NNT : 2017SACLL009

THÈSE DE DOCTORAT

de

L'UNIVERSITÉ PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : Telecom SudParis

Établissement d'accueil : Telecom SudParis

Laboratoire d'accueil : Services répartis, architectures, modélisation, validation, administration des réseaux, UMR 5157 CNRS

Spécialité de doctorat : Mathématiques appliquées

Ivan GORYNIN

Bayesian state estimation in partially observable Markov processes

PhD dissertation written in English with a French summary

Date de soutenance : 13 décembre 2017

Après avis des rapporteurs : CHRISTOPHE ANDRIEU (Bristol University)
PHILIPPE VANHEEGHE (École centrale de Lille)

Jury de soutenance :	CHRISTOPHE ANDRIEU	(Bristol University) Rapporteur
	PHILIPPE VANHEEGHE	(École Centrale de Lille) Rapporteur
	CHRISTOPHETTE BLANCHET	(École Centrale de Lyon) Examinateur
	EMMANUEL GOBET	(École Polytechnique) Président
	MADALINA OLTEANU	(Université Paris 1) Examinateur
	WOJCIECH PIECZYNSKI	(Telecom SudParis) Directeur de thèse
	EMMANUEL MONFRINI	(Telecom SudParis) Codirecteur de thèse

universite
PARIS-SACLAY

École doctorale
de mathématiques
Hadamard (EDMH)

Titre : Estimation bayésienne dans les modèles de Markov partiellement observés

Mots Clefs : Systèmes non-linéaires cachés, filtrage optimal, inférence paramétrique, systèmes à saut, volatilité stochastique, approximations stochastiques.

Résumé : Cette thèse porte sur l'estimation bayésienne d'état dans les séries temporelles modélisées à l'aide des variables latentes hybrides, c'est-à-dire dont la densité admet une composante discrète-finie et une composante continue. Des algorithmes généraux d'estimation des variables d'états dans les modèles de Markov partiellement observés à états hybrides sont proposés et comparés avec les méthodes de Monte-Carlo séquentielles sur un plan théorique et appliqué. Le résultat principal est que ces algorithmes permettent de réduire significativement le coût de calcul par rapport aux méthodes de Monte-Carlo séquentielles classiques.

Title : Bayesian state estimation in partially observable Markov processes

Keys words : Nonlinear state-space model, jump systems, optimal filter, stochastic volatility, parameter inference, stochastic approximations.

Abstract : This thesis addresses the Bayesian estimation of hybrid-valued state variables in time series. The probability density function of a hybrid-valued random variable has a finite-discrete component and a continuous component. Diverse general algorithms for state estimation in partially observable Markov processes are introduced. These algorithms are compared with the sequential Monte-Carlo methods from a theoretical and a practical viewpoint. The main result is that the proposed methods require less processing time compared to the classic Monte-Carlo methods.

Contents

Acknowledgments

I would like to express my sincere gratitude to my supervisors Prof. Pieczynski and Dr. Monfrini for the continuous support of my PhD study and related research, for their patience, motivation, enthusiasm, and immense knowledge whilst allowing me the room to work in my own way. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better or friendlier supervisors than Prof. Pieczynski and Dr. Monfrini.

Besides my supervisors, I would like to thank Prof. Andrieu and Prof. Vanheeghe for their insightful reports on my work, which incited me to widen my research from various perspectives. I also wish to thank Prof. Gobet who accepted to preside over the thesis committee and who was thorough and rigorous in his mission.

My sincere thanks also go to the rest of my thesis committee: Dr. Blanchet-Scalliet and Dr. Olteanu, for their interest in my researches and encouragement, as well as their questions.

I would like also thank Dr. Petetin for very fruitful discussions that I have had together with him and Prof. Pieczynski and Dr. Monfrini. Last but not the least, I would like to thank the rest of my colleagues in the department CITI (Communications, Images et Traitement de l'Information) in Telecom SudParis: Ms. Bonnet, M. Uro , Dr. Castella, Prof. Desbouvries, Prof. Lehmann, Prof. Douc, Dr. Leclere, Dr. Simon and Prof. Letrou, for their welcome and support.

List of symbols and abbreviations

List of symbols

$\mathbb{N}$	Set of natural numbers
$\mathbb{N}^*$	Set of nonzero natural numbers
$\{1:N\}$	Set of natural numbers ranging from 1 to N inclusive, $N \in \mathbb{N}$
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^*$	Set of nonzero real numbers
$\mathbb{R}_+$	Set of positive real numbers
$\mathbb{R}_+^*$	Set of strictly positive real numbers
$\mathbb{R}^a$	Set of a-dimensional vectors with coefficients in $\mathbb{R}$, $a \in \mathbb{N}^*$
$\mathbb{R}^{a\times b}$	Set of matrices of dimension $a \times b$ with coefficients in $\mathbb{R}$, $a, b \in \mathbb{N}^*$
$\mathcal{S}_{++}^d$	Set of positive definite matrices in $\mathbb{R}^{d\times d}$, $d \in \mathbb{N}^*$
$\mathbb{1}_S$	Indicator function of a set S
$\mathrm{Card}(S)$	The number of distinct elements in a finite set S
$\mathcal{X} \times \mathcal{Y}$	Cartesian product of two sets $\mathcal{X}$ and $\mathcal{Y}$
$\mathcal{F}(\mathcal{X} \to \mathcal{Y})$	Set of functions from $\mathcal{X}$ to $\mathcal{Y}$
$\boldsymbol{M}$	A bold uppercase italic symbol refers to a matrix in $\mathbb{R}^{a\times b}$, where $a, b \in \mathbb{N}^*$
$\mathbf{X}$	A bold uppercase non-italic symbol refers to a vector-valued random variable
$\boldsymbol{f}$, $\boldsymbol{v}$	A bold lowercase italic symbol refers to a vector-valued function or to a column vector in $\mathbb{R}^d$, where $d \in \mathbb{N}^*$
$\mathbf{x}$	A bold lowercase non-italic symbol refers to a realization of a vector-valued random variable
X	A non-bold uppercase non-italic symbol refers to a scalar random variable
z, f	A non-bold lowercase italic symbol refers to a scalar variable or to a scalar-valued function

x	A non-bold lowercase non-italic symbol refers to a realization of a scalar random variable
$\mathbf{X}_{1:N}$	Random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$, $N \in \mathbb{N}$
$\mathbf{x}_{1:N}$	Realizations of random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$, $N \in \mathbb{N}$
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$
$\lvert\boldsymbol{M}\rvert$	The determinant of matrix $\boldsymbol{M}$
$\mathrm{tr}(\boldsymbol{M})$	The trace of matrix $\boldsymbol{M}$
$p(.)$	Probability distribution
$\mathbb{E}\left[.\right]$	Expected value operator
$\mathbb{P}\left[.\right]$	Probability operator
$p_{\boldsymbol{\theta}}(.)$	Probability distribution parameterized by $\boldsymbol{\theta}$
$\mathbb{E}_{\boldsymbol{\theta}}\left[.\right]$	Expected value operator parameterized by $\boldsymbol{\theta}$
$\mathbb{P}_{\boldsymbol{\theta}}\left[.\right]$	Probability operator parameterized by $\boldsymbol{\theta}$
$p(.\lvert.)$	Conditional probability distribution
$\mathbb{E}\left[.\lvert.\right]$	Conditional expected value operator
$\mathbb{P}\left[.\lvert.\right]$	Conditional probability operator
$p_{\boldsymbol{\theta}}(.\lvert.)$	Conditional probability distribution parameterized by $\boldsymbol{\theta}$
$\mathbb{E}_{\boldsymbol{\theta}}\left[.\lvert.\right]$	Conditional expected value operator parameterized by $\boldsymbol{\theta}$
$\mathbb{P}_{\boldsymbol{\theta}}\left[.\lvert.\right]$	Conditional probability operator parameterized by $\boldsymbol{\theta}$
$\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{S}\right)$	The normal probability distribution defined by the mean vector $\boldsymbol{\mu}$ and the variance matrix $\boldsymbol{S}$
$\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{S}\right)$	The value at $\boldsymbol{x}$ of the normal probability density function defined by the mean vector $\boldsymbol{\mu}$ and the variance matrix $\boldsymbol{S}$: $\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{S}\right) = \lvert 2\pi\boldsymbol{S}\rvert^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{S}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$

List of abbreviations

ARIMA	Autoregressive Integrated Moving Average
ASV	Asymmetric Stochastic Volatility
AVP	Asymmetric Volatility Phenomenon
CGF	Conditional Gaussian Filter
CGHF	Conditional Gauss-Hermite Filter
CGLSSM	Conditionally Gaussian Linear State-Space Model
CGOMSM	Conditionally Gaussian Observed Markov Switching Model

CGPMSM	Conditionally Gaussian Pairwise Markov Switching Model
CMSHLM	Conditionally Markov Switching Hidden Linear Model
D-graph	Dependency graph (directed graph representing dependencies of several random variables towards each other)
EKF	Extended Kalman Filter
EM	Expectation-Maximization
ERM	Empirical Risk Minimization
E-step	Expectation step of the EM algorithm
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
GF	Gaussian Filter
GHF	Gauss-Hermite Filter
GPF	Gaussian Particle Filter
GSF	Gaussian Sum Filter
GSUKF	Gaussian Sum Unscented Kalman Filter
h-concat	Horizontal concatenation of row vectors: $\begin{bmatrix}\boldsymbol{v}_1^\top & \boldsymbol{v}_2^\top\end{bmatrix}$
HMM	Hidden Markov Model
HMM-CN	Hidden Markov Model With Conditionally Correlated Noise
HMM-IN	Hidden Markov Model With Conditionally Independent Noise
ICE	Iterative Conditional Estimation
LCGOMSMF	Learned Conditionally Gaussian Observed Markov Switching Model Filter
LCGOMSMS	Learned Conditionally Gaussian Observed Markov Switching Model Smoother
LSTM	Local Switching Trend Model
LTM	Local Trend Model
MCMC	Markov chain Monte Carlo
MGF	Markovian Grid-Based Filter
MGS	Markovian Grid-Based Smoother
MGSE	Markovian Grid-Based State Estimator
MME	Mean Misclassification Error
MPM	Maximum Posterior Mode
MSE	Mean Squared Error
M-step	Maximization step of the EM algorithm

MSSV	Markov Switching Stochastic Volatility
pdf	Probability Density Function
PF	Particle Filter
PMM	Pairwise Markov Model
PMM-CN	Pairwise Markov Model With Conditionally Correlated Noise
PMM-IN	Pairwise Markov Model With Conditionally Independent Noise
POMP	Partially Observable Markov Process
PS	Particle Smoother
PSO	Particle Swarm Optimization
QKF	Quadrature Kalman Filter
RHS	Right Hand Side Term
RMSE	Relative Mean Squared Error
RSKF	Reverse Switching Kalman Filter
SCGF	Switching Conditional Gaussian Filter
SCGHF	Switching Conditional Gauss-Hermite Filter
SCGPMSM	Stationary Conditionally Gaussian Pairwise Markov Switching Model
SEM	Stochastic Expectation-Maximization
SIR	Sampling Importance Resampling
SKF	Switching Kalman Filter
SLDS	Switching Linear Dynamical System
STMM	Simplified Triplet Markov Model
SV	Stochastic Volatility
TMM	Triplet Markov Model
TMM-IN	Triplet Markov Model With Independent Noise
UKF	Unscented Kalman Filter
UT	Unscented Transform
v-concat	Vertical concatenation of column vectors: $\left[\boldsymbol{v}_1^\top \; \boldsymbol{v}_2^\top\right]^\top$

Introduction générale

Le modèle de Markov caché, connu comme Hidden Markov Model (HMM), est un modèle mathématique omniprésent dans le traitement statistique des données. Ce modèle se réfère à une analyse basée sur les concepts de signal et d'état. Le signal est l'objet principal de la modélisation et représente un processus stochastique dont une réalisation est visible à l'analyste. Les exemples typiques des signaux sont l'évolution d'un indice boursier, du PIB ou du taux d'intérêt. L'état est un processus stochastique auxiliaire qui aide à caractériser l'évolution du signal et qui n'est pas directement observable. Les exemples typiques d'états comprennent la tendance et la volatilité. Par définition, l'état est de Markov dans les HMMs.

Les HMMs ont été généralisés aux modèles semi-Markoviens cachés, modèles de Markov couples, modèles de Markov triplets et modèles de Markov à sauts. Ces modèles ont un aspect en commun. Nous justifions que dans tous ces modèles, le processus couple état-signal est de Markov, c'est pourquoi nous disons que ces modèles sont des cas particuliers du modèle de Markov partiellement observé, connus comme le Partially Observable Markov Process (POMP). En effet, le processus état-signal est de Markov et la partie qui correspond à l'état n'est pas observable.

Dans cette thèse, nous classifions les POMPs suivant la nature de l'état: nous distinguerons entre les POMPs à états discrets finis, POMPs à états continus et POMPs à états hybrides (continus-discrets finis). L'estimation exacte rapide bayésienne de l'état n'étant généralement pas possible dans les POMPs, l'objectif de ce rapport est de présenter les méthodes d'estimation qui ont été développées pendant les études doctorales de l'auteur. Conformément aux consignes officielles de l'école doctorale, la suite de ce chapitre est décomposée en sections séparées, rédigées en français, qui contiennent des résumés détaillés des chapitres de la thèse, qui sont rédigés en anglais.

Résumé du chapitre 1

Le chapitre 1 présente la théorie générale des POMPs. Nous formalisons les problèmes d'estimation des paramètres des POMPs et les problèmes d'inférence bayésienne dans les POMPs, qui incluent le filtrage, le lissage et la prédiction. Nous présentons aussi les méthodes de Monte-Carlo séquentielles, qui sont des méthodes largement utilisées d'inférence bayésienne dans les POMPs. Nous notons $\mathbf{H}_{1:N} = (\mathbf{H}_1, \ldots, \mathbf{H}_N)$ une série temporelle des variables d'état à valeurs dans $\mathscr{H} = \mathbb{R}^d \times \Omega$, avec $\Omega = \{1 : K\}$ ensemble discret fini. La série temporelle des variables du signal correspondant est notée $\mathbf{Y}_{1:N}$ et est à valeurs dans $\mathbb{R}^{d'}$.

Pour tout N dans $\mathbb{N}^*$, le couple $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ est un modèle de Markov partiellement observé (POMP) si sa distribution vérifie:

$$p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathbf{h}_1, \mathbf{y}_1\right) p\left(\mathbf{h}_2, \mathbf{y}_2 \left| \mathbf{h}_1, \mathbf{y}_1\right.\right) \ldots p\left(\mathbf{h}_N, \mathbf{y}_N \left| \mathbf{h}_{N-1}, \mathbf{y}_{N-1}\right.\right), \tag{1}$$

ce qui signifie que $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ est de Markov.

La décomposition de $\mathbf{H}_{1:N}$ en une partie continue $\mathbf{X}_{1:N}$ et une partie discrète fini $\mathrm{R}_{1:N}$, est :

$$\forall n \in \{1:N\}, \mathbf{H}_n = (\mathbf{X}_n, \mathrm{R}_n). \tag{2}$$

Nous considérons la classification suivante des POMPs:

- $\mathrm{Card}(\Omega) = 1$ et $d > 0$. Dans ce cas, $\mathscr{H} = \mathbb{R}^d$ à une bijection près et le modèle est dit POMP à états continus. Dans ce cas, le processus état-signal est noté $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$;
- $1 \leq \mathrm{Card}(\Omega) < \infty$ et $d = 0$. Dans ce cas, $\mathscr{H} = \Omega$ à une bijection près et le modèle est dit POMP à états discrets finis. Dans ce cas, le processus état-signal est noté $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$;
- $0 \leq \mathrm{Card}(\Omega) < \infty$ et $d \geq 0$. Dans ce cas, le modèle est dit POMP à états hybrides. Dans la littérature, ces modèles sont appelés également des modèles à sauts. Dans ce cas, le processus état-signal est noté $(\mathrm{R}_{1:N}, \mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$.

Les POMPs abordés dans ce rapport sont reportés dans la Figure 1.

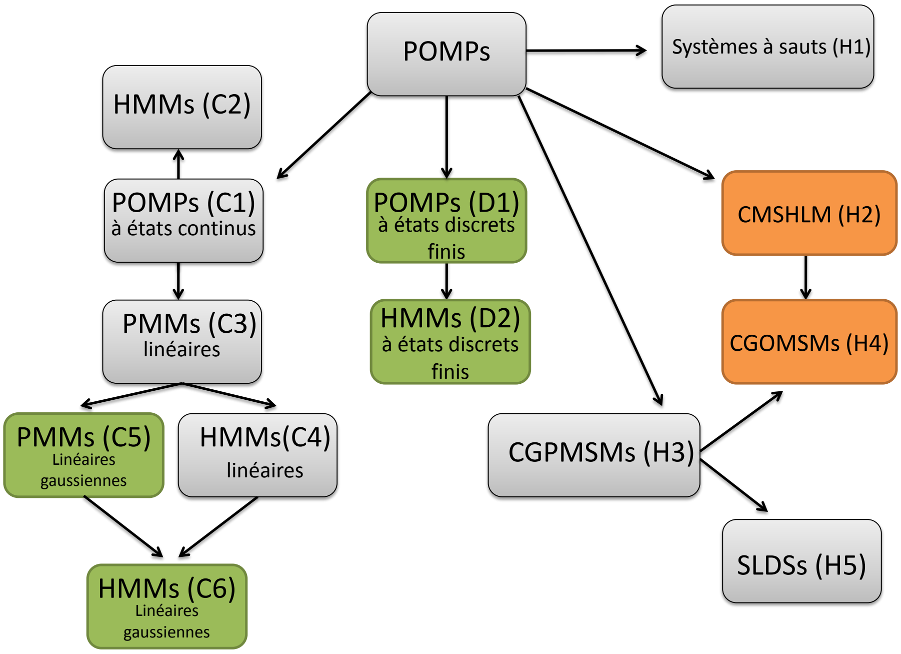

POMPs
Systèmes à sauts (H1)
HMMs (C2)
POMPs (C1) à états continus
POMPs (D1) à états discrets finis
CMSHLM (H2)
PMMs (C3) linéaires
HMMs (D2) à états discrets finis
CGOMSMs (H4)
PMMs (C5) Linéaires gaussiennes
HMMs(C4) linéaires
CGPMSMs (H3)
SLDSs (H5)
HMMs (C6) Linéaires gaussiennes

Figure 1: Modèles de Markov partiellement observés usuels. **A** $\longrightarrow$ **B** signifie que le modèle **B** est un cas particulier de **A**. Les modèles dans lesquels les distributions exactes de filtrage et de lissage ne sont pas calculables en général sont représentés par des rectangles gris. Les modèles dans lesquels les distributions exactes de filtrage et de lissage sont calculables sont représentés par des rectangles verts. Les modèles dans lesquels uniquement les moments exacts de la distribution de filtrage et de lissage sont calculables sont représentés par des rectangles oranges.

La distribution $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right)$ est appelée la distribution de filtrage à l'instant n. Cette distribution joue un grand rôle dans le traitement statistique à l'aide des POMPs. Formellement, cette distribution est donnée par récurrence grâce à la markovianité de $(\mathbf{H}_{1:n}, \mathbf{Y}_{1:n})$:

— Initialisation: $p\left(\mathbf{h}_1 \left| \mathbf{y}_1\right.\right) = \frac{p(\mathbf{h}_1, \mathbf{y}_1)}{\int p(\mathbf{h}_1, \mathbf{y}_1) \mathrm{d}\mathbf{h}_1}$.

— Récurrence: $p\left(\mathbf{h}_{n+1} \left| \mathbf{y}_{1:n+1}\right.\right)$ est calculée à partir de $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right)$ en trois étapes :

1. Calculer la distribution anticipée à un pas :

$$p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n}\right.\right) = \int p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right) p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{h}_n, \mathbf{y}_n\right.\right) \mathrm{d}\mathbf{h}_n; \tag{3}$$

2. Calculer le facteur de vraisemblance à $n+1$:

$$c_{n+1} = p\left(\mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) = \int p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) \mathrm{d}\mathbf{h}_{n+1}; \tag{4}$$

3. Faire la mise à jour :

$$p\left(\mathbf{h}_{n+1} \left| \mathbf{y}_{1:n+1} \right.\right) = \frac{p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right)}{c_{n+1}}. \tag{5}$$

Ce calcul récursif permet aussi de calculer la log-vraisemblance de la séquence observée $\mathbf{y}_{1:N}$:

$$\log p\left(\mathbf{y}_{1:N}\right) = \log \Big(p\left(\mathbf{y}_1\right) \prod_{n=1}^{N-1} p\left(\mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) \Big) = \log p\left(\mathbf{y}_1\right) + \sum_{n=1}^{N-1} \log c_{n+1}. \tag{6}$$

Cette possibilité permet d'envisager une estimation des paramètres des POMPs par maximisation de la log-vraisemblance avec des méthodes numériques. L'estimateur du maximum de vraisemblance est défini par :

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}\left(\mathbf{y}_{1:N}\right). \tag{7}$$

Les méthodes de Monte-Carlo séquentielles permettent un calcul approché de la distribution de filtrage. L'idée est de tirer $M \in \mathbb{N}^*$ réalisations $\{\mathbf{h}_n^{(m)}\}_{1 \leq m \leq M, n \in \mathbb{N}^*}$ dans le but d'approcher empiriquement la distribution de filtrage. Les réalisations tirées dans les méthodes de Monte-Carlo séquentielles sont appelées des particules. Les méthodes de Monte-Carlo séquentielles de filtrage peuvent être très coûteuses en temps de calcul lorsque la dimension de l'espace d'état est grande [Snyder et al., 2008, Ades and Van Leeuwen, 2015, Rebeschini et al., 2015]. Ces méthodes sont fondées sur le principe d'échantillonnage d'importance [Geweke, 1989], qui consiste à tirer les particules selon une distribution d'importance, puis à leur attribuer des poids afin de corriger l'écart entre la distribution de filtrage et la distribution d'importance. Cependant, l'application directe de ce principe dans les POMPs échoue en pratique, car la plupart des poids se rapprochent de zéro, alors que seulement quelques particules ont des poids non-négligeables. En conséquence, l'échantillonnage d'importance seul devient de plus en plus inefficace, car beaucoup de puissance de calcul est dépensée à l'échantillonnage des particules qui ne contribuent pas à l'estimation de la distribution de filtrage. Ce phénomène est connu sous le nom de la dégénérescence des poids [Cappé et al., 2005, Del Moral and Jacod, 2001]. L'approche classique contre la dégénérescence des poids consiste à re-échantillonner les particules à chaque itération ou selon un critère tel que le nombre de particules efficaces [Cornebise et al., 2008, Doucet and Johansen, 2011], c'est-à-dire de re-tirer chaque particule avec une probabilité égale à son poids. Cela donne la classe des algorithmes basés sur l'approche Sampling Importance Resampling (SIR) [Doucet et al., 2000, Douc and Cappe, 2005]. Dans cette approche, la phase de ré-échantillonnage supprime les particules avec les poids faibles et les particules avec des poids significatifs sont ré-échantillonnées plusieurs fois. Le filtre particulaire SIR classique est définie par :

1. Pour m dans $\{1:M\}$, tirer $\widetilde{\mathbf{h}}_{n+1}^{(m)}$ suivant la distribution $p\left(\mathbf{h}_{n+1} \middle| \mathbf{h}_n^{(m)}, \mathbf{y}_n\right)$ si $n > 0$, sinon tirer $\widetilde{\mathbf{h}}_1^{(m)}$ suivant $p\left(\mathbf{h}_1\right)$;

2. Pour m dans $\{1:M\}$, calculer $\eta_{n+1}^{(m)} = p\left(\mathbf{y}_{n+1} \middle| \widetilde{\mathbf{h}}_{n+1}^{(m)}, \mathbf{h}_n^{(m)}, \mathbf{y}_n\right)$;

3. Obtenir $\{\mathbf{h}_{n+1}^{(m)}\}_{1 \leq m \leq M}$ en tirant M particules de $\{\widetilde{\mathbf{h}}_{n+1}^{(m)}\}_{1 \leq m \leq M}$ avec les probabilités proportionnelles à $\{\eta_{n+1}^{(m)}\}_{1 \leq m \leq M}$;

Ainsi, pour tout n dans $\{1:N\}$, la distribution de filtrage $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right)$ est approximée par :

$$p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right) \approx \frac{1}{M} \sum_{m=1}^{M} \delta\Big(\mathbf{h}_n - \mathbf{h}_n^{(m)}\Big),$$

où δ est la distribution de Dirac.

