolvida para o cálculo da distribuição de probabilidade do estado de Markov. Aliando o sorteio do estado de Markov a uma decomposição mais refinada da distribuição de probabilidade, temos um ferramental cada vez mais desenvolvido na busca de uma melhor estimativa dos estados.

3.3.3 Algoritmo Metropolis-Hastings/*Data Augmentation* com *Simulated Annealing*

Este algoritmo é utilizado na obtenção de uma sequência, no sentido MAP, de estimativas dos estados $\theta_{1:T}$. O *Data Augmentation* com *Simulated Annealing* (DASA) é uma versão não homogênea do algoritmo DA que depende de uma regra de resfriamento determinística $\{T_r(\ell); \ell \in \mathbb{N}\}$, de tal modo que

$$T_r(\ell + 1) \leq T_r(\ell) \text{ e } \lim_{\ell \rightarrow +\infty} T_r(\ell) = 0. \quad (3.81)$$

O algoritmo Metropolis-Hastings/*Data Augmentation* com *Simulated Annealing* (Doucet et al., 2000) pode ser resumido no procedimento descrito a seguir:

1. Inicialização: Seleção aleatória $(\theta_{1:T}^{(0)}, x_{1:T}^{(0)})$ com $\ell = 1$.
2. Iteração: Dado $(\theta_{1:T}^{(\ell-1)}, x_{1:T}^{(\ell-1)})$, compute $(\theta_{1:T}^{(\ell)}, x_{1:T}^{(\ell)})$ para ℓ -ésima iteração segundo:

Simule $x_{1:T}^{(\ell)}$ a partir de

$$x_{1:T}^{(\ell)} \sim p\left(x_{1:T} | y_{1:T}, \theta_{1:T}^{(\ell-1)}\right). \quad (3.82)$$

Simule o candidato $\theta_{1:T}^c$ a partir de

$$\theta_{1:T}^c \sim p\left(\theta_{1:T}|y_{1:T}, x_{1:T}^{(\ell)}\right). \quad (3.83)$$

3. Cálculo da taxa de aceitação $\alpha(\theta_{1:T}^{(\ell-1)}, \theta_{1:T}^c)$:

$$\alpha_\ell(\theta_{1:T}^{(\ell-1)}, \theta_{1:T}^c) = \min \left\{ \left[\frac{p(\theta_{1:T}^c|y_{1:T})}{p(\theta_{1:T}^{(\ell-1)}|y_{1:T})} \right]^{1/T_r(\ell)-1}, 1 \right\} \quad (3.84)$$

4. Simule uma variável aleatória u em $[0,1]$, $u \sim \mathcal{U}(0,1)$.

5. Se $u < \alpha(\theta_{1:T}^{(\ell-1)}, \theta_{1:T}^c)$ então deixe $\theta_{1:T}^{(\ell)} = \theta_{1:T}^c$, caso contrário deixe $\theta_{1:T}^{(\ell)} = \theta_{1:T}^{(\ell-1)}$.

Para o cálculo da taxa de aceitação (3.84), nós precisamos calcular $p(\theta_{1:T}|y_{1:T})$ sobre uma constante de normalização. Nós temos que a distribuição $p(\theta_{1:T}|y_{1:T}) \propto p(\theta_{1:T})\bar{p}(y_{1:T}|\theta_{1:T})$. O primeiro termo é a distribuição conhecida *a priori* da realização da cadeia de Markov,

$$p(\theta_{1:T}) = \pi(\theta_1)p(\theta_2|\theta_1) \cdots p(\theta_T|\theta_{T-1}), \quad (3.85)$$

e o segundo, é a verossimilhança calculada através da sequência ponderada de inovações dadas pelo filtro de Kalman. A inovação ponderada é dada por

$$\bar{p}(y_k|y_{1:k-1}, \theta_{1:k-1}) = \sum_{m=1}^s \bar{p}(y_k|y_{1:k-1}, \theta_{1:k-1}, \theta_k = m)p(\theta_k = m|\theta_{k-1}) \quad (3.86)$$

em que

- $\bar{p}(y_k|y_{1:k-1}, \theta_{1:k-1}, \theta_k = m)$ é uma distribuição normal $\mathcal{N}(\tilde{y}_{k|k-1}(\theta_{1:k-1}, \theta_k = m), S_k(\theta_{1:k-1}, \theta_k = m))$ com $\tilde{y}_{k|k-1}(\theta_{1:k-1}, \theta_k = m)$ e $S_k(\theta_{1:k-1}, \theta_k = m)$ representando, respectivamente, a inovação e a covariância predita de um passo a frente da observação y_k condicionada à $\theta_{1:k}$ com $\theta_k = m$. Essas informações podem ser obtidas através do filtro de Kalman apresentado na seção 3.1;
- $p(\theta_k = m|\theta_{k-1})$ é dado pela matriz $\mathbb{P}$.

Duas inovações foram implementadas no *Data Augmentation*, a primeira diz respeito à velocidade de convergência do algoritmo, que está associada ao *Simulated Annealing*. Esta técnica permite que inicialmente o algoritmo tenha uma maior liberdade de escolha e que com o passar do tempo, esta liberdade vai sendo restringida pela taxa de resfriamento. A outra inovação está relacionada à utilização de uma seleção, feita pelo algoritmo de Metropolis,

que tem como principal objetivo aceitar ou rejeitar uma nova sequência aleatória. Se a nova sequência sorteada possuir uma probabilidade muito menor do que a atual, esta sequência é rejeitada, caso contrário ela passa por um processo de seleção aleatória, podendo ainda ser rejeitada. Se for aceita, ela passa a ser a sequência atual. Devido ao SA, fica cada vez mais difícil aceitar uma nova sequência e esta sequência só será aceita se ela possuir um probabilidade de ocorrência muito maior do que a anteriormente aceita. Para um melhor entendimento sobre o método de amostragem de Metropolis vide apêndice A.

3.3.4 Algoritmo do Mínimo Erro Quadrático Médio

O algoritmo do Mínimo Erro Quadrático utiliza os métodos de Monte Carlo (Doucet e Andrieu, 2001) para amostrar a distribuição marginal $p(\theta_{1:T}|y_{1:T})$, ao contrário dos algoritmos anteriores que se baseiam na amostragem da distribuição de probabilidade conjunta $p(\theta_{1:T}, x_{1:T}|y_{1:T})$. Aliando a facilidade de recuperar o estado contínuo x_k a partir do estado de Markov θ_k ao conhecimento da evolução do estado de Markov representado pela probabilidade inicial π e pela matriz de transição $\mathbb{P}$, os algoritmos tenderam a utilizar o estado de Markov em detrimento do estado contínuo.

O algoritmo para obter as estimativas do Mínimo Erro Quadrático Médio (MMSE) dos estados $\theta_{1:T}$ e $x_{1:T}$ pode ser descrito por:

1. Inicialização: Sorteie aleatoriamente $\theta_{1:T}^{(0)} \in \mathbb{R}$.
2. ℓ -ésima iteração, $\ell \geq 1$
 - para $k = 1, \dots, T$, amostre $\theta_k^{(\ell)} \sim p(\theta_k|y_{1:T}, \theta_{-k}^{(\ell)})$, onde $\theta_{-k}^{(\ell)} \triangleq (\theta_1^{(\ell)}, \dots, \theta_{k-1}^{(\ell)}, \theta_{k+1}^{(\ell-1)}, \dots, \theta_T^{(\ell-1)})$.

Após N iterações, a estimativa MMSE baseada no estimador (3.65) pode ser calculada

$$\bar{x}_{1:T}(N) \triangleq \frac{1}{N} \sum_{\ell=0}^{N-1} x_{1:T}^{(\ell)}. \quad (3.87)$$

Outra grande evolução deste método é o aprimoramento da decomposição da distribuição $p(\theta_k|y_{1:T}, \theta_{-k}^{(\ell)})$, que leva em consideração os estados de Markov $\theta_1^{(\ell)}, \dots, \theta_{k-1}^{(\ell)}$ da iteração ℓ e os estados de Markov $\theta_{k+1}^{(\ell-1)}, \dots, \theta_T^{(\ell-1)}$ da iteração $\ell - 1$.

A seguinte decomposição de $\bar{p}(y_{1:T}|\theta_{1:T})$ permite o cálculo eficiente da distribuição $p(\theta_k|y_{1:T}, \theta_{-k})$ para $k = 1, \dots, T$

$$\bar{p}(y_{1:T}|\theta_{1:T}) = \bar{p}(y_{1:k-1}|\theta_{1:k-1})\bar{p}(y_k|y_{1:k-1}, \theta_{1:k}) \int \bar{p}(y_{k+1:T}|\theta_{k+1:T}, x_k)p(x_k|y_{1:k}, \theta_{1:k}) \quad (3.88)$$

em que

$$\bar{p}(y_{k:T}|\theta_{1:T}, x_{k-1}) = \int \bar{p}(y_{k+1:T}|\theta_{k+1:T}, x_k) \bar{p}(y_k, x_k|\theta_k, x_{k-1}) dx_k. \quad (3.89)$$

Os dois primeiros termos do lado direito de (3.88) podem ser calculados através da utilização da recursão em avanço baseada no filtro de Kalman. Já o terceiro termo pode ser calculado, utilizando uma recursão em retrocesso desenvolvida abaixo.

1. Inicialização

$$P_{T|T}'^{-1}(\theta_T) = C'(\theta_T) [D(\theta_T) D'(\theta_T)]^{-1} C(\theta_T); \quad (3.90)$$

$$P_{T|T}'^{-1} m'_{T|T}(\theta_T) = C'(\theta_T) [D(\theta_T) D'(\theta_T)]^{-1} [y_T - G(\theta_T) u_T]. \quad (3.91)$$

2. Recursão retrocedente: Para $k = T - 1, \dots, 1$,

$$\Delta_{k+1} = \left[I_{n_v} + B'(\theta_{k+1}) P_{k+1|k+1}'^{-1}(\theta_{k+1:T}) B(\theta_{k+1}) \right]^{-1}; \quad (3.92)$$

$$\begin{aligned} P_{k|k+1}'^{-1}(\theta_{k+1:T}) &= A'(\theta_{k+1}) P_{k+1|k+1}'^{-1}(\theta_{k+1:T}) [I_{n_x} - B(\theta_{k+1}) \Delta_{k+1}(\theta_{k+1:T}) \\ &\quad \times B'(\theta_{k+1}) P_{k+1|k+1}'^{-1}(\theta_{k+1:T})] A(\theta_{k+1}); \end{aligned} \quad (3.93)$$

$$\begin{aligned} P_{k|k+1}'^{-1}(\theta_{k+1:T}) m'_{k|k+1}(\theta_{k+1:T}) &= A'(\theta_{k+1}) [I_{n_x} - P_{k+1|k+1}'^{-1}(\theta_{k+1:T}) B(\theta_{k+1}) \Delta_{k+1}(\theta_{k+1:T}) \\ &\quad \times B'(\theta_{k+1})] P_{k+1|k+1}'^{-1}(\theta_{k+1:T}) [m'_{k+1|k+1}(\theta_{k+1:T}) \\ &\quad - F(\theta_{k+1}) u_{k+1}]; \end{aligned} \quad (3.94)$$

$$P_{k|k}'^{-1}(\theta_{1:T}) = P_{k|k+1}'^{-1}(\theta_{k+1:T}) + C'(\theta_k) [D(\theta_k) D'(\theta_k)]^{-1} C(\theta_k); \quad (3.95)$$

$$\begin{aligned} P_{k|k}'^{-1} m'_{k|k}(\theta_{1:T}) &= P_{k|k+1}'^{-1}(\theta_{k+1:T}) m'_{k|k+1}(\theta_{k+1:T}) C'(\theta_k) [D(\theta_k) D'(\theta_k)]^{-1} \\ &\quad \times [y_k - G(\theta_k) u_k]. \end{aligned} \quad (3.96)$$

Combinando a recursão retrocedente com (3.88), temos que a expressão de $p(\theta_k|y_{1:T}, \theta_{-k})$ para $P_{k|k}(\theta_{1:k}) = 0_{n_x \times n_x}$ e $k = 2, \dots, T - 1$, é dada por

$$\begin{aligned} p(\theta_k|y_{1:T}, \theta_{-k}) &\propto p_{\theta_{k-1}, \theta_k} p_{\theta_k, \theta_{k+1}} \mathcal{N}(\tilde{y}_{k|k-1}(\theta_{1:k}), S_k(\theta_{1:k})) \exp \left(-\frac{1}{2} m'_{k|k}(\theta_{1:k}) \right. \\ &\quad \times P_{k|k+1}'^{-1}(\theta_{k+1:T}) m_{k|k}(\theta_{1:k}) - 2 m'_{k|k}(\theta_{1:k}) P_{k|k+1}'^{-1}(\theta_{k+1:T}) \\ &\quad \left. m_{k|k+1}(\theta_{k+1:T}) \right). \end{aligned} \quad (3.97)$$

Se $P_{k|k}(\theta_{1:k}) \neq 0_{n_x \times n_x}$, então existe uma matriz $\tilde{\Pi}_{k|k}(\theta_{1:k})$ diagonal $n_k \times n_k$, $1 \leq n_k \leq n_x$, com os autovalores diferentes de zero da matriz $P_{k|k}(\theta_{1:k})$ e uma matriz $\tilde{Q}_{k|k}(\theta_{1:k})$ tal que $P_{k|k}(\theta_{1:k}) = \tilde{Q}_{k|k}(\theta_{1:k})\tilde{\Pi}_{k|k}(\theta_{1:k})\tilde{Q}_{k|k}'(\theta_{1:k})$. Daí, temos

$$\begin{aligned}
 p(\theta_k|y_{1:T}, \theta_{-k}) &\propto p_{\theta_{k-1}, \theta_k} p_{\theta_k, \theta_{k+1}} \mathcal{N}\left(\tilde{y}_{k|k-1}(\theta_{1:k}), S_k(\theta_{1:k})\right) |\tilde{\Pi}_{k|k}(\theta_{1:k})\tilde{Q}_{k|k}'(\theta_{1:k}) \\
 &\quad \times P_{k|k+1}'^{-1}(\theta_{k+1:T})\tilde{Q}_{k|k}(\theta_{1:k}) + I_{n_k}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}m_{k|k}'(\theta_{1:k})\right. \\
 &\quad \times P_{k|k+1}'^{-1}(\theta_{k+1:T})m_{k|k}(\theta_{1:k}) - 2m_{k|k}'(\theta_{1:k})P_{k|k+1}'^{-1}(\theta_{k+1:T}) \\
 &\quad \times m_{k|k+1}'(\theta_{k+1:T}) - [m_{k|k+1}'(\theta_{k+1:T}) - m_{k|k}'(\theta_{1:k})]' \\
 &\quad \left. \times P_{k|k+1}'^{-1}(\theta_{k+1:T})R_{k|k}(\theta_{1:k})[m_{k|k+1}'(\theta_{k+1:T}) - m_{k|k}'(\theta_{1:k})]\right)
 \end{aligned} \tag{3.98}$$

em que

$$R_{k|k}(\theta_{1:k}) = \tilde{Q}_{k|k}(\theta_{1:k}) \left[\tilde{\Pi}_{k|k}^{-1}(\theta_{1:k}) + \tilde{Q}_{k|k}'(\theta_{1:k})P_{k|k+1}'^{-1}(\theta_{k+1:T})\tilde{Q}_{k|k}(\theta_{1:k}) \right]^{-1} \tilde{Q}_{k|k}'(\theta_{1:k}). \tag{3.99}$$

Os valores $m_{k|k-1}(\theta_{1:k})$, $P_{k|k-1}(\theta_{1:k})$, $m_{k|k}(\theta_{1:k})$, $P_{k|k}(\theta_{1:k})$, $\tilde{y}(\theta_{1:k})$ e $S_k(\theta_{1:k})$ são respectivamente, predição e a covariância de um passo a frente de x_k , estimativa e covariância filtrada de x_k , a inovação no instante k e a covariância da inovação. Essas quantidades são dadas pelo filtro de Kalman aplicado ao sistema $\mathcal{G}$, apresentado na seção 3.1.

Para finalizar, amostrar $p(\theta_k|y_{1:T}, \theta_{-T})$ requer primeiro o cálculo da informação retrocedente do filtro e segundo, o cálculo de $p(\theta_k|y_{1:T}, \theta_{-T})$, que combina esta informação às quantidades obtidas pelo filtro de Kalman. Finalmente, $p(\theta_k|y_{1:T}, \theta_{-T})$ é amostrada e os valores atualizados $m_{k|k}(\theta_{1:k-1}^{(k)})$ e $P_{k|k}(\theta_{1:k-1}^{(k)}, \theta_k^{(k)})$ são armazenados.

Encontramos neste algoritmo uma decomposição eficiente de $p(\theta_k|y_{1:T}, \theta_{-k}^{(k)})$ que leva em consideração tanto os estados de Markov anteriores quanto os posteriores em relação ao instante em que o sorteio é realizado. Podemos perceber que o estado contínuo passa a ser uma consequência do sorteio do estado de Markov. Além do mais, esta decomposição não exige que a matriz de covariância seja estritamente positiva e nem que a matriz de transição do modelo de espaço de estados seja regular.

3.3.5 Estimativa de $\theta_{1:T}$ no Sentido MAP Marginal

O algoritmo determinístico para o cálculo da estimativa de $\theta_{1:T}$ no sentido *Maximum a Posteriori* Marginal (MMAP) (Doucet e Andrieu, 2001) utiliza um método de coordenada ascendente para maximizar iterativamente e sucessivamente $p(\theta_k|y_{1:T}, \theta_{-k})$ sobre θ_k para

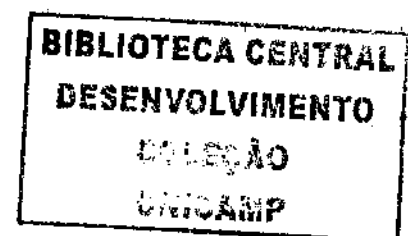
$k = 1, \dots, T$. Um algoritmo estocástico não homogêneo, utilizando o *Simulated Annealing* (SA) também é proposto como uma versão estocástica do algoritmo determinístico. Sua não homogeneidade da cadeia de Markov está relacionada a dependência do kernel de transição na iteração ℓ com uma taxa de resfriamento $\{\gamma_\ell; \ell \in \mathbb{N}\}$, tal que $\gamma_{\ell+1} \geq \gamma_\ell$, $\lim_{\ell \rightarrow +\infty} \gamma_\ell = +\infty$ e $\gamma_0 > 0$. O algoritmo determinístico e o estocástico para estimar MMAP da sequência $\theta_{1:T}$ seguem abaixo:

1. Inicialização. Sorteio aleatório de $\theta_{1:T}^{(0)} \in R$.
2. Iteração ℓ , $\ell \geq 1$
 - Algoritmo Determinístico (MMAP1): Para $k = 1, \dots, T$,

$$\theta_k^{(\ell)} = \underset{1, \dots, s}{\operatorname{argmax}} p(\theta_k | y_{1:T}, \theta_{-k}^{(\ell)}).$$
 - Algoritmo Estocástico (MMAP2): Para $k = 1, \dots, T$,
 amostre $\theta_k^{(\ell)} \sim \bar{p}^{\gamma_\ell}(\theta_k | y_{1:T}, \theta_{-k}^{(\ell)})$.

Lembrando que $\bar{p}^{\gamma_\ell}(\theta_k | y_{1:T}, \theta_{-k}^{(\ell)}) \propto [p(\theta_k | y_{1:T}, \theta_{-k}^{(\ell)})]^{\gamma_\ell}$.

Tanto o algoritmo estocástico quanto o determinístico utilizam a decomposição eficiente apresentada na seção anterior e o método de Monte Carlo para Cadeia de Markov. O algoritmo determinístico, ao invés de sorteio, escolhe o estado que possui a maior probabilidade de ocorrência e o estocástico, por sua vez, utiliza-se do *Simulated Annealing*.



Capítulo 4

Filtragem Recursiva

Em muitos problemas, uma estimativa dos estados do sistema é necessária a cada instante de tempo em que uma nova observação é incorporada, sendo conveniente o uso de um filtro recursivo. Nesta abordagem, nem todo o conjunto de informações precisa ser armazenado e, além do mais, não é necessário reprocessar a informação existente quando uma nova observação estiver disponível. No filtro recursivo toda a informação recebida é processada sequencialmente.

O filtro recursivo consiste essencialmente de dois estágios: predição e atualização. O passo de predição utiliza o modelo do sistema para prever a função de distribuição de probabilidade dos estados a frente entre uma medição e outra. Como os estados contínuos estão sujeitos a perturbações desconhecidas, modeladas como ruído aleatório, a predição geralmente translada, alarga ou deforma a verdadeira função de distribuição de probabilidade. A operação de atualização utiliza-se da última medição para modificar e corrigir a função distribuição de probabilidade. Isso pode ser obtido de maneira precisa através do teorema de Bayes, que é o mecanismo de atualização do conhecimento sobre os estados baseado na nova informação adquirida.

Atualmente, um filtro recursivo muito em voga é o filtro de partículas utilizado tanto em abordagens lineares quanto não lineares. O filtro de partículas de Monte Carlo desenvolvido por Doucet et al. (2001) será apresentado em toda a sua plenitude nas próximas seções para que fique claro o seu funcionamento e simplicidade.

4.1 Filtro de Partículas de Monte Carlo

Filtro de partículas de Monte Carlo é um algoritmo recursivo de filtragem baseado em simulações de Monte Carlo para calcular a distribuição condicional estimada dos estados x_k e θ_k dado o histórico de observações $y_{1:k}$. Este filtro de partículas pode ser visto aproxima-

mente como uma grade aleatória adaptativa, onde as partículas (valores da grade) evoluem aleatoriamente no tempo de acordo com uma regra de simulação. Cada partícula possui um peso associado à verossimilhança que é atualizado de acordo com a regra de Bayes a cada observação. Os procedimentos para o cálculo dos pesos e as regras de simulação serão explicitados nas próximas seções.

O filtro de partículas combina com propriedade a amostragem sequencial por importância e os métodos de Monte Carlo para cadeia de Markov (MCMC) associados a um esquema seletivo que replica/descarta partículas a cada instante de tempo. Além de sua simplicidade, o que torna este filtro ainda mais atrativo é que a taxa com que o erro das estimativas converge para zero independente da dimensão do estado de Markov.