Dans ce rapport, nous détaillons les méthodes alternatives à celles de Monte-Carlo qui ont fait l'objet d'étude de cette thèse. Cependant, les méthodes de Monte-Carlo séquentielles ont servi comme une référence pour quantifier la précision des méthodes proposées.

Résumé du chapitre 2

Dans le chapitre 2, nous revoyons le modèle Conditionally Gaussian Observed Markov Switching Model (CGOMSM), qui est un POMP à états hybrides.

Soit $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ un processus stationnaire état-signal. Le CGOMSM est un triplet $(\mathrm{R}_{1:N}, \mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ tel que, pour tout $\mathrm{r}_{n:n+1}$ dans Ω^2, nous avons :

— La distribution de $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left| \mathrm{r}_{n:n+1}\right.\right)$ est gaussienne de moyenne $\boldsymbol{\Upsilon}(\mathrm{r}_{n:n+1})$ et de matrice de covariance $\boldsymbol{\Xi}(\mathrm{r}_{n:n+1})$.

— La moyenne de $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left| \mathrm{r}_{n:n+1}\right.\right)$ est de la forme :

$$\boldsymbol{\Upsilon}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \mathbf{M}(\mathrm{r}_n) \\ \mathbf{M}(\mathrm{r}_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbb{E}\left[[\mathbf{X}_n^\top \mathbf{Y}_n^\top]^\top \left| \mathrm{R}_n = \mathrm{r}_n\right.\right] \\ \mathbb{E}\left[[\mathbf{X}_{n+1}^\top \mathbf{Y}_{n+1}^\top]^\top \left| \mathrm{R}_{n+1} = \mathrm{r}_{n+1}\right.\right] \end{bmatrix}; \tag{8}$$

— La matrice de covariance de $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left| \mathrm{r}_{n:n+1}\right.\right)$ est de la forme :

$$\boldsymbol{\Xi}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \boldsymbol{S}(\mathrm{r}_n) & \boldsymbol{\Sigma}(\mathrm{r}_{n:n+1}) \\ \boldsymbol{\Sigma}^\top(\mathrm{r}_{n:n+1}) & \boldsymbol{S}(\mathrm{r}_{n+1}) \end{bmatrix}; \tag{9}$$

— $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left| \mathrm{r}_{n:n+1}\right.\right)$ est soumise à la contrainte :

$$p\left(\mathbf{y}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_{n:n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n:n+1}, \mathbf{y}_n\right.\right).$$

Ce modèle permet une implémentation pratique d'algorithme de filtrage et de lissage exact [Abbassi et al., 2015, Abbassi et al., 2011, Gorynin et al., 2015, Gorynin et al., 2017a, Gorynin et al., 2017b].

Il a été observé dans [Derrode and Pieczynski, 2013] que la Probability Density Function (pdf) de $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ dans le CGOMSM stationnaire est de la forme :

$$p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2\right) = \sum_{1 \leq i,j \leq K} \alpha_{ij} p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2), \tag{10}$$

avec $\{\alpha_{ij}\}_{1 \leq i,j \leq K}$ réels positifs et pour tout (i,j) dans $\{1:K\}^2$, $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ est une distribution gaussienne qui vérifie

$$p_{ij}\left(\mathbf{y}_2 \left| \mathbf{x}_1, \mathbf{y}_1\right.\right) = p_{ij}\left(\mathbf{y}_2 \left| \mathbf{y}_1\right.\right).$$

L'idée est alors d'approcher un processus de Markov stationnaire $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ arbitraire par la distribution marginale (10) de CGOMSM. En effet, la distribution d'un processus de Markov stationnaire $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ est donnée par $p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2\right)$, et nous voyons que le CGOMSM permet de représenter $p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2\right)$ comme un mélange de gaussiennes contraintes par $p_{ij}\left(\mathbf{y}_2 \left| \mathbf{x}_1, \mathbf{y}_1\right.\right) = p_{ij}\left(\mathbf{y}_2 \left| \mathbf{y}_1\right.\right)$. Or, il est connu qu'un mélange de gaussiennes est très flexible et permet d'approcher d'aussi prêt qu'on le souhaite les distributions suffisamment régulières. Cela a permis de concevoir le Learned Conditionally Gaussian Observed Markov Switching Model Filter (LCGOMSMF) [Gorynin et al., 2017a] et Learned Conditionally Gaussian Observed Markov Switching Model Smoother (LCGOMSMS) [Gorynin et al., 2017b], qui fonctionnent de la manière suivante :

1. Considérer une séquence d'apprentissage $(\mathbf{x}^{\star}_{1:N'}, \mathbf{y}^{\star}_{1:N'})$ issue d'un processus de Markov stationnaire arbitraire. Cette séquence définit une distribution empirique du quadruplet $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$;

2. Approcher la distribution empirique du quadruplet $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ par un CGOMSM;

3. Procéder au filtrage ou au lissage des donnés réelles $(\mathbf{x}_{1:N}, \mathbf{y}_{1:N})$.

Le chapitre 2 détaille toutes les étapes de la construction du CGOMSM, le modèle Conditionally Markov Switching Hidden Linear Model (CMSHLM), les algorithmes de filtrage et de lissage exact et la contribution principale de l'auteur, qui est la conception d'algorithme de type Expectation-Maximization (EM) pour l'estimation des paramètres du modèle CGOMSM à partir d'une séquence d'apprentissage.

Résumé du chapitre 3

Dans le chapitre 3, nous présentons une approche générale de filtrage et de lissage dans les POMPs à états hybrides. Cette approche utilise les grilles d'intégration numérique. L'idée de l'application de ces grilles au problème d'estimation bayésienne est la suivante. Pour toute transformation mesurable $\boldsymbol{f}$ de la variable aléatoire $\mathbf{H}_n$, le calcul de l'espérance de $\boldsymbol{f}(\mathbf{H}_n)$ sachant le signal observé $\mathbf{y}_{1:N}$ peut se résumer à un calcul d'intégrales :

$$\mathbb{E}\left[\boldsymbol{f}(\mathbf{H}_n)\left|\mathbf{y}_{1:N}\right.\right] = \int \boldsymbol{f}(\mathbf{h}_n) p\left(\mathbf{h}_n\left|\mathbf{y}_{1:N}\right.\right) \mathrm{d}\mathbf{h}_{1:N} = \frac{\int \boldsymbol{f}(\mathbf{h}_n) p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \mathrm{d}\mathbf{h}_{1:N}}{\int p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \mathrm{d}\mathbf{h}_{1:N}},$$

sous réserve de l'existence de $\mathbb{E}\left[\boldsymbol{f}(\mathbf{H}_n)\left|\mathbf{y}_{1:N}\right.\right]$, où $\mathrm{d}\mathbf{h}_{1:N}$ dénote une mesure-hybride formée par les mesures de Dirac et Lebesgue.

Un calcul approché des intégrales au dénominateur et au numérateur peut se faire grâce à des grilles d'intégration. Une grille d'intégration permet de définir un sous-ensemble discret fini $\Lambda \subset \Omega \times \mathbb{R}^d$ et une fonction de masse $\pi^{(N)} \in \mathscr{F}\left(\Lambda^N \to \mathbb{R}\right)$ pour avoir :

$$\sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda^N} \boldsymbol{f}\left(\boldsymbol{\gamma}^{(n)}_{1:N}\right) \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) \approx \int \boldsymbol{f}(\mathbf{h}_n) p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \mathrm{d}\mathbf{h}_{1:N}; \tag{11a}$$

$$\sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda^N} \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) \approx \int p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \mathrm{d}\mathbf{h}_{1:N}, \tag{11b}$$

où $\boldsymbol{\gamma}_{1:N} = \begin{bmatrix} \boldsymbol{\gamma}^{(1)}_{1:N} & \boldsymbol{\gamma}^{(2)}_{1:N} & \dots & \boldsymbol{\gamma}^{(N)}_{1:N} \end{bmatrix}$ et les coefficients $\{\boldsymbol{\gamma}^{(i)}_{1:N}\}_{1 \leq i \leq N}$ sont dans Λ. Cependant, le calcul direct de (11) serait de complexité exponentielle $\mathscr{O}\left(\mathrm{Card}(\Lambda)^N\right)$, ce qui n'est pas compatible avec la majorité des applications pratiques. L'auteur introduit les grilles markoviennes, qui ont été développées dans le cadre de ce projet. Une grille markovienne est telle qu'il existe des fonctions $q_1, q_2, \dots q_{N-1}$ dans $\mathscr{F}\left(\Lambda^2 \to \mathbb{R}\right)$ telles que la fonction de masse π de la grille vérifie:

$$\forall \boldsymbol{\gamma}_{1:N} \in \Lambda^N, \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) = q_1\left(\boldsymbol{\gamma}^{(1)}_{1:N}, \boldsymbol{\gamma}^{(2)}_{1:N}\right) q_2\left(\boldsymbol{\gamma}^{(2)}_{1:N}, \boldsymbol{\gamma}^{(3)}_{1:N}\right) \dots q_{N-1}\left(\boldsymbol{\gamma}^{(N-1)}_{1:N}, \boldsymbol{\gamma}^{(N)}_{1:N}\right). \tag{12}$$

Dans le chapitre 3, nous démontrons que les grilles markoviennes permettent d'évaluer (11) avec une complexité $\mathscr{O}\left(N\mathrm{Card}(\Lambda)^2\right)$ qui est linéaire en N. De plus, nous considérons des séquences des grilles markoviennes de type $\left\{\Lambda^N_L, \pi^{(N)}_L\right\}_{L \in \mathbb{N}^*}$ où $\mathrm{Card}(\Lambda_L)$ augmente avec L.

Nous fournissons des conditions suffisantes qui garantissent la consistance de la méthode, c'est-à-dire qui assurent la propriété :

$$\lim_{L\to\infty}\frac{\sum\limits_{\gamma_{1:N}\in\Lambda_L^N}\boldsymbol{f}\left(\gamma_{1:N}^{(n)}\right)\pi_L^{(N)}(\gamma_{1:N})}{\sum\limits_{\gamma_{1:N}\in\Lambda_L^N}\pi_L^{(N)}(\gamma_{1:N})}=\mathbb{E}\left[\boldsymbol{f}(\mathbf{H}_n)\left|\mathbf{y}_{1:N}\right.\right] \tag{13}$$

pour toute fonction $\boldsymbol{f}$ développable en série entière au voisinage de chacun des points de son domaine de définition.

Nous appelons cette méthode d'estimation de $\mathbb{E}\left[\boldsymbol{f}(\mathbf{H}_n)\left|\mathbf{y}_{1:N}\right.\right]$ Markovian Grid-Based State Estimator (MGSE). Le MGSE appliqué au calcul approché de $\mathbb{E}\left[\mathbf{H}_N\left|\mathbf{y}_{1:N}\right.\right]$ est appelé Markovian Grid-Based Filter (MGF). Le MGSE appliqué au calcul approché de $\mathbb{E}\left[\mathbf{H}_n\left|\mathbf{y}_{1:N}\right.\right]$ pour $n < N$ est appelé Markovian Grid-Based Smoother (MGS).

Dans la littérature, nous pouvons trouver des méthodes analogues à MGSE : par exemple [Gospodinov and Lkhagvasuren, 2014, Farmer and Toda, 2017, Terry and Knotek, 2011, Tauchen, 1986, Lo et al., 2016]. La valeur ajoutée de la contribution de l'auteur par rapport aux résultats existants est la suivante :

- Nous prouvons que MGSE converge vers la valeur de l'espérance a posteori ;
- L'algorithme MGSE est donné dans le cas le plus général, c'est-à-dire dans le cas des POMPs à états hybrides ;
- Le MGSE utilisé avec des grilles creuses (Sparse grids) permet d'estimer efficacement l'état de grande dimension.

Résumé du chapitre 4

Le chapitre 4 est consacré à une étude de comparaison des performances des estimateurs optimaux d'états basés sur les sous-modèles des Pairwise Markov Model (PMM)s et Triplet Markov Model (TMM)s. Les PMMs sont vus comme une généralisation des HMMs. La contribution de l'auteur a consisté à conduire et rapporter des séries multiples d'expériences de comparaison des performances des estimateurs sur des données réelles et synthétiques.

Rappelons qu'un HMM $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ a les propriétés suivantes :

- $\mathrm{R}_{1:N}$ est une chaîne de Markov ;
- les éléments de $\mathbf{Y}_{1:N}$ sont indépendants conditionnellement à $\mathrm{R}_{1:N}$;
- pour tout n dans $\{1 : N\}$, $p\left(\mathbf{y}_n\left|\mathrm{r}_{1:N}\right.\right) = p\left(\mathbf{y}_n\left|\mathrm{r}_n\right.\right)$.

Dans un PMM, $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ est de Markov :

$$p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathrm{r}_1, \mathbf{y}_1\right) p\left(\mathrm{r}_2, \mathbf{y}_2\left|\mathrm{r}_1, \mathbf{y}_1\right.\right) \ldots p\left(\mathrm{r}_N, \mathbf{y}_N\left|\mathrm{r}_{N-1}, \mathbf{y}_{N-1}\right.\right). \tag{14}$$

Nous pouvons voir que la pdf de $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ dans un HMM est de la forme (14), car nous avons dans un HMM :

$$p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathrm{r}_1\right) p\left(\mathbf{y}_1\left|\mathrm{r}_1\right.\right) p\left(\mathrm{r}_2\left|\mathrm{r}_1\right.\right) p\left(\mathbf{y}_2\left|\mathrm{r}_2\right.\right) \ldots p\left(\mathrm{r}_N\left|\mathrm{r}_{N-1}\right.\right) p\left(\mathbf{y}_N\left|\mathrm{r}_N\right.\right), \tag{15}$$

et la densité $p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1}\left|\mathrm{r}_n, \mathbf{y}_n\right.\right)$ peut être identifiée à :

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1}\left|\mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{y}_{n+1}\left|\mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right.\right). \tag{16}$$

Ainsi, un PMM est un HMM si et seulement si pour tout n dans $\{1 : N-1\}$, nous avons :

$$p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n\right.\right); \tag{17a}$$

$$p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}\right.\right). \tag{17b}$$

Les équations (17) sont en effet des hypothèses implicites qui sont admises lorsqu'un système est modélisé par un HMM. Les PMMs permettent de relâcher ces hypothèses supplémentaires.

Nous considérons quatre sous-modèles des PMMs.

— Hidden Markov Model With Conditionally Independent Noise (HMM-IN) est le HMM "classique". La densité de transition dans un HMM-IN est de la forme :

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n\right.\right) p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}\right.\right). \tag{18}$$

— Hidden Markov Model With Conditionally Correlated Noise (HMM-CN) est un PMM où $\mathrm{R}_{1:N}$ est de Markov, les éléments de $\mathbf{Y}_{1:N}$ sont corrélés sachant $\mathrm{R}_{1:N}$ et qui n'est pas un HMM-IN (ce qui est schématisé dans la Figure 4.1). La densité de transition dans un HMM-CN est de la forme :

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n\right.\right) p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}, \mathbf{y}_n\right.\right). \tag{19}$$

— Pairwise Markov Model With Conditionally Independent Noise (PMM-IN) est un PMM où $\mathrm{R}_{1:N}$ n'est pas de Markov, les éléments de $\mathbf{Y}_{1:N}$ sont indépendants sachant $\mathrm{R}_{1:N}$ et qui n'est pas un HMM-IN. La densité de transition dans un PMM-IN est de la forme :

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}, \mathrm{r}_n\right.\right). \tag{20}$$

— Pairwise Markov Model With Conditionally Correlated Noise (PMM-CN) est un PMM où $\mathrm{R}_{1:N}$ n'est pas de Markov, les éléments de $\mathbf{Y}_{1:N}$ sont corrélés sachant $\mathrm{R}_{1:N}$ et qui n'est pas un HMM-IN, PMM-IN où HMM-CN. La densité de transition dans un PMM-CN est de la forme générale :

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right.\right). \tag{21}$$

Les graphes de dépendance de ces sous-modèles de PMM sont présentés à la Figure 4.2.

Dans ce chapitre, on cherche à quantifier dans quelle mesure le fait de relâcher les deux hypothèses de (17) contribue à améliorer les performances de l'estimateur du Maximum Posterior Mode (MPM) du PMM en comparaison avec l'estimateur du MPM du HMM. Pour cela, on a défini les modèles HMM-CN et PMM-IN qui sont intermédiaires entre le HMM-IN et PMM-CN. Puis, l'auteur a proposé une technique d'approximation d'un PMM-CN par un HMM-IN, HMM-CN et PMM-IN en utilisant la méthode des moments. L'étude a consisté principalement à simuler une réalisation de PMM-CN et à restaurer les états cachés avec les estimateurs du MPM qui correspondent aux quatre sous-modèles dans le but de quantifier les gains possibles liés à l'utilisation des PMMs. Nous avons considéré trois cas de distributions de $\mathbf{Y}_{1:N}$ sachant $\mathrm{R}_{1:N}$: gaussienne, exponentielle et gamma. Dans tous ces cas, nous avons montré que les deux hypothèses (17) du HMM contribuent indépendamment à la dégradation de la qualité de restauration, ce qui a confirmé expérimentalement la préférence des PMMs aux HMMs.

Une grande partie du chapitre est consacrée à une validation expérimentale sur des données réelles.

Les HMMs et PMMs permettent d'étendre le modèle de Black-Scholes utilisé en modélisation financière des rendements des actifs. Le modèle classique de Black-Scholes suppose que le logarithme du rendement d'un actif sur une durée fixe a une distribution normale, c'est-à-dire que :

$$\mathrm{Y}_n = \mu + \sigma \mathrm{U}_n, \tag{22}$$

où $\{\mathrm{Y}_n\}_{1\leq n\leq N}$ sont les log-rendements sur une durée fixe et $\{\mathrm{U}_n\}_{1\leq n\leq N}$ des variables gaussiennes indépendantes identiquement distribuées. Les HMMs et PMMs permettent d'introduire un processus caché $\mathrm{R}_{1:N}$ à valeurs dans un ensemble discret fini Ω et supposer une dépendance de μ et σ des valeurs prises par les variables cachées et donc poser :

$$\mathrm{Y}_n = \mu(\mathrm{r}_n) + \sigma(\mathrm{r}_n)\mathrm{U}_n. \tag{23}$$

Prenons par exemple $\Omega = \{\omega_1, \omega_2\}$, alors ω_1 peut être associé à un état baissier du marché et ω_2 peut être associer à un état haussier du marché. En supposant que $\mathrm{R}_{1:N}$ est de Markov, (23) permet de définir un HMM qui modélise les log-rendements.

Afin de proposer un modèle PMM compatible avec (23) et qui serait plus général que le HMM, l'auteur a proposé de modéliser la loi de $\mathrm{Y}_{1:N}$ sachant $\mathrm{R}_{1:N}$ par celle d'un processus autorégressif d'ordre 1, c'est-à-dire :

$$\mathrm{U}_{n+1} = \rho(\mathrm{R}_n, \mathrm{R}_{n+1})\mathrm{U}_n + \sqrt{1-\rho(\mathrm{R}_n, \mathrm{R}_{n+1})^2}\mathrm{V}_{n+1}, \tag{24}$$

avec $\mathrm{U}_0, \{\mathrm{V}_n\}_{n>0}$ des variables gaussiennes indépendantes identiquement distribuées et pour tout $i, j \in \Omega, |\rho(i,j)| < 1$.

Ensuite, l'auteur a proposé de modéliser le lien probabiliste possible entre R_{n+1} et Y_n sachant R_n en utilisant la fonction logistique. Dans le cas où $\Omega = \{\omega_1, \omega_2\}$, cela donne :

$$p\left(\mathrm{r}_{n+1} = \omega_1 \left| \mathrm{r}_n, \mathrm{u}_n \right.\right) = \frac{1}{1+e^{-a(\mathrm{r}_n)-b(\mathrm{r}_n)\mathrm{u}_n}}, \tag{25}$$

avec $a(\omega) \in \mathbb{R}, b(\omega) \in \mathbb{R}$ pour tout $\omega \in \Omega$.

Finalement, le modèle proposé des log-rendements est donné par :

$$p\left(\mathrm{y}_1 \left| \mathrm{r}_1 \right.\right) = \mathscr{N}\Big(\mathrm{y}_1; \mu(\mathrm{r}_1), \sigma^2(\mathrm{r}_1)\Big); \tag{26a}$$

$$p\left(\mathrm{r}_{n+1} = \omega_1 \left| \mathrm{r}_n, \mathrm{y}_n \right.\right) = \frac{1}{1+e^{-a(\mathrm{r}_n)-\frac{b(\mathrm{r}_n)}{\sigma(\mathrm{r}_n)}(\mathrm{y}_n-\mu(\mathrm{r}_n))}}; \tag{26b}$$

$$p\left(\mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathrm{y}_n \right.\right) = \mathscr{N}\Big(\mathrm{y}_{n+1}; \mu(\mathrm{r}_{n+1}) + \frac{\rho(\mathrm{r}_n, \mathrm{r}_{n+1})\sigma(\mathrm{r}_{n+1})}{\sigma(\mathrm{r}_n)}\left(\mathrm{y}_n - \mu(\mathrm{r}_n)\right), \sigma(\mathrm{r}_{n+1})^2(1-\rho(\mathrm{r}_n, \mathrm{r}_{n+1})^2)\Big). \tag{26c}$$

Ce modèle a été implémenté et appliqué à des données historiques dans le cadre d'une simulation de trading (backtesting). Cette étude a mis en évidence les améliorations apportées par le passage du HMM au PMM.

Résumé du chapitre 5

Le chapitre 5 cherche à analyser les insuffisances du Gaussian Filter (GF), qui ont été corrigées par le Conditional Gaussian Filter (CGF). La contribution de l'auteur est de proposer une extension du CGF applicable dans le contexte des POMPs à états hybrides. Cette extension est appelée le Switching Conditional Gaussian Filter (SCGF).

Le GF et le CGF s'appliquent dans le cadre d'un POMP à états continus, donné par le processus état-signal $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ de la forme :

$$\mathbf{X}_{n+1} = \boldsymbol{f}_{n+1}(\mathbf{X}_n, \mathbf{U}_{n+1}), \qquad n \in \mathbb{N}^*, n < N; \tag{27a}$$

$$p\left(\mathbf{y}_n \left| \mathbf{x}_n \right.\right) \propto h_n(\mathbf{y}_n, \mathbf{x}_n), \qquad n \in \mathbb{N}^*, n \leq N, \tag{27b}$$

avec $\mathbf{X}_{1:N}$ un processus de Markov dans $\mathbb{R}^d$ et les éléments de $\mathbf{Y}_{1:N}$ dans $\mathbb{R}^{d'}$ indépendantes conditionnellement à $\mathbf{X}_{1:N}$. Les éléments de $\mathbf{U}_{1:N}$ sont des variables gaussiennes centrées réduites indépendantes dans $\mathbb{R}^q$.

Le principe du GF est implémenté dans l'Extended Kalman Filter (EKF), l'Unscented Kalman Filter (UKF) et le Gauss-Hermite Filter (GHF). L'idée du GF consiste à supposer que pour tout n, la densité $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n}\right.\right)$ est gaussienne :

$$\forall n \in \mathbb{N}, p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n}\right.\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_{n+1} \\ \mathbf{y}_{n+1}\end{bmatrix}; \begin{bmatrix}\widehat{\mathbf{x}}_{n+1|n} \\ \widehat{\mathbf{y}}_{n+1|n}\end{bmatrix}, \begin{bmatrix}\mathbf{P}^{\mathbf{xx}}_{n+1|n} & \mathbf{P}^{\mathbf{xy}}_{n+1|n} \\ \mathbf{P}^{\mathbf{yx}}_{n+1|n} & \mathbf{P}^{\mathbf{yy}}_{n+1|n}\end{bmatrix}\right), \tag{28}$$

avec $\widehat{\mathbf{x}}_{n+1|n} \in \mathbb{R}^d$, $\widehat{\mathbf{y}}_{n+1|n} \in \mathbb{R}^{d'}$, $\mathbf{P}^{\mathbf{xx}}_{n+1|n} \in \mathbb{R}^{d\times d}$, $\mathbf{P}^{\mathbf{xy}}_{n+1|n} \in \mathbb{R}^{d\times d'}$, $\mathbf{P}^{\mathbf{yx}}_{n+1|n} \in \mathbb{R}^{d'\times d}$ et $\mathbf{P}^{\mathbf{yy}}_{n+1|n} \in \mathbb{R}^{d'\times d'}$.

Cela implique que :

$$\forall \in \mathbb{N}^*, p_{n|n}\left(\mathbf{x}_n\right) = p\left(\mathbf{x}_n \left| \mathbf{y}_{1:n}\right.\right) = \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n}\right); \tag{29a}$$

$$\forall \in \mathbb{N}, p_{n+1|n}\left(\mathbf{x}_{n+1}\right) = p\left(\mathbf{x}_{n+1} \left| \mathbf{y}_{1:n}\right.\right) = \mathcal{N}\left(\mathbf{x}_{n+1}; \widehat{\mathbf{x}}_{n+1|n}, \widehat{\mathbf{\Gamma}}_{n+1|n}\right), \tag{29b}$$

où $\widehat{\mathbf{x}}_{n|n} \in \mathbb{R}^d$, $\widehat{\mathbf{\Gamma}}_{n+1|n} \in \mathbb{R}^{d\times d}$ et $(\widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n})$ sont obtenus par le conditionnement gaussien de (28):

$$\widehat{\mathbf{x}}_{n|n} = \widehat{\mathbf{x}}_{n|n-1} + \mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1}(\mathbf{y}_n - \widehat{\mathbf{y}}_{n|n-1}); \tag{30a}$$

$$\widehat{\mathbf{\Gamma}}_{n|n} = \widehat{\mathbf{\Gamma}}_{n|n-1} - \mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1}\mathbf{P}^{\mathbf{yx}}_{n|n-1}. \tag{30b}$$

Le GF calcule $\widehat{\mathbf{x}}_{n+1|n+1}$ et $\widehat{\mathbf{\Gamma}}_{n+1|n+1}$ à partir de $\widehat{\mathbf{x}}_{n|n}$, $\widehat{\mathbf{\Gamma}}_{n|n}$ et $\mathbf{y}_{n+1}$:

1. Prédiction

$$\widehat{\mathbf{x}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})p_{n|n}\left(\mathbf{x}_n\right)p\left(\mathbf{u}_{n+1}\right)\mathrm{d}\mathbf{x}_n\mathrm{d}\mathbf{u}_{n+1}; \tag{31a}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})\boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})^\top p_{n|n}\left(\mathbf{x}_n\right)p\left(\mathbf{u}_{n+1}\right)\mathrm{d}\mathbf{x}_n\mathrm{d}\mathbf{u}_{n+1} - \widehat{\mathbf{x}}_{n+1|n}\widehat{\mathbf{x}}^\top_{n+1|n}. \tag{31b}$$

2. Mise à jour

$$\widehat{\mathbf{y}}_{n+1|n} = \int \mathbf{y}_{n+1}h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1})p_{n+1|n}\left(\mathbf{x}_{n+1}\right)\mathrm{d}\mathbf{x}_{n+1}\mathrm{d}\mathbf{y}_{n+1}; \tag{32a}$$

$$\mathbf{P}^{\mathbf{xy}}_{n+1|n} = \int (\mathbf{x}_{n+1} - \widehat{\mathbf{x}}_{n+1|n})(\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|n})^\top h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1})p_{n+1|n}\left(\mathbf{x}_{n+1}\right)\mathrm{d}\mathbf{x}_{n+1}\mathrm{d}\mathbf{y}_{n+1}. \tag{32b}$$

$$\mathbf{P}^{\mathbf{yy}}_{n+1|n} = \int \mathbf{y}_{n+1}\mathbf{y}^\top_{n+1}h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1})p_{n+1|n}\left(\mathbf{x}_{n+1}\right)\mathrm{d}\mathbf{x}_{n+1}\mathrm{d}\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|n}\widehat{\mathbf{y}}^\top_{n+1|n}. \tag{32c}$$

Ensuite, $\widehat{\mathbf{x}}_{n+1|n+1}$ et $\widehat{\mathbf{\Gamma}}_{n+1|n+1}$ sont obtenus en appliquant la formule (30) à $\widehat{\mathbf{x}}_{n+1|n}$, $\widehat{\mathbf{y}}_{n+1|n}$, $\widehat{\mathbf{\Gamma}}_{n+1|n}$, $\mathbf{P}^{\mathbf{xy}}_{n+1|n}$, $\mathbf{P}^{\mathbf{yy}}_{n+1|n}$ et $\mathbf{P}^{\mathbf{yx}}_{n+1|n} = (\mathbf{P}^{\mathbf{xy}}_{n+1|n})^\top$.

Les insuffisances du GF viennent de la forme d'approximation (28). Nous notons que :

- L'approximation (28) peut induire la divergence du filtre dans le cas où la distribution de $\mathbf{Y}_n$ sachant $\mathbf{X}_n$ est à queue lourde *cf.* [Roth et al., 2013]. Cela vient du fait que seuls les deux premiers moments sont considérés dans l'approximation de la distribution jointe de $(\mathbf{X}_n, \mathbf{Y}_n)$.

- Si la distribution de $\mathbf{Y}_n$ sachant $\mathbf{X}_n$ est de variance infinie, ou si les variables $\mathbf{Y}_n$ et $\mathbf{X}_n$ sont décorrélées mais pas indépendantes, alors nous avons $\mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1} = \mathbf{0}$. Dans ce cas, l'étape de la mise à jour échoue systématiquement et le GF n'extrait aucune information de $\mathbf{Y}_{1:N}$ sur la distribution de $\mathbf{X}_{1:N}$.

L'approche du CGF permet de corriger ces défauts. L'idée est de supposer une hypothèse moins forte que celle du GF, qui est :

$$p_{n|n}\left(\mathbf{x}_n\right) = p\left(\mathbf{x}_n \left|\mathbf{y}_{1:n}\right.\right) = \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n}\right); \tag{33a}$$

$$p_{n+1|n}\left(\mathbf{x}_{n+1}\right) = p\left(\mathbf{x}_{n+1} \left|\mathbf{y}_{1:n}\right.\right) = \mathcal{N}\left(\mathbf{x}_{n+1}; \widehat{\mathbf{x}}_{n+1|n}, \widehat{\mathbf{\Gamma}}_{n+1|n}\right). \tag{33b}$$

L'algorithme du CGF est le suivant :

1. Prédiction

$$\widehat{\mathbf{x}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) p_{n|n}\left(\mathbf{x}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1}; \tag{34a}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})^\top p_{n|n}\left(\mathbf{x}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1} - \widehat{\mathbf{x}}_{n+1|n} \widehat{\mathbf{x}}_{n+1|n}^\top. \tag{34b}$$

2. Mise à jour

$$c_{n+1} = \int h_{n+1}(\mathbf{y}_{n+1}; \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1}; \tag{35a}$$

$$\widehat{\mathbf{x}}_{n+1|n+1} = \int \mathbf{x}_{n+1} \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right)}{c_{n+1}} \mathrm{d}\mathbf{x}_{n+1}; \tag{35b}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n+1} = \int \mathbf{x}_{n+1} \mathbf{x}_{n+1}^\top \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right)}{c_{n+1}} \mathrm{d}\mathbf{x}_{n+1} - \widehat{\mathbf{x}}_{n+1|n+1} \widehat{\mathbf{x}}_{n+1|n+1}^\top. \tag{35c}$$

Par construction, le CGF n'a pas les défauts annoncés du GF, car il évite de faire une approximation de la distribution de $\mathbf{Y}_n$ sachant $\mathbf{X}_n$. Cela permet de justifier l'intérêt à étendre le CGF pour pouvoir l'appliquer dans le contexte des POMPs à états hybrides.

Publications de l'auteur

Ce travail a fait l'objet de plusieurs articles dans des revues internationales (publiés, acceptés ou soumis) et dans des conférences internationales. Nous présentons ici la liste des différentes publications de l'auteur. Leurs liens avec les différentes sections sont donnés dans la Table 1.

Articles de revues internationales avec comité de lecture

Articles publiés

- [CSDA17] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini and Wojciech Pieczynski (2017). Fast smoothing in switching approximations of non-linear and non-Gaussian models. *Computational Statistics & Data Analysis* vol. **114** no. **1**, pp. 38–46.

- [TAC17] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini and Wojciech Pieczynski (2017). Fast Filtering in Switching Approximations of Nonlinear Markov Systems With Applications to Stochastic Volatility. *IEEE Transactions on Automatic Control* vol. **62** no. **2**, pp. 853–862.

Articles acceptés

- [SP17b] Ivan Gorynin, Hugo Gangloff, Emmanuel Monfrini and Wojciech Pieczynski (2017). Assessing the segmentation performance of pairwise and triplet Markov models. *Signal Processing*, accepted with minor changes.

Articles soumis

- [SS17] Ivan Gorynin and Emmanuel Monfrini (2017). Convergent Markovian Grid Approximations for Inference in Jump Markov Systems. *Statistical Science*, submitted for publication.
- [JMLR17] Ivan Gorynin and Wojciech Pieczynski (2017). Bayesian-Assimilation-Based Smoothing in Switching Systems with Application to Financial Trend Analysis. *Journal of Machine Learning Research*, submitted for publication.
- [SP17a] Ivan Gorynin and Wojciech Pieczynski (2017). Switching conditional Gauss-Hermite filter with application to jump volatility model. *Signal Processing*, submitted for publication.

Articles de conférences internationales avec actes et comité de lecture

- [EUSIPCO17] Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski. Pairwise Markov Models for Stock Index Forecasting. *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Kos Island, Greece, August 28 - September 2, 2017.
- [ICASSP17] Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski. Unsupervised learning of asymmetric high-order autoregressive stochastic volatility model. *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 5-9 March, 2017.
- [MLSP16] Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski. Unsupervised learning of Markov-switching stochastic volatility with an application to market data. *Proceedings of the 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Vietri sul Mare, Salerno, Italy, September 13-16, 2016.
- [SSP16] Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski. Fast filtering with new sparse transition Markov chains. *Proceedings of the 2016 IEEE Workshop on Statistical Signal Processing (SSP 16)*, Palma de Mallorca, Spain, June 26-29, 2016.
- [EUSIPCO15] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini and Wojciech Pieczynski. Exact fast smoothing in switching models with application to stochastic volatility. *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, Nice, France, August 31 - September 4, 2015.

Articles de conférences nationales avec actes et comité de lecture

- [GRETSI17] Ivan Gorynin, Emmanuel Monfrini and Wojciech Pieczynski. Estimation de la variance stochastique multivariée avec un filtre gaussien basé sur la méthode de Laplace. *XXVI'eme Colloque GRETSI*, Juan-Les-Pins, France, 5-8 Septembre, 2017.
- [GRETSI15] Ivan Gorynin, Stéphane Derrode, Emmanuel Monfrini and Wojciech Pieczynski. Lissage rapide dans des modèles non linéaires et non gaussiens. *XXV'ème Colloque GRETSI*, Lyon, France, 8-11 septembre, 2015.

Paragraphe	Section	Articles
Conditionally Gaussian observed Markov switching models	Section 2.2: Exact Bayesian state estimation	EUSIPCO15 CSDA17
Conditionally Gaussian observed Markov switching models	Section 2.3: Parameter estimation by the EM algorithm.	EUSIPCO15 CSDA17 TAC17
Markovian grid-based Bayesian state estimation	Section 3.2: Markovian grid-based state estimators	SSP16 SS17
Bayesian state estimation in partially observed Markov processes with discrete state space	Section 4.3: Performance comparison across PMM estimators	SP17b
Bayesian state estimation in partially observed Markov processes with discrete state space	Section 4.4: Stock forecasting with PMMs	EUSIPCO17
Bayesian state estimation in partially observed Markov processes with hybrid state space	Section 5.1: Bayesian smoothing in conditionally linear POMPs with hybrid state space	JMLR17
Bayesian state estimation in partially observed Markov processes with hybrid state space	Section 5.2: Bayesian filtering in non-linear non-Gaussian POMPs with hybrid state space	SP17a MLSP16

Table 1: Correspondance entre les différentes publications de l'auteur et des sections du document.

Chapter 1

Introduction

This chapter is a general presentation of the Partially Observable Markov Process (POMP). We outline the general Bayesian state estimation procedure which is used in the Bayesian filtering, smoothing and forecasting. We also present the sequential Monte-Carlo methods, which are widely used in the context of POMPs. In this chapter and for the rest of the report, we denote by $\mathbf{H}_{1:N} = \mathbf{H}_1, \ldots, \mathbf{H}_N$ a time series, where for each n in $\{1 : N\}$, $\mathbf{H}_n$ is a state vector and takes values in $\mathscr{H} = \mathbb{R}^d \times \Omega$, where $\Omega = \{1 : K\}$ is a finite discrete set. The corresponding observed time series is denoted by $\mathbf{Y}_{1:N}$ and each $\mathbf{Y}_n$ takes values in $\mathbb{R}^{d'}$.

1.1 Partially observed Markov process

Here we present the POMPs and their categorization.

Definition 1. *Partially observed Markov process (POMP)*

Let N be in $\mathbb{N}^*$, the pair $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ is a Partially observed Markov process (POMP) if:

$$p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathbf{h}_1, \mathbf{y}_1\right) p\left(\mathbf{h}_2, \mathbf{y}_2 \left| \mathbf{h}_1, \mathbf{y}_1\right.\right) \ldots p\left(\mathbf{h}_N, \mathbf{y}_N \left| \mathbf{h}_{N-1}, \mathbf{y}_{N-1}\right.\right), \tag{1.1}$$

which means that the pair $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ is Markovian.

Let us present a categorization of POMPs:

- $\mathrm{Card}(\Omega) = 1$ and $d > 0$. In this case, we have $\mathscr{H} = \mathbb{R}^d$ up to a bijection and such a POMP is called a continuous-state POMP;
- $1 \leq \mathrm{Card}(\Omega) < \infty$ and $d = 0$. In this case, we have $\mathscr{H} = \Omega$ up to a bijection and such a POMP is called a finite-discrete-state POMP;
- $0 \leq \mathrm{Card}(\Omega) < \infty$ and $d \geq 0$. Such a POMP is called a hybrid-state POMP. In the literature, these models may also be called switching processes (systems), jump processes (systems), interacting multimodels and so on.

For the rest of the report, we consider the following decomposition of $\mathbf{H}_{1:N}$ into a continuous-valued component $\mathbf{X}_{1:N}$ and a finite-discrete-valued component $\mathrm{R}_{1:N}$:

$$\forall n \in \{1 : N\}, \mathbf{H}_n = (\mathbf{X}_n, \mathrm{R}_n). \tag{1.2}$$

Therefore,

- In a continuous-state POMP, the state-signal process is denoted as $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$;

- In a finite-discrete-state POMP, the state-signal process is denoted as $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$;
- In a hybrid-state POMP, the state-signal process is denoted as $(\mathrm{R}_{1:N}, \mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$.

In the literature, the finite-discrete-state POMP is known as the Pairwise Markov Model (PMM), [Pieczynski, 2003]. The PMM may also be seen as a generalization of the Hidden Markov Model (HMM). The hybrid-state POMPs are sometimes referred as "triplet systems". They may be seen as a simultaneous generalization of the continuous-state and finite-discrete-state POMPs.

1.2 Bayesian state estimation

Here we consider a hybrid-state POMP $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ defined by the distribution of the pair $(\mathbf{H}_1, \mathbf{Y}_1)$ and the transition kernel

$$\forall n \in \{1 : N-1\}, p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{h}_n, \mathbf{y}_n \right.\right). \tag{1.3}$$

We present general algorithms of Bayesian filtering, smoothing and forecasting.

1.2.1 Bayesian filtering

For each n in $\mathbb{N}^*$, let $(\mathbf{H}_{1:n}, \mathbf{Y}_{1:n})$ be a POMP. The Probability Density Function (pdf) $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n} \right.\right)$ is called the filtering distribution. One may make use of the Markovianity of $(\mathbf{H}_{1:n}, \mathbf{Y}_{1:n})$ in order to compute the filtering distribution as follows:

— Initialization: we have

$$p\left(\mathbf{h}_1 \left| \mathbf{y}_1 \right.\right) = \frac{p\left(\mathbf{h}_1, \mathbf{y}_1\right)}{\int p\left(\mathbf{h}_1, \mathbf{y}_1\right) \mathrm{d}\mathbf{h}_1}. \tag{1.4}$$

— Iterative part : suppose that $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n} \right.\right)$ is given, then $p\left(\mathbf{h}_{n+1} \left| \mathbf{y}_{1:n+1} \right.\right)$ is classically computed in three steps:

1. Compute the following one-step predictive distribution:

$$p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) = \int p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n} \right.\right) p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{h}_n, \mathbf{y}_n \right.\right) \mathrm{d}\mathbf{h}_n; \tag{1.5}$$

2. Compute the likelihood coefficient at $n+1$:

$$c_{n+1} = p\left(\mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) = \int p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) \mathrm{d}\mathbf{h}_{n+1}; \tag{1.6}$$

3. Update the filtering distribution:

$$p\left(\mathbf{h}_{n+1} \left| \mathbf{y}_{1:n+1} \right.\right) = \frac{p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right)}{c_{n+1}}. \tag{1.7}$$

This iterative method allows computing the log-likelihood of $\mathbf{y}_{1:N}$. We have:

$$\log p\left(\mathbf{y}_{1:N}\right) = \log \Big(p\left(\mathbf{y}_1\right) \prod_{n=1}^{N-1} p\left(\mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) \Big) = \log p\left(\mathbf{y}_1\right) + \sum_{n=1}^{N-1} \log c_{n+1}. \tag{1.8}$$

This allows a maximum likelihood parameter estimation of POMPs by using the tools of the numerical analysis. Recall that the maximum likelihood estimator is defined by:

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}\left(\mathbf{y}_{1:N}\right). \tag{1.9}$$

Such an estimator is convergent and asymptotically efficient [Wasserman, 2004, Douc et al., 2004, Douc and Matias, 2001, Douc et al., 2011].

The filtering distribution is generally not available exactly. Indeed, diverse POMP submodels presented in Figure 1 allow the following cases :

- The models of type (C6) are classic linear Gaussian state-space systems. The Kalman filter allows exact Bayesian filtering in these models [Cappé et al., 2005];

- The models of type (C5) are pairwise-linear Gaussian models [Gorynin et al., 2016a]. They may be seen as a generalization of (C6). A modified version of the Kalman filter allows exact Bayesian filtering in these models;

- The models of type (C4) are linear, non-Gaussian state-space systems [Harvey and Luati, 2014]. The filtering distribution in such models is generally not available exactly [Cappé et al., 2005];

- The models of type (C3) are pairwise-linear, non-Gaussian state-space systems. The filtering distribution in such models is generally not available exactly;

- The models of type (C2) are non-linear non-Gaussian state-space systems. The filtering distribution in such models is generally not available exactly. The stochastic volatility model [Jacquier et al., 1994, Jacquier et al., 2002] is an example of a model of type (C2);

- The models of type (C1) are pairwise-non-linear and non-Gaussian. The filtering distribution in such models is generally not available exactly. The asymmetric stochastic volatility model [Centeno and Salido, 2009] is an example of a model of type (C1);

- The models of type (D2) are classic HMMs with a finite-discrete state space. The forward-backward algorithm allows exact Bayesian state estimation in such models [Cappé et al., 2005];

- The models of type (D1) are known as PMMs. They can be seen as a generalization of (D2). A modified version of the forward-backward algorithm allows exact Bayesian state estimation in such models [Pieczynski, 2003];

- A model of type (H5) is a hybrid-state POMP which is linear Gaussian state-space conditional on $\mathrm{R}_{1:N}$. Such a model is also known as a Switching Linear Dynamical System (SLDS) and Conditionally Gaussian Linear State-Space Model (CGLSSM) [Cappé et al., 2005]). The filtering distribution is generally not available exactly in such models;

- A model of type (H3) is a hybrid-state POMP which is pairwise-linear Gaussian conditional on $\mathrm{R}_{1:N}$. Such a model is also known as Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM) [Abbassi et al., 2015]. The filtering distribution is generally not available exactly in such models;

- Models of type (H2) represent the Conditionally Markov Switching Hidden Linear Model (CMSHLM), where one can compute exactly $p\left(\mathrm{r}_n \left|\mathbf{y}_{1:n}\right.\right)$, $p\left(\mathrm{r}_n \left|\mathbf{y}_{1:N}\right.\right)$ and the first two moments of $p\left(\mathbf{x}_n \left|\mathrm{r}_n, \mathbf{y}_{1:n}\right.\right)$, $p\left(\mathbf{x}_n \left|\mathrm{r}_n, \mathbf{y}_{1:N}\right.\right)$ [Pieczynski, 2011a].

- Models of type (H4), represent the Conditionally Gaussian Observed Markov Switching Model (CGOMSM). They are submodels of (H2) and (H3) simultaneously [Abbassi et al., 2015, Gorynin et al., 2017a].

- A model of type (H1) is a hybrid-state POMP which is a non-linear, non-Gaussian state-space system conditional on $\mathrm{R}_{1:N}$. The filtering distribution in such models is generally not available exactly. The switching stochastic volatility model [So et al., 1998, Carvalho and Lopes, 2007] is an example of a model of type (H1);

- Finally, the most general POMPs are hybrid-state POMPs which are not necessarily Gaussian nor linear conditional on $\mathrm{R}_{1:N}$. The switching asymmetric stochastic volatility model [Gorynin et al., 2016c] is an example of such a model.

1.2.2 Bayesian smoothing

Let N in $\mathbb{N}^*$, $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ be a POMP. The pdf $p(\mathbf{h}_{n:N} | \mathbf{y}_{1:N})$ is called the smoothing distribution.

One may make use of the Markovianity of $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ in order to compute the smoothing distribution by the following recursion: given the filtering distribution $p(\mathbf{h}_n | \mathbf{y}_{1:n})$ and the smoothing distribution $p(\mathbf{h}_{n+1:N} | \mathbf{y}_{1:N})$, we compute $p(\mathbf{h}_{n:N} | \mathbf{y}_{1:N})$ by:

- Compute :

$$p(\mathbf{h}_n | \mathbf{h}_{n+1}, \mathbf{y}_{1:n+1}) = \frac{p(\mathbf{h}_n | \mathbf{y}_{1:n}) \, p(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} | \mathbf{h}_n, \mathbf{y}_n)}{\int p(\mathbf{h}_n | \mathbf{y}_{1:n}) \, p(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} | \mathbf{h}_n, \mathbf{y}_n) \, \mathrm{d}\mathbf{h}_n}; \tag{1.10}$$

- Compute :

$$p(\mathbf{h}_{n:N} | \mathbf{y}_{1:N}) = p(\mathbf{h}_n | \mathbf{h}_{n+1}, \mathbf{y}_{1:n+1}) \, p(\mathbf{h}_{n+1:N} | \mathbf{y}_{1:N}). \tag{1.11}$$

This recursion iterates backward and is initialized by filtering distribution $p(\mathbf{h}_N | \mathbf{y}_{1:N})$. The exact smoothing distribution is available in the same POMPs where an exact filtering distribution is available.

Bayesian smoothing is an essential component of diverse parameter estimation methods such as the Expectation-Maximization (EM), Stochastic Expectation-Maximization (SEM) and Iterative Conditional Estimation (ICE) [Banga et al., 1992, Delmas, 1995].

1.2.3 Bayesian forecasting

The predictive distribution in a POMP at the horizon $T \in \mathbb{N}^*$ is defined as the pdf $p(\mathbf{h}_{n+1:n+T}, \mathbf{y}_{n+1:n+T} | \mathbf{y}_{1:n})$.