Como apresentado na seção 3.3.2, se nós pudéssemos amostrar $N = N_p$ i.i.d. amostras aleatórias $\{\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)}; \ell = 1, \dots, N_p\}$, denominadas aqui de partículas, de acordo com $p(\theta_{1:k}, x_{1:k}|y_{1:k})$, uma estimativa para o valor esperado da função $\varphi_{k|k}(\theta_k, x_k)$ seria dada por

$$\begin{aligned} \bar{E}_{N_p}(\varphi_{k|k}) &= \int \varphi_{k|k}(\theta_k, x_k) \bar{p}_{N_p}(\theta_{1:k}, x_{1:k}|y_{1:k}) d\theta_{1:k} dx_{1:k} \\ &= \frac{1}{N_p} \sum_{\ell=1}^{N_p} \varphi_{k|k}(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)}). \end{aligned} \quad (4.1)$$

Infelizmente, é impossível amostrar eficientemente da distribuição “alvo” $p(\theta_{1:k}, x_{1:k}|y_{1:k})$ em qualquer tempo k . Associamos então o método de Monte Carlo para cadeia de Markov à técnica de amostragem por importância, em que uma distribuição arbitrária proposta $\Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k})$, chamada distribuição de amostragem ou densidade por importância, é introduzida, tal que $p(\theta_{1:k}, x_{1:k}|y_{1:k}) > 0$ implica $\Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k}) > 0$. Isso significa que a medida Υ é absolutamente contínua em relação a p . Além disso, podemos dizer que Υ é dominada por p e que esta relação é indicada por $\Upsilon \ll p$. Para facilitar a obtenção de amostras, multiplicamos e dividimos $E(\varphi_{k|k})$ por $\Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k})$,

$$\begin{aligned} E(\varphi_{k|k}) &= \int \varphi_{k|k}(\theta_k, x_k) \frac{p(\theta_{1:k}, x_{1:k}|y_{1:k})}{\Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k})} \Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k}) d\theta_{1:k} dx_{1:k} \\ &= \int \varphi_{k|k}(\theta_k, x_k) w(\theta_{1:k}, x_{1:k}) \Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k}) d\theta_{1:k} dx_{1:k} \end{aligned} \quad (4.2)$$

com o peso por importância igual à

$$w(\theta_{1:k}, x_{1:k}) = \frac{p(\theta_{1:k}, x_{1:k}|y_{1:k})}{\Upsilon(\theta_{1:k}, x_{1:k}|y_{1:k})}. \quad (4.3)$$

Uma outra maneira de representar o valor esperado de $\varphi_{k|k}$ pode ser dado por

$$\begin{aligned}
 E(\varphi_{k|k}) &= \frac{\int \varphi_{k|k}(\theta_k, x_k) \frac{p(\theta_{1:k}, x_{1:k} | y_{1:k})}{\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})} \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}}{\int p(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}} \\
 &= \frac{\int \varphi_{k|k}(\theta_k, x_k) \frac{p(\theta_{1:k}, x_{1:k} | y_{1:k})}{\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})} \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}}{\int \frac{p(\theta_{1:k}, x_{1:k} | y_{1:k})}{\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})} \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}} \\
 &= \frac{\int \varphi_{k|k}(\theta_k, x_k) w(\theta_{1:k}, x_{1:k}) \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}}{\int w(\theta_{1:k}, x_{1:k}) \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) d\theta_{1:k} dx_{1:k}}. \tag{4.4}
 \end{aligned}$$

Temos ainda que $E(\varphi_{k|k})$ também pode ser expresso por

$$E(\varphi_{k|k}) = \frac{E_{\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})} [\varphi_{k|k}(\theta_k, x_k) w(\theta_{1:k}, x_{1:k})]}{E_{\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})} [w(\theta_{1:k}, x_{1:k})]} \tag{4.5}$$

com

$$E_{p(a|b)} [f(a)] = \int f(a) p(a|b) da. \tag{4.6}$$

Se tivermos N_p i.i.d. amostras aleatórias $\{\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)}; \ell = 1, \dots, N_p\}$ distribuídas de acordo com $\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k})$, então uma estimativa de Monte Carlo de $E(\varphi_{k|k})$, combinando (4.1) e (4.4) com $\bar{p}_{N_p} = \Upsilon$, é dada por

$$\begin{aligned}
 \hat{E}_{N_p}^1(\varphi_{k|k}) &= \frac{\sum_{\ell=1}^{N_p} \varphi_{k|k}(\theta_k^{(\ell)}, x_k^{(\ell)}) w(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)})}{\sum_{\ell=1}^{N_p} w(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)})} \\
 &= \sum_{\ell=1}^{N_p} \tilde{w}_{1:k}^{(\ell)} \varphi_{k|k}(\theta_k^{(\ell)}, x_k^{(\ell)}) \tag{4.7}
 \end{aligned}$$

em que os pesos por importância normalizados $\tilde{w}_{1:k}^{(\ell)}$ são iguais a

$$\tilde{w}_{1:k}^{(\ell)} = \frac{w(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)})}{\sum_{j=1}^{N_p} w(\theta_{1:k}^{(j)}, x_{1:k}^{(j)})} \tag{4.8}$$

Este método é equivalente a seguinte aproximação pontual de massa da distribuição $p(\theta_{1:k}, x_{1:k} | y_{1:k})$:

$$\hat{p}_{N_p}(\theta_{1:k}, x_{1:k} | y_{1:k}) = \frac{1}{N_p} \sum_{\ell=1}^{N_p} \tilde{w}_{1:k}^{(\ell)} \delta_{(\theta_{1:k}^{(\ell)}, x_{1:k}^{(\ell)})}(\theta_{1:k}, x_{1:k}) \tag{4.9}$$

e

$$\hat{p}_{N_p}(\theta_k, x_k | y_{1:k}) = \frac{1}{N_p} \sum_{\ell=1}^{N_p} \tilde{w}_{1:k}^{(\ell)} \delta_{(\theta_k^{(\ell)}, x_k^{(\ell)})}(\theta_k, x_k). \quad (4.10)$$

No caso de simulação “perfeita”, i.e., $\Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) = p(\theta_{1:k}, x_{1:k} | y_{1:k})$, teríamos $\tilde{w}_{1:k}^{(\ell)} = N_p^{-1}$ para todo ℓ . Na prática, iremos tentar selecionar a distribuição por importância mais perto possível da distribuição alvo. Como explicitado na seção 3.3.2, para N_p finito, $\hat{E}_{N_p}^1(\varphi_{k|k})$ é tendencioso (razão das estimativas), mas assintoticamente de acordo com LFGN, $\hat{E}_{N_p}(\varphi_{k|k})$ converge quase certamente para $E(\varphi_{k|k})$. Sobre restrições adicionais, a convergência via TLC em (3.68) também é garantida. Entretanto, iremos mostrar primeiro como a variância do estimador $\hat{E}_{N_p}^1(\varphi_{k|k})$ pode ser diminuída.

4.2 Redução da Variância

A distribuição $p(\theta_{1:k}, x_k | y_{1:k})$ pode ser decomposta por

$$p(\theta_{1:k}, x_k | y_{1:k}) = p(\theta_{1:k} | y_{1:k}) p(x_k | \theta_{1:k}, y_{1:k}), \quad (4.11)$$

em que $p(x_k | \theta_{1:k}, y_{1:k})$ é uma distribuição Gaussiana cujos parâmetros são dados pelo Filtro de Kalman. Portanto, se tivermos uma aproximação de $p(\theta_{1:k} | y_{1:k})$, nós teremos diretamente uma aproximação de $p(\theta_{1:k}, x_k | y_{1:k})$.

Além disso, se $E_{p(x_k | y_{1:k}, \theta_{1:k})}[\varphi_{k|k}(\theta_k, x_k)]$ puder ser calculada numa expressão de forma fechada, então a seguinte alternativa bayesiana de amostra por importância da estimativa de $E(\varphi_{k|k})$ pode ser proposta

$$\hat{E}_{N_p}^2(\varphi_{k|k}) = \frac{\sum_{\ell=1}^{N_p} E_{p(x_k | y_{1:k}, \theta_{1:k}^{(\ell)})}[\varphi_{k|k}(\theta_k^{(\ell)}, x_k)] w(\theta_{1:k}^{(\ell)})}{\sum_{\ell=1}^{N_p} w(\theta_{1:k}^{(\ell)})} \quad (4.12)$$

em que

$$w(\theta_{1:k}) = \frac{p(\theta_{1:k} | y_{1:k})}{\Upsilon(\theta_{1:k} | y_{1:k})} \quad (4.13)$$

e

$$\Upsilon(\theta_{1:k} | y_{1:k}) = \int \Upsilon(\theta_{1:k}, x_{1:k} | y_{1:k}) dx_{1:k}. \quad (4.14)$$

Intuitivamente, para alcançar uma dada precisão, $\hat{E}_{N_p}^2$ irá requisitar um número N_p reduzido de amostras se comparado com $\hat{E}_{N_p}^1$, pois precisamos amostrar a partir de uma distribuição de dimensão menor. Isso pode ser provado através da proposição 1 apresentada

em (Doucet et al., 2001). Dado este resultado, focaremos na amostragem por importância para conseguir uma aproximação de $p(\theta_{1:k}|y_{1:k})$ e $E(\varphi_k|k)$, usando uma distribuição por importância $\Upsilon(\theta_{1:k}|y_{1:k})$.

4.3 Amostragem Sequencial por Importância

Como θ_k é um processo Markoviano, podemos sempre decompor a função por importância no tempo k através de

$$\Upsilon(\theta_{1:k}|y_{1:k}) = \Upsilon(\theta_1|y_{1:k}) \prod_{\ell=2}^k \Upsilon(\theta_\ell|y_{1:k}, \theta_{1:\ell-1}). \quad (4.15)$$

Para desenvolver um filtro recursivo como descrito na introdução deste capítulo é preciso substituir $\Upsilon(\theta_\ell|y_{1:k}, \theta_{1:\ell-1})$ no lado direito de (4.15) por $\Upsilon(\theta_\ell|y_{1:\ell}, \theta_{1:\ell-1})$. Esta aproximação se faz necessária na obtenção, em qualquer instante k , de uma estimativa da distribuição $p(\theta_{1:k}|y_{1:k})$ e na propagação desta estimativa no tempo sem modificar, subsequente, as trajetórias passadas simuladas $\{\theta_{1:k}^{(\ell)}; \ell = 1, \dots, N_p\}$. Ao fazermos uso desta aproximação, estamos diminuindo o tempo computacional requerido para o filtro e desconsiderando parte do conjunto de observação $y_{\ell+1:k}$.

As funções por importância aproximadas são da forma

$$\Upsilon(\theta_{1:k}|y_{1:k}) \approx \Upsilon(\theta_1|y_1) \prod_{\ell=2}^k \Upsilon(\theta_\ell|y_{1:\ell}, \theta_{1:\ell-1}). \quad (4.16)$$

e permitem o cálculo recursivo dos pesos por importância através de

$$w(\theta_{1:k}) = w_k w(\theta_{1:k-1}) \quad (4.17)$$

em que w_k é o peso incremental. Combinando (4.13) com a restrição das funções por importância descrita acima, temos

$$w(\theta_{1:k}) = \frac{p(\theta_{1:k}|y_{1:k})}{\Upsilon(\theta_{1:k}|y_{1:k})} = \frac{p(\theta_k|y_{1:k}, \theta_{1:k-1})p(\theta_{1:k-1}|y_{1:k-1})}{\Upsilon(\theta_k|y_{1:k}, \theta_{1:k-1})\Upsilon(\theta_{1:k-1}|y_{1:k-1})}. \quad (4.18)$$

Ao compararmos (4.17) com (4.18), temos que o peso incremental w_k é dado por

$$\begin{aligned} w_k &= \frac{p(\theta_k|y_{1:k}, \theta_{1:k-1})}{\Upsilon(\theta_k|y_{1:k}, \theta_{1:k-1})} \\ &= \frac{p(y_k|y_{1:k-1}, \theta_{1:k})p(\theta_k|\theta_{k-1})}{\Upsilon(\theta_k|y_{1:k}, \theta_{1:k-1})} \\ &\propto \frac{p(y_k|y_{1:k-1}, \theta_{1:k})p(\theta_k|\theta_{k-1})}{\Upsilon(\theta_k|y_{1:k}, \theta_{1:k-1})}. \end{aligned} \quad (4.19)$$

4.3.1 Escolha da Distribuição por Importância

Duas distribuições $\Upsilon(\theta_{1:k}|y_{1:k})$ que minimizam a variância dos pesos por importância no instante k serão apresentadas.

- Distribuição de amostragem ótima: A distribuição ótima satisfaz

$$p(\theta_k = m|y_{1:k}, \theta_{1:k-1}) = \frac{p(y_k|y_{1:k-1}, \theta_{1:k-1}, \theta_k = m)p(\theta_k = m|\theta_{k-1})}{p(y_k|y_{1:k-1}, \theta_{1:k-1})} \quad (4.20)$$

e o peso por importância associado w_k é proporcional à

$$p(y_k|y_{1:k-1}, \theta_{1:k-1}) = \sum_{m=1}^s p(y_k|y_{1:k-1}, \theta_{1:k-1}, \theta_k = m)p(\theta_k = m|\theta_{k-1}) \quad (4.21)$$

em que

- $p(y_k|y_{1:k-1}, \theta_{1:k-1}, \theta_k = m)$ é uma distribuição normal $\mathcal{N}(\tilde{y}_{k|k-1}(\theta_{1:k-1}, \theta_k = m), S_k(\theta_{1:k-1}, \theta_k = m))$ com $\tilde{y}_{k|k-1}(\theta_{1:k})$ e $S_k(\theta_{1:k-1}, \theta_k = m)$ representando respectivamente a inovação e a covariância predita de um passo a frente da observação y_k condicionada à $\theta_{1:k}$ com $\theta_k = m$. Essas informações podem ser obtidas através do filtro de Kalman apresentado na seção 3.1;
- $p(\theta_k = m|\theta_{k-1})$ é dado pela matriz $\mathbb{P}$.

Calcular $p(y_k|y_{1:k-1}, \theta_{1:k-1})$ requer o cálculo de s passos de um passo a frente do Filtro de Kalman. Isso acarretará em um elevado custo computacional se s for grande.

- Distribuição *a priori*: se utilizarmos a distribuição *a priori* $p(\theta_k|\theta_{k-1})$ como a distribuição por importância, o peso por importância será proporcional a $p(y_k|y_{1:k-1}, \theta_{1:k}) = \mathcal{N}(\tilde{y}_{k|k-1}(\theta_{1:k}), S_k(\theta_{1:k}))$ com $\tilde{y}_{k|k-1}(\theta_{1:k})$ e $S_k(\theta_{1:k-1}, \theta_k = m)$ representando, respectivamente, a inovação e a covariância predita de um passo a frente da observação y_k condicionada à $\theta_{1:k}$ com $\theta_k = m$. O que permitirá o cálculo de apenas um passo do filtro de Kalman.

4.3.2 Degeneração do Algoritmo

(Doucet et al., 2001, proposição 4) mostra que para funções por importância na forma (4.15), a variância de $w(\theta_{1:k})$ cresce com tempo, o que provoca a degeneração do fenômeno. Praticamente, após algumas iterações do algoritmo, todos menos um dos pesos por importância normalizados estarão perto de zero e um custo computacional muito grande é utilizado

para atualizar as trajetórias cujas contribuições para o estado final são quase nulas. Por isso, é de crucial importância introduzir um passo de seleção no algoritmo.

O objetivo do passo de seleção é descartar as partículas $\theta_{1:k}$ com pequeno peso por importância normalizado $\tilde{w}(\theta_{1:k})$ e replicar as com alto $\tilde{w}(\theta_{1:k})$. Isso evita a degeneração do algoritmo, além de selecionar uma região mais interessante do espaço amostral. A cada instante em que o passo de seleção é utilizado, os pesos retornam para N_p^{-1} . Ao mecanismo de geração de partículas filhas a partir das partículas originais se dá o nome de reamostragem.

4.3.3 Reamostragem

Seja $\tilde{\theta}_{1:k}^{(\ell)}$ ($\ell = 1, \dots, N_p$) o conjunto de partículas antes da reamostragem e $\theta_{1:k}^{(\ell)}$ ($\ell = 1, \dots, N_p$) o conjunto depois. A reamostragem associa a cada partícula $\tilde{\theta}_{1:k}^{(\ell)}$ ($\ell = 1, \dots, N_p$) um número de filhos $n_k^{(\ell)} \in \mathbb{N}$, tal que

$$\sum_{\ell=1}^{N_p} n_k^{(\ell)} = N_p, \quad (4.22)$$

para obter N_p novas partículas $\theta_{1:k}^{(\ell)}$. Se $n_k^{(\ell)} = 0$, então a partícula $\tilde{\theta}_{1:k}^{(\ell)}$ é descartada, caso contrário, ela possuirá $n_k^{(\ell)}$ filhos no instante k .

Na literatura sobre filtro de partículas quatro algoritmos de reamostragem não tendenciosos são comumente utilizados:

1. Reamostragem Aleatória Simples

- Para $h = N_p, \dots, 1$ calcule u_h

$$u_h = u_{h+1} \tilde{u}_h^{\frac{1}{N_p}}, \quad u_{N_p} = \tilde{u}_{N_p}^{\frac{1}{N_p}} \text{ com } \tilde{u}_h \sim U[0,1];$$

- Alocar $n_k^{(\ell)}$ filhos de cada partícula de acordo com a distribuição

$$n_k^{(\ell)} = n^Q \text{ de valores de } u_h \in \left(\sum_{s=1}^{\ell-1} \tilde{w}_k^{(s)}, \sum_{s=1}^{\ell} \tilde{w}_k^{(s)} \right].$$

2. Reamostragem Estratificada

- Para $h = 1, \dots, N_p$ calcule u_h

$$u_h = \frac{(h-1) + \tilde{u}_h}{N_p} \text{ com } \tilde{u}_h \sim U[0,1];$$

- Alocar $n_k^{(\ell)}$ filhos de cada partícula de acordo com a distribuição

$$n_k^{(\ell)} = n^{\mathcal{Q}} \text{ de valores de } u_h \in \left(\sum_{s=1}^{\ell-1} \tilde{w}_k^{(s)}, \sum_{s=1}^{\ell} \tilde{w}_k^{(s)} \right].$$

3. Reamostragem Sistemática

- Para $h = 1, \dots, N_p$ calcule u_h

$$u_h = \frac{(h-1) + \tilde{u}}{N_p} \text{ com } \tilde{u} \sim U[0,1];$$

- Alocar $n_k^{(\ell)}$ filhos de cada partícula de acordo com a distribuição

$$n_k^{(\ell)} = n^{\mathcal{Q}} \text{ de valores de } u_h \in \left(\sum_{s=1}^{\ell-1} \tilde{w}_k^{(s)}, \sum_{s=1}^{\ell} \tilde{w}_k^{(s)} \right].$$

4. Reamostragem Residual

- Alocar $n_k^{(\ell)'} = \lfloor N_p \tilde{w}_k^{(\ell)} \rfloor$ cópias da partícula ℓ para a nova distribuição;
- Adicionalmente, reamostrar $m_k = N_p - \sum_{\ell=1}^{N_p} n_k^{(\ell)'}$ partículas, fazendo $n_k^{(\ell)''}$ filhos da partícula ℓ com probabilidade proporcional a $\hat{w}_k^{(\ell)} = N_p \tilde{w}_k^{(\ell)} - n_k^{(\ell)'}$, utilizando um dos esquemas de amostragem utilizados anteriormente;
- Calcular $n_k^{(\ell)} = n_k^{(\ell)'} + n_k^{(\ell)''}$.

A principal diferença entre as reamostragens está relacionada à variância do número de filhos de uma partícula $n_k^{(\ell)}$ em cada iteração k . As reamostragens estratificada, sistemática e residual possuem menores variâncias se comparadas à reamostragem simples aleatória, pois utilizam métodos de redução de variância. A reamostragem que possui a menor variância é a sistemática.

Em relação ao custo computacional temos que a reamostragem aleatória simples é a que possui o maior custo computacional, seguida da estratificada e finalmente, da sistemática. A residual, por sua vez, tem uma parte das partículas determinadas deterministicamente e a outra por um dos algoritmos de reamostragem já especificados. Portanto, o custo computacional pode ser maior ou menor do que o custo do algoritmo de reamostragem utilizado para complementar a parte determinística.

Duas estimativas de $E(\varphi_{k|k})$ podem ser propostas antes $\tilde{E}_{N_p}(\varphi_{k|k})$ e depois $\hat{E}_{N_p}(\varphi_{k|k})$ da reamostragem no instante k , em que

$$\tilde{E}_{N_p}(\varphi_{k|k}) = \int E_{p(x_k|y_{1:k}, \theta_{1:k})} [\varphi_{k|k}(\theta_k, x_k)] \cdot \tilde{p}_{N_p}(\theta_{1:k}|y_{1:k}) d\theta_{1:k} \quad (4.23)$$

$$\hat{E}_{N_p}(\varphi_{k|k}) = \int E_{p(x_k|y_{1:k}, \theta_{1:k})} [\varphi_{k|k}(\theta_k, x_k)] \cdot \hat{p}_{N_p}(\theta_{1:k}|y_{1:k}) d\theta_{1:k} \quad (4.24)$$

com

$$\tilde{p}_{N_p}(\theta_{1:k}|y_{1:k}) = \sum_{\ell=1}^{N_p} \tilde{w}_k^{(\ell)} \delta_{\tilde{\theta}_{1:k}^{(\ell)}}(\theta_{1:k}) \quad (4.25)$$

e

$$\hat{p}_{N_p}(\theta_{1:k}|y_{1:k}) = \frac{1}{N_p} \sum_{\ell=1}^{N_p} \delta_{\theta_{1:k}^{(\ell)}}(\theta_{1:k}). \quad (4.26)$$

Mais informações sobre reamostragem podem ser obtidas em (Bolíc et al., 2004).

4.4 Algoritmo do Filtro de Partículas de Monte Carlo

Supondo que no instante $k - 1$, N_p amostras aleatórias $\theta_{1:k-1}^{(\ell)}, \ell = 1, \dots, N_p$, distribuídas aproximadamente de acordo com $p(\theta_{1:k-1}|y_{1:k-1})$, o filtro de partículas de Monte Carlo evolui no instante k conforme o procedimento descrito abaixo.

- Passo da amostragem sequencial por importância

Para $\ell = 1, \dots, N_p$, amostre $\tilde{\theta}_k^{(\ell)} \sim \Upsilon(\theta_k|y_{1:k}, \theta_{1:k-1}^{(\ell)})$, e faça $\tilde{\theta}_{1:k}^{(\ell)} \triangleq (\theta_{1:k-1}^{(\ell)}, \tilde{\theta}_k^{(\ell)})$.