Given the filtering distribution $p(\mathbf{h}_n | \mathbf{y}_{1:n})$, the predictive distribution at the horizon 1 is given by (1.5). One may make use of the Markovianity of $(\mathbf{H}_{1:n+t}, \mathbf{Y}_{1:n+t})$ in order to compute the predictive distribution as follows. For each t in $\{1 : T-1\}$, the filtering distribution at the horizon $t+1$ is given by :

$$\begin{aligned} & p(\mathbf{h}_{n+1:n+t+1}, \mathbf{y}_{n+1:n+t+1} | \mathbf{y}_{1:n}) = \\ & \quad p(\mathbf{h}_{n+1:n+t}, \mathbf{y}_{n+1:n+t} | \mathbf{y}_{1:n}) \, p(\mathbf{h}_{n+t+1}, \mathbf{y}_{n+t+1} | \mathbf{h}_{n+t}, \mathbf{y}_{n+t}). \end{aligned} \tag{1.12}$$

The ability of a POMP to accurately forecast the signal is extremely important for practical applications. The accuracy of the forecast of the model is often the main criterion of the model selection.

1.3 Parameter estimation

In this section, we consider a POMP $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ defined by the pdf $p_{\boldsymbol{\theta}}(\mathbf{h}_1, \mathbf{y}_1)$ and the transition kernel

$$\forall n \in 1 : N-1, p_{\boldsymbol{\theta}}(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} | \mathbf{h}_n, \mathbf{y}_n), \tag{1.13}$$

where $p_{\boldsymbol{\theta}}(.)$, $p_{\boldsymbol{\theta}}(.|.)$ mean that the value of the pdf depends upon the value of $\boldsymbol{\theta}$, which is the parameter vector of the model. In this subsection, we recall diverse computational approaches for estimating $\boldsymbol{\theta}$ from $\mathbf{y}_{1:N}$ or $(\mathbf{h}_{1:N}, \mathbf{y}_{1:N})$. We distinguish the following two cases:

— A supervised estimation consists in estimating $\boldsymbol{\theta}$ from $(\mathbf{h}_{1:N}, \mathbf{y}_{1:N})$;

— An unsupervised estimation consists in estimating $\boldsymbol{\theta}$ from $\mathbf{y}_{1:N}$.

1.3.1 Supervised estimation

A supervised estimator can be a maximum likelihood estimator, defined by:

$$\widehat{\boldsymbol{\theta}}_{\text{SUP}}(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}) = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right). \tag{1.14}$$

In the case where this estimator is not available exactly, one can maximize the model's likelihood by using the tools of the numerical analysis. Specifically, it consists in defining the objective function

$$\boldsymbol{\theta} \to \log p_{\boldsymbol{\theta}}\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) = \log p_{\boldsymbol{\theta}}\left(\mathbf{h}_1, \mathbf{y}_1\right) + \sum_{n=1}^{N-1} \log p_{\boldsymbol{\theta}}\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left|\mathbf{h}_n, \mathbf{y}_n\right.\right). \tag{1.15}$$

in order to maximize it over a given set Θ of acceptable parameter values.

1.3.2 Unsupervised estimation

In the context of the unsupervised estimation, the maximum likelihood estimator is defined by :

$$\widehat{\boldsymbol{\theta}}_{\text{UNSUP}}(\mathbf{y}_{1:N}) = \arg\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}\left(\mathbf{y}_{1:N}\right). \tag{1.16}$$

This estimator is generally not available exactly. However, one can define the following objective function, known as the log-likelihood function:

$$\boldsymbol{\theta} \to \log p_{\boldsymbol{\theta}}\left(\mathbf{y}_1\right) + \sum_{n=1}^{N-1} \log c_{n+1}(\boldsymbol{\theta}), \tag{1.17}$$

where for each n in $\{1 : N-1\}$, $c_{n+1}(\boldsymbol{\theta}) = p_{\boldsymbol{\theta}}\left(\mathbf{y}_{n+1} \left|\mathbf{y}_{1:n}\right.\right)$ is computed by using (1.6) as an output of the Bayesian filtering procedure. Next, one maximizes this function by using the tools of the numerical analysis.

Alternatively, there exist iterative unsupervised estimation algorithms such as the EM [Dempster et al., 1977], SEM [Celeux and Govaert, 1992] and ICE [Banga et al., 1992]. All these methods require an initial guess, denoted by $\widehat{\boldsymbol{\theta}}^{(0)}$, which may be chosen at random or determined somehow from $\mathbf{y}_{1:N}$. These methods are based on the fixed-point principle which means that they look for a value (or for a pdf in the case of the stochastic ICE and SEM) invariant to the transformation of the form:

$$\boldsymbol{\theta} \to \kappa\Big(\boldsymbol{\theta}, \mathbf{y}_{1:N}\Big). \tag{1.18}$$

depending on the method considered.

The sequence of parameter estimates $\Big(\widehat{\boldsymbol{\theta}}^{(1)}, \widehat{\boldsymbol{\theta}}^{(2)}, \ldots, \widehat{\boldsymbol{\theta}}^{(q)}\Big)$ defined by :

$$\forall k \geq 0, \widehat{\boldsymbol{\theta}}^{(k+1)} = \kappa\Big(\widehat{\boldsymbol{\theta}}^{(k)}, \mathbf{y}_{1:N}\Big), \tag{1.19}$$

is supposed to converge to or to hover around some value which is then seen as the parameter estimate produced by the method.

— In the case of the EM algorithm, the transformation (1.18) is :

$$\kappa\Big(\widehat{\boldsymbol{\theta}}^{(k)}, \mathbf{y}_{1:N}\Big) = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\widehat{\boldsymbol{\theta}}^{(k)}}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{H}_{1:N}, \mathbf{y}_{1:N}\right) \left|\mathbf{y}_{1:N}\right.\right]. \tag{1.20}$$

— In the case of the SEM algorithm, the transformation (1.18) is :

$$\kappa\Big(\widehat{\boldsymbol{\theta}}^{(k)}, \mathbf{y}_{1:N}\Big) = \widehat{\boldsymbol{\theta}}_{\text{SUP}}(\widetilde{\mathbf{H}}_{1:N}, \mathbf{y}_{1:N}), \qquad \widetilde{\mathbf{H}}_{1:N} \sim p_{\widehat{\boldsymbol{\theta}}^{(k)}}\left(\mathbf{h}_{1:N} \left|\mathbf{y}_{1:N}\right.\right). \tag{1.21}$$

where $\widehat{\boldsymbol{\theta}}_{\text{SUP}}$ is the maximum likelihood supervised estimator.

– In the case of the ICE algorithm, the transformation (1.18) is :

$$\kappa\left(\widehat{\boldsymbol{\theta}}^{(k)}, \mathbf{y}_{1:N}\right) = \mathbb{E}_{\widehat{\boldsymbol{\theta}}^{(k)}}\left[\widehat{\boldsymbol{\theta}}_{\text{SUP}}(\mathbf{H}_{1:N}, \mathbf{y}_{1:N})\,|\mathbf{y}_{1:N}\right], \tag{1.22}$$

where $\widehat{\boldsymbol{\theta}}_{\text{SUP}}$ is a supervised estimator. In the case where the above expression cannot be computed exactly, one may use a Monte-Carlo method as an approximation, which defines a stochastic ICE.

Let us also outline the Markov chain Monte Carlo (MCMC) methods, which are particularly efficient in the context of machine learning [Andrieu et al., 2003b, Andrieu et al., 2010].

1.4 Sequential Monte-Carlo methods

In this section, we present the sequential Monte-Carlo methods [Doucet and Johansen, 2011, Ristic et al., 2004, Carpenter et al., 1999, Andrieu and Doucet, 2002].

The sequential Monte-Carlo methods are used in POMPs where an exact Bayesian state estimation is not possible. The idea is to sample $M \in \mathbb{N}^*$ particles $\{\mathbf{h}_n^{(m)}\}_{1\leq m\leq M, n\in\mathbb{N}^*}$ in order to approximate the filtering or smoothing distribution. These methods do generally converge to the exact Bayesian solution when M tends towards infinity.

These methods realize the principle of importance sampling [Geweke, 1989], which is to sample particles according to a proposal density and then to attribute weights to them in order to correct the deviation of the proposal density from the posterior density. However, the sequential Monte-Carlo methods do not apply this principle directly, since most of the weights tend to zero and only few of them have significant weights. Thus, the importance sampling becomes less and less efficient due to the necessity to process the particles which do not contribute to estimating the posterior density. This effect is known as the weight degeneracy [Cappé et al., 2005, Del Moral and Jacod, 2001]. The most widely used approach to overcome the weight degeneracy is to implement the Sampling Importance Resampling (SIR)[Doucet et al., 2000], which means to resample each particle with the probability proportional to its weight. This produces a range of SIR-based sequential Monte-Carlo methods [Douc and Cappe, 2005, Li et al., 2015]. Indeed, the resampling stage removes the particle with low weights, and resamples the others multiple times, which creates a sort of dependency among the resampled particles and increases the variance of the estimate. Several approaches can overcome this increase of variance [Beskos et al., 2017, Verge et al., 2015, Lindsten et al., 2017]. Moreover, the sequential Monte-Carlo methods may have a heavy computational load in the case of high-dimensional state estimation [Snyder et al., 2008, Ades and Van Leeuwen, 2015, Rebeschini et al., 2015].

1.4.1 Particle filter

Here we describe a simple SIR-based Particle Filter (PF), which is a widely used sequential Monte-Carlo method to access the filtering distribution in a POMP. Let $M \in \mathbb{N}^*$ be the number of particles to sample according to $p(\mathbf{h}_n\,|\mathbf{y}_{1:n})$, the SIR-PF consists in repeating the following steps. For each $n \geq 0$:

1. For each m in $\{1 : M\}$, sample $\widetilde{\mathbf{h}}_{n+1}^{(m)}$ from $p\left(\mathbf{h}_{n+1}\left|\mathbf{h}_n^{(m)}, \mathbf{y}_n\right.\right)$ if $n > 0$, otherwise sample $\widetilde{\mathbf{h}}_1^{(m)}$ from $p(\mathbf{h}_1)$;

2. For each m in $\{1 : M\}$, compute $\eta_{n+1}^{(m)} = p\left(\mathbf{y}_{n+1}\left|\widetilde{\mathbf{h}}_{n+1}^{(m)}, \mathbf{h}_n^{(m)}, \mathbf{y}_n\right.\right)$;

3. Sample $\{\mathbf{h}_{n+1}^{(m)}\}_{1\leq m\leq M}$ by resampling $\{\widetilde{\mathbf{h}}_{n+1}^{(m)}\}_{1\leq m\leq M}$ with probabilities proportional to $\{\eta_{n+1}^{(m)}\}_{1\leq m\leq M}$;

Thus, for each n in $\{1:N\}$, the filtering distribution $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right)$ is approximated by

$$p\left(\mathbf{h}_n \left| \mathbf{y}_{1:n}\right.\right) \approx \frac{1}{M}\sum_{m=1}^{M} \delta\Big(\mathbf{h}_n - \mathbf{h}_n^{(m)}\Big),$$

where δ denotes the Dirac distribution.

1.4.2 Particle smoothing

A Particle Smoother (PS) is a sequential Monte-Carlo method to access the smoothing distribution $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:N}\right.\right)$ in a POMP. It approximates, for each n in $\{1:N\}$, the smoothing distribution by

$$p\left(\mathbf{h}_n \left| \mathbf{y}_{1:N}\right.\right) \approx \frac{1}{M}\sum_{m=1}^{M} \delta\Big(\mathbf{h}_n - \mathbf{h}_n'^{(m)}\Big),$$

where $M \in \mathbb{N}^*$ is the number of particles to sample according to $p\left(\mathbf{h}_n \left| \mathbf{y}_{1:N}\right.\right)$.

Here we consider the most known PS which is the forward-backward smoother [Briers et al., 2010]. We suppose that we have already sampled particles $\{\mathbf{h}_n^{(m)}\}_{1\leq m\leq M, 1\leq n\leq N}$ according to the filtering distribution as it was presented previously.

The particles $\{\mathbf{h}_n'^{(m)}\}_{1\leq m\leq M, 1\leq n\leq N}$ representing the smoothing distribution are obtained as follows:

— Initialization: For each m in $\{1:M\}$, let

$$\mathbf{h}_N'^{(m)} = \mathbf{h}_N^{(m)}.$$

— For each n in $\{1:N-1\}$, iterate:

1. For each m in $1:M$, compute $\eta_n'^{(m)} = p\left(\mathbf{h}_{n+1}'^{(m)}) \left| \mathbf{h}_n^{(m)}, \mathbf{y}_n\right.\right)$;
2. Sample $\{\mathbf{h}_n'^{(m)}\}_{1\leq m\leq M}$ by resampling $\{\mathbf{h}_n^{(m)}\}_{1\leq m\leq M}$ with probabilities proportional to $\{\eta_n'^{(m)}\}_{1\leq m\leq M}$.

1.4.3 Monte-Carlo forecasting

The Monte-Carlo forecasting is an approach to approximate, for each n in $\mathbb{N}^*$, the predictive distribution at horizon $T \in \mathbb{N}^*$ defined by $p\left(\mathbf{h}_{n+1:n+T}, \mathbf{y}_{n+1:n+T} \left| \mathbf{y}_{1:n}\right.\right)$, as follows:

$$p\left(\mathbf{h}_{n+1:n+T}, \mathbf{y}_{n+1:n+T} \left| \mathbf{y}_{1:n}\right.\right) \approx \frac{1}{M}\sum_{m=1}^{M} \delta\Big(\mathbf{h}_{n+1:n+T} - \mathbf{h}_{n+1:n+T}''^{(m)}\Big)\delta\Big(\mathbf{y}_{n+1:n+T} - \mathbf{y}_{n+1:n+T}''^{(m)}\Big),$$

where $M \in \mathbb{N}^*$ is the number of particles to sample according to $p\left(\mathbf{h}_{n+1:n+T}, \mathbf{y}_{n+1:n+T} \left| \mathbf{y}_{1:n}\right.\right)$. Here, each particle represents a trajectory of type $(\mathbf{h}_{n+1:n+T}, \mathbf{y}_{n+1:n+T})$.

In the POMP framework, we suppose that we have already sampled $\{\mathbf{h}_n^{(m)}\}_{1\leq m\leq M, 1\leq n\leq N}$ according to the filtering distribution as it was presented previously.

The particles $\{(\mathbf{h}_{n+1:n+T}''^{(m)}, \mathbf{y}_{n+1:n+T}''^{(m)})\}_{1\leq m\leq M, 1\leq n\leq N}$ are sampled as follows:

— Initialisation: For each m in $\{1:M\}$, sample $(\mathbf{h}_{n+1:n+1}''^{(m)}, \mathbf{y}_{n+1:n+1}''^{(m)})$ from $p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{h}_n^{(m)}, \mathbf{y}_n\right.\right)$;

— For each t in $\{1:T-1\}$, sample the particles according to the predictive distribution at horizon $t+1$ by using those which were sampled according to the predictive distribution at horizon t. Thus, for each m in $\{1:M\}$, sample $(\mathbf{h}_{n+1:n+t+1}''^{(m)}, \mathbf{y}_{n+1:n+t+1})$ from $p\left(\mathbf{h}_{n+t+1}, \mathbf{y}_{n+t+1} \left| \mathbf{h}_{n+t}''^{(m)}, \mathbf{y}_{n+t}''^{(m)}\right.\right)$.

1.5 Conclusion

We have presented the POMP framework and the sequential Monte-Carlo methods, which are widely used Bayesian state estimation approaches. These methods are generally based on the SIR principle. They are convergent asymptotically, but may need a considerable computational cost. The rest of the report is devoted to the alternative methods of state estimation in POMPs, which should allow an accurate state estimation for a low computational cost. The accuracy of these methods will be compared with that of sequential Monte-Carlo methods.

Chapter 2

Conditionally Gaussian observed Markov switching models

Let $\mathrm{R}_{1:N}$ be a random sequence taking its values in $\Omega = \{1 : K\}$, $\mathbf{X}_{1:N}$ and $\mathbf{Y}_{1:N}$ random sequences taking their values in $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ respectively, with $d \in \mathbb{N}^*$, $d' \in \mathbb{N}^*$. $\mathrm{R}_{1:N}$ and $\mathbf{X}_{1:N}$ are hidden and $\mathbf{Y}_{1:N}$ is observed.

In this chapter we first define the Conditionally Markov Switching Hidden Linear Model (CMSHLM) [Pieczynski, 2011a], then we move on to the Conditionally Gaussian Observed Markov Switching Model (CGOMSM) proposed in [Abbassi et al., 2015]. Next, we present the related exact Bayesian state estimation algorithms in Section 2.2. The author's contribution is given in Section 2.3. Extensive experiments on synthetic and real-world data are presented in Section 2.4.

2.1 Model definition and properties

Definition 2. *CMSHLM*

Let $\mathbf{X}_{1:N}$, $\mathrm{R}_{1:N}$ and $\mathbf{Y}_{1:N}$ be random sequences as specified previously. The triplet $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is said to be a CMSHLM if

$$(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N}) \text{ is Markovian;} \tag{2.1a}$$

$$\forall n \in \{1 : N-1\}, p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_n, \mathbf{y}_n \right.\right) = p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n \right.\right); \tag{2.1b}$$

$$\begin{aligned}&\forall n \in \{1 : N-1\}, \mathbf{X}_{n+1} = \mathbf{F}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})\mathbf{X}_n + \\ &\quad \mathbf{G}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})\boldsymbol{W}_{n+1} + \mathbf{T}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1}),\end{aligned} \tag{2.1c}$$

with $\mathbf{F}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$, $\mathbf{G}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$ matrices of appropriate dimensions, $\boldsymbol{W}_{1:N}$ is a zero-mean white noise and $\mathbf{T}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$ vectors of appropriate dimension.

Figure 2.1 represents the dependency graph of CMSHLM.

The next definition concerns a particular CGOMSM used in this report. The general definition of the CGOMSM is given in [Abbassi et al., 2015].

Definition 3. *CGOMSM*

Let $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ be a stationary Markov triplet and define, for each n in $\{1 : N\}$,

$$\mathbf{Z}_n = \begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix}; \tag{2.2}$$

We say that $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is a CGOMSM if for each n in $\{1 : N-1\}$, $\mathrm{r}_{n:n+1}$ in Ω^2,

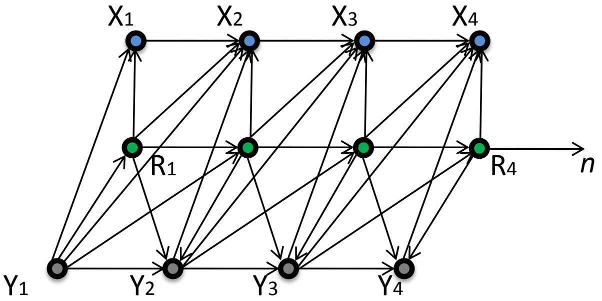


Figure 2.1: Dependency graph of CMSHLM.

— $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left|\mathrm{r}_{n:n+1}\right.\right)$ is Gaussian:

$$p(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1}|\mathrm{r}_{n:n+1}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{z}_n^\top \ \mathbf{z}_{n+1}^\top\end{bmatrix}^\top, \boldsymbol{\Upsilon}(\mathrm{r}_{n:n+1}), \boldsymbol{\Xi}(\mathrm{r}_{n:n+1})\right); \tag{2.3}$$

— the mean of $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left|\mathrm{r}_{n:n+1}\right.\right)$ is of the form

$$\boldsymbol{\Upsilon}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \mathbf{M}(\mathrm{r}_n) \\ \mathbf{M}(\mathrm{r}_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbb{E}\left[\mathbf{Z}_n \left|\mathrm{r}_n\right.\right] \\ \mathbb{E}\left[\mathbf{Z}_{n+1} \left|\mathrm{r}_{n+1}\right.\right] \end{bmatrix}; \tag{2.4}$$

— the variance matrix of $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left|\mathrm{r}_{n:n+1}\right.\right)$ is of the form

$$\boldsymbol{\Xi}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \boldsymbol{S}(\mathrm{r}_n) & \boldsymbol{\Sigma}(\mathrm{r}_{n:n+1}) \\ \boldsymbol{\Sigma}^\top(\mathrm{r}_{n:n+1}) & \boldsymbol{S}(\mathrm{r}_{n+1}) \end{bmatrix}; \tag{2.5}$$

— $p\left(\mathbf{x}_{n:n+1}, \mathbf{y}_{n:n+1} \left|\mathrm{r}_{n:n+1}\right.\right)$ is such that

$$p\left(\mathbf{y}_{n+1} \left|\mathbf{x}_n, \mathrm{r}_{n:n+1}, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n:n+1}, \mathbf{y}_n\right.\right). \tag{2.6}$$

Proposition 1. *A CGOMSM is a CMSHLM with* $\mathbf{F}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$, $\mathbf{G}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$ *and* $\mathbf{T}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$ *given by* (2.15).

Proof. $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is Markovian in CGOMSM, thus (2.1a) is verified. According to (2.3) - (2.5), we have $p\left(\mathrm{r}_{n+1} \left|\mathbf{x}_n, \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left|\mathrm{r}_n\right.\right)$. We then use (2.6) to prove that a CGOMSM has property (2.1b) of the CMSHLM.

To find out the corresponding $\mathbf{F}_{n+1}$, $\mathbf{G}_{n+1}$ and $\mathbf{T}_{n+1}$ in (2.1c), we set

$$\boldsymbol{A}(\mathrm{r}_{n:n+1}) = \boldsymbol{\Sigma}^\top(\mathrm{r}_{n:n+1})\ \boldsymbol{S}^{-1}(\mathrm{r}_n), \tag{2.7}$$

and consider $\boldsymbol{B}(\mathrm{r}_{n:n+1})$ and $\boldsymbol{Q}(\mathrm{r}_{n:n+1})$ such that

$$\boldsymbol{B}(\mathrm{r}_{n:n+1})\boldsymbol{B}^\top(\mathrm{r}_{n:n+1}) = \boldsymbol{\Sigma}^\top(\mathrm{r}_{n:n+1})\boldsymbol{S}^{-1}(\mathrm{r}_n)\boldsymbol{\Sigma}(\mathrm{r}_{n:n+1}), \tag{2.8}$$

$$\boldsymbol{Q}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \boldsymbol{Q}_1(\mathrm{r}_{n:n+1}) & \boldsymbol{Q}_2(\mathrm{r}_{n:n+1}) \\ \boldsymbol{Q}_3(\mathrm{r}_{n:n+1}) & \boldsymbol{Q}_4(\mathrm{r}_{n:n+1}). \end{bmatrix} = \boldsymbol{B}(\mathrm{r}_{n:n+1})\boldsymbol{B}^\top(\mathrm{r}_{n:n+1}). \tag{2.9}$$

Equation (2.6) induces that the matrix $\boldsymbol{A}(\mathrm{r}_{n:n+1})$ has the following form:

$$\boldsymbol{A}(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \boldsymbol{A}_1(\mathrm{r}_{n:n+1}) & \boldsymbol{A}_2(\mathrm{r}_{n:n+1}) \\ \mathbf{0} & \boldsymbol{A}_4(\mathrm{r}_{n:n+1}) \end{bmatrix}. \tag{2.10}$$

Hence, we can state that the discrete time process $\mathbf{Z}_{1:N}$ satisfies the following recursion equation:

$$\mathbf{Z}_{n+1} = \boldsymbol{A}(\mathrm{R}_{n:n+1})\Big(\mathbf{Z}_n - \boldsymbol{M}(\mathrm{R}_n)\Big) + \boldsymbol{B}(\mathrm{R}_{n:n+1})\boldsymbol{W}_{n+1} + \boldsymbol{M}(\mathrm{R}_{n+1}), \tag{2.11}$$

where $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_N$ are Gaussian unit-variance white noise vectors.

We split $\mathbf{M}(\mathrm{r}_n)$ as $\mathbf{M}(\mathrm{r}_n) = \left[\boldsymbol{M}_1(\mathrm{r}_n)^\top \ \boldsymbol{M}_2(\mathrm{r}_n)^\top\right]^\top$. Next, $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_{n:n+1}, \mathbf{y}_n \right.\right)$ is a multivariate normal distribution with variance matrix $\boldsymbol{Q}(\mathrm{r}_{n:n+1})$ and mean vector given by

$$\begin{aligned}\boldsymbol{A}(\mathrm{r}_{n:n+1}) &\begin{bmatrix}\mathbf{x}_n \\ \mathbf{y}_n\end{bmatrix} + \begin{bmatrix}\boldsymbol{N}_1(\mathrm{r}_{n:n+1}) \\ \boldsymbol{N}_2(\mathrm{r}_{n:n+1})\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{A}_1(\mathrm{r}_{n:n+1})\mathbf{x}_n + \boldsymbol{A}_2(\mathrm{r}_{n:n+1})\mathbf{y}_n + \boldsymbol{N}_1(\mathrm{r}_{n:n+1}) \\ \boldsymbol{A}_4(\mathrm{r}_{n:n+1})\mathbf{y}_n + \boldsymbol{N}_2(\mathrm{r}_{n:n+1})\end{bmatrix},\end{aligned} \tag{2.12}$$

where we set

$$\begin{aligned}\boldsymbol{N}_1(\mathrm{r}_{n:n+1}) &= \boldsymbol{M}_1(\mathrm{r}_{n+1}) - \boldsymbol{A}_1(\mathrm{r}_{n:n+1})\boldsymbol{M}_1(\mathrm{r}_n) - \\ &\quad \boldsymbol{A}_2(\mathrm{r}_{n:n+1})\boldsymbol{M}_2(\mathrm{r}_n), \\ \boldsymbol{N}_2(\mathrm{r}_{n:n+1}) &= \boldsymbol{M}_2(\mathrm{r}_{n+1}) - \boldsymbol{A}_4(\mathrm{r}_{n:n+1})\boldsymbol{M}_2(\mathrm{r}_n).\end{aligned}$$

$p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1} \right.\right)$ is also a multivariate normal probability density function with mean vector

$$\begin{aligned}&\boldsymbol{Q}_2(\mathrm{r}_{n:n+1})\boldsymbol{Q}_4^{-1}(\mathrm{r}_{n:n+1})(\mathbf{y}_{n+1} - \boldsymbol{A}_4(\mathrm{r}_{n:n+1})\mathrm{y}_n - \boldsymbol{N}_2(\mathrm{r}_{n:n+1})) \\ &\quad + \boldsymbol{A}_1(\mathrm{r}_{n:n+1})\mathbf{x}_n + \boldsymbol{A}_2(\mathrm{r}_{n:n+1})\mathbf{y}_n + \boldsymbol{N}_1(\mathrm{r}_{n:n+1}),\end{aligned} \tag{2.13}$$

and variance matrix

$$\boldsymbol{Q}_1(\mathrm{r}_{n:n+1}) - \boldsymbol{Q}_2(\mathrm{r}_{n:n+1})\boldsymbol{Q}_4^{-1}(\mathrm{r}_{n:n+1})\boldsymbol{Q}_3(\mathrm{r}_{n:n+1}). \tag{2.14}$$

This allows to complete the proof and to specify $\mathbf{F}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$, $\mathbf{G}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$ and $\mathbf{T}_{n+1}(\mathrm{R}_{n:n+1}, \mathbf{Y}_{n:n+1})$:

$$\mathbf{F}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) = \boldsymbol{A}_1(\mathrm{r}_{n:n+1}), \tag{2.15a}$$

$$\begin{aligned}\mathbf{T}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) &= \boldsymbol{A}_2(\mathrm{r}_{n:n+1})\mathbf{y}_n + \mathbf{N}_1(\mathrm{r}_{n:n+1}) + \\ &\quad \boldsymbol{Q}_2(\mathrm{r}_{n:n+1})\boldsymbol{Q}_4^{-1}(\mathrm{r}_{n:n+1})(\mathbf{y}_{n+1} - \boldsymbol{A}_4(\mathrm{r}_{n:n+1})\mathbf{y}_n - \mathbf{N}_2(\mathrm{r}_{n:n+1})),\end{aligned} \tag{2.15b}$$

$$\begin{aligned}\mathbf{G}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1})\mathbf{G}_{n+1}^T(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) &= \\ \boldsymbol{Q}_1(\mathrm{r}_{n:n+1}) - \boldsymbol{Q}_2(\mathrm{r}_{n:n+1})\boldsymbol{Q}_4^{-1}(\mathrm{r}_{n:n+1})\boldsymbol{Q}_3(\mathrm{r}_{n:n+1}).\end{aligned} \tag{2.15c}$$

□

According to (2.3) - (2.5), we may state that for all n in $\{1 : N\}$,

$$p\left(\mathbf{x}_n, \mathbf{y}_n \left| \mathrm{r}_n, \mathrm{r}_{n+1} \right.\right) = p\left(\mathbf{x}_n, \mathbf{y}_n \left| \mathrm{r}_n \right.\right),$$

and thus $p\left(\mathrm{r}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_n, \mathbf{y}_n \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n \right.\right)$. This ensures that in CGOMSM seen as a subcase of CMSHLM, $\mathrm{R}_{1:N}$ is a Markov chain. Figure 2.2 represents the dependency graph of CGOMSM. In contrast with the dependency graph of CMSHLM, we have $p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n \right.\right)$, what removes the line between $\mathbf{Y}_n$ and R_{n+1}.

Proposition 2. *A CGOMSM can be represented as*

$$\mathbf{Y}_{n+1} = \boldsymbol{D}(\mathrm{r}_{n:n+1})\mathbf{Y}_n + \boldsymbol{H}(\mathrm{r}_{n:n+1}) + \boldsymbol{\Lambda}(\mathrm{r}_{n:n+1})\boldsymbol{V}_{n+1}; \tag{2.16a}$$

$$\begin{aligned}\mathbf{X}_{n+1} &= \boldsymbol{A}(\mathrm{r}_{n:n+1})\mathbf{X}_n + \boldsymbol{B}(\mathrm{r}_{n:n+1})\mathbf{Y}_n + \boldsymbol{C}(\mathrm{r}_{n:n+1})\mathbf{Y}_{n+1} \\ &\quad + \boldsymbol{F}(\mathrm{r}_{n:n+1}) + \boldsymbol{\Pi}(\mathrm{r}_{n:n+1})\boldsymbol{U}_{n+1},\end{aligned} \tag{2.16b}$$

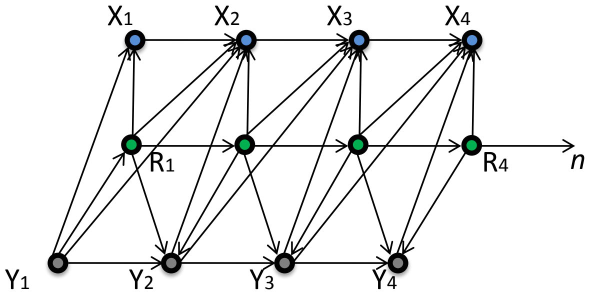


Figure 2.2: Dependency graph of CGOMSM.

where $\mathrm{R}_{1:N}$ *is a Markov chain,* $\boldsymbol{D}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{H}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{\Lambda}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{A}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{B}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{C}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{F}(\mathrm{r}_{n:n+1})$*,* $\boldsymbol{\Pi}(\mathrm{r}_{n:n+1})$ *are matrices defined by* (2.21)-(2.24) *and* $\boldsymbol{U}_{1:N}, \boldsymbol{V}_{1:N}$ *are standard independent and identically distributed Gaussian random vectors.*

Proof. First, $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is a Hidden Markov Model With Conditionally Correlated Noise (HMM-CN) with discrete state space. Thus,

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n \right.\right) p\left(\mathbf{y}_{n+1} \left| \mathrm{r}_{n:n+1}, \mathbf{y}_n \right.\right). \tag{2.17}$$

Second, (2.6) is equivalent to

$$p\left(\mathbf{x}_n \left| \mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1} \right.\right) = p\left(\mathbf{x}_n \left| \mathrm{r}_n, \mathbf{y}_n \right.\right), \tag{2.18}$$

Since the distribution $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n, \mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1} \right.\right)$ is Gaussian, $\mathbf{X}_{n+1}$ is Gaussian conditional on the pair $(\mathrm{R}_n, \mathrm{R}_{n+1})$ and on a linear combination of $\mathbf{X}_n$, $\mathbf{Y}_n$ and $\mathbf{Y}_{n+1}$. A similar reasoning holds for $p\left(\mathbf{y}_{n+1} \left| \mathbf{y}_n, \mathrm{r}_{n:n+1} \right.\right)$ and summarizing, we have (2.16).

Let us set $\boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n} = \mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n \right.\right]$ and $\boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n} = \mathbb{E}\left[\mathbf{Y}_n \left| \mathrm{r}_n \right.\right]$. It follows from (2.11) that $\left[\mathbf{X}^\top_{n+1}\ \mathbf{Y}^\top_{n+1}\right]^\top$ is normally distributed given $\left[\mathbf{X}^\top_n\ \mathbf{Y}^\top_n\right]^\top$ and $\mathrm{R}_{n:n+1}$. From (2.11), we find that the conditional mean of $\left[\mathbf{X}^\top_{n+1}\ \mathbf{Y}^\top_{n+1}\right]^\top$ is

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_{n+1}} \\ \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_{n+1}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{a}_1(\mathrm{r}_{n:n+1}) & \boldsymbol{a}_2(\mathrm{r}_{n:n+1}) \\ \boldsymbol{a}_3(\mathrm{r}_{n:n+1}) & \boldsymbol{a}_4(\mathrm{r}_{n:n+1}) \end{bmatrix} \begin{bmatrix} \mathbf{x}_n - \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n} \\ \mathbf{y}_n - \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_{n+1}} + \boldsymbol{a}_1(\mathrm{r}_{n:n+1})(\mathbf{x}_n - \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n}) + \boldsymbol{a}_2(\mathrm{r}_{n:n+1})(\mathbf{y}_n - \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n}) \\ \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_{n+1}} + \boldsymbol{a}_3(\mathrm{r}_{n:n+1})(\mathbf{x}_n - \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n}) + \boldsymbol{a}_4(\mathrm{r}_{n:n+1})(\mathbf{y}_n - \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n}) \end{bmatrix},
\end{aligned} \tag{2.19}$$

and that the conditional variance matrix of $\left[\mathbf{X}^\top_{n+1}\ \mathbf{Y}^\top_{n+1}\right]^\top$ is $\boldsymbol{b}(\mathrm{r}_{n:n+1})\boldsymbol{b}^\top(\mathrm{r}_{n:n+1})$, written in block-form as

$$\boldsymbol{b}(\mathrm{r}_{n:n+1})\boldsymbol{b}^\top(\mathrm{r}_{n:n+1}) = \begin{bmatrix} \boldsymbol{\gamma}_1(\mathrm{r}_{n:n+1}) & \boldsymbol{\gamma}_2(\mathrm{r}_{n:n+1}) \\ \boldsymbol{\gamma}_3(\mathrm{r}_{n:n+1}) & \boldsymbol{\gamma}_4(\mathrm{r}_{n:n+1}) \end{bmatrix}. \tag{2.20}$$

Since $\boldsymbol{a}_3(\mathrm{r}_{n:n+1}) = 0$ for each $\mathrm{r}_{n:n+1}$ in Ω^2, equation (2.16a) holds for

$$\boldsymbol{D}(\mathrm{r}_{n:n+1}) \quad = \boldsymbol{a}_4(\mathrm{r}_{n:n+1}), \tag{2.21a}$$

$$\boldsymbol{H}(\mathrm{r}_{n:n+1}) \quad = -\boldsymbol{a}_4(\mathrm{r}_{n:n+1})\boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n} + \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_{n+1}} \tag{2.21b}$$

and for some matrix $\boldsymbol{\Lambda}(\mathrm{r}_{n:n+1})$ such that

$$\boldsymbol{\Lambda}(\mathrm{r}_{n:n+1})\boldsymbol{\Lambda}^\top(\mathrm{r}_{n:n+1}) = \boldsymbol{\gamma}_4(\mathrm{r}_{n:n+1}). \tag{2.22}$$

Likewise, $\mathbf{X}_{n+1}$ is also normally distributed given $\mathbf{X}_n$, $\mathrm{R}_{n:n+1}$ and $\mathbf{Y}_{n:n+1}$. The conditional variance of $\mathbf{X}_{n+1}$ is

$$\boldsymbol{\gamma}_1(\mathrm{r}_{n:n+1}) - \boldsymbol{\gamma}_2(\mathrm{r}_{n:n+1})\boldsymbol{\gamma}_4^{-1}(\mathrm{r}_{n:n+1})\boldsymbol{\gamma}_2^\top(\mathrm{r}_{n:n+1}),$$

and its conditional mean is

$$\begin{aligned}&\boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_{n+1}} + \boldsymbol{a}_1(\mathrm{r}_{n:n+1})(\mathbf{x}_n - \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n}) + \boldsymbol{a}_2(\mathrm{r}_{n:n+1})(\mathbf{y}_n - \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n})\\ &\quad + \boldsymbol{\gamma}_2(\mathrm{r}_{n:n+1})\boldsymbol{\gamma}_4^{-1}(\mathrm{r}_{n:n+1})\big\{\mathbf{y}_{n+1} - (\boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_{n+1}} + \boldsymbol{a}_3(\mathrm{r}_{n:n+1})(\mathbf{x}_n - \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n})\\ &\quad + \boldsymbol{a}_4(\mathrm{r}_{n:n+1})(\mathbf{y}_n - \boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n}))\big\}.\end{aligned}$$

Term-by-term identification of (2.16b) with the equation above gives

$$\boldsymbol{C}(\mathrm{r}_{n:n+1}) \quad = \boldsymbol{\gamma}_2(\mathrm{r}_{n:n+1})\boldsymbol{\gamma}_4^{-1}(\mathrm{r}_{n:n+1}) \tag{2.23a}$$

$$\boldsymbol{A}(\mathrm{r}_{n:n+1}) \quad = \boldsymbol{a}_1(\mathrm{r}_{n:n+1}) - \boldsymbol{C}(\mathrm{r}_{n:n+1})\boldsymbol{a}_3(\mathrm{r}_{n:n+1}) \tag{2.23b}$$

$$\boldsymbol{B}(\mathrm{r}_{n:n+1}) \quad = \boldsymbol{a}_2(\mathrm{r}_{n:n+1}) - \boldsymbol{C}(\mathrm{r}_{n:n+1})\boldsymbol{a}_4(\mathrm{r}_{n:n+1}) \tag{2.23c}$$

$$\boldsymbol{F}(\mathrm{r}_{n:n+1}) \quad = \boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_{n+1}} - \boldsymbol{A}(\mathrm{r}_{n:n+1})\boldsymbol{M}^{\mathbf{X}}_{\mathrm{r}_n} - \boldsymbol{B}(\mathrm{r}_{n:n+1})\boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_n} - \boldsymbol{C}(\mathrm{r}_{n:n+1})\boldsymbol{M}^{\mathbf{Y}}_{\mathrm{r}_{n+1}} \tag{2.23d}$$

and $\boldsymbol{\Pi}(\mathrm{r}_{n:n+1})$ is a matrix such that

$$\boldsymbol{\Pi}(\mathrm{r}_{n:n+1})\boldsymbol{\Pi}^{\top}(\mathrm{r}_{n:n+1}) = \boldsymbol{\gamma}_1(\mathrm{r}_{n:n+1}) - \boldsymbol{C}(\mathrm{r}_{n:n+1})\boldsymbol{\gamma}_2^{\top}(\mathrm{r}_{n:n+1}). \tag{2.24}$$

□

The distribution of $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ in stationary CGOMSM is of the form

$$p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) = \sum_{1\le i,j\le K} \alpha_{ij} p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2), \tag{2.25}$$

where $\{\alpha_{ij}\}_{1\le i,j\le K}$ are positive scalars which sum up to one and for each (i,j) in $\{1:K\}^2$, $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ is a Gaussian pdf.

Proposition 3. *For each (i,j) in $\{1:K\}^2$, $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ in (2.25) satisfies one of the two following equivalent properties*

$$p_{ij}(\mathbf{y}_2 \,|\mathbf{x}_1, \mathbf{y}_1) = p_{ij}(\mathbf{y}_2 \,|\mathbf{y}_1); \tag{2.26}$$

$$\boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_2}(ij) = \boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_1}(ij)\,\boldsymbol{\Gamma}^{-1}_{\mathbf{Y}_1}(ij)\,\boldsymbol{\Sigma}_{\mathbf{Y}_1\mathbf{Y}_2}(ij), \tag{2.27}$$

where $\boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_2}(ij) \in \mathbb{R}^{d\times d'}$, $\boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_1}(ij) \in \mathbb{R}^{d\times d'}$, $\boldsymbol{\Gamma}^{-1}_{\mathbf{Y}_1}(ij) \in \mathbb{R}^{d'\times d'}$, $\boldsymbol{\Sigma}_{\mathbf{Y}_1\mathbf{Y}_2}(ij) \in \mathbb{R}^{d'\times d'}$ are sub-matrices of the variance matrix $\boldsymbol{\Gamma}_{ij} \in \mathbb{R}^{(d+d')\times(d+d')}$ of $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ such that

$$\boldsymbol{\Gamma}_{ij} = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathbf{X}_1}(ij) & \boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_1}(ij) & \boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{X}_2}(ij) & \boldsymbol{\Sigma}_{\mathbf{X}_1\mathbf{Y}_2}(ij) \\ \boldsymbol{\Sigma}^{\top}_{\mathbf{X}_1\mathbf{Y}_1}(ij) & \boldsymbol{\Gamma}_{\mathbf{Y}_1}(ij) & \boldsymbol{\Sigma}_{\mathbf{Y}_1\mathbf{X}_2}(ij) & \boldsymbol{\Sigma}_{\mathbf{Y}_1\mathbf{Y}_2}(ij) \\ \boldsymbol{\Sigma}^{\top}_{\mathbf{X}_1\mathbf{X}_2}(ij) & \boldsymbol{\Sigma}^{\top}_{\mathbf{Y}_1\mathbf{X}_2}(ij) & \boldsymbol{\Gamma}_{\mathbf{X}_2}(ij) & \boldsymbol{\Sigma}_{\mathbf{X}_2\mathbf{Y}_2}(ij) \\ \boldsymbol{\Sigma}^{\top}_{\mathbf{X}_1\mathbf{Y}_2}(ij) & \boldsymbol{\Sigma}^{\top}_{\mathbf{Y}_1\mathbf{Y}_2}(ij) & \boldsymbol{\Sigma}^{\top}_{\mathbf{X}_2\mathbf{Y}_2}(ij) & \boldsymbol{\Gamma}_{\mathbf{Y}_2}(ij) \end{bmatrix}. \tag{2.28}$$

Proof. By stationarity assumption on $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$, $p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1})$ does not depend on n, *i.e.* for any n, $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ is equal in distribution to $(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2)$:

$$p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1}) = p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2). \tag{2.29}$$

$p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ can obtained by marginalizing $(\mathrm{r}_1, \mathrm{r}_2)$ out from $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2, \mathrm{r}_1, \mathrm{r}_2)$:

$$p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) = \sum_{\mathrm{r}_1,\mathrm{r}_2\in\Omega} p(\mathrm{r}_1, \mathrm{r}_2)\, p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2). \tag{2.30}$$

Moreover, for each $(\mathrm{r}_1, \mathrm{r}_2)$ in Ω^2, we have $p(\mathbf{y}_2 \,|\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1, \mathrm{r}_2) = p(\mathbf{y}_2 \,|\mathbf{y}_1, \mathrm{r}_1, \mathrm{r}_2)$ by CGOMSM property (2.6). Thus, $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ is of form (2.25) with

$$\begin{aligned}&\forall i,j \in \{1:K\}, \alpha_{ij} = p(\mathrm{r}_1 = i, \mathrm{r}_2 = j),\\ &\quad p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) = p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2 \,|\mathrm{r}_1 = i, \mathrm{r}_2 = j).\end{aligned}$$

Let us show (2.26) and (2.27). For each (i, j) in $\{1 : K\}^2$, (2.26) is the same as

$$p_{ij}\left(\mathbf{y}_2, \mathbf{x}_1 \left| \mathbf{y}_1 \right.\right) = p_{ij}\left(\mathbf{y}_2 \left| \mathbf{y}_1 \right.\right) p_{ij}\left(\mathbf{x}_1 \left| \mathbf{y}_1 \right.\right),$$

that is to say that $\mathbf{Y}_2$ and $\mathbf{X}_1$ are independent given $\mathbf{Y}_1$. Since $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ is Gaussian, we apply Lemma 1 from Appendix A to show that this is equivalent to (2.27). □

Let us remember that CGOMSMs can be very close to the classic Conditionally Gaussian Linear State-Space Model (CGLSSM) [Derrode and Pieczynski, 2013, Petetin and Desbouvries, 2014]. The interest of this remarks is that the which does not offer the CGLSSMs do not offer the possibility of a fast exact Bayesian smoothing [Cappé et al., 2005], as opposed to the CGOMSMs. Exact Bayesian smoothing and filtering algorithms for CGOMSMs are detailed in the following section.

2.2 Exact Bayesian state estimation

In this section, we present and prove exact Bayesian inference algorithms for the CMSHLM. By Bayesian inference we mean computing posterior distribution of $p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right)$ and posterior moments $\mathbb{E}\left[\mathbf{x}_n \left| \mathbf{y}_{1:n} \right.\right]$ in the case of filtering, and $p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right)$, $\mathbb{E}\left[\mathbf{x}_n \left| \mathbf{y}_{1:N} \right.\right]$ in the case of smoothing. We have the following general result [Pieczynski, 2011a].

Proposition 4. *Let* $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ *be a CMSHLM. Then, for each* n *in* $\{1 : N\}$ *and* $\mathrm{r}_n \in \Omega$*,* $\mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]$ *and* $\mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]$ *are computable with a complexity linear in* N*.*

Proof. Since for all n in $\{1 : N-1\}$,

$$\begin{aligned}
&\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right] = \\
&\quad \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right) \Big(\mathbf{F}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right] + \mathbf{T}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \Big)
\end{aligned} \tag{2.31}$$

and

$$\begin{aligned}
&\mathbb{E}\left[\mathbf{X}_{n+1} \mathbf{X}_{n+1}^\top \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right] = \\
&\quad \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right) \Big(\mathbf{F}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right] \mathbf{F}_{n+1}^\top(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \\
&\quad + \mathbf{F}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right] \mathbf{T}_{n+1}^\top(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) + \\
&\quad \mathbf{T}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbb{E}\left[\mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right] \mathbf{F}_{n+1}^\top(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \\
&+ \mathbf{G}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbf{G}_{n+1}^\top(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) + \mathbf{T}_{n+1}(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \mathbf{T}_{n+1}^\top(\mathrm{r}_{n:n+1}, \mathbf{y}_{n:n+1}) \Big),
\end{aligned} \tag{2.32}$$

thus $\mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]$ and $\mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]$ can be computed recursively.

Besides, it follows from hypothesis (2.1b) that $\boldsymbol{V}_{1:N} = (\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is Markovian. We can therefore calculate the needed probabilities

$$p\left(\mathrm{r}_n \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right) = \frac{p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n \right.\right) p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right)}{\sum_{\mathrm{r}_n^* \in \Omega} p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n^*, \mathbf{y}_n \right.\right) p\left(\mathrm{r}_n^* \left| \mathbf{y}_{1:n} \right.\right)}$$

since $p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_n \right.\right)$ are known and $p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right)$, $p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right)$ can be computed by using the outputs of the classic forward-backward algorithm, which are $\alpha_n(\mathrm{r}_n) = p\left(\mathrm{r}_n, \mathbf{y}_{1:n}\right)$

and $\beta_n(\mathrm{r}_n) = p\left(\mathbf{y}_{n+1:N} \left| \mathbf{v}_n \right.\right)$. More precisely, we have:

$$\begin{aligned}
&\alpha_1(\mathrm{r}_1) = p\left(\mathbf{v}_1\right); \\
&\alpha_{n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n \in \Omega} \alpha_n(\mathrm{r}_n) p\left(\mathbf{v}_{n+1} \left| \mathbf{v}_n \right.\right); \qquad (2.33) \\
&\beta_N(r_N) = 1; \\
&\beta_n(\mathrm{r}_n) = \sum_{\mathrm{r}_{n+1} \in \Omega} \beta_{n+1}(\mathrm{r}_{n+1}) p\left(\mathbf{v}_{n+1} \left| \mathbf{v}_n \right.\right). \qquad (2.34)
\end{aligned}$$

Then

$$p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right) = \frac{\alpha_n(\mathrm{r}_n)}{\sum_{\mathrm{r}_n^* \in \Omega} \alpha_n(\mathrm{r}_n^*)}, \qquad (2.35)$$

and

$$p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) = \frac{\alpha_n(\mathrm{r}_n)\beta_n(\mathrm{r}_n)}{\sum_{\mathrm{r}_n^* \in \Omega} \alpha_n(\mathrm{r}_n^*)\beta_n(\mathrm{r}_n^*)}. \qquad (2.36)$$

□

2.2.1 Filtering

For each n in $\{1:N\}$, $\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_{1:n} \right.\right]$ and $\mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathbf{y}_{1:n} \right.\right]$ can be computed recursively with a complexity linear in n by

$$\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_{1:n} \right.\right] = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right) \mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]; \qquad (2.37)$$

$$\mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathbf{y}_{1:n} \right.\right] = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:n} \right.\right) \mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]. \qquad (2.38)$$

2.2.2 Smoothing

We have the following general result [Pieczynski, 2011b].