Para $\ell = 1, \dots, N_p$, calcule os pesos incrementais sobre uma constante de normalização

$$w_k^{(\ell)} \propto \frac{p(y_k|y_{1:k}, \tilde{\theta}_{1:k}^{(\ell)}) p(\tilde{\theta}_k^{(\ell)}|\tilde{\theta}_{k-1}^{(\ell)})}{\Upsilon(\tilde{\theta}_k^{(\ell)}|y_{1:k}, \tilde{\theta}_{1:k-1}^{(\ell)})}. \quad (4.27)$$

Para $\ell = 1, \dots, N_p$, normalize os pesos incrementais

$$\tilde{w}_k^{(\ell)} = \left[\sum_{j=1}^N w_k^{(j)} \right]^{-1} w_k^{(\ell)}; \quad (4.28)$$

- Passo de Seleção

Obter N_p partículas filhas $(\theta_{1:k}^{(\ell)}; \ell = 1, \dots, N_p)$ a partir da replicação/descarte das partículas $\tilde{\theta}_{1:k}^{(\ell)}; \ell = 1, \dots, N_p$, considerando os pesos incrementais normalizados $\tilde{w}_k^{(\ell)}$;

- Cálculo das estimativas

$$\begin{aligned}
 E[\theta_k = i | y_{1:k}] &= E[1_{\{\theta_k = i\}} | y_{1:k}] \\
 &\approx \frac{\sum_{\ell=1}^{N_p} 1_{\{\theta_k^{(\ell)} = i\}}}{N_p}; \tag{4.29}
 \end{aligned}$$

$$\begin{aligned}
 E[x_k | y_{1:k}] &= \sum_{i=1}^s E[x_k | \theta_k = i, y_{1:k}] E[1_{\{\theta_k = i\}} | y_{1:k}] \\
 &\approx \sum_{i=1}^s \frac{\sum_{\ell=1}^{N_p} 1_{\{\theta_k^{(\ell)} = i\}} \sum_{\ell=1}^{N_p} m_{k|k}^{(\ell)} 1_{\{\theta_k^{(\ell)} = i\}}}{N_p \sum_{h=1}^{N_p} 1_{\{\theta_k^{(h)} = i\}}} \\
 &\approx \frac{\sum_{i=1}^s \sum_{\ell=1}^{N_p} m_{k|k}^{(\ell)} 1_{\{\theta_k^{(\ell)} = i\}}}{N_p} \\
 &\approx \frac{\sum_{\ell=1}^{N_p} m_{k|k}^{(\ell)}}{N_p}; \tag{4.30}
 \end{aligned}$$

$$\begin{aligned}
 E[x_k x_k' | \theta_{1:k}, y_{1:k}] &= \sum_{i=1}^s E[x_k x_k' | \theta_{1:k-1}, \theta_k = i, y_{1:k}] E[1_{\{\theta_k = i\}} | y_{1:k}] \\
 &\approx \sum_{i=1}^s \frac{\sum_{\ell=1}^{N_p} 1_{\{\theta_k^{(\ell)} = i\}} \sum_{\ell=1}^{N_p} [m_{k|k}^{(\ell)} (m_{k|k}^{(\ell)})' + P_{k|k}^{(\ell)}] 1_{\{\theta_k^{(\ell)} = i\}}}{N_p \sum_{h=1}^{N_p} 1_{\{\theta_k^{(h)} = i\}}} \\
 &\approx \frac{\sum_{i=1}^s \sum_{\ell=1}^{N_p} [m_{k|k}^{(\ell)} (m_{k|k}^{(\ell)})' + P_{k|k}^{(\ell)}] 1_{\{\theta_k^{(\ell)} = i\}}}{N_p} \\
 &\approx \frac{\sum_{\ell=1}^{N_p} [m_{k|k}^{(\ell)} (m_{k|k}^{(\ell)})' + P_{k|k}^{(\ell)}]}{N_p}. \tag{4.31}
 \end{aligned}$$

sendo N_p , $m_{k|k}^{(\ell)}$ e $P_{k|k}^{(\ell)}$, o número total de partículas, a estimativa e a variância do estado contínuo, no instante k da partícula ℓ , respectivamente.

Podemos também calcular o valor esperado de x_k e $x_k x'_k$ condicionado a θ_k dado por

$$E[x_k | \theta_k = i, y_{1:k}] \approx \frac{\sum_{\ell=1}^{N_p} m_{k|k}^{(\ell)} 1_{\{\theta_k^{(\ell)}=i\}}}{\sum_{h=1}^{N_p} 1_{\{\theta_k^{(h)}=i\}}} \quad (4.32)$$

$$E[x_k x'_k | \theta_k = i, y_{1:k}] \approx \frac{\sum_{\ell=1}^{N_p} [m_{k|k}^{(\ell)} (m_{k|k}^{(\ell)})' + P_{k|k}^{(\ell)}] 1_{\{\theta_k^{(\ell)}=i\}}}{\sum_{h=1}^{N_p} 1_{\{\theta_k^{(h)}=i\}}}, \quad (4.33)$$

sendo N_p , $m_{k|k}^{(\ell)}$ e $P_{k|k}^{(\ell)}$, o número total de partículas, a estimativa e a variância do estado contínuo no instante k da partícula ℓ , respectivamente.

4.5 Questões de Implementação

A primeira vista, pode parecer que precisamos manter na memória os caminhos de todas as trajetórias $\tilde{\theta}_{1:k}^{(i)}, i = 1, \dots, N_p$ para $\forall k$. Neste caso, o requisito de armazenamento iria aumentar linearmente com o tempo. Como $p(\theta_k | \theta_{k-1})$ e $p(\theta_k | y_{1:k}, \theta_{1:k-1})$ geram a restrição padrão de que $\Upsilon(\theta_k | y_{1:k}, \theta_{1:k-1})$ depende apenas de $\theta_{1:k-1}$, nós só precisamos manter na memória o conjunto estatístico de baixa dimensão caracterizado por $\tilde{\theta}_{k-1}^{(i)}$, $m_{k|k-1}^{(i)}$ e $P_{k|k-1}^{(i)}$ para $i = 1, \dots, N_p$. Portanto, os requisitos de armazenamento são $O(N_p)$ e não crescem com o tempo.

Para verificar a influência do número de partículas N_p no filtro, utilizamos o exemplo do rastreamento de alvo móvel que será apresentada na seção 5.2. Tendo com base um conjunto com 400 amostras, variamos o número de partículas de acordo com a sequência $N_p = \{25, 50, 75, 100, 250, 500, 750, 1000\}$ e realizamos 29 simulações/estimativas para cada número de partículas da sequência. Na figura 4.1 podemos ver como o erro quadrático médio (RMS) da estimativa diminui exponencialmente com o aumento do número de partículas. Além disso, podemos detectar a diminuição da variância com o aumento do número de partículas, aumentando a confiabilidade da amostra.

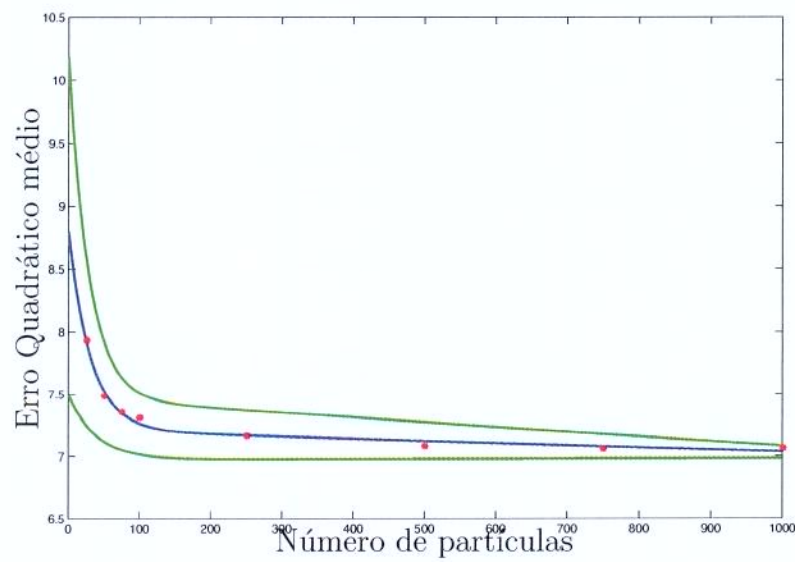


Figura 4.1: Evolução do erro quadrático médio com o aumento do número de partículas com confiabilidade de 95%.

Já na figura 4.2 podemos ver o aumento do tempo computacional devido ao aumento do número de partículas utilizado.

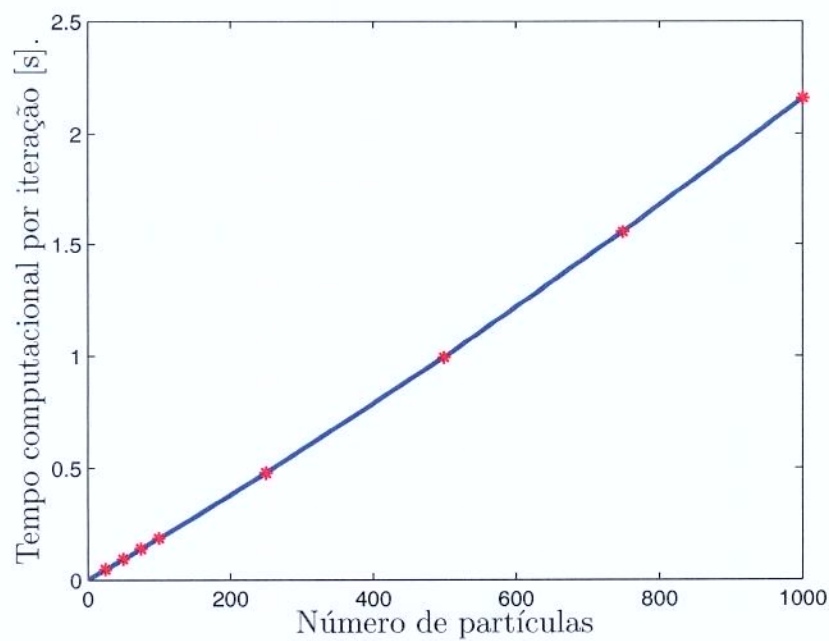


Figura 4.2: Comportamento do tempo computacional por iteração com o aumento do número de partículas.

Capítulo 5

Aplicações

Neste capítulo, apresentamos duas aplicações dos algoritmos de filtragem sem observação de estados estudados nos capítulos 3 e 4: a identificação de fácies litológicas a partir de um perfil de resposta real (Ruanaidh e Fitzgerald, 1996; Salomão, 2006) e o rastreamento de alvo móvel (Doucet et al., 2000). A segunda aplicação, por sua vez, foi simulada, permitindo uma análise comparativa criteriosa associada à eficiência e robustez destes algoritmos.

5.1 Identificação de Fácies Litológicas

Na indústria do petróleo é muito importante conhecer com precisão se uma fácies litológica dentro de uma formação geológica é ou não produtora de petróleo. Como os custos para realizar testes de produção são muito elevados, a análise preliminar desta fácies é feita através de perfis litológicos.

Inúmeras ferramentas de perfilagem que levam em conta a salinidade, a resistividade nuclear-magnética e até a velocidade de propagação do som nas formações (perfis sônicos) estão disponíveis para análise dos perfis. Com base nestes perfis, um geólogo consegue identificar com razoável precisão as características litológicas das várias fácies de uma determinada formação geológica, suas propriedades físicas, bem como a natureza dos fluidos por ventura nelas contidos.

Uma característica muito importante na geologia é que dentro de uma mesma fácies, as propriedades das rochas tendem a ser muito semelhantes. Ao se passar de uma fácies para outra, mudanças abruptas nestas propriedades são detectadas, permitindo a aplicação do modelo de cadeia escondida apresentado na seção 2.2.

Conseguir identificar, em tempo real, qual fácies litológica está sendo perfurada é uma outra importante aplicação dos perfis que vem sendo estudada recentemente. Para que isso possa ocorrer, é necessário que os perfis sejam obtidos de maneira *on-line*. Ao detectar mudanças

litológicas, poderemos otimizar os parâmetros de perfuração, modificando a velocidade de rotação da broca, a força aplicada sobre a broca, ou até mesmo identificar uma zona de alta pressão e temperatura, minimizando custos operacionais.

Um exemplo ilustrativo de aplicação de sistemas SLSM à identificação de fácies litológicas é a utilização do perfil de resposta nuclear-magnética (Ruanaidh e Fitzgerald, 1996; Salomão, 2006) cujos os ruídos podem ser modelados por distribuições Gaussianas. Como temos acesso apenas às observações ruidosas provenientes do perfil, podemos recuperar estimativas dos estados contínuos (resistividade nuclear-magnética), assim como detectar mudanças de fácies.

A resistividade nuclear-magnética de uma formação A , conjunto de fácies litológicas, pode ser representada por uma distribuição normal $\mathcal{N}(rad^{(A)}, var^{(A)})$ com $rad^{(A)}$ e $var^{(A)}$, representando a média e a variância da resistividade da formação. Considerando que a resistividade nuclear-magnética dentro de uma mesma fácies é constante e que a resistividade de uma fácies não influencia a das fácies vizinhas, variáveis independentes, temos que a resistividade da fácies z_1 contida na formação geológica A é dada por

$$rad_{z_1}^{(A)} \sim \mathcal{N}(rad^{(A)}, var^{(A)}). \quad (5.1)$$

Para completar o modelo vamos considerar ora um ruído branco gaussiano de observação com variância $var^{(obs)}$ e ora uma observação errônea (*outlier*) dada pelo mau funcionamento da ferramenta de perfilagem. Podemos caracterizar o *outlier* por uma gaussiana $\mathcal{N}(rad^{(out)}, var^{(out)})$ com média e variância dadas por $rad^{(out)}$ e $var^{(out)}$, respectivamente. Quatro estados de Markov foram utilizados no modelo:

- Estado 1 - Evolução da resistividade dentro de uma mesma fácies com observação ruidosa

$$x_{k+1} = x_k \quad (5.2)$$

$$y_k = x_k + var^{(obs)} z_k; \quad (5.3)$$

- Estado 2 - Evolução da resistividade dentro de uma mesma fácies com *outlier*

$$x_{k+1} = x_k \quad (5.4)$$

$$y_k = rad^{(out)} + var^{(out)} z_k; \quad (5.5)$$

- Estado 3 - Mudança de fácies com observação ruidosa

$$x_{k+1} = rad^{(A)} + var^{(A)} v_k \quad (5.6)$$

$$y_k = x_k + var^{(obs)} z_k; \quad (5.7)$$

- Estado 4 - Mudança de fácies com *outlier*

$$x_{k+1} = rad^{(A)} + var^{(A)}v_k \quad (5.8)$$

$$y_k = rad^{(out)} + var^{(out)}z_k. \quad (5.9)$$

Embora não haja ação de controle, nós definimos u_k para $\forall k$ dada por

$$u_k = \begin{bmatrix} rad^{(A)} \\ rad^{(out)} \end{bmatrix}$$

como artifício matemático para inserir as médias não nulas do *outlier* $rad^{(out)}$ e da resistividade nuclear-magnética da formação $rad^{(A)}$.

As matrizes do sistema $\mathcal{G}$ são dadas por

$$\begin{aligned} A_1 = A_2 = 1, \quad A_3 = A_4 = 0, \quad B_1 = B_2 = 0, \quad B_3 = B_4 = var^{(A)}, \\ C_1 = C_3 = 1, \quad C_2 = C_4 = 0, \quad D_1 = D_3 = var^{(obs)}, \quad D_2 = D_4 = var^{(out)}, \\ F_1 = F_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}', \quad F_3 = F_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}', \\ G_1 = G_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}', \quad G_2 = G_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}'. \end{aligned}$$

As médias, variâncias e a matriz de transição do modelo dependem da formação e da ferramenta de perfilagem que será utilizada. Consideramos um perfil real (Ruanaidh e Fitzgerald, 1996) originário da *Schlumberger* que foi coletado durante uma operação de perfuração para óleo *Log-While-Drilling* cujos parâmetros identificados por (Salomão, 2006) são dados por $rad^{(A)} = 115.000$, $var^{(A)} = 10.000$, $var^{(obs)} = 2.500$, $rad^{(out)} = 85.000$ e $var^{(out)} = 12.500$.

Ao invés de um cadeia de Markov com quatro estados, Salomão (2006) utiliza duas cadeias de Markov com dois estados cada uma. Por isso, tivemos que transformar as duas matrizes de transições em uma única, considerando que a cadeia do modelo do sistema é independente da de medição. A matriz de transição para a cadeia de Markov com quatro estados, considerando as transições identificadas em (Salomão, 2006) é dada por

$$\mathbb{P} = \begin{bmatrix} 0,992813 & 0,003487 & 0,003687 & 0,000013 \\ 0,199260 & 0,797040 & 0,000740 & 0,002960 \\ 0,992813 & 0,003487 & 0,003687 & 0,000013 \\ 0,199260 & 0,797040 & 0,000740 & 0,002960 \end{bmatrix}.$$

O perfil real possui $T = 4050$ com T representando o número total de amostras.

Para finalizar o modelo, consideramos que o primeiro estado de Markov seja sempre igual ao estado 1, isto é, $\pi = [1 \ 0 \ 0 \ 0]'$, $x_0 = rad^{(A)} = 115.000$ e $P_0 = var^{(A)} = 10.000$. Lembramos que x_k representa a resistividade nuclear-magnética na amostra k .

Com a velocidade constante e a posição inicial da ferramenta podemos transformar as amostras de tempo em posições no espaço. Os gráficos contendo os perfis apresentarão as amostras com sinal negativo para representar a passagem da ferramenta de cima para baixo.

Em cada um dos algoritmos estudados alguns parâmetros precisam ser definidos. No *Data Augmentation* (DA) o tempo de queima foi igual a 100 e $N = 300$ com N representando o número total de iterações. No *Data Augmentation* com *Simulated Annealing* (DASA) utilizamos $N = 200$ com a taxa de resfriamento, para $\ell = 1, \dots, N$, dada por

$$Tr(\ell) = 0,98^\ell. \quad (5.10)$$

Tanto no Mínimo Erro Quadrático (MMSE), quanto nos algoritmos no sentido MAP Marginal, determinístico (MMAP1) e estocástico (MMAP2), utilizamos $N = 200$ com a taxa de resfriamento do MMAP2, para $\ell = 1, \dots, N$, dada por

$$\gamma_\ell = 0,1 + \ell \frac{9,9}{N}. \quad (5.11)$$

Por último, no filtro de partículas (FP) consideramos $N_p = 500$ com N_p representando o número total de partículas.

Para comparar a eficiência entre os diversos algoritmos foram calculados o erro quadrático para $k = 1, \dots, T$

$$EQY_k = (y_k - \bar{x}_k)'(y_k - \bar{x}_k), \quad (5.12)$$

o erro quadrático médio

$$\overline{EQY} = \frac{1}{T} \sum_{k=1}^T EQY_k, \quad (5.13)$$

o desvio padrão do erro quadrático

$$\sigma_{EQY} = \sqrt{\frac{1}{T-1} \sum_{k=1}^T (EQY_k - \overline{EQY})'(EQY_k - \overline{EQY})}, \quad (5.14)$$

e o tempo de simulação, tendo como base o mesmo processador.

O erro quadrático médio, seu desvio e o tempo de simulação para cada algoritmo podem ser vistos na tabela 5.1. Podemos verificar também nesta tabela que os erros quadráticos médios baseados na observação y_k , além de serem da mesma ordem de grandeza, apresentam valores muito próximos, o que dificulta uma análise comparativa minuciosa entre os algoritmos.

Tabela 5.1: Erro quadrático médio, desvio padrão do erro quadrático e tempo computacional dos algoritmos de filtragem sem observação.

Algoritmo	$EQY[10^7]$	$\sigma_{EQY}[10^8]$	Tempo[s]
DA	2,562	1,620	1.562,28
DASA	3,780	1,684	591,61
MMSE	2,065	1,622	1.851,36
MMAP1	2,022	1,592	1.854,43
MMAP2	3,071	1,718	1.826,65
FP	1,703	1,362	2.838,81

Procuramos então uma outra maneira de comparar as estimativas obtidas. Primeiramente, tentamos plotar o erro quadrático em função das amostra, mas infelizmente visualmente não conseguimos distinguir uma estimativa da outra.

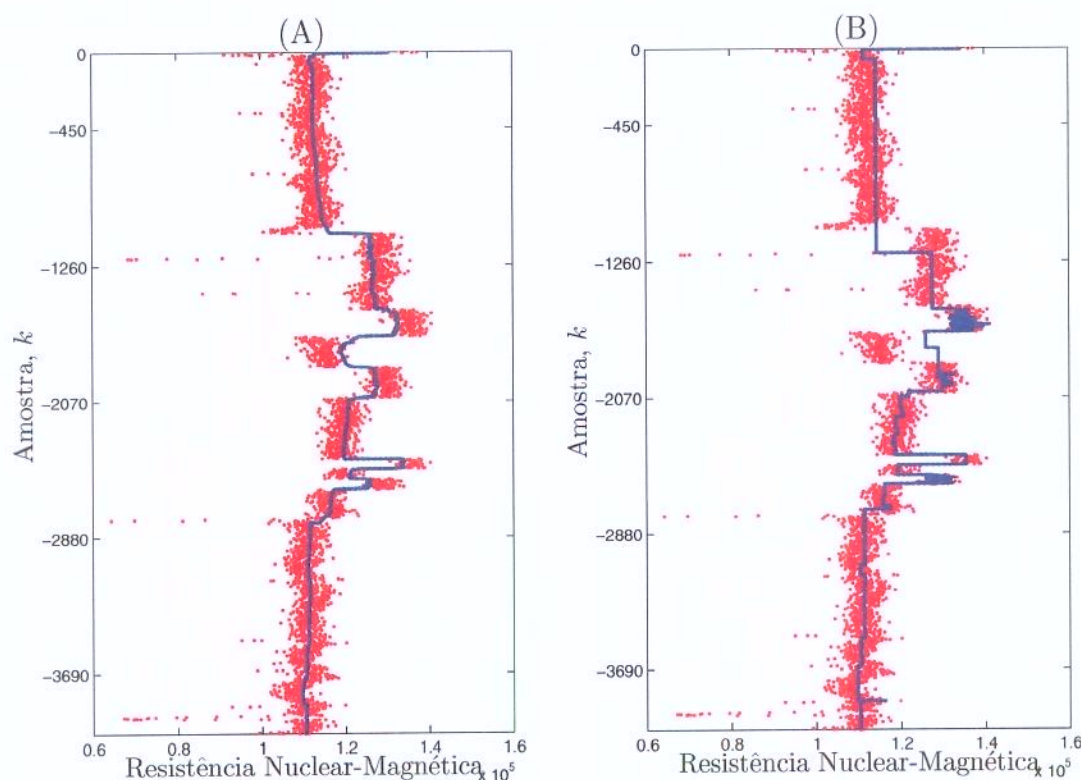


Figura 5.1: Perfil radioativo associado à estimativa via (A) *Data Augmentation* e (B) *Data Augmentation* com *Simulated Annealing*.

Outra alternativa foi plotarmos sobre o mesmo gráfico a observação y_k e o estado contínuo estimado $\bar{x}_k$. Neste caso, com a observação y_k representada pelos pontos vermelhos e estado contínuo estimado $\bar{x}_k$ pelo traço contínuo em azul, conseguimos através de inspeção visual

evidenciar algumas diferenças entre os algoritmos. Para facilitar a comparação entre os diversos algoritmos, suas estimativas foram comparadas duas a duas.

Como pode ser visto na figura 5.1, o *Data Augmentation* (DA) detecta as mudanças entre fácies litológicas, mas apresenta um *offset* entre o estado estimado $\bar{x}_k$ e a observação y_k . Já sua vertente com *Simulated Annealing* (DASA) falha na detecção de algumas mudanças de fácies próximo à iteração $k = -1000$ e $k = -1700$.