Proposition 5. *Let $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ be a CMSHLM. Then, for each n in $\{1:N\}$,*

$$\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_{1:N} \right.\right] = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) \mathbb{E}\left[\mathbf{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right]; \qquad (2.39)$$

$$\mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathbf{y}_{1:N} \right.\right] = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) \mathbb{E}\left[\mathbf{X}_n \mathbf{X}_n^\top \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right], \qquad (2.40)$$

both expectations being computable with a complexity linear in N.

Proof. Let us show (2.39) and (2.40). For all n in $\{1:N-1\}$, $\mathbf{X}_n$ and $\mathbf{Y}_{n+1}$ are independent given $(\mathrm{R}_n, \mathbf{Y}_n) = (\mathrm{r}_n, \mathbf{y}_n)$ *cf.* (2.1b). It follows that the variables $\mathbf{X}_n$ and $(\mathrm{R}_{n+1:N}, \mathbf{Y}_{n+1:N})$ are also independent given $(\mathrm{R}_n, \mathbf{Y}_n) = (\mathrm{r}_n, \mathbf{y}_n)$. Thus, $p\left(\mathbf{x}_n \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right) = p\left(\mathbf{x}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right)$. We have (2.39) and (2.40) from

$$p\left(\mathbf{x}_n \left| \mathbf{y}_{1:N} \right.\right) = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) p\left(\mathbf{x}_n \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right) = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) p\left(\mathbf{x}_n \left| \mathrm{r}_n, \mathbf{y}_{1:n} \right.\right).$$

□

The fact that $\mathbf{X}_n$ and $(\mathrm{R}_{n+1:N}, \mathbf{Y}_{n+1:N})$ are independent given $(\mathrm{R}_n, \mathbf{Y}_n) = (\mathrm{r}_n, \mathbf{y}_n)$ could appear as somewhat limiting. However, this kind of assumptions is widespread. For example, in the classic Hidden Markov Model (HMM) $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ the variables R_n and $\mathbf{Y}_{n+1}$ are independent given $\mathrm{R}_{n+1} = \mathrm{r}_{n+1}$, but they are not independent without this conditioning and it is well known that $\mathbf{Y}_{n+1:N}$ can bring a large deal of information on R_n.

2.3 Parameter estimation by the Expectation-Maximization (EM) algorithm

One can see from (2.16) that the problem of estimation of CGOMSM from $\mathbf{y}_{1:N}$ is ill-specified: there is no way of estimating parameters in (2.16b) considering $\mathbf{y}_{1:N}$ only. Thus, here we consider the problem of estimating the parameters of CGOMSM from $(\mathbf{x}_{1:N}, \mathbf{y}_{1:N})$. This section presents an iterative estimation algorithm derived by the author and proven to be an EM algorithm. The related proof can be found in Appendix C.

We propose an iterative estimation technique described in Algorithm 1. At the q-th iteration of the algorithm, the parameters of CGOMSM is denoted by $\boldsymbol{\theta}^{(q)}$, defined by:

$$\boldsymbol{\theta}^{(q)} = \left\{ \boldsymbol{\mu}_i^{(q)}, \boldsymbol{\Gamma}_i^{(q)}, p_{ij}^{(q)}, \boldsymbol{A}_{ij}^{(q)}, \boldsymbol{B}_{ij}^{(q)}, \boldsymbol{C}_{ij}^{(q)}, \boldsymbol{D}_{ij}^{(q)}, \boldsymbol{F}_{ij}^{(q)}, \boldsymbol{H}_{ij}^{(q)}, \boldsymbol{\Pi}_{ij}^{(q)}, \boldsymbol{\Lambda}_{ij}^{(q)} \,|1 \le i, j \le K \right\},$$

where:

- for each i in Ω, $\boldsymbol{\mu}_i^{(q)}$ and $\boldsymbol{\Gamma}_i^{(q)}$ define pdf $p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{x}_1, \mathbf{y}_1 \,|\mathrm{r}_1 = i\right)$;
- for each i, j in Ω, $p_{ij}^{(q)} = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_1 = i, \mathrm{r}_2 = j\right)$ and $\boldsymbol{A}_{ij}^{(q)}$, $\boldsymbol{B}_{ij}^{(q)}$, $\boldsymbol{C}_{ij}^{(q)}$, $\boldsymbol{D}_{ij}^{(q)}$, $\boldsymbol{F}_{ij}^{(q)}$, $\boldsymbol{H}_{ij}^{(q)}$, $\boldsymbol{\Pi}_{ij}^{(q)}$, $\boldsymbol{\Lambda}_{ij}^{(q)}$ are defined *cf.* (2.16).

Algorithm 1. *Parameter estimation of CGOMSM*

1. *Make an initial guess*

$$\boldsymbol{\theta}^{(0)} = \left\{ \boldsymbol{\mu}_i^{(0)}, \boldsymbol{\Gamma}_i^{(0)}, p_{ij}^{(0)}, \boldsymbol{A}_{ij}^{(0)}, \boldsymbol{B}_{ij}^{(0)}, \boldsymbol{C}_{ij}^{(0)}, \boldsymbol{D}_{ij}^{(0)}, \boldsymbol{F}_{ij}^{(0)}, \boldsymbol{H}_{ij}^{(0)}, \boldsymbol{\Pi}_{ij}^{(0)}, \boldsymbol{\Lambda}_{ij}^{(0)} \,|1 \le i, j \le K \right\}$$

as follows:

(a) *Apply the K-means clustering method to* $\mathbf{x}_{1:N}$*. We will denote by* $\kappa_n(i)$ *the function which assigns* 1 *if* $\mathbf{x}_n$ *is within the* i^{th} *cluster, and* 0 *otherwise. We also note* $\delta_n(i,j) = \kappa_n(i)\kappa_{n+1}(j)$*;*

(b) *For each* i *in* Ω*,* $\boldsymbol{\mu}_i^{(0)}$ *and* $\boldsymbol{\Gamma}_i^{(0)}$ *are given by*

$$\boldsymbol{\mu}_i^{(0)} = \frac{\sum_{n=1}^{N} \mathbf{z}_n \kappa_n(i)}{\sum_{n=1}^{N} \kappa_n(i)}; \tag{2.41a}$$

$$\boldsymbol{\Gamma}_i^{(0)} = \frac{\sum_{n=1}^{N} \left(\mathbf{z}_n - \boldsymbol{\mu}_i^{(0)}\right)\left(\mathbf{z}_n - \boldsymbol{\mu}_i^{(0)}\right)^{\top} \kappa_n(i)}{\sum_{n=1}^{N} \kappa_n(i)}, \tag{2.41b}$$

where $\mathbf{z}_n^{\top} = \left[\mathbf{x}_n^{\top} \; \mathbf{y}_n^{\top}\right]$*, and for each* (i,j) *in* Ω^2*,* $p_{ij}^{(0)}$ *is given by*

$$p_{ij}^{(0)} = \frac{1}{N-1} \sum_{n=1}^{N-1} \delta_n(i,j). \tag{2.42}$$

(c) *Compute intermediate matrices* $\boldsymbol{E}_{ij}^{(0)}$, $\boldsymbol{S}_{ij}^{(0)}$, $\boldsymbol{\chi}_{ij}^{(0)}$, $\boldsymbol{\Phi}_{ij}^{(0)}$, $\boldsymbol{G}_{ij}^{(0)}$, $\boldsymbol{P}_{ij}^{(0)}$, $\boldsymbol{\xi}_{ij}^{(0)}$ *and* $\boldsymbol{T}_{ij}^{(0)}$

as follows:

$$\boldsymbol{E}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \begin{bmatrix} \mathbf{z}_n \\ \mathbf{y}_{n+1} \end{bmatrix} \delta_n(i,j); \tag{2.43a}$$

$$\boldsymbol{S}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \begin{bmatrix} \mathbf{z}_n \mathbf{z}_n^\top & \mathbf{z}_n \mathbf{y}_{n+1}^\top \\ \mathbf{y}_{n+1} \mathbf{z}_n^\top & \mathbf{y}_{n+1} \mathbf{y}_{n+1}^\top \end{bmatrix} \delta_n(i,j); \tag{2.43b}$$

$$\boldsymbol{\chi}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \begin{bmatrix} \mathbf{x}_{n+1} \mathbf{z}_n^\top & \mathbf{x}_{n+1} \mathbf{y}_{n+1}^\top \end{bmatrix} \delta_n(i,j); \tag{2.43c}$$

$$\boldsymbol{\Phi}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{x}_{n+1} \delta_n(i,j); \tag{2.43d}$$

$$\boldsymbol{G}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{y}_n \delta_n(i,j); \tag{2.43e}$$

$$\boldsymbol{P}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{y}_n \mathbf{y}_n^\top \delta_n(i,j); \tag{2.43f}$$

$$\boldsymbol{\xi}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{y}_{n+1} \mathbf{y}_n^\top \delta_n(i,j); \tag{2.43g}$$

$$\boldsymbol{T}_{ij}^{(0)} = \frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{y}_{n+1} \delta_n(i,j). \tag{2.43h}$$

(d) *For each i,j in Ω, $\boldsymbol{A}_{ij}^{(0)}$, $\boldsymbol{B}_{ij}^{(0)}$, $\boldsymbol{C}_{ij}^{(0)}$, $\boldsymbol{D}_{ij}^{(0)}$, $\boldsymbol{F}_{ij}^{(0)}$, $\boldsymbol{H}_{ij}^{(0)}$, $\boldsymbol{\Pi}_{ij}^{(0)}$ and $\boldsymbol{\Lambda}_{ij}^{(0)}$ are given by*

$$\left[\boldsymbol{F}_{ij}^{(0)} \boldsymbol{A}_{ij}^{(0)} \boldsymbol{B}_{ij}^{(0)} \boldsymbol{C}_{ij}^{(0)}\right] = \left[\boldsymbol{\Phi}_{ij}^{(0)} \boldsymbol{\chi}_{ij}^{(0)}\right] \begin{bmatrix} N-1 & \left(\boldsymbol{E}_{ij}^{(0)}\right)^\top \\ \boldsymbol{E}_{ij}^{(0)} & \boldsymbol{S}_{ij}^{(0)} \end{bmatrix}^{-1}; \tag{2.44a}$$

$$\left[\boldsymbol{H}_{ij}^{(0)} \ \boldsymbol{D}_{ij}^{(0)}\right] = \left[\boldsymbol{T}_{ij}^{(0)} \boldsymbol{\xi}_{ij}^{(0)}\right] \begin{bmatrix} N-1 & \left(\boldsymbol{G}_{ij}^{(0)}\right)^\top \\ \boldsymbol{G}_{ij}^{(0)} & \boldsymbol{P}_{ij}^{(0)} \end{bmatrix}^{-1}; \tag{2.44b}$$

$$\begin{aligned} &(N-1)\boldsymbol{\Lambda}_{ij}^{(0)} \left(\boldsymbol{\Lambda}_{ij}^{(0)}\right)^\top = \\ &\frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{y}_{n+1} \mathbf{y}_{n+1}^\top \delta_n(i,j) - \boldsymbol{H}_{ij}^{(0)} \left(\boldsymbol{T}_{ij}^{(0)}\right)^\top - \boldsymbol{D}_{ij}^{(0)} \left(\boldsymbol{\xi}_{ij}^{(0)}\right)^\top; \end{aligned} \tag{2.44c}$$

$$\begin{aligned} &(N-1)\boldsymbol{\Pi}_{ij}^{(0)} \left(\boldsymbol{\Pi}_{ij}^{(0)}\right)^\top = \\ &\frac{1}{p_{ij}^{(0)}} \sum_{n=1}^{N-1} \mathbf{x}_{n+1} \mathbf{x}_{n+1}^\top \delta_n(i,j) - \boldsymbol{F}_{ij}^{(0)} \left(\boldsymbol{\Phi}_{ij}^{(0)}\right)^\top - \left[\boldsymbol{A}_{ij}^{(0)} \quad \boldsymbol{B}_{ij}^{(0)} \quad \boldsymbol{C}_{ij}^{(0)}\right] \left(\boldsymbol{\chi}_{ij}^{(0)}\right)^\top. \end{aligned} \tag{2.44d}$$

2. *Find the new set of parameters $\boldsymbol{\theta}^{(q+1)}$ as follows:*

(a) *For each i in Ω, compute posterior probabilities*

$$\phi_n^{(q)}(i) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_n = i \,|\mathbf{x}_{1:N}, \mathbf{y}_{1:N}\right),$$

and for each i, j in Ω compute

$$\psi_n^{(q)}(i,j) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_n = i, \mathrm{r}_{n+1} = j \,|\mathbf{x}_{1:N}, \mathbf{y}_{1:N}\right)$$

cf. (2.48)*;*

(b) *For each i in Ω, compute $\boldsymbol{\mu}_i^{(q+1)}$ and $\boldsymbol{\Gamma}_i^{(q+1)}$ by substitution $\phi_n^{(q)}(i)$ for $\kappa_n(i)$ in* (2.41)*;*

(c) *For each i, j in Ω, $p_{ij}^{(q+1)}$ is given by*

$$p_{ij}^{(q+1)} = \frac{1}{N-1} \sum_{n=1}^{N-1} \psi_n^{(q)}(i,j). \tag{2.45}$$

Then compute intermediate matrices $\boldsymbol{E}_{ij}^{(q+1)}$, $\boldsymbol{S}_{ij}^{(q+1)}$, $\boldsymbol{\chi}_{ij}^{(q+1)}$, $\boldsymbol{\Phi}_{ij}^{(q+1)}$, $\boldsymbol{G}_{ij}^{(q+1)}$, $\boldsymbol{P}_{ij}^{(q+1)}$, $\boldsymbol{\xi}_{ij}^{(q+1)}$ and $\boldsymbol{T}_{ij}^{(q+1)}$ by substituting $\psi_n^{(q)}(i,j)$, $p_{ij}^{(q+1)}$ with $\delta_n(i,j)$, $p_{ij}^{(0)}$ in (2.43)*. Finally, compute $\boldsymbol{A}_{ij}^{(q+1)}$, $\boldsymbol{B}_{ij}^{(q+1)}$, $\boldsymbol{C}_{ij}^{(q+1)}$, $\boldsymbol{D}_{ij}^{(q+1)}$, $\boldsymbol{F}_{ij}^{(q+1)}$, $\boldsymbol{H}_{ij}^{(q+1)}$, $\boldsymbol{\Pi}_{ij}^{(q+1)}$ and $\boldsymbol{\Lambda}_{ij}^{(q+1)}$ by substituting $\boldsymbol{E}_{ij}^{(q+1)}$, $\boldsymbol{S}_{ij}^{(q+1)}$, $\boldsymbol{\chi}_{ij}^{(q+1)}$, $\boldsymbol{\Phi}_{ij}^{(q+1)}$, $\boldsymbol{G}_{ij}^{(q+1)}$, $\boldsymbol{P}_{ij}^{(q+1)}$, $\boldsymbol{\xi}_{ij}^{(q+1)}$, $\boldsymbol{T}_{ij}^{(q+1)}$ with $\boldsymbol{E}_{ij}^{(0)}$, $\boldsymbol{S}_{ij}^{(0)}$, $\boldsymbol{\chi}_{ij}^{(0)}$, $\boldsymbol{\Phi}_{ij}^{(0)}$, $\boldsymbol{G}_{ij}^{(0)}$, $\boldsymbol{P}_{ij}^{(0)}$, $\boldsymbol{\xi}_{ij}^{(0)}$, $\boldsymbol{T}_{ij}^{(0)}$ in* (2.44)*.*

(The algorithm ends here)

Let us recall the formulas for $\phi_n^{(q)}(i)$ and $\psi_n^{(q)}(i,j)$. Let us pose $\mathbf{t}_n = (\mathbf{x}_n, \mathrm{r}_n, \mathbf{y}_n)$, $\alpha_n(\mathrm{r}_n) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_n, \mathbf{z}_{1:n}\right)$ and $\beta_n(\mathrm{r}_n) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{z}_{n+1:N} \,|\mathbf{t}_n\right)$. Then, the forward-backward algorithm computes recursively $\alpha_n(\mathrm{r}_n)$ and $\beta_n(\mathrm{r}_n)$ as follows:

- $\alpha_1(\mathrm{r}_1) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_1\right)$ and

$$\alpha_{n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n \in \Omega} \alpha_n(\mathrm{r}_n) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_{n+1} \,|\mathbf{t}_n\right) \tag{2.46}$$

for n in $\{1 : N-1\}$.

- $\beta_N(r_N) = 1$ and

$$\beta_n(\mathrm{r}_n) = \sum_{\mathrm{r}_{n+1} \in \Omega} \beta_{n+1}(\mathrm{r}_{n+1}) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_{n+1} \,|\mathbf{t}_n\right) \tag{2.47}$$

for n in $\{1 : N-1\}$.

where

$$
\begin{aligned}
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_1\right) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_1\right) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{z}_1 \left|\mathrm{r}_1\right.\right)\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_{n+1} \left|\mathbf{t}_n\right.\right) = p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_{n+1} \left|\mathrm{r}_n\right.\right) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_{n:n+1}\right.\right)\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_1 = i\right) = \sum_{j\in\Omega} p_{ij}^{(q)}\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{z}_1 \left|\mathrm{r}_1 = i\right.\right) = \mathscr{N}\left(\mathbf{z}_1; \boldsymbol{\mu}_i^{(q)}, \boldsymbol{\Gamma}_i^{(q)}\right)\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_{n+1} = j \left|\mathrm{r}_n = i\right.\right) = \frac{p_{ij}^{(q)}}{p_{\boldsymbol{\theta}^{(q)}}\left(\mathrm{r}_1 = i\right)}\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{y}_{n+1} \left|\mathbf{y}_n, \mathrm{r}_{n:n+1} = (i,j)\right.\right) =\\
&\quad \mathscr{N}\left(\mathrm{y}_{n+1}; \boldsymbol{D}_{ij}^{(q)}\mathbf{y}_n + \boldsymbol{H}_{ij}^{(q)}, \boldsymbol{\Lambda}_{ij}^{(q)}\left(\boldsymbol{\Lambda}_{ij}^{(q)}\right)^\top\right)\\
&p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{x}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_{n:n+1}, \mathrm{r}_{n:n+1} = (i,j)\right.\right) =\\
&\quad \mathscr{N}\left(\mathbf{x}_{n+1}; \boldsymbol{A}_{ij}^{(q)}\mathbf{x}_n + \boldsymbol{B}_{ij}^{(q)}\mathbf{y}_n + \boldsymbol{C}_{ij}^{(q)}\mathbf{y}_{n+1} + \boldsymbol{F}_{ij}^{(q)}, \boldsymbol{\Pi}_{ij}^{(q)}\left(\boldsymbol{\Pi}_{ij}^{(q)}\right)^\top\right).
\end{aligned}
$$

Thus,

$$
\psi_n^{(q)}(i,j) = \frac{\alpha_n(\mathrm{r}_n) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_{n+1} \left|\mathbf{t}_n\right.\right) \beta_{n+1}(\mathrm{r}_{n+1})}{\sum_{\mathrm{r}_n^\star, \mathrm{r}_{n+1}^\star} \alpha_n(\mathrm{r}_n^\star) p_{\boldsymbol{\theta}^{(q)}}\left(\mathbf{t}_{n+1}^\star \left|\mathbf{t}_n^\star\right.\right) \beta_{n+1}(r_{n+1}^\star)}, \tag{2.48}
$$

with $\mathbf{t}_n^\star = (\mathbf{x}_n, \mathrm{r}_n^\star, \mathbf{y}_n)$.

Proposition 6. *Algorithm 1 is an EM algorithm of estimation of CGOMSM parameters in form* (2.16)*.*

Proof. See Appendix C. □

It is noteworthy that one can see Algorithm 1 as an EM algorithm for parameter estimation of Gaussian mixture (2.25) with the variance of mixands constrained to (2.27). That is why it is not a classic EM algorithm of estimating a Gaussian mixture, since in its classic version, the mixands' variances are not constrained.

Estimating mixture (2.25) is a necessary stage of the CGOMSM application to the Bayesian state estimation in non-linear non-Gaussian models, as detailed in the next section.

2.4 Application to Bayesian state estimation in non-linear non Gaussian models

Let us consider a Partially Observable Markov Process (POMP) with continuous state space $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$, which can possibly be non-linear or/and non-Gaussian. For each n in $\{1:N\}$, $\mathbf{X}_n$ takes its value in $\mathbb{R}^d$ and $\mathbf{Y}_n$ takes its value in $\mathbb{R}^{d'}$ with $d \in \mathbb{N}^*$, $d' \in \mathbb{N}^*$. In this Section, we consider an application of CGOMSMs to the problem of Bayesian inference, which consists in the sequential search of $\mathbf{X}_{1:N}$ from $\mathbf{Y}_{1:N}$. We suppose that $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ is stationary, which means that for any n, $(\mathbf{X}_n, \mathbf{Y}_n, \mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ is equal in distribution to $(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2)$, what we note by $p\left(\mathbf{x}_n, \mathbf{y}_n, \mathbf{x}_{n+1}, \mathbf{y}_{n+1}\right) = p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2\right)$. Bayesian filter and smoother based on CGOMSMs are called the Learned Conditionally Gaussian Observed Markov Switching Model Filter (LCGOMSMF) and the Learned Conditionally Gaussian Observed Markov Switching Model Smoother (LCGOMSMS) respectively.

LCGOMSMF and LCGOMSMS approximate the corresponding Bayesian solution in the POMP considered. Specifically, since $(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N})$ is stationary, its distribution derives from $p\left(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2\right)$, as the latter provides $p\left(\mathbf{x}_1, \mathbf{y}_1\right)$ and $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left|\mathbf{x}_n, \mathbf{y}_n\right.\right)$ for each

n in $\{1 : N\}$. $p(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ can be approximated using a mixture of K^2 components of form (2.25), where the mixands $p_{ij}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ are such that their variance matrices satisfy (2.27).

Definition 4. *LCGOMSMF*

We call LCGOMSMF the following algorithm:

1. Generate an artificial sample $(\mathbf{x}^\star_{1:N'}, \mathbf{y}^\star_{1:N'})$ according to a given model of type $(\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}) = \mathscr{T}(\mathbf{X}_n, \mathbf{Y}_n, \boldsymbol{W}_n)$, where $N' \in \mathbb{N}^*$, $\mathscr{T}$ is a model transition kernel and $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{N'}$ are independent variables;

2. Estimate CGOMSM parameters from $(\mathbf{x}^\star_{1:N'}, \mathbf{y}^\star_{1:N'})$ by Algorithm 1;

3. Filtering: when a new measurement $\mathbf{y}_{n+1}$ is received, compute $p(\mathrm{r}_{n+1} | \mathbf{y}_{1:n+1})$, $\mathbb{E}[\mathbf{X}_{n+1} | \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1}]$ and $\mathbb{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top | \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1}\right]$ by using (2.31), (2.32), then $\mathbb{E}[\mathbf{X}_{n+1} | \mathbf{y}_{1:n+1}]$ and $\mathbb{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top | \mathbf{y}_{1:n+1}\right]$ are given by (2.37), (2.38).

The LCGOMSMS is defined in the same way and uses (2.39), (2.40) to compute $\mathbb{E}[\mathbf{X}_{n+1} | \mathbf{y}_{1:N}]$ and $\mathbb{E}\left[\mathbf{X}_{n+1}\mathbf{X}_{n+1}^\top | \mathbf{y}_{1:N}\right]$.

For an estimated scalar signal $\widehat{\mathrm{x}}_{1:N}$ obtained from $\mathbf{y}_{1:N}$, the Mean Squared Error (MSE) is defined by

$$\mathrm{MSE} = \sum_{n=1}^{N} (\mathrm{x}_n - \widehat{\mathrm{x}}_n)^2, \tag{2.49}$$

where $\mathrm{x}_{1:N}$ is the "ground-truth" simulated and unknown to inference algorithms. MSE is a useful performance criterion for comparing the effectiveness of Bayesian inference algorithms.

The following subsections are examples of applications of LCGOMSMF and LCGOMSMS to different models belonging to the class of POMP with continuous state space. All the results presented below are averaged over 100 equivalent independent experiments, each of them being computed using $N = 1000$ simulated data points.

2.4.1 Bayesian state estimation in the stochastic volatility model

Here we consider the standard Stochastic Volatility (SV) model [Jacquier et al., 1994], usually presented as follows:

$$\mathrm{X}_1 = \mu + \mathrm{U}_1; \tag{2.50a}$$

$$\forall n \in \mathbb{N}^*, \mathrm{X}_{n+1} = \mu + \phi(\mathrm{X}_n - \mu) + \sigma \mathrm{U}_{n+1}; \tag{2.50b}$$

$$\forall n \in \mathbb{N}^*, \mathrm{Y}_n = \beta \exp(\mathrm{X}_n/2)\mathrm{V}_n, \tag{2.50c}$$

where $\mathrm{U}_{1:N}, \mathrm{V}_{1:N}$ are independent standard Gaussian variables in $\mathbb{R}$ and $\mu \in \mathbb{R}$, $\phi \in]-1, 1[$, $\beta \in \mathbb{R}_+^*$, $\sigma \in \mathbb{R}_+^*$ are fixed.

We compare the performance of the LCGOMSMF with that of the Particle Filter (PF) and Gaussian Sum Filter (GSF) [Simandl and Kralovec, 2000], in the case of filtering in model (2.50). We set $\mu = 0.5$, $\beta = 0.5$, and consider four different cases for ϕ and σ such that $\phi^2 + \sigma^2 = 1$ (that is to ensure that the common variance of the variables X_n is unitary). The results are reported in Table 2.1.

The details of each filtering method used in the experiments are the following:

- For the LCGOMSMF, we test out different values of K and we infer the CGOMSM from an independently generated sample $(\mathrm{x}_{1:N'}, \mathrm{y}_{1:N'})$ of size $N' = 20000$, performing 100 EM iterations. See Figure 2.4 for an example of trajectories.

Cases	ϕ	σ^2	LCGOMSMF				PF	GSF
			$K=2$	$K=3$	$K=5$	$K=7$		
1	0.99	0.0199	0.41	0.27	0.20	0.19	0.18	0.21
2	0.90	0.1900	0.55	0.49	0.47	0.46	0.46	0.50
3	0.80	0.3600	0.63	0.59	0.58	0.57	0.57	0.60
4	0.50	0.7500	0.72	0.71	0.70	0.70	0.70	0.72

Table 2.1: Average MSE results for different SV models defined by ϕ and σ ($\mu = 0.5$, $\beta = 0.5$).

- The PF implementation is that of Section 1.4.1 and uses $M = 1500$ particles. We found out empirically that PF behaves asymptotically for this number of particles or greater.

- In order to use the GSF, we linearize the SV model by taking the logarithm of both sides of (2.50c) to get

$$\mathrm{X}_1 = \mu + \mathrm{U}_1; \tag{2.51a}$$

$$\mathrm{X}_{n+1} = \mu + \phi(\mathrm{X}_n - \mu) + \sigma \mathrm{U}_{n+1}; \tag{2.51b}$$

$$\mathrm{Y}'_n = \mathrm{X}_n + \mathrm{V}'_n, \tag{2.51c}$$

  where $Y'_n = \log(Y_n^2) - 2\log\beta$ and $\mathrm{V}'_{1:N}$ are independent, non-Gaussian variables, such that $\exp\left(\frac{\mathrm{V}'_1}{2}\right)$, ..., $\exp\left(\frac{\mathrm{V}'_N}{2}\right)$ are standard Gaussians. Then, for each n in $\mathbb{N}^*$, the Probability Density Function (pdf) of V'_n is $p(\mathrm{v}'_n) = \exp\left(\frac{\mathrm{v}'_n}{2}\right)\mathcal{N}\left(\exp\left(\frac{\mathrm{v}'_n}{2}\right); 0, 1\right)$. Following the general principle of the GSF, we approximate the latter pdf by a Gaussian mixture of r components : $p(v'_n) \approx \sum_{m=1}^{r} \gamma_n \mathcal{N}(v'_n; \hat{v}'_m, R_m)$. We found that when $r \geq 5$, the approximation is accurate enough to achieve a negligible residual effect. Since the number ξ_n of mixands in the filtering pdf

$$p(x_n | y_{1:n}) = \sum_{j=1}^{\xi_n} \alpha_{nj} \mathcal{N}(x_n; \hat{x}_{nj}, P_{nj}) \tag{2.52}$$

  grows exponentially with n, a reduction technique is implemented to keep computational demands of the algorithm within reasonable bounds.

  For the experiments, we classically reduce the number of terms as follows: when ξ_n becomes greater than r, we keep the r mixands in (2.52) which have the greatest weight coefficients α_{nj}, and we discard the remaining. Therefore, we impose the constraint that $\xi_n = r$. We found out empirically that GSF behaves asymptotically for $r \geq 3$, but does not attain the optimal MSE.

  We note that since the model (2.51) is linear, there is no reason for considering the extensions of the GSF for non-linear systems, such as the Gaussian Sum Unscented Kalman Filter (GSUKF) [Straka et al., 2011].

Contrary to the LCGOMSMF which makes use of a single global approximation, the GSUKF relies on multiple approximations:

- an approximation of the noise terms with a Gaussian mixture;

- some reduction technique to keep the number of mixands of the filtering pdf within reasonable bounds.

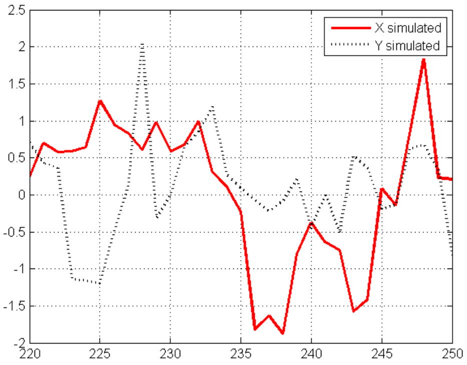


Figure 2.3: Simulated log-volatility trajectory with an SV model (red, plain), simulated log-returns (black, dotted).

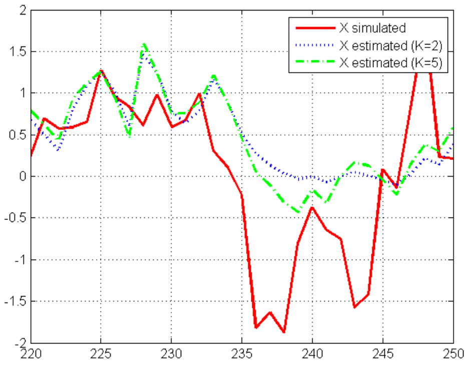


Figure 2.4: Log-volatility estimates computed using $K = 2$ classes (blue, dotted), and $K = 5$ classes (green, dashed).

Additionally, when the model is non-linear, the GSUKF uses the Unscented Transform (UT) for computing the approximate means and covariances. The UT relies, in turn, on its scaling parameters. Our experiments show that computing a single global approximation may be advantageous and helps to avoid the cumulative residual effect. However, unlike then LCGOMSMF, the GSUKF may be used for non-stationary systems.

2.4.2 Filtering in the asymmetric stochastic volatility model

Here we consider the Asymmetric Stochastic Volatility (ASV) model [Omori and Watanabe, 2008], which may be presented as follows:

$$\mathrm{X}_1 = \mu + \mathrm{U}_1; \tag{2.53a}$$

$$\mathrm{X}_{n+1} = \mu + \phi(\mathrm{X}_n - \mu) + \sigma\left(\frac{\rho \mathrm{Y}_n}{\beta \exp(\mathrm{X}_n/2)} + \lambda \mathrm{U}_{n+1}\right); \tag{2.53b}$$

$$\mathrm{Y}_n = \beta \exp(\mathrm{X}_n/2)\mathrm{V}_n. \tag{2.53c}$$

Here, we compare the performance of the LCGOMSMF with that of the PF only, since the GSF and GSUKF would not take into account the value of the volatility asymmetry coefficient ρ and therefore they are not suitable for this model. The experimental configuration is identical to the previous one. For the sake of consistency with the Asymmetric Volatility Phenomenon (AVP), ρ should be assumed negative.

We set $\mu = 0.5$, $\beta = 0.5$, and consider five different cases for ρ and λ such that

$$\rho^2 + \lambda^2 = 1;$$
$$\phi^2 + \sigma^2 = 1,$$

to ensure that for each n in $\mathbb{N}^*$, the variance of X_n is unitary. The results are reported in Table 2.2 for $\phi = 0.5$ and in Table 2.4 for $\phi = 0.8$. Figure 2.6 shows an ASV trajectory, and its restoration with the LCGOMSMF for $K = 2$ and $K = 5$ classes. Table 2.3 contains indicative processing time required for the LCGOMSMF and PF to process a data sequence of length $N = 1000$, or to learn the CGOMSM parameters from a sequence of length $N' = 20000$. This table is provided on an indicative basis only, since the processing time depends on the PC system configuration, processor type and settings, PF implementation and compilation details, software specifications and so on.

Cases	ρ	λ^2	LCGOMSMF			PF
			$K=2$	$K=3$	$K=5$	
1	-0.9	0.19	0.23	0.22	0.20	0.20
2	-0.8	0.36	0.36	0.35	0.34	0.33
3	-0.5	0.75	0.59	0.58	0.58	0.57
4	-0.3	0.91	0.68	0.67	0.66	0.65
5	0.0	1.00	0.72	0.71	0.70	0.70

Table 2.2: Average MSE results for different ASV models defined by ρ and λ ($\mu = 0.5$, $\beta = 0.5$, and $\sigma^2 + \phi^2 = 1$), for $\phi = 0.5$.

Measure type	LCGOMSMF			PF
	$K=2$	$K=3$	$K=5$	
Filtering time (s.)	0.003	0.004	0.010	0.20
EM time (s.)	8.05	10.70	19.88	N/A

Table 2.3: Average computation time for the LCGOMSMF and PF.

Cases	ρ	λ^2	LCGOMSMF			PF
			$K=2$	$K=3$	$K=5$	
1	-0.9	0.19	0.22	0.21	0.19	0.18
2	-0.8	0.36	0.33	0.31	0.29	0.29
3	-0.5	0.75	0.52	0.49	0.48	0.47
4	-0.3	0.91	0.59	0.55	0.54	0.54
5	0.0	1.00	0.63	0.59	0.58	0.57

Table 2.4: Average MSE results for different ASV models defined by ρ and λ ($\mu = 0.5$, $\beta = 0.5$, and $\sigma^2 + \phi^2 = 1$), for $\phi = 0.8$.

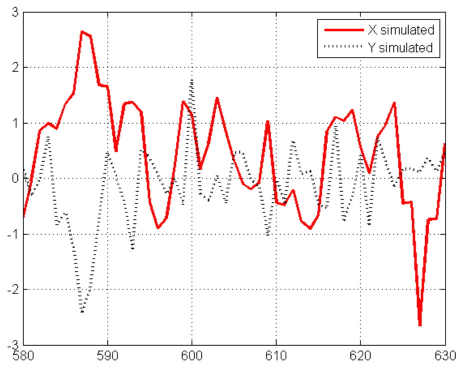


Figure 2.5: Simulated log-volatility trajectory with an ASV model (red, plain), simulated log-returns (black, dotted).

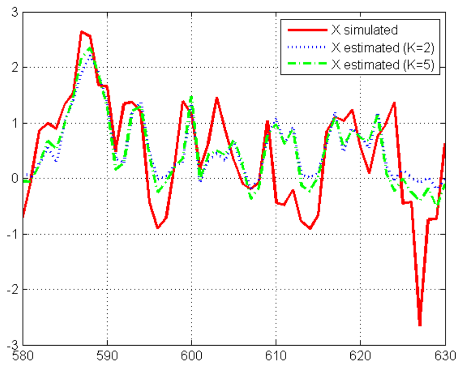


Figure 2.6: Log-volatility estimates computed using $K = 2$ classes (blue, dotted), and $K = 5$ classes (green, dashed).

According to these results, the LCGOMSMF is efficient for both SV and ASV models, and attains the same asymptotic performances as the PF. Regarding the processing time, we find that after having it adjusted to the SV model, the LCGOMSMF is nearly five times faster than the PF.

At the moment, we have no computational technique to select the minimum number of classes allowing to obtain asymptotic performances. We only note the trade-off between the computational cost and the variance of the resulting estimates. Indeed, with a greater number of classes the former increases, while the latter decreases. In practice, five classes seem to be enough for most of situations.

2.4.3 Filtering real-world data

Here we propose an application of the LCGOMSMF to recover volatility estimates of a real-world stock chart. Let us remind that if P_{n-1} denotes the stock price at the beginning of the previous trading day and if P_n denotes the stock price at the beginning of the current trading day, then :

- $R_n = \frac{P_n - P_{n-1}}{P_{n-1}}$ is the current daily return on the stock investment;
- $u_n = \log(1 + R_n) = \log\left(\frac{P_n}{P_{n-1}}\right)$ is the *continuously compounded daily return*. It is also often called the *log-return*.

To see why u_n is called the continuously compounded return, take the exponential of both sides to get $\exp(u_n) = \frac{P_n}{P_{n-1}}$. Rearranging, we get $P_n = P_{n-1}\exp(u_n)$ so that u_n is the continuously compounded growth rate in prices between the beginning of the previous and the current trading days. This has to be contrasted with R_n, which is the simple growth rate in prices P_{n-1} and P_n without any compounding.

Following [Durham, 2006] to examine the performance of the LCGOMSMF on the stock market data, we compute the log-returns u_n over the daily Standard & Poor's 500 (S&P) index data from Jun. 23, 1980 to Aug. 30, 2002 ($N = 5604$), then we calculate $y_n = u_n^\star - \mu_r$, where μ_r is given in [Durham, 2007] and $u_n^\star$ denotes pre-processed log-return [Durham, 2006, Durham, 2007]. Next, we use the LCGOMSMF to compute the filtered volatility estimates within the ASV model, whose parameters are given in [Durham, 2007] and reported in Table 2.5. Our result is shown in Figure 2.7.

We find that the volatility estimates produced by the LCGOMSMF are consistent with the log-return process: as we can see in Figure 2.7, the intervals where the fluctuation of log-returns are low (*e.g.* between 1991 and 1995) match the intervals where the log-volatility

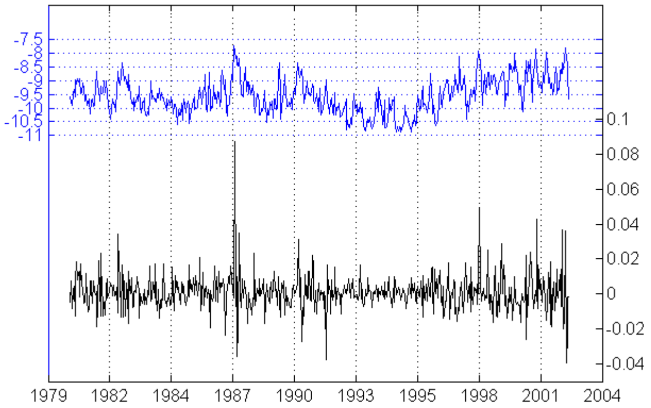


Figure 2.7: Trajectories of the S&P log-returns (down) and log-volatility estimates (up). The x-axis represents the dates for both trajectories, the y-axis labelling on the left concerns the log-volatility values, and the y-axis labelling on the right is related to the log-return values.

Parameter	μ_r	μ	ϕ	σ	ρ	β
Value	$7 \cdot 10^{-5}$	-9.54	0.98	0.17	-0.43	1.00

Table 2.5: The parameters of the ASV model for the stock market data.

is low, and vice versa. Moreover, we calculated the mean squared distance between the LCGOMSMF volatility estimates and those of the PF, and we find that this distance is negligible compared to the variance of the log-volatility process. Furthermore, when the number K of classes in the LCGOMSMF increases, this distance decreases as shown in Table 2.6.

2.4.4 Smoothing in dynamic beta models

The dynamic beta regression allows modeling monthly unemployment rate [Da-Silva et al., 2011]. More precisely, let $N \in \mathbb{N}^*$, Y_n in $[0,1]$ be the unemployment rate at time n, the dynamic beta model [Lopes and Tsay, 2011] for $\mathrm{Y}_{1:N}$ is:

$$\begin{aligned} &\mathrm{Y}_n \sim \text{Beta}\left(\frac{1}{c(1+\exp(\mathrm{X}_n))}, \frac{\exp(\mathrm{X}_n)}{c(1+\exp(\mathrm{X}_n))}\right); \\ &\mathrm{X}_{n+1} = \mu + \phi(\mathrm{X}_n - \mu) + \sigma \mathrm{U}_{n+1}, \end{aligned} \tag{2.54}$$

where μ, ϕ, σ and c are fixed and $\mathrm{U}_{1:N}$ are independent standard Gaussian vectors. We recall that for α, β in $\mathbb{R}_+^*$, $\text{Beta}(\alpha, \beta)$ denotes the beta distribution:

$$\text{Beta}(x; \alpha, \beta) = \begin{cases} \frac{x^{\alpha-1}(1-x)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases}, \tag{2.55}$$

$K = 5$	$K = 7$	$K = 9$
0.0022	0.0017	0.0015

Table 2.6: Mean square distances between the LCGOMSMF volatility estimates and those from the PF, with different number of classes. Here, $Var[\mathrm{X}_n] = 0.6145$.

where Γ denotes the Gamma function $\Gamma(x) = \int\limits_{0}^{\infty} t^{x-1} \exp(-t)dt$.

If $|\phi| < 1$ and $\mathrm{X}_1 \sim \mathcal{N}(\mu, \sigma_0^2)$ with $\sigma_0 = \frac{\sigma}{\sqrt{1-\phi^2}}$, then the autoregressive process of $\mathrm{X}_{1:N}$ is stationary [Dickey and Fuller, 1979], as well as $(\mathrm{X}_{1:N}, \mathrm{Y}_{1:N})$.

The conditional distribution of Y_n is generally skewed. Besides, we have:

$$\mathbb{E}\left[\mathrm{Y}_n \,|\mathrm{X}_n\right] = \frac{1}{1+\exp(\mathrm{X}_n)}; \tag{2.56}$$

$$\mathrm{Var}\left[\mathrm{Y}_n \,|\mathrm{X}_n\right] = \frac{\exp(\mathrm{X}_n)}{(1+\exp(\mathrm{X}_n))^2}\left(1 - \frac{1}{c+1}\right), \tag{2.57}$$

which means that c can be seen as a "noise level" of the observation of X_n made through Y_n. When $c = 0$, Y_n is a deterministic bijective function of X_n, and when c tends to infinity, the conditional variance of Y_n tends to its maximum. See Figure 2.8 for an illustration.

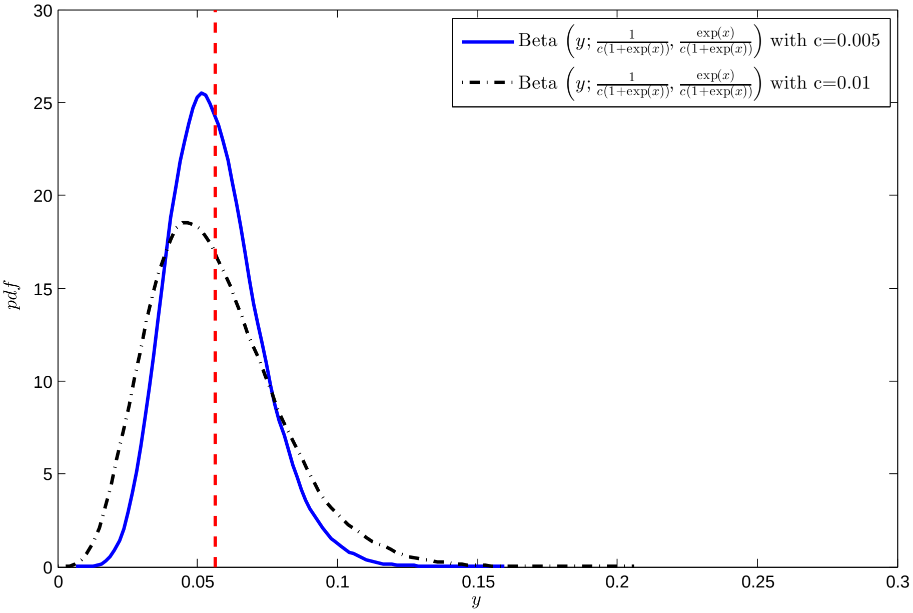


Figure 2.8: Distribution of Y_1 given $\mathrm{x}_1 = -2.82$, for different values of the "noise level" c. The vertical red line locates the common mean of both distributions.

The parameter ϕ is the lag-one autocorrelation of the latent process.

The dynamic beta regression is a particular case of the dynamic generalized linear model [Lopes and Tsay, 2011, West et al., 1985], where the latent process is Gaussian autoregressive and the observational distribution belongs to the exponential family.

Bayesian inference in model (2.54) is an established part of econometric and social analyses. We calibrated this model to a real-world data of the US monthly unemployment rate data from March 2002 to December 2015. The rounded values of the parameters are $\mu = -2.82, \phi = 0.95, \sigma_0 = 0.17$ and $c = 0.005$. In order to test the performance of LCGOMSMS in the case of model (2.54), we consider estimating $\mathrm{X}_{1:N}$ from $\mathrm{Y}_{1:N}$ when observed variables arise from (2.54) for various values of c and ϕ.

We use LCGOMSMS with different number of states K to estimate the latent variables from the $N = 1000$ observable ones, and we report our results in terms of the Relative Mean Squared Error (RMSE) for the mean of 100 independent experiments. The RMSE is relative to the common variance of variables in $\mathrm{X}_{1:N}$ which is σ_0^2. The results are in Table 2.7.

The dimensions of the latent variables and the observable ones are $a = b = 1$, the training sample size is $N' = 20000$, and $Q = 100$ is the number of EM iterations. For

			K					
	ϕ	c	2	3	5	7	PS	PF
1	0.95	0.005	0.38	0.32	0.31	0.29	0.29	0.41
2	0.95	0.01	0.54	0.43	0.39	0.38	0.38	0.50
3	0.99	0.005	0.40	0.21	0.16	0.16	0.16	0.22
4	0.99	0.01	0.42	0.37	0.31	0.24	0.23	0.28

Table 2.7: The RMSE of smoothing in model (2.54) with $\mu = -2.82$, $\sigma_0 = 0.17$ and four different values of lag-one autocorrelation ϕ and noise level c coefficients. The RMSE values for asymptotically optimal PF and PS are present as a reference.

comparison purpose, a similar outcome using a PF and Particle Smoother (PS) with $M = 1500$ particles is also given. We use the PS presented in Section 1.4.2.

We observe that for moderate values of K (*e.g.*, $K = 5$), the accuracy of the LCGOMSMS is satisfactory. When the latent process is highly persistent (ϕ close to 1) and when the "noise level" c is significant, one needs a greater number of states to estimate the latent process accurately.

The complexity of the particle smoother is $N \times m \times T$ while the complexity of the LCGOMSMS is $N \times K^2$. In practice, the computation time of our method is quite the same as the one consumed by a particle smoother using K^2 particles, which is rather a small number of particles. As a consequence, one may use a large value of K if needed.

2.4.5 Smoothing in asymmetric stochastic volatility model

Here, we provide results of experiments of Bayesian smoothing in the ASV model 2.53. The experiment protocol consists in estimating $\mathrm{X}_{1:N}$ from $\mathrm{Y}_{1:N}$ by using LCGOMSMS with different number of states K, $N = 1000$ observable ones, and we report our results in terms of RMSE for the mean of 100 independent experiments. The RMSE is relative to the common variance of variables in $\mathrm{X}_{1:N}$ which is $\frac{\sigma^2}{1-\phi^2}$. The results are provided in Table 2.8.