Na figura 5.2, podemos ver que tanto o algoritmo determinístico (MMA1) quanto o estocástico (MMA2), que calculam as estimativas baseadas na distribuição marginal *maximum a posteriori*, conseguem detectar os saltos com maior sensibilidade, evidenciando pequenas modificações dentro de uma mesma fácies (sub-fácies). Este detalhamento permite uma análise geológica mais precisa, pois evidencia com mais clareza a sequência de processos físico-químicos que deram origem à formação estudada. Ambos os algoritmos possuem uma boa performance computacional, mas a principal diferença entre eles é que a vertente estocástica não consegue eliminar alguns *outliers* como evidenciado nas primeiras iterações e na iteração próxima a $k = -4000$.

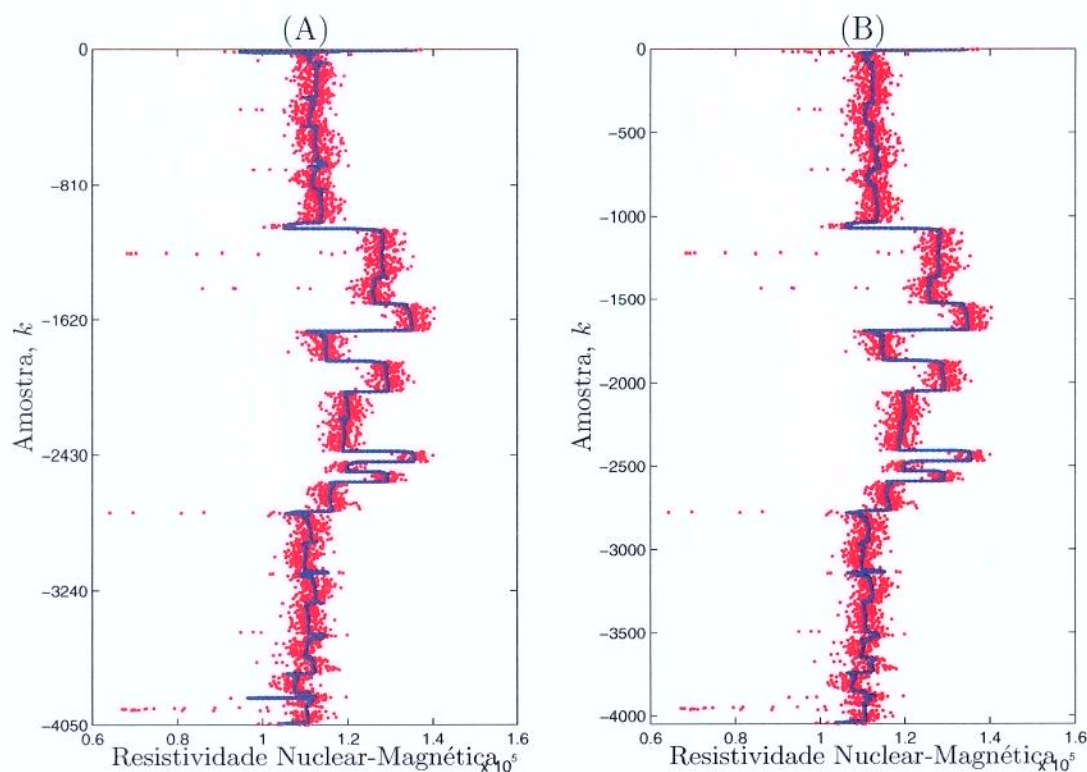


Figura 5.2: Perfil radioativo associado à estimativa via (A) MMA2 versão estocástica e (B) MMA1 versão determinística.

Na figura 5.3, podemos comparar o funcionamento do filtro de partículas (FP) com o algoritmo do mínimo erro quadrático (MMSE). Tanto o FP como o MMSE conseguem detectar com maior sensibilidade os saltos, evidenciando as sub-fácies como os algoritmos baseados na distribuição marginal *maximum a posteriori* (MMAP). Neste caso, verificamos que o filtro de partículas tem uma maior dificuldade em relação aos algoritmos em *batch* de eliminar os *outliers*. Isso se deve a característica *on line* do filtro de partículas que leva em consideração apenas as observações $y_{1:k}$ obtidas até um dado instante k , ao contrário dos algoritmos em *batch* que utilizam uma janela de observações $y_{1:T}$. Em relação ao MMSE e os algoritmos MMAP, podemos perceber que o primeiro consegue suavizar e minimizar a presença dos *outliers*, pois as estimativas são calculadas através da média das iterações.

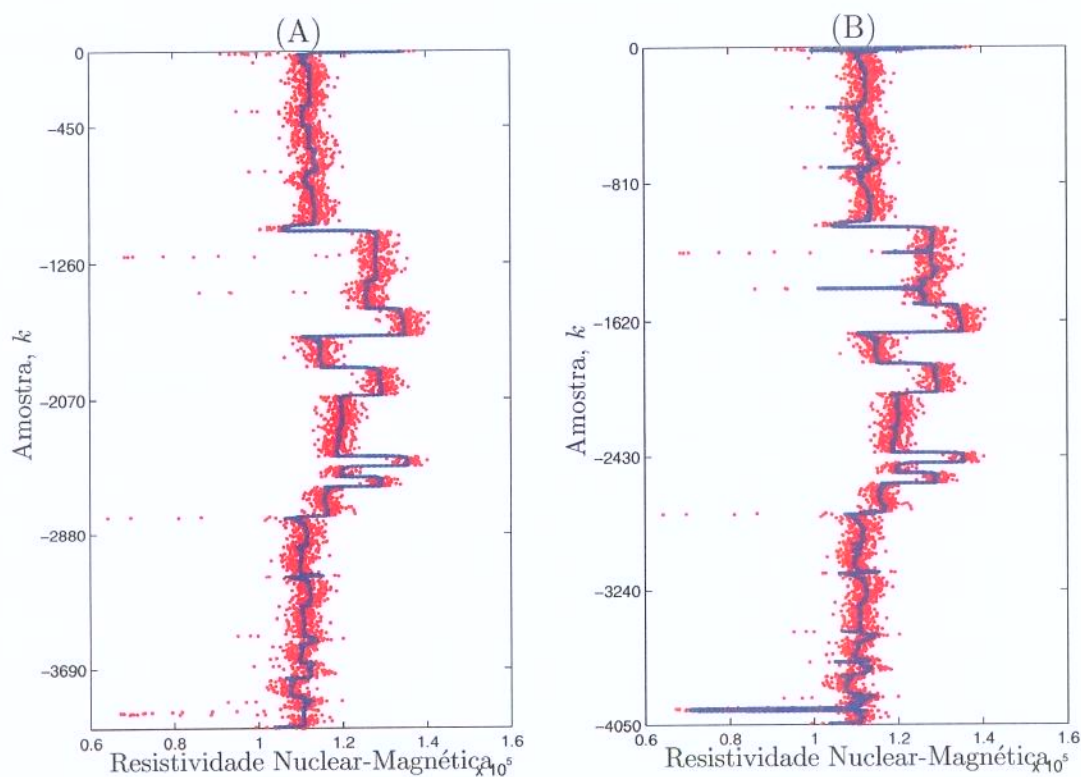


Figura 5.3: Perfil radioativo associado à estimativa via (A) Mínimo Erro Quadrático e (B) Filtro de Partículas.

Em relação à aplicação dos algoritmos de filtragem sem observação na identificação de fácies litológicas podemos concluir que o MMSE é o que melhor se aplica e que tanto o DA quanto o DASA não são indicados ou por recuperarem a resistividade nuclear-magnética com erro ou por não detectarem a mudança de fácies apropriadamente. Se for necessário, o uso de uma aplicação *on-line* o filtro de partículas poderá ser utilizado. Embora ele consiga detectar

os saltos com precisão, as estimativas das propriedades, os estado contínuos x_k , logo após os saltos, podem não representar o valor da propriedade real. Após algumas iterações, estas estimativas tendem ao valor real.

5.2 Rastreamento de Alvo Móvel

Outro exemplo de aplicação é o rastreamento de um alvo móvel imerso em ruído (Doucet et al., 2000) cuja parcela contínua do vetor de estados no instante k é representada por $x_k \triangleq (s_{x,k}, \mathcal{V}_{x,k}, s_{y,k}, \mathcal{V}_{y,k})'$, sendo $s_{x,k}/s_{y,k}$ a posição e $\mathcal{V}_{x,k}/\mathcal{V}_{y,k}$ a velocidade do alvo nas direções x e y . A dificuldade deste problema está associada à incerteza no comando de manobra que dirige o alvo em movimento.

Para modelar estes comandos, consideramos uma cadeia de Markov de três estados que correspondem, respectivamente, a três possíveis comandos de manobra:

- Estado 1 - Andar para frente;
- Estado 2 - Virar à esquerda;
- Estado 3 - Virar à direita;

Com isso, o sistema $\mathcal{G}$ evolui de acordo com o modelo linear sujeito a saltos Markovianos com os seguintes parâmetros (Bar-Shalom e Li, 1995),

$$A = \begin{bmatrix} 1 & \rho & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \rho \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = 0.1I_4, \quad C = I_4, \quad G = 0_{4 \times 1}$$

$$D = \sqrt{3} \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} -1,225 \\ -0,35 \\ 1,225 \\ 0.350 \end{bmatrix} \quad e \quad F_3 = \begin{bmatrix} 1.225 \\ 0.35 \\ -1.225 \\ -0.350 \end{bmatrix}$$

com $A_1 = A_2 = A_3 = A$, $B_1 = B_2 = B_3 = B$, $C_1 = C_2 = C_3 = C$, $D_1 = D_2 = D_3 = D$ e $G_1 = G_2 = G_3 = G$. A ação de controle determinística u_k é igual a 1 para $\forall k$. Por notação, o número máximo de amostras será representado por T .

Além disso, consideramos a mesma probabilidade de ocorrência inicial para todos os estados da cadeia $\pi = [\frac{1}{3} \ \frac{1}{3} \ \frac{1}{3}]'$. Assumimos que o ponto inicial é conhecido e dado por $\hat{x}_0 = (-500 \ -20 \ 500 \ 10)'$ e $P_0 = 0_{4 \times 4}$.

Com o conjunto original

$$\lambda \triangleq \{\pi, \mathbb{P}, A, B, C, D, F, G, \hat{x}_0, P_0\}. \quad (5.15)$$

simulamos com $T = 400$ e $\rho = 0.3$ uma trajetória e armazenamos as seqüências de estados contínuos $x_{1:T}$, de estados de Markov $\theta_{1:T}$ e de observações $y_{1:T}$, conjunto este denominado de realização. Podemos ver na figura 5.4 o caminho percorrido pelo alvo móvel representado pela linha contínua e sua observação dada pelos pontos em vermelho para esta realização.

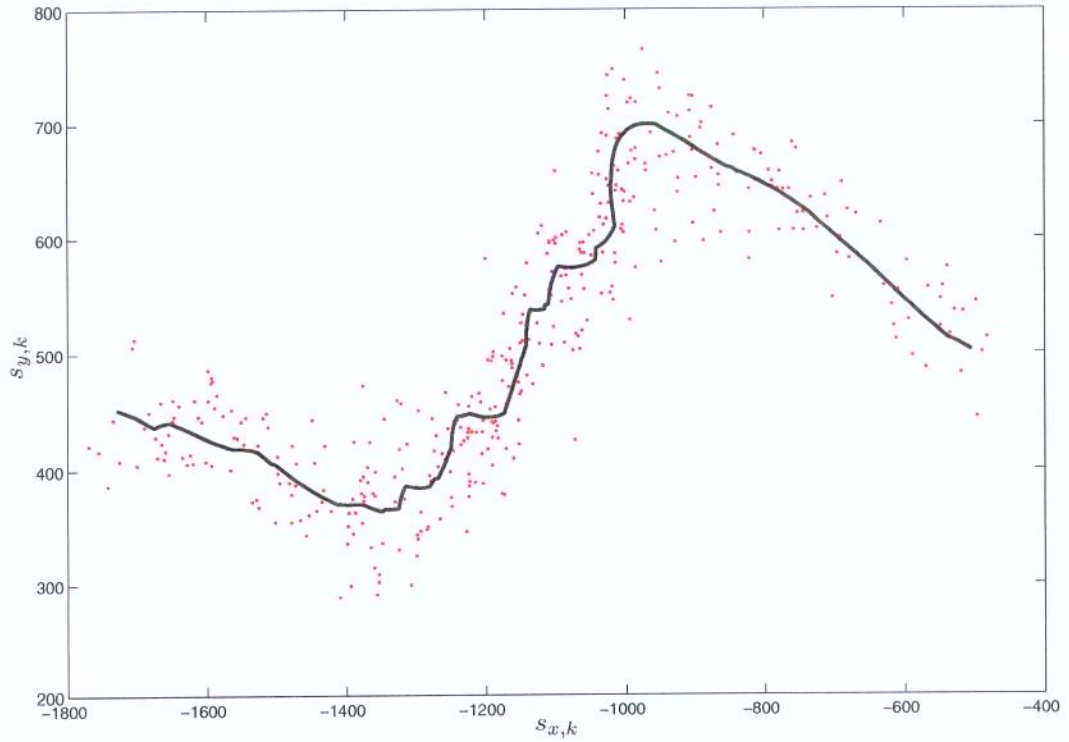


Figura 5.4: Trajetória do alvo móvel e sua observação com $s_{x,0} = -500$ e $s_{y,0} = 500$.

Calculamos a velocidade absoluta $|\mathcal{V}_k|$ e a velocidade absoluta observada $\mathcal{V}_k^{(o)}$, para $k = 1, \dots, T$, segundo

$$|\mathcal{V}_k| = \sqrt{\mathcal{V}_{x,k}^2 + \mathcal{V}_{y,k}^2} \quad (5.16)$$

$$\mathcal{V}_k^{(o)} = \sqrt{y_k(2)^2 + y_k(4)^2} \quad (5.17)$$

com $y_k = (y_k(1) \ y_k(2) \ y_k(3) \ y_k(4))'$. Na figura 5.5 podemos visualizar a velocidade absoluta representada pela linha contínua, e a observada, pelos pontos em vermelho.

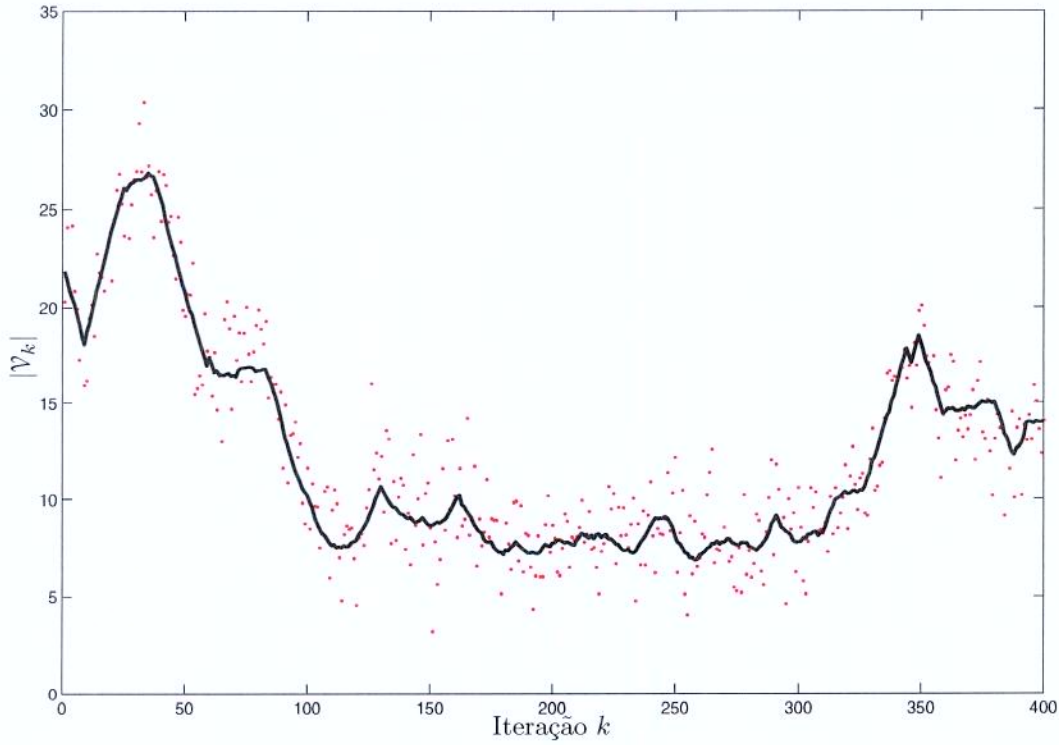


Figura 5.5: Módulo da velocidade e sua observação em função da amostra k com $\mathcal{V}_{x,0} = -20$ e $\mathcal{V}_{y,0} = 10$.

Para que pudéssemos ter um dado estatístico consistente, cada algoritmo de filtragem foi executado 50 vezes. Por notação, o número máximo de execuções será representado por T_s .

No *Data Augmentation* (DA) utilizamos o tempo de queima igual a 50 e o número de iterações, a 150. Ao utilizar o *Simulated Annealing* (DASA), consideramos $N = 100$ e a taxa de resfriamento igual a

$$Tr(\ell) = 0,98^\ell, \quad (5.18)$$

para $\ell = 1, \dots, N$. Nos algoritmos Mínimo Erro Médio Quadrático (MMSE) e nas vertentes determinísticas (MMA1) e estocástica (MMA2) do *maximum a posteriori* marginal, $N = 100$ foi considerado. A taxa de resfriamento do MMA2, para $\ell = 1, \dots, N$, foi dada por

$$\gamma_k = 0,1 + \ell \frac{9,9}{N}. \quad (5.19)$$

Nas simulações com o filtro de partícula utilizamos $N_p = 500$.

Para cada um dos algoritmos foram calculados, para $k = 1 \dots T$, a média das estimativas

$$\bar{x}_k = \frac{1}{T_s} \sum_{s=1}^{T_s} \bar{x}_k^{(s)}, \quad (5.20)$$

o desvio padrão das estimativas

$$\sigma_k = \sqrt{\frac{1}{T_s - 1} \sum_{s=1}^{T_s} (\bar{x}_k^{(s)} - \bar{x}_k)' (\bar{x}_k^{(s)} - \bar{x}_k)}, \quad (5.21)$$

o erro quadrático

$$EQ_k = (x_k - \bar{x}_k)' (x_k - \bar{x}_k), \quad (5.22)$$

o erro quadrático médio

$$\overline{EQ} = \frac{1}{T} \sum_{k=1}^T EQ_k, \quad (5.23)$$

e o desvio padrão médio

$$\bar{\sigma} = \frac{1}{T} \sum_{k=1}^T \sigma_k \quad (5.24)$$

em que $\bar{x}_k^{(s)}$ representa a estimativa do estado contínuo x_k , no instante k e na s -ésima execução de um dos algoritmo.

Na tabela 5.2 podemos encontrar o erro quadrático médio, o desvio padrão médio e o tempo de execução, tendo por base o mesmo computador, e na figura 5.6, o erro quadrático para o conjunto λ .

Tabela 5.2: Erro quadrático médio, desvio padrão médio e tempo computacional dos algoritmos de filtragem sem observação para o conjunto λ .

Algoritmo	$\overline{EQ}$	$\bar{\sigma}$	Tempo[s]
DA	344,4	407,3	53,8
DASA	384,2	136,8	39,9
MMSE	35,1	27,4	81,3
MMAPI	44,2	26,4	81,8
MMAPI2	38,1	28,9	79,9
FP	50,4	36,9	230,3

Os algoritmos baseados no *Data Augmentation*, embora possuam o menor custo computacional, possuem os maiores erro quadrático e desvio padrão médios como apresentado na tabela 5.2. Este elevado desvio garante uma grande variabilidade por simulação das estimativas dos estados contínuos $\bar{x}_{1:T}^{(s)}$.

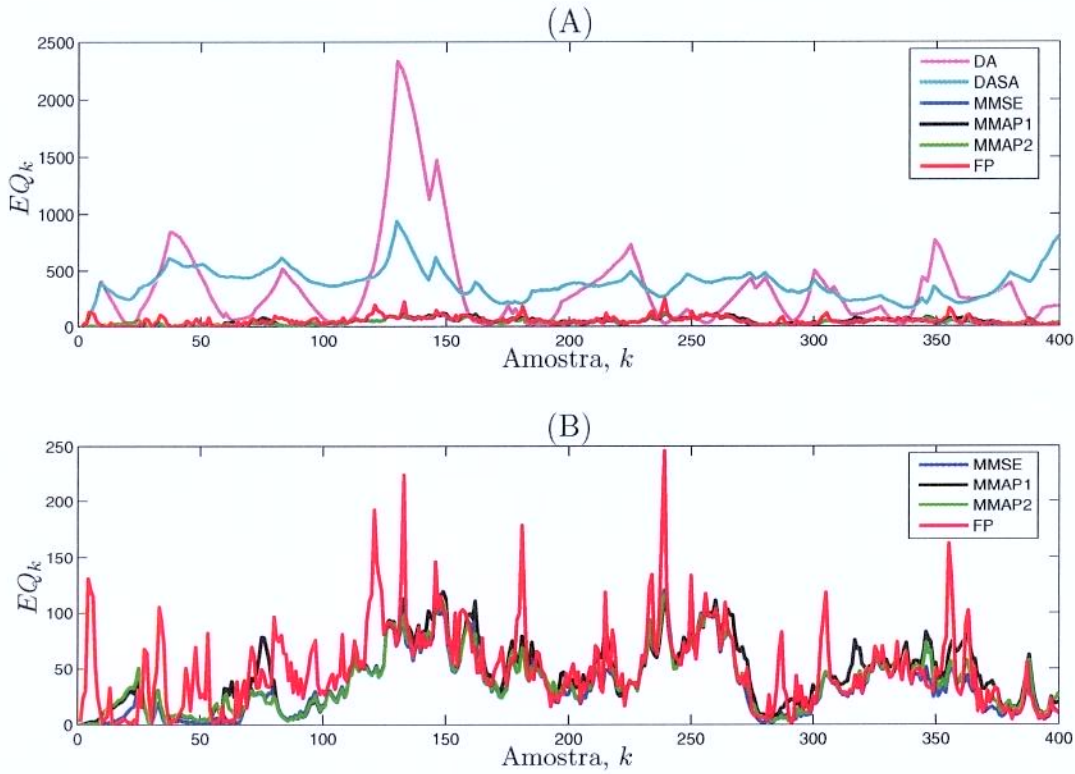


Figura 5.6: (A) Evolução do erro quadrático para λ no cálculo das estimativas. (B) Detalhe de (A).

Podemos comparar os demais algoritmos com o auxílio da tabela 5.2 e do detalhe (B) da figura 5.6. Os melhores algoritmos são o mínimo erro quadrático médio (MMSE) e as vertentes determinística (MMAP1) e estocástica (MMAP2) do MAP marginal, pois tanto seus erros quadráticos médios quanto suas variâncias médias são pequenas. Embora o filtro de partículas (FP) também possua essas mesmas propriedades, ele possui um elevado custo computacional se comparado aos outros. Este custo só é compensado pelo seu uso em aplicações *on-line*.

5.3 Sensibilidade Paramétrica

Como os algoritmos de filtragem assumem que o conjunto de parâmetros λ do modelo $\mathcal{G}$ é conhecido, testes de sensibilidade paramétrica são necessários para verificar o efeito de falhas na identificação destes parâmetros, principalmente das matrizes de transição e dos ruídos associados, no cálculo das estimativas dos estados contínuos $\bar{x}_{1:T}$.

Para testar a sensibilidade paramétrica dos algoritmos de filtragem sem observação man-

tivemos a realização obtida na seção anterior e modificamos alguns elemento do conjunto λ utilizado pelo filtro no cálculo das estimativas.

Estes testes poderão aprimorar as técnicas de identificação dos parâmetros dos sistemas lineares com saltos através da aplicação dos algoritmos de filtragem. Nas seções subseqüentes serão apresentados os resultados obtidos para cada um dos parâmetros modificados.

5.3.1 Ponto Inicial

O primeiro conjunto alterado é dado por

$$\lambda_1 \triangleq \{\pi, \mathbb{P}, A, B, C, D, F, G, x_\alpha, P_\alpha\}.$$

A nossa intenção é verificar o comportamento dos algoritmos para erros de identificação no ponto inicial. Primeiramente, escolhemos um ponto inicial diferente do conjunto original $x_\alpha^{(a)} = -\hat{x}_0$, mas com variância nula $P_\alpha^{(a)} = 0_{4 \times 4}$.

Baseados no conjunto $\lambda_1^{(a)}$, a média das estimativas, o erro quadrático e o erro quadrático médio foram calculados segundo (5.20), (5.22) e (5.23). Com o intuito de compararmos o comportamento do filtro para os conjuntos λ e $\lambda_1^{(a)}$, apresentamos o erro quadrático médio destes conjuntos na tabela 5.3. Para analisarmos o comportamento do filtro para $\lambda_1^{(a)}$ ao longo do número de amostras, o erro quadrático para cada um dos algoritmos de filtragem sem observação foi plotado na figura 5.7.

Tabela 5.3: Erro quadrático médio dos algoritmos de filtragem sem observação para $\lambda_1^{(a)}$ e λ .

Algoritmo	$\overline{EQ}[\lambda_1^{(a)}]$	$\overline{EQ}[\lambda]$
DA	291.690,0	344,4
DASA	276.580,0	384,2
MMSE	415.650,0	35,1
MMAP1	415.630,0	44,2
MMAP2	415.590,0	38,1
FP	449.570,0	50,4

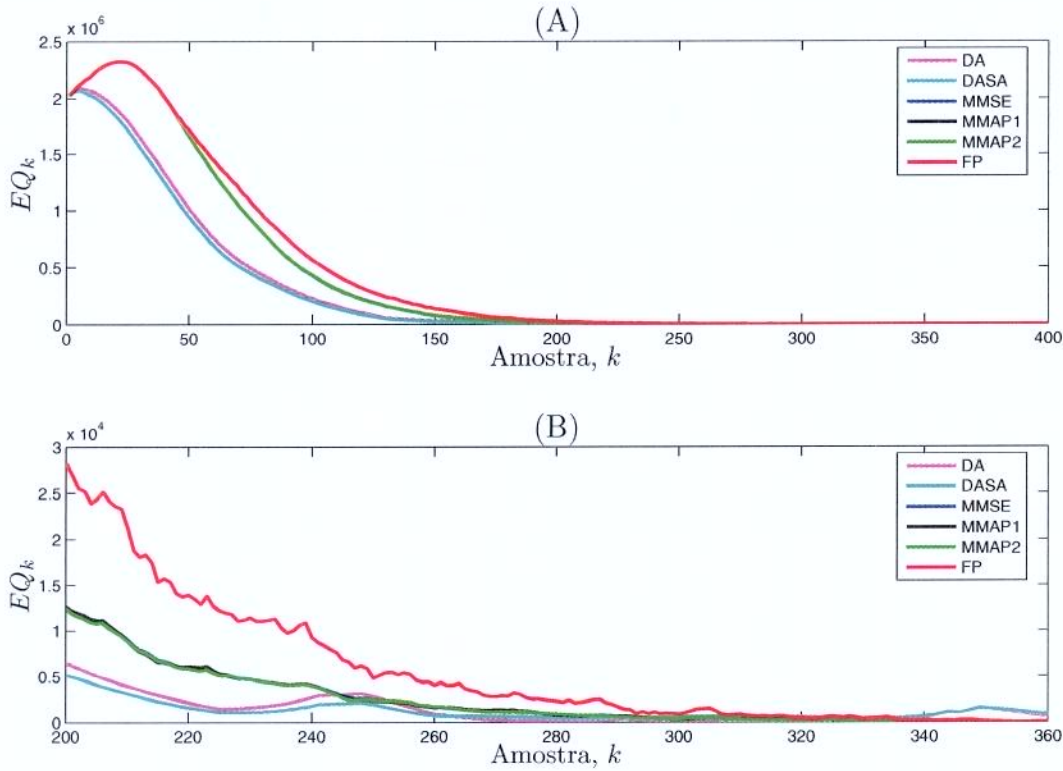


Figura 5.7: (A) Evolução do erro quadrático para $\lambda_1^{(a)}$ no cálculo das estimativas. (B) Detalhe de (A).

Na figura 5.7 podemos perceber a transição entre dois regimes: o transiente e o permanente. O primeiro é caracterizado pelo decaimento do erro quadrático e o segundo, por sua vez, pelo intervalo em que este erro se assemelha ao comportamento do erro associado ao filtro quando se tem acesso ao conjunto original λ .

Embora os algoritmos relacionados ao *Data Augmentation* apresentem elevados erros associados às estimativas dos estados contínuos $\bar{x}_{1:T}$ no regime permanente, eles, no regime transiente, conseguem corrigir rapidamente as trajetórias estimadas ocasionadas por falhas de identificação do ponto inicial ou, até mesmo, por perturbações localizadas. A rapidez com que o erro quadrático no regime transiente converge para o permanente corrobora o uso destes algoritmos nas aplicações associadas a falhas de identificação do ponto inicial com variâncias nulas ou pequenas, assim como nas aplicações em que perturbações não modeladas atuam sobre o sistema.

Seguindo a proposta de escolher um ponto inicial diferente do original $x_\alpha^{(b)} = -\hat{x}_0$, mas com uma variância elevada $P_\alpha^{(b)} = 100000I_{4 \times 4}$, calculamos a média das estimativas, o erro quadrático e o erro quadrático médio segundo (5.20), (5.22) e (5.23) baseados no conjunto

$\lambda_1^{(b)}$. Na tabela 5.4 podemos encontrar os erros quadráticos médios de cada algoritmo para os conjuntos $\lambda_1^{(b)}$ e λ . Os erros quadráticos nos instantes $k = 1, \dots, 50$ para $\lambda^{(b)}$ podem ser visualizados na figura 5.8. Neste caso, o transiente é menor se comparado ao caso anterior e nos algoritmos relacionados ao *Data Augmentation*, ele nem é perceptível.

Tabela 5.4: Erro quadrático médio dos algoritmos de filtragem sem observação para $\lambda_1^{(b)}$ e λ .

Algoritmo	$\overline{EQ}$	$\overline{EQ}[\lambda]$
DA	427,3	344,4
DASA	348,1	384,2
MMSE	45,5	35,1
MMAP1	53,9	44,2
MMAP2	48,3	38,1
FP	58,2	50,4

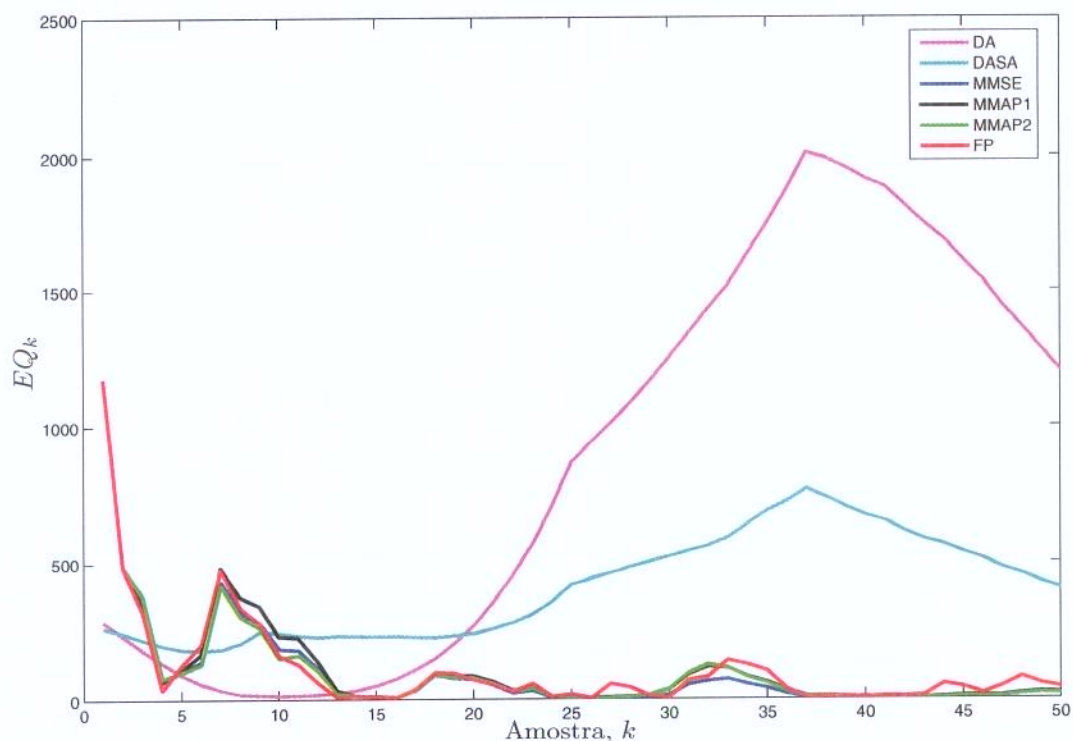


Figura 5.8: Evolução do erro quadrático para $\lambda_1^{(b)}$ no cálculo das estimativas.

Ao aumentarmos a variância do ponto inicial P_α , permitimos a excursão dos algoritmos sobre uma maior área de busca determinada pela Gaussiana $\mathcal{N}(x_\alpha, P_\alpha)$, ocasionando um

pequeno transiente, diferentemente do caso anterior em que se considerava a variância do ponto inicial nula $P_\alpha^{(a)} = 0_{4 \times 4}$.

Se tivermos acesso ao estado inicial x_α , poderemos ter $P_\alpha = 0_{n_x \times n_x}$, caso contrário, teremos que definir P_α diferente de zero. Quanto maior a confiabilidade sobre a identificação do ponto inicial, menor poderá ser a variância estimada deste ponto. Lembramos que falhas de identificação associadas a pequenas variâncias acarretam elevados erros entre as estimativas e o valor real do conjunto de estados contínuos $x_{1:T}$.

5.3.2 Ruído Associado

Nesta seção verificaremos como erros de identificação das matrizes associadas aos ruídos afetam os algoritmos de filtragem sem observação. Quando assumimos um erro menor do que ele realmente é, estamos sub-dimensionando as matrizes B e D . No extremo oposto, se considerarmos o erro maior, estaremos super-dimensionando essas matrizes. Para testar o funcionamento dos algoritmos, definimos quatro novos conjuntos de parâmetros, utilizando dois fatores de escala 0,1 e 10. Os conjuntos associados a alteração da matriz B são

$$\lambda_3 \triangleq \{\pi, \mathbb{P}, A, 0,1 \times B, C, D, F, G, \hat{x}_0, P_0\} \quad (5.25)$$

$$\lambda_4 \triangleq \{\pi, \mathbb{P}, A, 10 \times B, C, D, F, G, \hat{x}_0, P_0\} \quad (5.26)$$

e os da matriz D são

$$\lambda_5 \triangleq \{\pi, \mathbb{P}, A, B, C, 0,1 \times D, F, G, \hat{x}_0, P_0\} \quad (5.27)$$

$$\lambda_6 \triangleq \{\pi, \mathbb{P}, A, B, C, 10 \times D, F, G, \hat{x}_0, P_0\}. \quad (5.28)$$

Para cada conjunto, a média das estimativas, o erro quadrático e o erro quadrático médio foram calculados segundo (5.20), (5.22) e (5.23), respectivamente. Na tabela 5.5 podemos encontrar os erros quadráticos médios de cada algoritmo para os conjuntos λ_3 até λ_6 e λ .

Pela tabela 5.5 temos que os algoritmos associados ao *Data Augmentation* são muito sensíveis tanto ao sub-dimensionamento da matriz B (λ_3) quanto ao super-dimensionamento da Matriz D (λ_6). Quando o filtro tem como parâmetros λ_3 , ele assume que o erro de identificação do modelo do sistema é menor do que o real, prejudicando o estágio de predição através da diminuição da variância da estimativa predita. Quando o filtro tem como parâmetros λ_6 , ele assume que o erro de identificação do modelo de medição é maior do que o real, prejudicando o estágio de atualização através da aumento da variância da estimativa atualizada.

Nestes algoritmos nos surpreendeu o fato do erro quadrático médio ser menor nos conjuntos λ_4 e λ_5 do que no conjunto original λ .

Tabela 5.5: Erro quadrático médio dos algoritmos de filtragem sem observação para λ e λ_3 até λ_6 .

Algoritmo	$\overline{EQ}[\lambda_3]$	$\overline{EQ}[\lambda_4]$	$\overline{EQ}[\lambda_5]$	$\overline{EQ}[\lambda_6]$	$\overline{EQ}[\lambda]$
DA	10.207,0	143,3	128,6	10.432,0	344,4
DASA	3.119,0	154,2	149,5	5.710,0	384,2
MMSE	60,0	379,0	58,9	182,0	35,1
MMAP1	101,0	379,6	58,9	453,0	44,2
MMAP2	82,0	379,6	58,8	194,0	38,1
FP	60,0	405,7	89,01	201,0	50,4

Em relação aos outros algoritmos podemos dizer que existe uma tendência comum entre eles, pois em todos os conjuntos testados o erro quadrático médio foi maior do que o erro do conjunto original λ . Observamos ainda que o erro médio associado às matrizes super-dimensionadas possui uma ordem de grandeza superior ao erro relacionado ao sub-dimensionamento das matrizes.

Baseados nas simulações obtidas, quando não se tem uma boa identificação do sistema é indicado super-dimensionar a matriz B do modelo do sistema, aumentando a variância predita das estimativas, mas deve se evitar super-dimensionar a matriz D do modelo de medição. Quando o sistema for identificado corretamente tanto o uso do MMSE quanto dos algoritmos relacionados ao MMAP poderão ser utilizados na obtenção das estimativas do conjunto de estados contínuos $x_{1:T}$. Se for necessário o uso de um procedimento *on-line*, o filtro de partículas poderá ser aplicado.

5.3.3 Matriz de Transição

Para finalizar a análise de sensibilidade modificamos a matriz de transição, mantendo a trajetória gerada pelo conjunto original. As matrizes de transição testadas foram

$$\mathbb{P}_1 = \begin{bmatrix} 0,6 & 0,2 & 0,2 \\ 0,2 & 0,6 & 0,2 \\ 0,2 & 0,2 & 0,6 \end{bmatrix}, \mathbb{P}_2 = \begin{bmatrix} 0,3 & 0,4 & 0,4 \\ 0,4 & 0,3 & 0,4 \\ 0,4 & 0,4 & 0,3 \end{bmatrix} \text{ e } \mathbb{P}_3 = \begin{bmatrix} 0,1 & 0,45 & 0,45 \\ 0,45 & 0,1 & 0,45 \\ 0,45 & 0,45 & 0,1 \end{bmatrix},$$

gerando os conjuntos de parâmetros

$$\lambda_7 \triangleq \{\pi, \mathbb{P}_1, A, B, C, D, F, G, \hat{x}_0, P_0\}$$

$$\lambda_8 \triangleq \{\pi, \mathbb{P}_2, A, B, C, D, F, G, \hat{x}_0, P_0\} \quad (5.29)$$

$$\lambda_9 \triangleq \{\pi, \mathbb{P}_3, A, B, C, D, F, G, \hat{x}_0, P_0\}. \quad (5.30)$$

Para compararmos quantitativamente como a mudança nos parâmetros afeta os algoritmos de filtragem sem observação, calculamos a média das estimativas, os erros quadrático e quadrático médio segundo (5.20), (5.22) e (5.23), respectivamente. Os erros quadráticos médios para os conjuntos λ_7 até λ_9 e λ podem ser visto na tabela 5.6.

Tabela 5.6: Erro quadrático médio dos algoritmos de filtragem sem observação para λ e λ_7 até λ_9 .

Algoritmo	$\overline{EQ}[\lambda_7]$	$\overline{EQ}[\lambda_8]$	$\overline{EQ}[\lambda_9]$	$\overline{EQ}[\lambda]$
DA	343,0	340,4	338,7	344,4
DASA	260,7	278,8	293,4	384,2
MMSE	37,94	46,6	63,1	35,1
MMAP1	39,6	38,4	58,5	44,2
MMAP2	36,9	38,7	58,3	38,1
FP	62,1	93,4	165,2	50,4

Na tabela 5.6 temos que o *Data Augmentation* (DA) é insensível a variação da matriz de transição. Já sua vertente com *Simulated Annealing* (DASA) apresenta uma pequena diminuição do erro quadrático médio com esta variação. Tanto o filtro de partículas (FP) quanto o mínimo erro quadrático (MMSE) apresentam um aumento do erro quadrático médio com o afastamento da matriz de transição de seu valor original. Os algoritmos MAP marginais apresentam características semelhantes de erros menores ou próximos do erro do conjunto original nos conjuntos λ_7 e λ_8 e maiores no conjunto λ_9 .

5.4 Trajetórias para Diferentes Matrizes de Transição

Para corroborar os resultados obtidos na seção anterior, simulamos uma trajetória para cada um dos conjuntos λ_7 , λ_8 e λ_9 , obtendo três novas realizações. Isso foi feito para verificar a adequação do filtro a mudanças nos parâmetros do sistema. Calculamos as estimativas dos estados para cada um dos conjuntos, aplicando a realização correspondente, para determinar o comportamento do filtro para diferentes matrizes de transições.

Para cada conjunto e realização, a média das estimativas, os erros quadrático e quadrático médio foram calculados de acordo com (5.20), (5.22) e (5.23). Podemos ver os erros médios para as quatro realizações na tabela 5.7 e os erros quadráticos associados aos conjuntos λ_7 , λ_8 e λ_9 nas figuras 5.9, 5.10 e 5.11, respectivamente.

Tabela 5.7: Erro quadrático médio dos algoritmos de filtragem para diferentes trajetórias para λ e λ_7 até λ_9 .

Algoritmo	$\overline{EQ}[\lambda_7]$	$\overline{EQ}[\lambda_8]$	$\overline{EQ}[\lambda_9]$	$\overline{EQ}[\lambda]$
DA	152,4	86,6	17,6	344,4
DASA	137,9	81,5	43,2	384,2
MMSE	35,4	30,3	25,0	35,1
MMAP1	40,5	36,1	29,1	44,2
MMAP2	40,2	35,9	29,2	38,1
FP	47,4	42,3	33,5	50,4

Embora o erro médio quadrático diminua para todos os algoritmos com a alteração da matriz de transição de $\mathbb{P}_1$ até $\mathbb{P}_3$ como mostra a tabela 5.7, os algoritmos associados ao DA apresentam uma diminuição mais acentuada. Estes algoritmos que possuíam elevados erros médios nos exemplos apresentados nas seções 5.2 e 5.3, dependendo da matriz de transição, apresentam erros compatíveis ou até menores se comparados aos outros algoritmos. Isso também pode ser detectado através das figuras 5.9, 5.10 e 5.11, onde podemos visualizar os erros quadráticos associados aos conjuntos λ_7 até λ_9 .

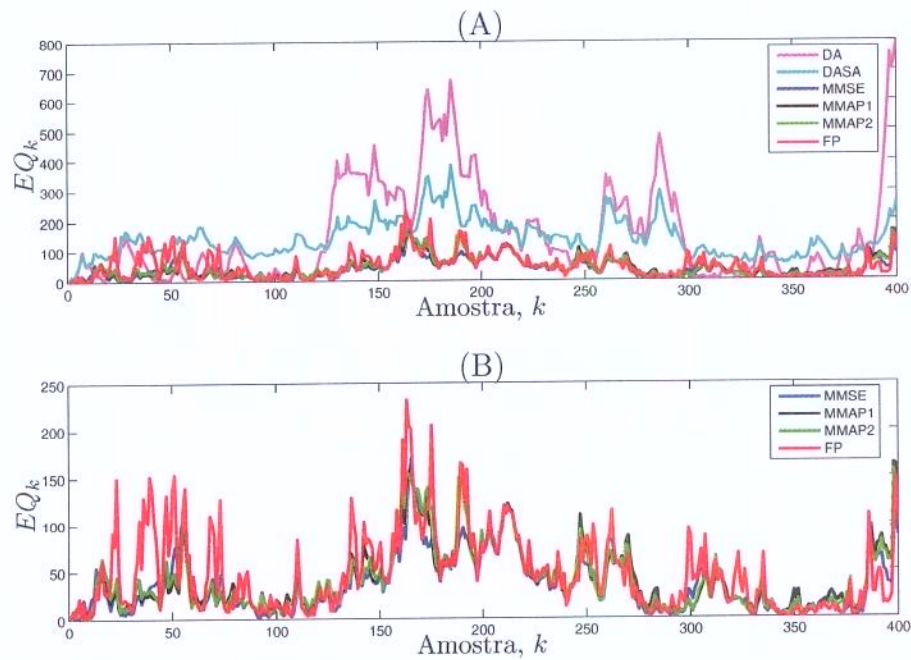


Figura 5.9: (A) Evolução do erro quadrático para λ_7 no cálculo das estimativas. (B) Detalhe de (A).

Aqui fica bem claro quando se deve utilizar os algoritmos associados ao *Data Augmentation*. Como já havíamos detectado na seção 5.3.1, estes algoritmos conseguem rapidamente passar do regime transiente para o permanente, mas quando as transições são sucessivas, o sistema se encontrará constantemente em regime transiente e nunca tenderá a permanecer em um mesmo estado de Markov, indicando o uso do *Data Augmentation*.

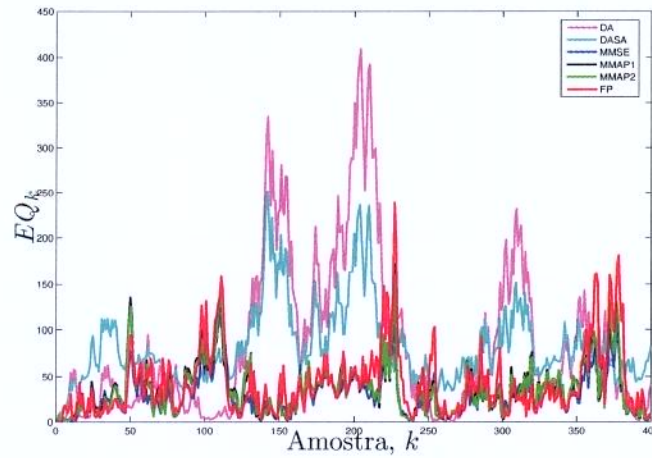


Figura 5.10: Evolução do erro quadrático para λ_8 no cálculo das estimativas.

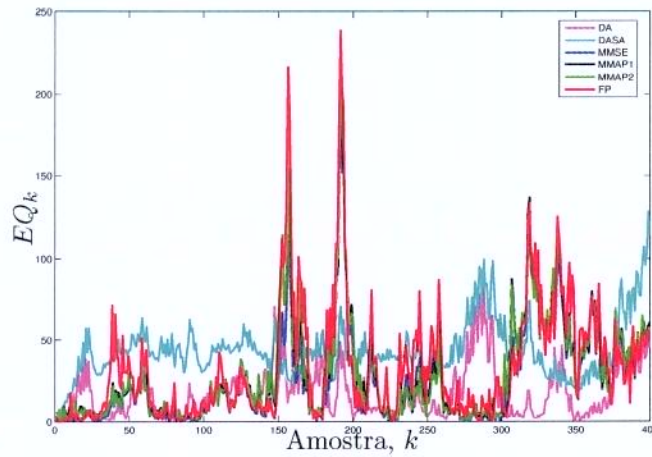


Figura 5.11: Evolução do erro quadrático para λ_9 no cálculo das estimativas.

Quando houver uma tendência de permanência em algum dos estados de Markov, minimizando as transições de estados, o sistema tenderá a sair do regime transiente para o permanente e nestes casos o MMSE, os algoritmos associados ao MMAP e o filtro de partículas devem ser utilizados.

Capítulo 6

Controle via Horizonte Retrocedente

Ao estudarmos as técnicas de filtragem nos deparamos com o desafio de controlar o sistema $\mathcal{S}$ a partir das estimativas dos estados do sistema obtidas a partir dos algoritmos apresentados nos capítulos 3 e 4. Por conveniência, buscamos selecionar uma ação de controle u_k condicionada ao vetor de observações do sistema $O_{1:k}$. Se temos acesso direto a todos os estados do sistema, podemos controlar o cenário 1 através da realimentação de $O_k = (\theta_k, x_k)$ com $u_k = U_k(\theta_k, x_k)$. No caso de observação parcial, em que não temos acesso direto a todos estes estados, demais cenários, o controle deve ser baseado em todo conjunto de observações disponíveis $O_{1:k}$ com $u_k = U_k(O_{1:k})$.

Nós dedicamos este capítulo a problemas de regulação cujo objetivo é levar os estados contínuos x_k para 0 sem utilizar muita energia associada às ações de controle u_k . Para isso, escolhemos como critério a minimização da esperança matemática de um funcional de custo na forma quadrática relacionado às variáveis de estado e de controle ao longo de um horizonte finito de tempo N (Soderstrom, 1994). Em termos práticos queremos uma estratégia de controle que minimize o custo médio ao longo de um conjunto de realizações. Este controle recebe o nome de controle via horizonte retrocedente.

Quando não temos acesso direto a todos os estados do sistema, o controle recai no conceito de controle dual, pois além de satisfazer a estratégia de controle proposta, a ação de controle deve excitar o sistema para melhorar as estimativas dos estados não observados (Davis e Vinter, 1985; Astrom, 1970). Dois objetivos completamente duais e opostos.

Contudo, devido à complexidade e ao esforço computacional requerido para a solução do controle dual, utilizaremos, por simplificação, o princípio de equivalência à certeza, em que a estratégia de controle é dividida em duas partes: um estimador que produz a melhor estimativa do vetor de estados do sistema a partir do conjunto de observações $O_{1:k}$ e uma lei de controle via realimentação cuja ação u_k é uma função linear das estimativas dos estados

contínuos $\hat{x}_k$. O cálculo da ação de controle para cada cenário será explicitado nas próximas seções.

Para avaliar a interação entre o filtro e o controle em diferentes situações de observação do estado contínuo x_k e do estado de Markov θ_k , dois exemplos ilustrativos foram implementados: o modelo linear de uma junta do *European Robot Arm (ERA)* (Yang e Blanke, 2000) modificado em (Vargas, 2004) e o exemplo 6.4 extraído de Ji e Chizeck (1990b).

6.1 Formulação do Problema

Ao estudarmos o controle via horizonte retrocedente nos deparamos com um modelo linear diferente do modelo encontrado nos artigos relacionados à filtragem, em que o cálculo de x_{k+1} depende de x_k e de θ_k e não mais de $\hat{x}_k$ e de $\hat{\theta}_{k+1}$. Para compatibilizar os algoritmos de filtragem e controle, resolvemos adotar o sistema $\mathcal{G}_c$ descrito por

$$\mathcal{G}_c : \begin{cases} x_{k+1} = A(\theta_k)x_k + B(\theta_k)v_k + F(\theta_k)u_k \\ y_k = C(\theta_k)x_k + D(\theta_k)z_k + G(\theta_k)u_k \end{cases} \quad (6.1)$$

em que $x_k \in \mathbb{R}^{n_x}$ é a parcela contínua do vetor de estados ou estado contínuo do sistema, $y_k \in \mathbb{R}^{n_y}$ é o vetor de observação no tempo k , $u_k \in \mathbb{R}^{n_u}$ é uma entrada conhecida determinística, $v_k \in \mathbb{R}^{n_v}$ é uma sequência de ruído branco gaussiano com média nula cuja variância é identidade I_{n_v} e $z_k \in \mathbb{R}^{n_z}$ é uma sequência de ruído branco gaussiano com média nula cuja variância é identidade I_{n_z} . A distribuição de probabilidade inicial deste sistema é dada por $\pi_i = p(\theta_0 = i)$ para $i \in S$, de tal forma que $\pi_i \geq 0$ para $\forall i \in S$ e $\sum_{i=1}^s \pi_i = 1$.

O controle por horizonte retrocedente busca uma estratégia de controle admissível que além de estabilizar o sistema $\mathcal{G}_c$ possa otimizar um determinado critério de desempenho em termos do valor esperado. Como não possuímos observações dos estados futuros, nos resta apenas calcular o valor esperado da evolução do sistema baseado em todas as informações conhecidas até o instante presente. O critério adotado neste trabalho minimiza o índice de desempenho, que corresponde a um funcional de custo esperado, num horizonte finito de N estágios condicionado a todas as informações conhecidas até o instante k . O funcional de custo considerado possui forma quadrática em x e u dado por

$$J^{k,N} := E \left[\sum_{\ell=0}^{N-1} (x'_{k+\ell} Q(\theta_{k+\ell}) x_{k+\ell} + u'_{k+\ell} R(\theta_{k+\ell}) u_{k+\ell}) + x'_{k+N} V(\theta_{k+N}) x_{k+N} \middle| \mathcal{O}_{1:k}, \lambda \right], \quad (6.2)$$

sendo Q e R matrizes associadas ao custo por estágio e V ao custo final.

A, B, C, D, F, G, Q, R e V representam coleções de matrizes que são funções do estado da cadeia de Markov θ_k e evoluem de acordo com a realização da cadeia de Markov finita. Assumimos $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$ e consideramos x_0, v_k e z_k mutuamente independentes para todo k . Os parâmetros do modelo são conhecidos e seu conjunto é dado por

$$\lambda \triangleq \{\pi, \mathbb{P}, A, B, C, D, F, G, Q, R, V, \hat{x}_0, P_0\},$$

de tal forma que se $\theta_k = i$, temos $A(\theta_k) = A_i, B(\theta_k) = B_i, C(\theta_k) = C_i, D(\theta_k) = D_i, F(\theta_k) = F_i, G(\theta_k) = G_i, Q(\theta_k) = Q_i, R(\theta_k) = R_i$ e $V(\theta_k) = V_i$.

Como veremos na seção 6.2, a ação de controle via horizonte retrocedente que minimiza o funcional de custo estará associada à distribuição μ_k , dada por

$$\mu_k(i) = p(\theta_k = i | O_{1:k}, \lambda), \quad (6.3)$$

e ao conjunto de matrizes de segundo momento de estado definidas por

$$X_i^{k,t} = E[x_{k+t} x'_{k+t} 1_{\{\theta_{k+t}=i\}} | O_{1:k}, \lambda], \forall i \in S, \quad (6.4)$$

para $t = 0, \dots, N$. Adotamos a notação semelhante à (do Val e Basar, 1999), com $1_{\mathcal{C}}$ representando função indicadora do conjunto $\mathcal{C}$.

6.2 Cálculo da Ação de Controle

Como a ação de controle via horizonte retrocedente depende do acesso às informações dos estados, uma metodologia específica para cada um dos cenários definidos na seção 2.5 será apresentada. Para facilitar o entendimento destas metodologias dividiremos esta seção em duas partes de acordo com a observação ou não do estado de Markov θ_k .

6.2.1 Cenários com Observação do estado de Markov

Tanto o cenário 1 quanto o 2 possuem observação direta do estado de Markov θ_k , mas diferem um do outro pela observação do estado contínuo x_k . Como no cenário 1, temos acesso ao estado contínuo x_k , poderemos aplicar controle via realimentação de estados dado por

$$u_k = K_{\theta_k}^k x_k, \quad (6.5)$$

para $\forall \theta_k \in S$. No cenário 2, como não temos acesso ao estado contínuo x_k , aplicaremos o princípio da equivalência à certeza e ao invés de utilizarmos o estado contínuo x_k no controle

por realimentação, utilizaremos $\hat{x}_k = E[x_k | \theta_{0:k-1}, \lambda]$, obtida através do banco de filtros de Kalman apresentado na seção 3.1. A ação de controle no instante k neste cenário é dada por

$$u_k = K_{\theta_k}^k \hat{x}_k, \quad (6.6)$$

para $\forall \theta_k \in S$.

Podemos perceber que tanto em (6.5) quanto em (6.6) a ação de controle u_k depende do vetor de realimentação K_{θ_k} que, por sua vez, é função do estado de Markov θ_k . Antes de calcularmos o valor de K_{θ_k} , vamos definir, para $\forall i \in S$, o operador $\mathcal{E}_i(\phi)$

$$\mathcal{E}_i(\phi) := \sum_{j=1}^n p_{ij} \phi_j. \quad (6.7)$$

Segundo (do Val e Basar, 1999) a solução ótima de K_{θ_k} que minimiza o funcional de custo para θ_k observável satisfaz, para $\forall i \in S$ e $t = 0, \dots, N-1$, a

$$K_i^{k,t} = -[R_i + F_i' \mathcal{E}_i(M^{k,t+1}) F_i]^{-1} F_i' \mathcal{E}_i(M^{k,t+1}) A_i. \quad (6.8)$$

Para $t = 0, \dots, N-1$, o conjunto de matrizes $M^{k,t}$ é determinado pelo conjunto de equações tipo Riccati

$$M_i^{k,t} = A_i' \mathcal{E}_i(M^{k,t+1}) \{I - F_i [R_i + F_i' \mathcal{E}_i(M^{k,t+1}) F_i]^{-1} F_i' \mathcal{E}_i(M^{k,t+1})\} A_i + Q_i \quad (6.9)$$

com $M_i^{k,N} = V_i$ para $\forall i \in S$.

O funcional de custo para o cenário 1 é dado por

$$J^{k,N} = x_k' M_{\theta_k}^{k,0} x_k \quad (6.10)$$

e para o cenário 2 por

$$J^{k,N} = \hat{x}_k' M_{\theta_k}^{k,0} \hat{x}_k + w_{\theta_k}^{k,0} \quad (6.11)$$

em que

$$w_i^{k,t} = \mathcal{E}_i(w^{k,t+1}) + \text{tr}\{\mathcal{E}_i(M^{k,t+1}) B_i B_i'\} \quad (6.12)$$

com $w_i^{k,N} = 0$ para $\forall i \in S$.

6.2.2 Cenários sem Observação do estado de Markov

Para representar, nos cenários sem observação do estado de Markov, o custo $J^{k,N}$ de maneira mais conveniente à otimização, alguns operadores definidos em (do Val e Basar, 1999)

serão aqui apresentados

$$\mathcal{E}_i(\phi) := \sum_{j=1}^n p_{ij} \phi_j, \forall i \in S \quad (6.13)$$

$$\mathcal{L}_i^{k,t}(\phi) := (A_i^{k,t})' \mathcal{E}_i(\phi) A_i^{k,t}, \forall i \in S \quad (6.14)$$

$$\mathcal{J}^{k,t}(\phi) := \sum_{j=1}^n p_{ji} A_j^{k,t} \phi_j (A_j^{k,t})', \forall i \in S \quad (6.15)$$

com $A_i^{k,t} = A_i + F_i K^{k,t}$.

Outro modo de representar o custo funcional $J^{k,N}$ demonstrado em (Vargas, 2004) é dado pela proposição 6.1.

Proposição 6.1 *O custo $J^{k,N}$ em (6.2) equivale à*

$$J^{k,t} = E[x'_{k+t} L_{\theta_{k+t}}^{k,t} x_{k+t} + w_{\theta_{k+t}}^{k,t}] = \langle L^{k,t}, X^{k,t} \rangle + \mu_{k+t} w^{k,t} \quad (6.16)$$

em que

$$L_i^{k,t} = Q_i + (K^{k,t})' R_i K^{k,t} + (A_i^{k,t})' \mathcal{E}_i(L^{k,t+1}) A_i^{k,t} \quad (6.17)$$

$$w_i^{k,t} = \mathcal{E}_i(w^{k,t+1}) + \text{tr}\{\mathcal{E}_i(L^{k,t+1}) B_i B_i'\} \quad (6.18)$$

$$\mu_{k+t} = \mu_k \mathbb{P}^t \quad (6.19)$$

com $L_i^{k,N} = V_i$ e $w_i^{k,N} = 0$ para $\forall i \in S$

Temos ainda que as matrizes do segundo momento associadas ao ruído descritas em (Vargas, 2004) são dadas por

$$\Psi_i^{k,t} := \sum_{j=1}^n p_{ji} \mu_{k,t}(j) B_j B_j', \forall i \in S. \quad (6.20)$$

Tanto no cenário 3 quanto no 4 precisamos de uma estimativa da distribuição de probabilidade μ_k dada por (6.3). Para o cenário 3, esta estimativa pode ser obtida a partir do filtro do modelo de cadeia escondida. Como o modelo de cadeia escondida já foi apresentado na seção 3.2, iremos apresentar nesta seção apenas sua adaptação para o sistema $\mathcal{G}_c$. A variável em avanço deste sistema é dada por

$$\alpha_\ell(i) = p(x_{1:\ell+1}, \theta_\ell = i | \lambda). \quad (6.21)$$

A variável em avanço α_ℓ pode ser calculada recursivamente através do procedimento:

1) Inicialização: Para $i = 1, \dots, s$, calcule

$$\alpha_0(i) = \pi_i p(x_1 | \theta_0 = i, \lambda); \quad (6.22)$$

2) Indução: Para $j = 1, \dots, s$ e $1 \leq \ell \leq k-1$, calcule

$$\begin{aligned} \alpha_\ell(j) &= \left[\sum_{i=1}^s \alpha_{\ell-1}(i) p(\theta_\ell = j | \theta_{\ell-1} = i) \right] p(x_{\ell+1} | x_{1:\ell}, \theta_\ell = j, \lambda) \\ &= \left[\sum_{i=1}^s \alpha_{\ell-1}(i) p_{ij} \right] p(x_{\ell+1} | x_{1:\ell}, \theta_\ell = j, \lambda). \end{aligned} \quad (6.23)$$

Se tivermos o valor de $\alpha_{k-1}(i)$ para $\forall i \in S$ em um dado instante $k-1$, podemos calcular a probabilidade do estado de Markov θ_{k-1} ser igual à i condicionada ao modelo λ e à sequência parcial de observações $x_{1:k} = \{x_1, \dots, x_k\}$

$$\begin{aligned} p(\theta_{k-1} = i | x_{1:k}, \lambda) &= \frac{p(x_{1:k}, \theta_{k-1} = i | \lambda)}{p(x_{1:k} | \lambda)} = \frac{p(x_{1:k}, \theta_{k-1} = i | \lambda)}{\sum_{j=1}^s p(x_{1:k}, \theta_{k-1} = j | \lambda)} \\ &= \frac{\alpha_{k-1}(i)}{\sum_{j=1}^s \alpha_{k-1}(j)}. \end{aligned} \quad (6.24)$$

Se substituirmos $p(x_k | x_{1:k-1}, \theta_{k-1} = i, \lambda)$ pela verossimilhança

$$\bar{p}(x_k | x_{1:k-1}, \theta_{k-1} = i, \lambda) = \frac{w_k^{(i)}}{\sum_{j=1}^s w_k^{(j)}} \quad (6.25)$$

com $w_k^{(i)}$ dado por (3.46) e $\alpha_k(i)$ por $\tilde{\alpha}_k(i)$ nas equações de (6.22) e (6.24), uma estimativa de $p(\theta_{k-1} = i | x_{1:k}, \lambda)$ pode ser obtida por

$$p(\theta_{k-1} = i | x_{1:k}, \lambda) \approx \frac{\tilde{\alpha}_{k-1}(i)}{\sum_{j=1}^s \tilde{\alpha}_{k-1}(j)}. \quad (6.26)$$

Com (6.26), temos

$$\begin{aligned} \mu_k(i) &= E[1_{\{\theta_k=i\}} | x_{1:k}, \lambda] = p(\theta_k = i | x_{1:k}, \lambda) = \sum_{j=1}^s p_{ji} p(\theta_{k-1} = j | x_{1:k}, \lambda) \\ &\approx \sum_{j=1}^s p_{ji} \tilde{\alpha}_{k-1}(j). \end{aligned} \quad (6.27)$$

A partir de (4.29), uma estimativa de μ_k para cenário 4 pode ser obtida através do filtro de partículas

$$\begin{aligned}\mu_k(i) &= E[1_{\{\theta_k=i\}}|y_{1:k},\lambda] = p(\theta_k = i|y_{1:k},\lambda) \\ &\approx \frac{\sum_{l=1}^{N_p} 1_{\{\theta_k^{(l)}=i\}}}{N_p},\end{aligned}\quad (6.28)$$

sendo N_p o número total de partículas e $\theta_k^{(l)}$ o estado de Markov da partícula l no instante k .

No cenário 3, como o estado contínuo x_k é observável teremos o vetor observação dado por $O_{1:k} = x_{1:k}$ e o conjunto de matrizes de segundo momento por

$$X_i^{k,t} = E[x_{k+t}x'_{k+t}1_{\{\theta_{k+t}=i\}}|x_{1:k},\lambda], \forall i \in S. \quad (6.29)$$

Se $t = 0$, as matrizes de segundo momento, para $\forall i \in S$, se resumem a

$$\begin{aligned}X_i^{k,0} &= E[x_kx'_k1_{\{\theta_k=i\}}|x_{1:k},\lambda] \\ &= E[x_kx'_k|\theta_k = i, x_{1:k},\lambda]E[1_{\{\theta_k=i\}}|x_{1:k},\lambda] \\ &= x_kx'_kE[1_{\{\theta_k=i\}}|x_{1:k},\lambda].\end{aligned}\quad (6.30)$$

Como

$$E[1_{\{\theta_k=i\}}|x_{1:k},\lambda] = p(\theta_k = i|x_{1:k},\lambda) = \mu_k(i), \quad (6.31)$$

a equação (6.30) é equivalente à

$$X_i^{k,0} = \mu_k(i)x_kx'_k. \quad (6.32)$$

A partir de (4.33), estimativas das matrizes do segundo momento para o cenário 4 podem ser obtidas através do filtro de partículas

$$\begin{aligned}X_i^{k,0} &= E[x_kx'_k|\theta_k = i, y_{1:k},\lambda] \\ &\approx \frac{\sum_{l=1}^{N_p} [m_{k|k}^{(l)}(m_{k|k}^{(l)})' + P_{k|k}^{(l)}]1_{\{\theta_k^{(l)}=i\}}}{\sum_{h=1}^{N_p} 1_{\{\theta_k^{(h)}=i\}}},\end{aligned}\quad (6.33)$$

sendo N_p , $m_{k|k}^{(h)}$ e $P_{k|k}^{(h)}$, o número total de partículas, a estimativa e a variância do estado contínuo no tempo k da partícula h , respectivamente.

Como supomos acesso ao estado contínuo x_k , cenário 3, aplicaremos controle por realimentação de estados dado por

$$u_k = K^k x_k, \quad (6.34)$$

para cada instante de tempo k .

No cenário 4, aplicaremos o princípio da equivalência à certeza e ao invés de utilizarmos o estado contínuo x_k , que é inacessível neste caso, no controle por realimentação, utilizaremos $\hat{x}_k = E[x_k|y_{1:k}, \lambda]$, cuja a estimativa (4.30) é obtida através do filtro de partícula

$$\begin{aligned}\hat{x}_k &= E[x_k|y_{1:k}, \lambda] \\ &\approx \frac{\sum_{l=1}^{N_p} m_{k|k}^{(l)}}{N_p}.\end{aligned}\tag{6.35}$$

A ação de controle no instante k neste cenário é dada por

$$u_k = K^k \hat{x}_k,\tag{6.36}$$

para cada instante de tempo k

Podemos perceber que tanto em (6.34) quanto em (6.36) a ação de controle u_k depende do vetor de realimentação K que, por sua vez, independe do estado de Markov θ_k .

Segundo (Vargas, 2004) temos que para cada $t = 0, \dots, N-1$, a relação

$$X_i^{k, t+1} = \mathcal{J}^{k, t}(X^{k, t}) + \Psi_i^{k, t}, \forall i \in S\tag{6.37}$$

é válida com $\mathcal{J}$ e $\Psi_i^{k, t}$ apresentados em (6.15) e (6.20), respectivamente. Pelo teorema 3.1 de (Vargas, 2004) o conjunto de vetores de realimentação $\mathbf{K} = \{K^{k, 0}, \dots, K^{k, N-1}\}$ que minimiza o custo satisfaz

$$\sum_{i \in S} [(R_i + F_i' \mathcal{E}_i(L^{k, t+1}) F_i) K^{k, t} + F_i' \mathcal{E}_i(L^{k, t+1}) A_i] X_i^{k, t} = 0, \forall t = 0, \dots, N\tag{6.38}$$

com $X_i^{k, t}$ e $L_i^{k, t}$ definidos conforme (6.37) e (6.17).

6.2.3 Método Variacional

A solução das equações matriciais acopladas (6.17), (6.18), (6.37) e (6.38) representa um problema complexo. Neste trabalho proveremos, adiante, um método variacional apresentado em (do Val e Basar, 1999; Vargas, 2004) para encontrar o vetor de realimentação que minimiza o funcional de custo para os cenários 3 e 4. Definimos $G^k(\eta) := \{G^{k, 0}(\eta), \dots, G^{k, N}(\eta)\}$ como uma sequência de controle. A partir desta sequência temos que

$$A_{i, \eta}^{k, t} := A_i + B_i G_i^{k, t}(\eta), \quad \Lambda_{i, \eta}^{k, t} := R_i + F_i' \mathcal{E}_i(L^{k, t}(\eta)) F_i, \forall i \in S,\tag{6.39}$$

que associado aos operadores $\mathcal{J}$ e $\mathcal{L}$ geram

$$\mathcal{J}_{i,\eta}^{k,t}(\phi) = \sum_{j=1}^n p_{ji} A_{i,\eta}^{k,t} \phi_j (A_{i,\eta}^{k,t})', \forall i \in S \quad (6.40)$$

$$\mathcal{L}_{i,\eta}^{k,t}(\phi) = (A_{i,\eta}^{k,t})' \mathcal{E}_i(\phi) A_{i,\eta}^{k,t}, \forall i \in S \quad (6.41)$$

$$(6.42)$$

Um algoritmo que envolve cálculos recursivos é apresentado a seguir:

Passo 1: Grave o contador de iterações $\eta = 0$. Escolha a sequência inicial $G^k(0)$;

Passo 2: Para $t = 1, \dots, N$, encontre $X^{k,t}(\eta)$, solução do seguinte conjunto de equações:

$$X_i^{k,t}(\eta) = \mathcal{J}_{i,\eta}^{k,t}(X^{k,t-1}(\eta)) + \Psi_i^{k,t-1}, \forall i \in S \quad (6.43)$$

com $X_i^{k,0}$ igual à (6.32) no cenário com observação do estado contínuo ou igual à (6.33) no cenário sem observação dos estados. Faça $\eta = \eta + 1$ e $t = N - 1$;

Passo 3: Encontre $G^{k,t}$ definido por

$$\sum_{i \in S} [(R_i + F_i' \mathcal{E}_i(L^{k,t+1}(\eta)) F_i) G^{k,t} + F_i' \mathcal{E}_i(L^{k,t+1}(\eta)) A_i] X_i^{k,t}(\eta - 1) = 0. \quad (6.44)$$

Calcule $L^{k,t}(\eta)$ e $w^{k,t}(\eta)$, através de

$$L_i^{k,t}(\eta) = Q_i + (G^{k,t}(\eta))' R_i G^{k,t}(\eta) + \mathcal{L}_{i,\eta}^{k,t}(L^{k,t+1}(\eta)), \forall i \in S \quad (6.45)$$

$$w_i^{k,t}(\eta) = \mathcal{E}_i(w^{k,t+1}) + \text{tr}\{\mathcal{E}_i(L^{k,t+1}(\eta)) B_i B_i'\}, \forall i \in S \quad (6.46)$$

com $L_i^{k,N}(\eta) = V_i$ e $w_i^{k,N}(\eta) = 0$, $\forall i \in S$. Grave $t = t - 1$; se $t \leq 0$, retorne ao início do **Passo 3**;

Passo 4: O critério de parada deve ser baseado na variação do custo entre $J^{k,N}(\eta)$ e $J^{k,N}(\eta - 1)$. Se o critério de parada não for satisfeito, então retorne ao início do **Passo 2**.

Vide Vargas (2004) para maiores detalhes como a prova de convergência deste método.

6.3 Critério de Comparação

Para gerar uma análise comparativa da aplicação controle via horizonte retrocedente nos diversos cenários podemos calcular o custo real no horizonte N a partir do instante k para $0 \leq k \leq T - N$

$$Custo^{k,N} := \sum_{\ell=0}^{N-1} (x'_{k+\ell} Q(\theta_{k+\ell}) x_{k+\ell} + u'_{k+\ell} R(\theta_{k+\ell}) u_{k+\ell}) + x'_{k+N} V(\theta_{k+N}) x_{k+N} \quad (6.47)$$

com T representando o número máximo de amostras.

Se definirmos o custo associado ao cenário 1 como sendo padrão, podemos comparar os custos associados à não observação de estados, aplicando uma determinada técnica de filtragem/controle, lembrando-se de utilizar a mesma realização da cadeia para todos os casos. Com isso podemos verificar quão distante este custo está do custo padrão, quanto mais próximo deste custo, melhor será a eficiência desta técnica.

6.4 Aplicações de Controle

Dois exemplos ilustrativos serão utilizados na comparação entre os cenários, o modelo linear de uma junta do *European Robot Arm (ERA)* (Yang e Blanke, 2000) modificado em (Vargas, 2004) e o exemplo 6.4 exemplo extraído de Ji e Chizeck (1990b). Nesta última, alteramos a matriz de transição $\mathbb{P}$ para verificar o comportamento do conjunto controle-filtro.

6.4.1 O Braço do Robô

O modelo linear de uma junta do *European Robot Arm (ERA)* (Yang e Blanke, 2000) adaptado por Vargas (2004), adotando $x' = [\Omega, \dot{\Omega}, \epsilon, \dot{\epsilon}]$ composto pelo ângulo Ω e velocidade angular $\dot{\Omega}$ da junta do eixo inercial e pelo ângulo ϵ e velocidade angular $\dot{\epsilon}$ da junta do eixo de saída, é descrito por

$$x(k+1) = A_i x(k) + B_i w(k) + F_i u(k) \quad (6.48)$$

em que as matrizes A_i e F_i são definidas conforme

$$A_i = \begin{bmatrix} 1 & 0,05 & a_1^i & a_2^i \\ 0 & 1 & a_3^i & a_4^i \\ 0 & 0 & a_5^i & a_6^i \\ 0 & 0 & a_7^i & a_8^i \end{bmatrix}, \quad F_i = \begin{bmatrix} f_1^i & f_1^i \\ f_2^i & f_2^i \\ f_3^i & f_3^i \\ f_4^i & f_4^i \end{bmatrix}, \quad \forall i \in S,$$

nos quais os valores $a_1^i, \dots, a_8^i, f_1^i, \dots, f_4^i$ são obtidos conforme Tabela 6.1.

Em Vargas (2004) adotou-se

$$Q_i \equiv \begin{bmatrix} 90.000 & 0 & 300 & 300 \\ 0 & 4 & 10 & 200 \\ 300 & 10 & 26 & 501 \\ 300 & 20 & 501 & 10.001 \end{bmatrix}, \quad R_i \equiv \begin{bmatrix} 1,02 & 0 \\ 0 & 1,02 \end{bmatrix}, \quad \forall i \in S,$$

$V_i \equiv I_4$ e $B_i \equiv 0,0707I_4, \forall i \in S$; $x_0 = [0,8808 \quad 1,0902 \quad -0,4943 \quad 1,4825]'$; $N = 5$; $\mu_0 =$

Tabela 6.1: Valores dos parâmetros sujeitos a saltos markovianos para robô ERA.

Parâmetros	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
a_1^i	1,3864	1,8277	1,3864	1,8277	1,3864	1,8277
a_2^i	0,028	0,0449	0,028	0,0449	0,028	0,0449
a_3^i	29,2528	3,2249	29,2528	3,2249	29,2528	3,2249
a_4^i	1,3864	1,8277	1,3864	1,8277	1,3864	1,8277
a_5^i	-0,6453	-0,9984	-0,6453	-0,9984	-0,6453	-0,9984
a_6^i	0,0168	0,0009	0,0168	0,0009	0,0168	0,0009
a_7^i	-34,7158	-3,5259	-34,7158	-3,5259	-34,7158	-3,5259
a_8^i	-0,6453	-0,9984	-0,6453	-0,9984	-0,6453	-0,9984
f_1^i	-0,0009	-0,0012	-0,0011	-0,0015	-0,0001	-0,0001
f_2^i	-0,0231	-0,0107	-0,0277	-0,0129	-0,0028	-0,0013
f_3^i	0,0008	0,0011	-0,0046	0,001	0,0001	0,0001
f_4^i	0,0176	0,0019	0,0211	0,0023	0,0021	0,0002

$[0,9 \ 0,05 \ 0,05 \ 0 \ 0 \ 0]'$, além de estabelecer a matriz de transição de probabilidade

$$\mathbb{P} = \begin{bmatrix} 0,9 & 0,07 & 0,03 & 0 & 0 & 0 \\ 0,7 & 0,1 & 0,05 & 0,15 & 0 & 0 \\ 0,85 & 0,05 & 0,1 & 0 & 0 & 0 \\ 0,6 & 0,05 & 0,05 & 0,25 & 0,05 & 0 \\ 0,6 & 0 & 0 & 0 & 0,35 & 0,05 \\ 0,5 & 0 & 0 & 0,05 & 0,15 & 0,3 \end{bmatrix}.$$

Para que pudéssemos estender o problema para todos os cenários, adicionamos a observação y_k ao sistema

$$y(k) = C_i x(k) + D_i v(k) + G_i u(k) \quad (6.49)$$

em que as matrizes $C_i \equiv I_4$, $D_i \equiv 0.01I_4$ e $G_i \equiv 0_{4 \times 1}$, $\forall i \in S$.

Com objetivo de obter dados estatísticos confiáveis obtivemos 50 simulações independentes com 50 amostras em cada cenário. O número máximo de simulações será representado por T_{sim} . De posse deste conjunto de dados calculamos, para cada cenário e $0 \leq k \leq T$, a estimativa média de x_k

$$\bar{x}_k = \frac{1}{T_{sim}} \sum_{\ell=1}^{T_{sim}} \hat{x}_k(\ell), \quad (6.50)$$

a ação de controle média

$$\bar{u}_k = \frac{1}{T_{sim}} \sum_{\ell=1}^{T_{sim}} u_k(\ell), \quad (6.51)$$

o funcional de custo médio

$$\bar{J}_k^N = \frac{1}{T_{sim}} \sum_{\ell=1}^{T_{sim}} J_k^N(\ell) \quad (6.52)$$

e o custo médio para $0 \leq k \leq T - N$

$$\overline{Custo}_k^N = \frac{1}{T_{sim}} \sum_{\ell=1}^{T_{sim}} Custo_k^N(\ell) \quad (6.53)$$

em que $r(\ell)$ representa o valor da variável r na simulação ℓ .

Para o horizonte N igual a 5, os valores médios do ângulo da junta inercial Ω_k e da ação de controle u_k podem ser visualizados na figura 6.1.

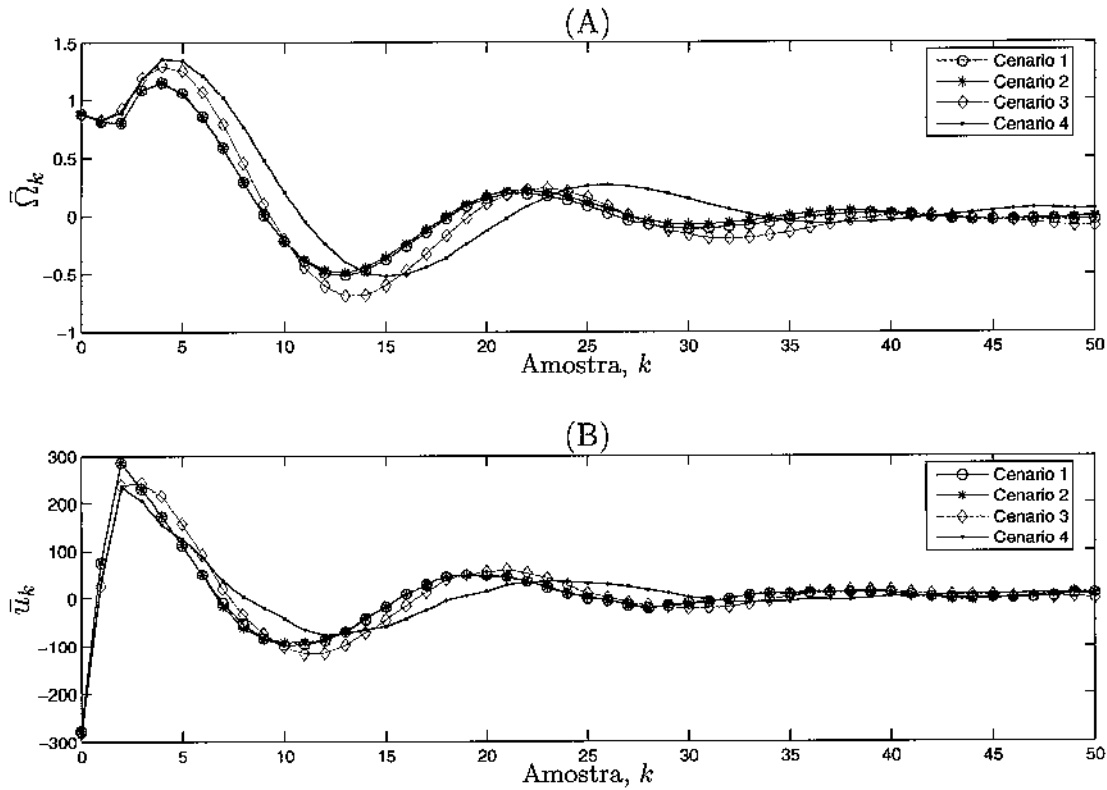


Figura 6.1: (A) Evolução do ângulo médio da junta inercial Ω_k ; (B) Evolução da ação de controle média $\bar{u}_k$ para $T_{sim} = 50$.

Na figura 6.1 podemos perceber que o tempo de estabilização é menor nos cenários 1 e 2 com observação do estado de Markov θ_k do que nos cenários 3 e 4 em que este estado não é observado. O cenário 4, por sua vez, apresenta o maior tempo de estabilização.

Já os valores médios das estimativas do funcional de custo J_k^N e do custo realizado $Custo_k^N$ podem ser vistos na figura 6.2.

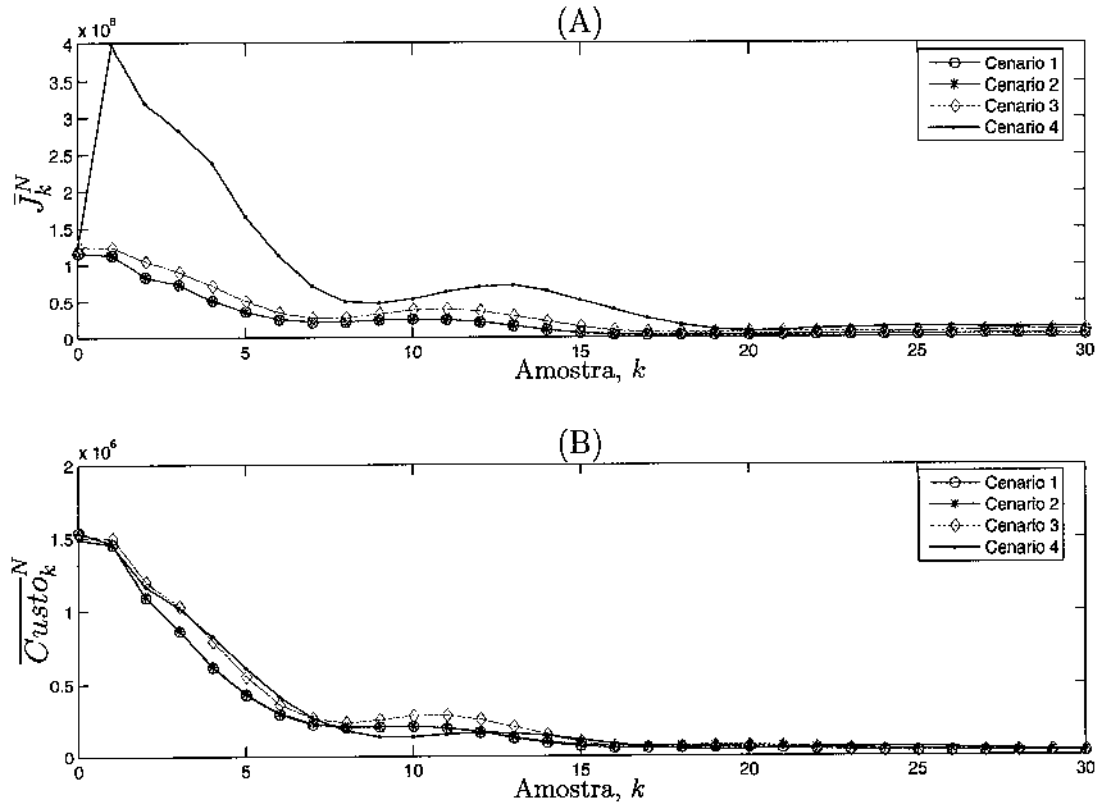


Figura 6.2: (A) Evolução da estimativa média do custo funcional J_k^N ; (B) Evolução da média do custo real realizado $Custo_k^N$ para $T_{sim} = 50$.

O cenário 1 apresentou o menor custo real, seguido pelos cenários 2 e 3. O cenário com maior custo real foi o 4. Isso significa que quanto menor for o acesso à informação dos estados, maior o custo real associado, observando que ter acesso ao estado de Markov acarreta um custo menor do que ter acesso ao estado contínuo.

O funcional de custo das primeiras amostras referente ao cenário 4 se diferencia dos outros funcionais, pois apresenta um custo superior aos demais.

6.4.2 Sensibilidade do Conjunto Filtro-Controlle a Alteração da $\mathbb{P}$

A seguir, apresentamos o exemplo 6.4 extraído de Ji e Chizeck (1990b) com o objetivo de ilustrar o comportamento do conjunto controle-filtro ao alterarmos a matriz de transição $\mathbb{P}$.

Considere um sistema com dois modos distintos de operação, sendo que o primeiro é controlável e representa as condições normais de operação e o segundo, além de não controlável, representa uma falha de atuador.

Os dados do sistema são

$$A_1 = \begin{bmatrix} 2 & 2 \\ 3 & 1 \end{bmatrix}, F_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0,25 & 0 \\ 0 & 0,2 \end{bmatrix}, Q_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, R_1 = 1, V_1 = 0,$$

$$A_2 = \begin{bmatrix} 1 & 0 \\ 0,5 & 1 \end{bmatrix}, F_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,4 \end{bmatrix}, Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_2 = 1, V_2 = 0.$$

com $N = 5$, $x'_0 = [-10 \ 6]$ e $\mu'_0 = [0,85 \ 0,15]$.

O conjunto de matrizes de transição testadas foram

$$\mathbb{P}_1 = \begin{bmatrix} 0,9 & 0,1 \\ 0,8 & 0,2 \end{bmatrix}, \mathbb{P}_2 = \begin{bmatrix} 0,5 & 0,5 \\ 0,5 & 0,5 \end{bmatrix} \text{ e } \mathbb{P}_3 = \begin{bmatrix} 0,1 & 0,9 \\ 0,2 & 0,8 \end{bmatrix}.$$

Para que pudéssemos comparar a aplicação do controle via horizonte retrocedente, realizamos uma única simulação para cada cenário.

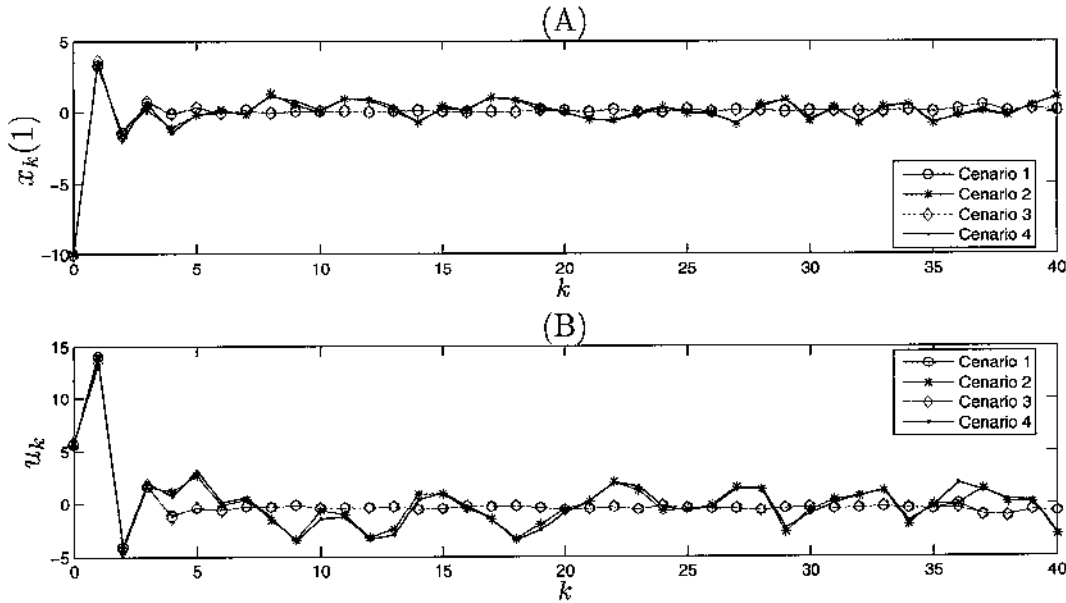


Figura 6.3: (A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_1$.

Com a matriz $\mathbb{P}_1$, o sistema foi capaz de regular e após algumas iterações, o custo tendeu a zero em todos os cenários. Através da evolução da primeira componente do vetor de estados

$x_k(1)$, apresentada na letra (A) da figura 6.3, podemos verificar que quando se tem acesso ao estado de Markov temos uma melhor regulação se comparada à observação do estado contínuo ou à não observação total dos estados.

Em relação a ação de controle, temos que a observabilidade do estado de Markov acarreta um menor esforço de controle se comparado à observabilidade do estado contínuo como pode ser visto na letra (B) da figura 6.3. Além disso, a ação de controle u_k nos cenários 1 e 2 só não será nula, quando o modo de funcionamento do sistema estiver operando normalmente, pois sabemos que no modo de falha o sistema não é controlável.

Na matriz $\mathbb{P}_2$, embora o sistema tenha regulado, houve uma perda qualitativa na regulação, devido ao aumento da região de excursão tanto dos estados quanto da ação de controle. A evolução de $x_k(1)$ e do esforço de controle $\bar{u}_k$ podem ser visualizados na figura 6.4.

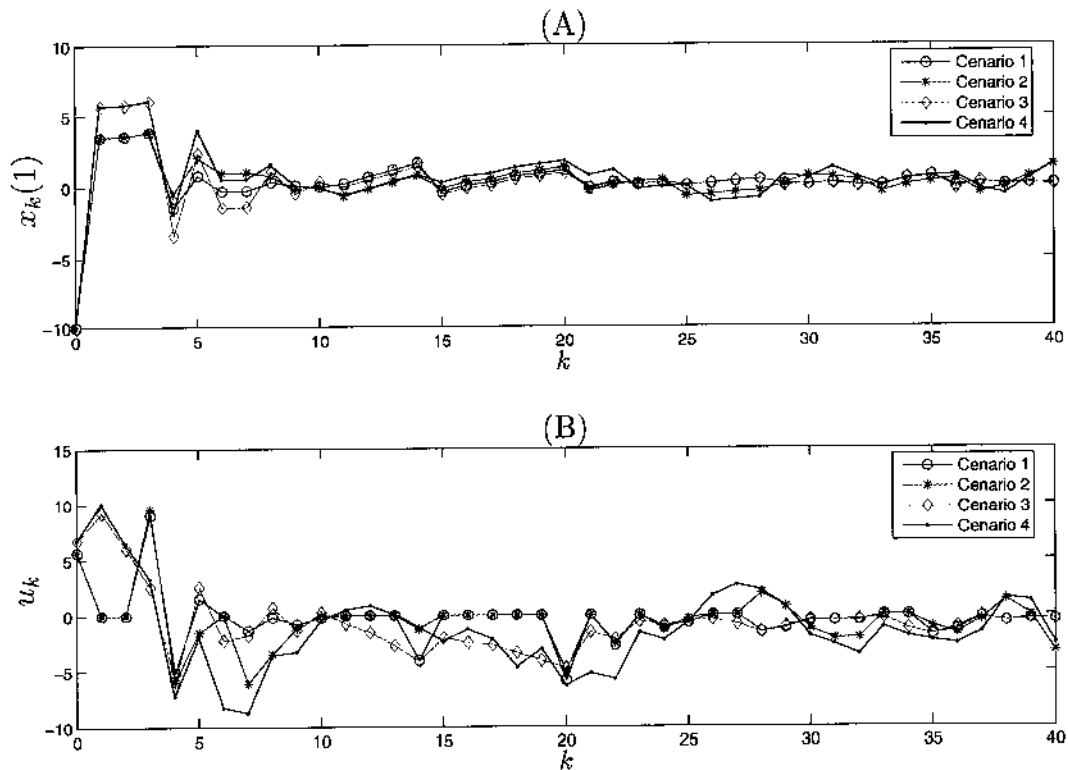


Figura 6.4: (A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_2$.

Ao utilizarmos a matriz $\mathbb{P}_3$, detectamos uma maior deterioração da regulação. A evolução de $x_k(1)$ e do esforço de controle u_k podem ser verificados na figura 6.5.

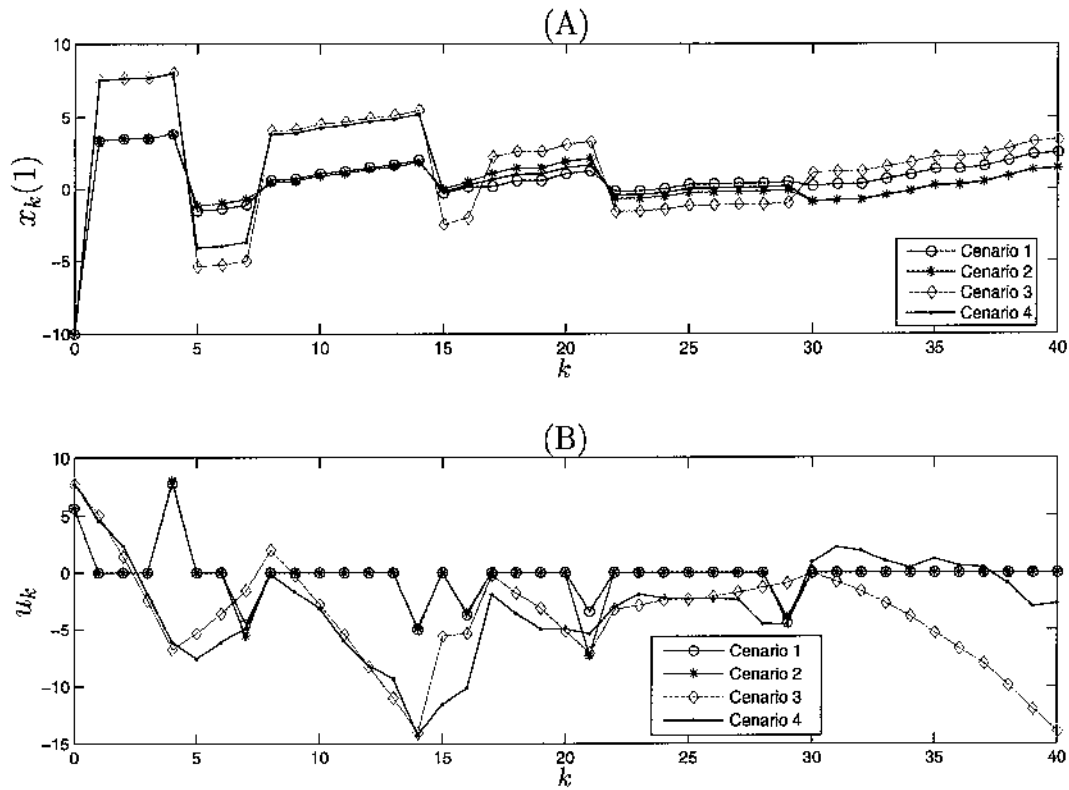


Figura 6.5: (A) Evolução da primeira componente do vetor de estados $x_k(1)$; (B) Evolução da ação de controle u_k com $\mathbb{P}_3$.

Analisando as figuras 6.3, 6.4 e 6.5 podemos perceber que quanto menor a probabilidade do sistema estar no modo de operação normal, pior é a regulação. Quando alteramos a matriz de transição de $\mathbb{P}_1$ até $\mathbb{P}_3$, o sistema perde controlabilidade no sentido estocástico, porque o estado controlável fica cada vez menos freqüente. Se a probabilidade de ocorrência do modo normal tender a zero, o sistema tenderá à instabilidade, pois o modo de falha não é controlável.

Capítulo 7

Conclusões

O estudo das técnicas de filtragem aplicadas a sistemas lineares sujeitos a saltos Markovianos sem observação de estados teve uma grande evolução a partir do uso dos métodos de Monte Carlo para cadeia de Markov, pois foi possível obter estimativas dos estados do sistema condicionadas ao conjunto disponível de observações ruidosas através de algoritmos estocásticos com custo linear pelo número de observações.

Ao estudarmos a bibliografia, nos deparamos com alguns algoritmos inoperantes ou que só funcionavam para uma determinada aplicação. Deixando de lado estes algoritmos, estudamos e analisamos nesta dissertação as principais técnicas de filtragem via métodos de Monte Carlo aplicadas a sistemas lineares com saltos Markovianos.

Os primeiros algoritmos (Logothetis e Krishnamurthy, 1999) eram determinísticos e utilizavam a combinação de dois estimadores, um associado ao estado de Markov e outro ao estado contínuo. Nos algoritmos em *batch*, os *smoothers* eram considerados e nos *on-line*, os filtros. O *smoother* ou filtro de Kalman (Anderson e Moore, 1979) era utilizado para calcular as estimativas do estado contínuo a partir da observação dos estados de Markov e *smoother* ou filtro do modelo de cadeia escondida (Rabiner, 1989), as estimativas do estado de Markov a partir da observação dos estados de contínuos. Os dois estimadores eram combinados através do método de coordenada descendente.

A passagem dos algoritmos determinísticos para os estocásticos ficou a cargo das técnicas de amostragem baseadas nos métodos de Monte Carlo para cadeia de Markov cuja evolução e aplicações são o cerne desta dissertação. A síntese de comparação entre os algoritmos de filtragem sem observação baseada nos resultados obtidos das simulações e apresentados nos capítulos 5 e 6, assim como alguns comentários e observações, serão apresentados a seguir.

Os algoritmos associados ao *Data Augmentation*, primeiros algoritmos estocásticos, são indicados para sistemas que apresentam um grande número de transições entre os estados de

Markov e que, além do mais, sofrem ação de perturbações não-modeladas. Isso se deve à alta capacidade de adaptação destes algoritmos, que conseguem minimizar rapidamente o erro associado a estimativa no regime transiente. Como estes algoritmos possuem uma elevada variância do erro quadrático em cada instante k , as estimativas de uma dada execução podem ser bem diferente das de outra execução.

Se, além do sistema ter sido identificado corretamente, houver uma tendência deste de permanecer em algum dos estados de Markov, minimizando as transições de estados, o sistema tenderá a sair do regime transiente para o permanente e nestes casos, o Mínimo Erro Quadrático (MMSE) (Doucet e Andrieu, 2001), os algoritmos associados ao *maximum a posteriori* marginal (MMAP) (Doucet e Andrieu, 2001) e o filtro de partículas (Doucet et al., 2001) devem ser considerados no cálculo das estimativas dos estados do sistema.

Ao compararmos o MMSE aos algoritmos MMAP, podemos perceber que o primeiro consegue suavizar e minimizar o erro associado as estimativas do estado contínuo, pois suas estimativas são calculadas através da média das estimativas de todas as iterações, ao contrário dos algoritmos MMAP que consideram as estimativas obtidas da última iteração. Além disso, o MMSE gera uma estimativa da distribuição de probabilidade de θ_k para $\forall k$, ao contrário do MMAP, que recupera uma sequência de estados de Markov. Esta sequência pode não representar a sequência real e sim uma sequência relacionada a um máximo local da distribuição *a posteriori* $p(\theta_{1:T}|y_{1:T})$. A escolha de uma única sequência provoca uma maior variação do erro ao longo das amostras e inviabiliza o uso das estimativas dos estados de Markov no cálculo das ações de controle, pois uma estimativa errada poderá levar o sistema à instabilidade.

O filtro de partículas, o único representante recursivo que pode ser aplicado em tempo real, apresenta tanto uma boa estimativa do estado contínuo, quanto uma boa estimativa da distribuição de probabilidade do estado de Markov θ_k para $\forall k$. Fica evidente que o filtro de partículas não consegue obter erros menores do que o Mínimo Erro Quadrático, pois os filtros em *batch* têm acesso a todo o conjunto de observações $O_{1:T}$, ao contrário do filtro *on line*, que num dado instante k , só tem acesso às observações até este instante $O_{1:k}$.

Para que o filtro de partículas (Doucet et al., 2001) pudesse ser desenvolvido foi necessário a evolução dos algoritmos, partindo da Maximização do Valor Esperado (Logothetis e Krishnamurthy, 1999) até o Mínimo Erro Quadrático (Doucet e Andrieu, 2001). No início, as estimativas eram baseadas na distribuição conjunta *a posteriori* $p(\theta_{1:k}, x_{1:k}|y_{1:k}, \lambda)$ e no uso de dois estimadores, um associado ao estado contínuo e outro ao estado de Markov. Com o tempo, foi se percebendo que o estado contínuo era uma consequência do estado

de Markov e passou-se a calcular as estimativas baseadas na distribuição marginal *a posteriori* $p(\theta_{1:k}|y_{1:k},\lambda)$. Esta evolução foi associada ao aperfeiçoamento da decomposição das distribuições de probabilidade conjunta *a posteriori*.

Tendo o estado de Markov como elemento chave da filtragem, foi possível desenvolver o filtro de partículas que simula inúmeras trajetórias simultaneamente e através do processo de seleção em que as partículas com baixos pesos são eliminadas e as com alto pesos são replicadas, é possível calcular as estimativas do estado contínuo e da distribuição de probabilidade do estado de Markov.

Com estas informações foi possível aplicar o controle via horizonte retrocedente para regular o sistema, além de verificar a variação do desempenho da regulação para os diversos cenários de observação. Detectamos que quanto menor o acesso às informações dos estados pior é o desempenho da regulação e que os melhores desempenhos foram atingidos com o acesso direto ao estado de Markov.

Uma das dificuldades encontradas ao longo do desenvolvimento desta dissertação foi a falta de uma notação comum entre os diversos artigos encontrados na literatura. Para contornar este problema, tentamos unificar da melhor maneira possível as notações, levando em consideração tanto as variáveis dos filtros quanto as de controle.

Duas propostas devem ser consideradas para trabalhos futuros, a primeira diz respeito a observabilidade e a segunda, a controlabilidade de Sistemas Lineares com Saltos Markovianos numa visão estocástica. Como C foi igual à matriz identidade para todos os estados de Markov, as aplicações de filtragem apresentadas se resumem a um problema de observação completa dos estados contínuos com ruído. Além da necessidade do desenvolvimento analítico dos filtros estocásticos, faz-se necessário verificar o comportamento dos filtros para C diferente da matriz identidade, tal como matrizes cujos elementos da diagonal principal são iguais a zero. Em relação a controlabilidade, deve-se determinar as condições necessárias e/ou suficientes para estabilizar e controlar o sistema, além de entender como filtro e o controle se relacionam.

Apêndice A

Algoritmo de Metropolis

O algoritmo de Metropolis é de grande simplicidade e poder e pode ser usado para amostrar essencialmente qualquer função de densidade não importando sua complexidade analítica ou dimensão. Por *sistema* nós queremos dizer aqui simplesmente um ponto x no espaço Ω que pode ser dito como uma possível descrição de um problema físico. Por *cinética* nós queremos representar a transição estocástica que governa a evolução do sistema: a função de densidade de probabilidade $K(X|Y)$ representa a evolução do sistema, que se sabe que está em Y , para a proximidade de X . K pode ser descrito como o modelo do processo físico pelo qual as mudanças do sistema ocorrem. Uma das condições para que o sistema caminhe para o equilíbrio e lá fique é do sistema mover, na média, tanto para uma vizinhança específica de X a partir da vizinhança de Y quanto de mover na direção reversa. Se a densidade de probabilidade de observar o sistema próximo de X em equilíbrio é $p(X)$, então a cinética precisa satisfazer:

$$K(X|Y)p(Y) = K(Y|X)p(X) \quad (\text{A.1})$$

Essa relação é chamada de equilíbrio detalhado. Em se tratando de um sistema físico, nós usualmente assumimos que $K(X|Y)$ é conhecido e nós temos a tarefa de encontrar $p(X)$. O algoritmo de Metropolis reverte isso: nós temos a tarefa de encontrar a correta e conveniente cinética que irá equilibrar o sistema para que uma dada $p(X)$ acabe representando a chance de observar o sistema próximo de X . Isso acaba sendo extremamente fácil e elegante com o dispositivo dado por Metropolis. Transições são propostas de Y para X' usando simplesmente qualquer distribuição $T(X'|Y)$. Então comparando $p(X')$ com $p(Y)$ e levando em conta também T , o sistema se move ou para X' (movimento aceito) ou retorna para Y (movimento rejeitado). A aceitação do movimento ocorre com probabilidade $A(X'|Y)$, que deve ser calculada para satisfazer o equilíbrio detalhado. Nós temos que

$$K(X|Y) = A(X|Y)T(X|Y). \quad (\text{A.2})$$

O equilíbrio detalhado requer que

$$A(X|Y)T(X|Y)p(Y) = A(Y|X)T(Y|X)p(X). \quad (\text{A.3})$$

Dado uma pdf $p(X)$, onde X é um vetor de muitas dimensões, a técnica de Metropolis estabelece um caminho aleatório cujos passos são designados de tal maneira que repetido várias vezes, a distribuição de X 's é $p(X)$. Suponha que $X_1, X_2, X_3, \dots, X_n$ são passos num caminho aleatório. Cada um dos X 's é uma variável aleatória e possui uma probabilidade associada a $\Phi_1(X), \Phi_2(X), \Phi_3(X),$

$\dots, \Phi_n(X)$, em que $\Phi_1(X)$ pode ser qualquer distribuição de X . O $\Phi_n(X)$ tem a propriedade que assintoticamente

$$\lim_{n \rightarrow \infty} \Phi_n(X) = p(X). \quad (\text{A.4})$$

A cada passo no caminho aleatório existe uma densidade de transição $T(X|Y)$ que é a função de densidade de probabilidade para um movimento de uma tentativa de mover de X para Y . A $T(X|Y)$ é normalizada de maneira que

$$\int T(X|Y)dX = 1 \quad (\text{A.5})$$

para todos os valores de Y . A quantidade $q(X|Y)$ é definida como

$$q(X|Y) = \frac{T(Y|X)p(X)}{T(X|Y)p(Y)} \geq 0, \quad (\text{A.6})$$

onde nós explicitamente assumimos que é possível mover de X para Y se é possível mover de Y para X e vice versa. De $q(X|Y)$ a probabilidade de aceitação do movimento pode ser calculado; uma possibilidade usada frequentemente é

$$A(X|Y) = \min(1, q(X, Y)). \quad (\text{A.7})$$

O algoritmo pode agora ser descrito concretamente. No passo n do caminho aleatório, o valor de X é X_n ; um possível valor para X , X'_{n+1} , é amostrado de $T(X'_{n+1}|X_n)$, e a probabilidade de aceitar X'_{n+1} é computada. Se $q(X'_{n+1}|X_n) > 1$, então $A(X'_{n+1}|X_n) = 1$; Se $q(X'_{n+1}|X_n) < 1$, então $A(X'_{n+1}|X_n) = q(X'_{n+1}|X_n)$, onde

$$q(X'_{n+1}|X_n) = \frac{T(X_n|X'_{n+1})p(X'_{n+1})}{T(X'_{n+1}|X_n)p(X_n)}. \quad (\text{A.8})$$

Com probabilidade $A(X'_{n+1}|X_n)$ nós temos que $X_{n+1} = X'_{n+1}$; caso contrário temos que $X_{n+1} = X_n$. Isso é, se $A(X'_{n+1}|X_n) > \xi$, então $X_{n+1} = X'_{n+1}$; caso contrário $X_{n+1} = X_n$. O caminho aleatório gera uma relação recursiva para as funções de densidade, dada por

$$\Phi_{n+1}(X) = \int A(X|Y)T(X|Y)\Phi_n(Y)dY + \int [1 - A(Y|X)]T(T|X)\Phi_n(X)dY. \quad (\text{A.9})$$

Bibliografia

- Anderson, B. D. O. e Moore, J. (1979). *Optimal Filtering*, Prentice-Hall, Englewood Cliffs-NJ.
- Astrom, K. J. (1970). *Introduction to Stochastic Control Theory*, Academic Press.
- Athans, M., Castanon, D., Dunn, K. P., Greene, C. S., Lee, W. H., Sandell, N. R. e Willsky, A. S. (1977). The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method - Part i: Equilibrium flight, *IEEE Transactions on Automatic Control* (22): 768–780.
- Bar-Shalom, Y. e Li, X. R. (1995). *Multitarget-Multisensor Tracking: Principles and Technics*, YBS.
- Bitmead, R. R., Gevers, M. e Wertz, V. (1990). *Adaptive Optimal Control: the thinking man's GPC*, Prentice Hall, Sydney, Australia.
- Bolic, M., Djuric, P. M. e Hong, S. (2004). Comparison of resampling algorithms, *EURASIP Journal on Applied Signal* 15: 2267–2277.
- Camacho, E. F. e Bordons, C. (1999). *Model Predictive Control*, Springer-Verlag, London.
- Costa, E. F. e do Val, J. B. R. (2000). Stability of receding horizon control of Markov jump linear systems without jump observations, *American Control Conference*, Chicago, USA, pp. 4289–4293.
- Costa, O. L. V. e Fragoso, M. D. (1993). Stability results for discrete-time linear systems with Markovian jumping parameters, *Journal of Mathematical Analysis and Applications* 179: 154–178.
- Costa, O. L. V. e Fragoso, M. D. (1995). Discrete-time LQ-optimal control problems for finite Markov jump parameters systems, *IEEE Transactions on Automatic Control* 40: 2076–2088.

- Costa, O. L. V., Fragoso, M. D. e Marques, R. P. (2005). *Discrete-Time Markov Jump Linear Systems*, Springer-Verlag, United States.
- Davis, M. H. A. e Vinter, R. B. (1985). *Stochastic Modelling and Control*, Chapman and Hall.
- do Val, J. B. R. e Basar, T. (1999). Receding horizon control of jump linear systems and macroeconomic policy problem, *Journal of Economic Dynamics and Control* **23**: 1099–1131.
- Doucet, A. e Andrieu, C. (2001). Iterative algorithms for state estimation of jump Markov linear systems, *IEEE Transactions on Signal Processing* **49**(6): 1216–1227.
- Doucet, A., Gordon, N. J. e Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems, *IEEE Transactions on Signal Processing* **49**(3): 613–624.
- Doucet, A., Logothetis, A. e Krishnamurthy, V. (2000). Stochastic sampling algorithms for state estimation of jump Markov linear systems, *IEEE Transactions on Automatic Control* **45**(1): 188–202.
- Ji, Y. e Chizeck, H. J. (1990a). Controllability, stabilizability and continuous-time Markovian jump linear quadratic control, *IEEE Transactions on Automatic Control* **35**(7): 777–788.
- Ji, Y. e Chizeck, H. J. (1990b). Jump linear quadratic Gaussian control: Steady-state solution and testable conditions, *Control-Theory and Advanced Technology* **6**(3): 289–319.
- Khanbaghi, M., Malhame, R. P. e Perrier, M. (2002). Optimal white water and broke recirculation policies in paper mills via jump linear quadratic control, *IEEE Transactions on Automatic Control* **10**(4): 578–588.
- Kwon, W. H., Bruckstein, A. M. e Kailath, T. (1983). Stabilizing state-feedback design via the moving horizon method, *International Journal of Control* **37**: 631–643.
- Logothetis, A. e Krishnamurthy, V. (1999). Expectation maximization algorithms for map estimation of jump Markov linear systems, *IEEE Transactions on Signal Processing* **47**(8): 2139–2156.
- Mayne, D. Q. e Michalska, H. (1990). Receding horizon control of nonlinear systems, *IEEE Transactions on Automatic Control* **35**: 814–824.

- Mazor, E., Averbuck, A., Bar-Shalon, Y. e Dayan, J. (1998). Interacting multiple model methods in target tracking: A survey, *IEEE Trans. Aerosp. Electron. Syst.* **34**: 103–123.
- Mendel, J. M. (1990). *Maximum -Likelihood Deconvolution: A Journey into Model-Based Signal Processing*, Springer-Verlag, New York.
- Mosca, E. (1995). *Optimal, Predictive, and Adaptive Control*, Prentice Hall.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected application in speech recognition, *Proceedings of the IEEE* (2): 257–286.
- Ristic, B., Arulampalam, A. e Gordon, N. (2004). *Beyond the Kalman filter - Particles filters for tracking applications*, Artech House: London.
- Ruanaidh, J. J. K. . e Fitzgerald, W. J. (1996). *Numerical Bayesian Methods Applied to Signal Processing*, Springer-Verlag, New York, NY.
- Salomão, R. (2006). *Filtro de Partículas para Análise de Sinal Geofísico Obtido Durante Perfuração de Poços de Petróleo*, UNICAMP.
- Saridis, G. N. (1983). Intelligent robotic control, *IEEE Transactions on Automatic Control* (28): 547–557.
- Soderstrom, T. (1994). *Discrete-time Stochastic Systems: Estimation and Control*, Prentice Hall.
- Sworder, D. D. e Rogers, R. O. (1983). An LQ-solution to a control problem associated with a solar thermal central receiver, *IEEE Transactions on Automatic Control* **10**(28): 971–978.
- Vargas, A. (2004). *Controle por Horizonte Retrocedente de Sistemas lineares com Saltos Markovianos e Ruído Aditivo*, UNICAMP.
- Yang, Z. e Blanke, M. (2000). The robust control mixer module method for control reconfiguration, *American Control Conference*.