The dimensions of the latent variables and the observable ones are $a = b = 1$, the training sample size is $N' = 20000$, and $Q = 100$ is the number of EM iterations. For comparison purpose, a similar outcome using a PF and PS with $M = 1500$ particles is also given. We use the PS presented in Section 1.4.2.

We observe that for moderate values of K (*e.g.*, $K = 5$), the accuracy of the LCGOMSMS is satisfactory.

			K				
	ρ	λ^2	2	3	5	7	PS
1	-0.90	0.19	0.23	0.21	0.20	0.20	0.19
2	-0.80	0.36	0.36	0.34	0.32	0.32	0.32
3	-0.50	0.75	0.57	0.55	0.55	0.55	0.54
4	-0.30	0.91	0.65	0.63	0.62	0.62	0.62
5	-0.00	1.00	0.70	0.67	0.66	0.66	0.66

Table 2.8: The MSE of smoothing in the ASV model with $\mu = 0.5$, $\beta = 0.5$, $\phi = 0.5$ and five different values of λ^2 and ρ such that $\lambda^2 + \rho^2 = 1$ and $\sigma^2 + \phi^2 = 1$ for a unitary unconditional variance of X_n.

2.4.6 Smoothing in Markov-switching stochastic volatility model

Let N in $\mathbb{N}^*$, the Markov Switching Stochastic Volatility (MSSV) model [So et al., 1998, Carvalho and Lopes, 2007] of $(\mathrm{X}_{1:N}, \mathrm{Y}_{1:N})$ reads as follows:

$$\forall n \in \{1 : N-1\}, \mathrm{X}_{n+1} = \gamma_1 + \sum_{j=2}^{q} \gamma_j \mathbb{1}_{[j;+\infty]}(\mathrm{S}_{n+1}) + \phi \mathrm{X}_n + \sigma \mathrm{U}_{n+1}; \tag{2.58}$$

$$\forall n \in \{1 : N\}, \mathrm{Y}_n = \exp(\mathrm{X}_n/2)\mathrm{V}_n, \tag{2.59}$$

where

— $\mathbb{1}_{\mathscr{A}}(.)$ is the indicator function of a set $\mathscr{A}$;

— $\mathrm{S}_{1:N}$ is a stationary discrete Markov chain with k states;

— for all n in $\{1 : N-1\}$, $p(\mathrm{s}_{n+1} | \mathrm{x}_{1;n}, \mathrm{y}_{1;n}, \mathrm{s}_{1:n}) = p(\mathrm{s}_{n+1} | \mathrm{s}_n)$;

— $\gamma_1, \ldots, \gamma_q, \phi$, σ are fixed parameters in $\mathbb{R}$

— $\mathrm{U}_{1:N}, \mathrm{V}_{1:N}$ are independent standard Gaussian vectors.

We set $q = 2$, $p_{11} = p(\mathrm{s}_{n+1} = 1 | \mathrm{s}_n = 1)$ and $p_{22} = p(\mathrm{s}_{n+1} = 2 | \mathrm{s}_n = 2)$. Since random sampling is straightforward within the MSSV framework, LCGOMSMS is applicable. Table 2.9 shows its results for some MSSV parameters. We use the PS presented in Section 1.4.2.

			K				
	p_{11}	p_{22}	2	3	5	7	PS
1	0.99	0.985	0.02	0.02	0.02	0.02	0.02
2	0.85	0.25	0.71	0.38	0.38	0.38	0.38
3	0.5	0.5	0.45	0.42	0 42	0.42	0.42

Table 2.9: MSE of smoothing in the MSSV model with $k = 2$, $\gamma_1 = -5.0$, $\gamma_2 = -3.0$, $\sigma^2 = 0.1$, $\phi = 0.5$ and three different values of p_{11} and p_{22}.

We observe that if K is large enough, the smoothed output of the LCGOMSMS is as good as the statistically optimal one, produced by the PS. Our smoothing procedure is riskless from the weight degeneracy phenomenon frequently encountered in particle methods and seems to be robust even in the case of the switching models.

2.5 Conclusion

CGOMSMs are POMPs with hybrid state space in which exact fast Bayesian inference is feasible. We presented the CGOMSM framework and the related algorithms of Bayesian inference. We also proposed LCGOMSMF and LCGOMSMS, which are CGOMSM-based methods for Bayesian inference in non-Gaussian non-linear systems. They rely on a single global approximation of the system done by the EM algorithm, the latter constitutes the major contribution of the author. LCGOMSMF and LCGOMSMS are very general and has several advantages over existing techniques. Their performances have been examined on synthetic samples related to SV models as well as on real data. We found that the LCGOMSMF attains the asymptotic performances of the PF, what could not be obtained with GSF and GSUKF.

The filtering procedure which is the object of the Section is applicable in general stationary (or asymptotically stationary) Markov dynamic systems, provided that one can sample its realizations. It is as fast as the standard Kalman filter, provided that one adjusts the filter to a particular model via *e.g.* the EM algorithm.

Chapter 3

Markovian grid-based Bayesian state estimation

The jump Markov system [Andrieu et al., 2003a], and, more generally, the hybrid-state Partially Observable Markov Process (POMP), allows modeling time series whose dynamics depend upon unknown exogenous discrete-valued factors. It applies in econometrics [Kim, 1994, Zhu and Rahman, 2015], finance [Azzouzi and Nabney, 1999, Panopoulou and Pantelidis, 2015], tracking [Weiss et al., 2004], speech recognition [Mesot and Barber, 2007, Rosti and Gales, 2003], pattern recognition [Pavlovic et al., 2001], among others [Ristic et al., 2004, Ghahramani and Hinton, 2000]. These models are also known as regime-switching models (processes) and interacting multiple models. Exact Bayesian state estimation in such a system is usually impossible [Lerner, 2002] unless the system is a hidden Markov chain with finite discrete state space [Andrieu et al., 2003a].

Switching filters are algorithms for Bayesian inference in hybrid-state POMPs. They include sampling-based approaches [Kim and Nelson, 1999, Doucet et al., 2001, Fong et al., 2002, Särkkä et al., 2012, Carter and Kohn, 1996] and deterministic ones [Zoeter and Heskes, 2006, Zhong et al., 2008]. Sampling-based filters rely on Monte Carlo and quasi-Monte Carlo methods [Caflisch, 1998, Niederreiter, 2010, Morokoff and Caflisch, 1995, Gerber and Chopin, 2015]. These filters are asymptotically optimal, but can be computationally intensive. Usual deterministic ones are modified versions of the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or Gauss-Hermite Filter (GHF) which handle the discrete-valued process of switches. EKF, UKF, GHF and their variants are discussed in [Afshari et al., 2017]. However, sampling-based methods may be computationally expensive in the case of high-dimensional state space, while deterministic ones are generally not proven to converge to the Bayesian solution.

In this chapter, we introduce a novel approach for Bayesian inference in stationary hybrid-state POMPs, which we call Markovian Grid-Based State Estimator (MGSE). This method allows using sparse grids [Bungartz and Griebel, 2004] for reducing the dimensionality effect, what we presented in [Gorynin et al., 2016b], which allowed efficient state estimation even in the case of high-dimensional state space. We extend our previous study [Gorynin et al., 2016b] by proving the convergence of the MGSE towards the Bayesian solution in the POMPs.

The chapter is organized as follows. The next section is a background on the grid methods and quadrature rules. The second section is the main contribution of the author and introduces Markovian grids as a computational tool for statistical inference in hidden Markov and switching hidden Markov systems. The third section contains experiments and the last one contains conclusions and perspectives.

3.1 Background

Definition 5. *Monomial in $\mathbb{R}^a$ of order t.*

Let a in $\mathbb{N}$, t in $\mathbb{N}$. We say that $\mathrm{m} \in \mathscr{F}\Big(\mathbb{R}^a \to \mathbb{R}\Big)$ is a monomial in $\mathbb{R}^a$ of order t if there exists an a-uplet $(\alpha_1, \ldots, \alpha_a)$ in $\mathbb{N}^a$ such that

$$\alpha_1 + \ldots + \alpha_a = t, \qquad \forall \mathbf{z} \in \mathbb{R}^a, \mathrm{m}(\mathbf{z}) = \mathbf{z}[1]^{\alpha_1} \ldots \mathbf{z}[a]^{\alpha_a},$$

where for each i in $\{1 : a\}$, $\mathbf{z}[i]$ denotes the i-th coefficient of vector $\mathbf{z}$.

The set of all monomials in $\mathbb{R}^a$ of order less than or equal to t is denoted by $\mathscr{M}_t\Big(\mathbb{R}^a\Big)$.

Definition 6. *Continuous-discrete domains.*

We say that a set Γ is a continuous-discrete domain if there exist a non empty finite discrete set Ω and $a \in \mathbb{N}$ such that $\Gamma = \Omega \times \mathbb{R}^a$. The set of continuous-discrete domains is denoted as $\mathscr{D}$ and is defined as

$$\mathscr{D} = \bigcup_{\substack{0<\mathrm{Card}(\Omega)<\infty \\ a\in\mathbb{N}}} \Omega \times \mathbb{R}^a. \tag{3.1}$$

Note that for any $a \in \mathbb{N}$ and Ω such that $\mathrm{Card}(\Omega) = 1$, we have there is a trivial bijection between $\Omega \times \mathbb{R}^a$ and $\mathbb{R}^a$. In this case, we pose for simplicity $\Omega \times \mathbb{R}^a = \mathbb{R}^a$, thus we have $\mathbb{R}^a \in \mathscr{D}$.

Definition 7. *Vector valued functions.*

Let $\Gamma \in \mathscr{D}$, the set of vector-valued functions on Γ is denoted by $\mathscr{F}(\Gamma)$, and is defined as

$$\mathscr{F}(\Gamma) = \bigcup_{d\in\mathbb{N}} \mathscr{F}\Big(\Gamma \to \mathbb{R}^d\Big). \tag{3.2}$$

Definition 8. *Analytic function on $\mathbb{R}^a$.*

Let $a, b \in \mathbb{N}$,

- *$\boldsymbol{f} \in \mathscr{F}(\mathbb{R} \to \mathbb{R}^b)$ is analytic on $\mathbb{R}$ if for each $\mathbf{x}_0$ in $\mathbb{R}$, there exists an open neighborhood of $\mathbf{x}_0$ in which $\boldsymbol{f}$ is equal to a convergent power series in $\mathbb{R}^b$ [Gunning, 1965];*
- *$\boldsymbol{f} \in \mathscr{F}(\mathbb{R}^a \to \mathbb{R}^b)$ is analytic on $\mathbb{R}^a$ if it is analytic in each variable separately, that is for any fixed $(a-1)$ coordinates, the restriction of $\boldsymbol{f}$ is an analytic function of the remaining coordinate [Pedrick, 1994].*

The set of analytic functions from $\mathbb{R}^a$ to $\mathbb{R}^b$ is denoted as $\mathscr{A}\Big(\mathbb{R}^a \to \mathbb{R}^b\Big)$. The subset of analytic functions in $\mathscr{F}(\mathbb{R}^a)$ is denoted as $\mathscr{A}\Big(\mathbb{R}^a\Big)$. Let us recall that the sums, products, and compositions of elements in $\mathscr{A}\Big(\mathbb{R}^a\Big)$ are also in $\mathscr{A}\Big(\mathbb{R}^a\Big)$.

Let us extend the concept of analytic functions to the continuous-discrete domains.

Definition 9. *Analytic function on Γ.*

Let $a, b \in \mathbb{N}$, Ω be a finite discrete set and $\Gamma = \Omega \times \mathbb{R}^a$. We say that $\boldsymbol{f} \in \mathscr{F}\Big(\Gamma \to \mathbb{R}^b\Big)$ is analytic on Γ if for each ω in Ω, $\boldsymbol{f}$ is analytic on $\{(\omega, \mathbf{z}) | \mathbf{z} \in \mathbb{R}^a\}$.

The remaining symbols are:
– For $\Gamma \in \mathscr{D}$, the set of positive measures on Γ is denoted by $\mathscr{U}(\Gamma)$;
– For $\Gamma \in \mathscr{D}$, $\mu \in \mathscr{U}(\Gamma)$ and $\boldsymbol{f} \in \mathscr{F}(\Gamma)$ μ-integrable, we denote

$$< \mu, \boldsymbol{f} >= \int_\Gamma \boldsymbol{f} \mathrm{d}\mu = \sum_{\omega\in\Omega} \int_{\mathbb{R}^a} \boldsymbol{f}(\omega, \mathbf{z}) \mu(\omega, \mathbf{z}) \mathrm{d}\mathbf{z}; \tag{3.3}$$

– For $\Gamma, \Gamma' \in \mathscr{D}$, the product measure of $\mu \in \mathscr{U}(\Gamma)$ and $\mu' \in \mathscr{U}(\Gamma')$, is denoted as $\mu \otimes \mu'$;

– For $\Gamma \in \mathscr{D}$ and $\mathbf{x} \in \Gamma$, the Dirac delta function is denoted by $\delta_{\mathbf{x}} \in \mathscr{U}(\Gamma)$. Recall that for each $\boldsymbol{f}$ in $\mathscr{F}\Big(\Gamma\Big)$ such that $\boldsymbol{f}(\mathbf{x})$ is finite, we have

$$< \delta_{\mathbf{x}}, \boldsymbol{f} >= \boldsymbol{f}(\mathbf{x});$$

– The indicator function of a set $\mathscr{S}$ is denoted by $\mathbb{1}_{\mathscr{S}}$;

– For $k, n \in \mathbb{N}$, $0 \leq k \leq n$, the binomial coefficient defined by $\frac{n!}{k!(n-k)!}$ is denoted by C_n^k.

Definition 10. *Tensor product of functions.*

Let $\mathscr{X}_1$ and $\mathscr{X}_2$ be two sets, $\boldsymbol{f}_1 \in \mathscr{F}\Big(\mathscr{X}_1 \to \mathbb{R}\Big)$ and $\boldsymbol{f}_2 \in \mathscr{F}\Big(\mathscr{X}_2 \to \mathbb{R}\Big)$. The tensor product of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ is an element of $\mathscr{F}\Big(\mathscr{X}_1 \times \mathscr{X}_2 \to \mathbb{R}\Big)$ denoted as $\boldsymbol{f}_1 \otimes \boldsymbol{f}_2$ and defined by

$$\forall \mathbf{x}_1 \in \mathscr{X}_1, \mathbf{x}_2 \in \mathscr{X}_2, (\boldsymbol{f}_1 \otimes \boldsymbol{f}_2)(\mathbf{x}_1, \mathbf{x}_2) = \boldsymbol{f}_1(\mathbf{x}_1)\boldsymbol{f}_2(\mathbf{x}_2). \tag{3.4}$$

Definition 11. Γ*-grid.*

Let $M \in \mathbb{N}^*$, $\Gamma \in \mathscr{D}$, $\Lambda = \{\gamma_1, \dots, \gamma_M\} \subset \Gamma$ and π in $\mathscr{F}\Big(\Lambda \to \mathbb{R}\Big)$. Then $\mathscr{I} = \{\Lambda, \pi\}$ is called a Γ-grid. $\{\gamma_i\}_{1 \leq i \leq M}$ are called grid nodes and $\{\pi(\gamma_i)\}_{1 \leq i \leq M}$ are called grid weights.

Definition 12. Γ*-grid measure.*

Let $\Gamma \in \mathscr{D}$ and $\mathscr{I} = \{\Lambda, \pi\}$ be a Γ-grid. The grid measure corresponding to $\mathscr{I}$ is defined as

$$T_{\mathscr{I}} = \sum_{\gamma \in \Lambda} \delta_{\gamma} \pi(\gamma). \tag{3.5}$$

Definition 13. *Quadrature rule induced by a grid.*

Let $a \in \mathbb{N}$, $\Gamma \in \mathscr{D}$, $\boldsymbol{f}$ in $\mathscr{F}\Big(\Gamma \to \mathbb{R}^a\Big)$ and $\mathscr{I} = \{\Lambda, \pi\}$ be a Γ-grid. The quadrature rule for $\boldsymbol{f}$ induced by $\mathscr{I}$ is defined as $< T_{\mathscr{I}}, \boldsymbol{f} >$, where $T_{\mathscr{I}}$ is the grid measure corresponding to $\mathscr{I}$. Specifically, we have

$$< T_{\mathscr{I}}, \boldsymbol{f} >= \sum_{\gamma \in \Lambda} \boldsymbol{f}(\gamma)\pi(\gamma). \tag{3.6}$$

For the sake of simplicity, we denote in the same way $< T_{\mathscr{I}}, \boldsymbol{f} >$ and $< \mathscr{I}, \boldsymbol{f} >$.

Definition 14. *Degree of precision of a Γ-grid.*

Let $a \in \mathbb{N}$, $t \in \mathbb{N}$, Ω be a finite discrete set, $\Gamma = \Omega \times \mathbb{R}^a$, $\mu \in \mathscr{U}(\Gamma)$ and $\mathscr{I} = \{\Lambda, \pi\}$ be a Γ-grid. We say that $\mathscr{I}$ has a degree of precision t with respect to μ if for each monomial m in $\mathscr{M}_t\Big(\mathbb{R}^a\Big)$ and each $\omega \in \Omega$, one has

$$\sum_{\boldsymbol{z} \in \{\mathbf{x} \in \mathbb{R}^a | (\omega, \mathbf{x}) \in \Lambda\}} \mathrm{m}(\boldsymbol{z})\pi(\omega, \boldsymbol{z}) = \mu(\omega) < \mu, \mathrm{m} > . \tag{3.7}$$

Additionally, for any g in $\mathscr{F}\Big(\Gamma \to \mathbb{R}_+\Big)$, we say that $\mathscr{I}$ has a degree of precision of t with respect to g if it has a degree of precision t with respect to $T_g \in \mathscr{U}(\Gamma)$ defined by

$$\forall \boldsymbol{f} \in \mathscr{F}\Big(\Gamma\Big), < T_g, \boldsymbol{f} >= \sum_{\omega \in \Omega} \int_{\mathbb{R}^a} \boldsymbol{f}(\omega, \mathbf{z}) g(\omega, \mathbf{z}) \mathrm{d}\mathbf{z}. \tag{3.8}$$

In the case where $\Omega = \emptyset$, we have $\Gamma = \mathbb{R}^a$ and we say that $\mathscr{I}$ has a degree of precision t with respect to μ if for each monomial m in $\mathscr{M}_t\Big(\mathbb{R}^a\Big)$, one has

$$< \mathscr{I}, \mathrm{m} >=< \mu, \mathrm{m} > . \tag{3.9}$$

Similarly, for any g in $\mathscr{F}\Big(\mathbb{R}^a \to \mathbb{R}_+\Big)$, we say that $\mathscr{I}$ has a degree of precision of t with respect to g if it has a degree of precision t with respect to $T_g \in \mathscr{U}(\mathbb{R}^a)$ defined by

$$\forall \boldsymbol{f} \in \mathscr{F}\Big(\mathbb{R}^a\Big), < T_g, \boldsymbol{f} >= \int\limits_{\mathbb{R}^a} \boldsymbol{f}(\mathbf{z}) g(\mathbf{z}) \mathrm{d}\mathbf{z}. \tag{3.10}$$

Definition 15. *Strongly arbitrarily precise sequence of Γ-grids.*

Let $\Gamma \in \mathscr{D}$, $\mu \in \mathscr{U}(\Gamma)$ and $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ be a sequence of Γ-grids. We say that $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to μ if for any $t \in \mathbb{N}$, there exists $L_t \in \mathbb{N}^*$ such that for all L greater or equal to L_t, $\mathscr{I}_L$ has a degree of precision t with respect to μ.

Additionally, for any g in $\mathscr{F}\Big(\Gamma \to \mathbb{R}_+\Big)$, we say that $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to g if it is strongly arbitrarily precise with respect to $T_g \in \mathscr{U}(\Gamma)$ defined by (3.8).

Definition 16. *Grid-by-scalar product.*

Let $\Gamma \in \mathscr{D}$, $\mathscr{I} = \{\Lambda, \pi\}$ be a Γ-grid and h in $\mathscr{F}\Big(\Gamma \to \mathbb{R}\Big)$. We define the grid-by-scalar product of $\mathscr{I}$ and h as follows:

$$\mathscr{I}h = \{\Lambda, h\pi\}.$$

Note that we also have,

$$T_{\mathscr{I}h} = T_{\mathscr{I}}h.$$

Let us now introduce the concept of consistency of a grid sequence with a measure in $\mathscr{U}(\Gamma)$.

Definition 17. *Consistent sequences of Γ-grids.*

Let $\Gamma \in \mathscr{D}$, $\mu \in \mathscr{U}(\Gamma)$ and $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ be a sequence of Γ-grids. We say that $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is consistent with μ if for any $\boldsymbol{f}$ in $\mathscr{A}\Big(\Gamma\Big)$, $(< \mathscr{I}_L, \boldsymbol{f} >)_{L\in\mathbb{N}^*}$ converges to $< \mu, \boldsymbol{f} >$ in the sense of convergence of numerical sequences.

Additionally, for any g in $\mathscr{F}\Big(\Gamma \to \mathbb{R}\Big)$, we say that $(\mathscr{I}_M)_{M\in\mathbb{N}^*}$ is consistent with g if it is consistent with $T_g \in \mathscr{U}(\Gamma)$ defined by (3.8).

Definition 18. *Union grid.*

Let $\Gamma \in \mathscr{D}$, $\mathscr{I} = \{\Lambda, \pi\}$ and $\mathscr{I}' = \{\Lambda', \pi'\}$ be two Γ-grids. We define the union grid $\mathscr{J}$ on Γ as

$$\mathscr{J} = \left\{\Lambda \cup \Lambda', \mathbb{1}_{\Lambda}\pi + \mathbb{1}_{\Lambda'}\pi'\right\},$$

and we note it as

$$\mathscr{J} = \mathscr{I} + \mathscr{I}'.$$

Note that we also have,

$$T_{\mathscr{I}+\mathscr{I}'} = T_{\mathscr{I}} + T_{\mathscr{I}'}.$$

Thus, for any $\boldsymbol{f}$ in $\mathscr{F}(\Gamma)$,

$$< \mathscr{I} + \mathscr{I}', \boldsymbol{f} >=< \mathscr{I}, \boldsymbol{f} > + < \mathscr{I}', \boldsymbol{f} > .$$

Definition 19. *Weekly arbitrarily precise sequence of Γ-grids.*

Let $\Gamma \in \mathscr{D}$, $\mu \in \mathscr{U}(\Gamma)$ and $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ be a sequence of Γ-grids. We say that $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is weekly arbitrarily precise with respect to μ if there exist F in $\mathbb{N}^*$, $h_1, \ldots, h_F$ in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $\mu_1, \ldots, \mu_F \in \mathscr{U}(\Gamma)$ and F sequences of Γ-grids $\left(\mathscr{I}_L^{(1)}\right)_{L\in\mathbb{N}^*}, \ldots, \left(\mathscr{I}_L^{(F)}\right)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\mu_1, \ldots, \mu_F$ respectively such that $\mu = \sum\limits_{f=1}^{F} \mu_f h_f$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum\limits_{f=1}^{F} \mathscr{I}_L^{(f)} h_f$.

Weekly arbitrarily precise grid sequences will be simply referred as arbitrarily precise further in the text.

The following two propositions result from an original research of the author.

Proposition 7. *Let* $\Gamma \in \mathscr{D}$*,* $\mu \in \mathscr{U}(\Gamma)$*,* h *in* $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$ *such that* μh *would be in* $\mathscr{U}(\Gamma)$*. Let* $(\mathscr{I}_L)_{L\in\mathbb{N}^*} = (\{\Lambda_L, \pi_L\})_{L\in\mathbb{N}^*}$ *be a sequence of* Γ*-grids arbitrarily precise with respect to* μ*. Then* $(\mathscr{I}_L h)_{L\in\mathbb{N}^*}$ *is arbitrarily precise with respect to* μh*.*

Proof. $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to μ, thus there exist F in $\mathbb{N}^*$, $h_1, \ldots, h_F$ in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $\mu_1, \ldots, \mu_F \in \mathscr{U}(\Gamma)$ and F sequences of Γ-grids $\Big(\mathscr{I}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{I}_L^{(F)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\mu_1, \ldots, \mu_F$ respectively such that $\mu = \sum\limits_{f=1}^{F} \mu_f h_f$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum\limits_{f=1}^{F} \mathscr{I}_L^{(f)} h_f$. Since for each f in $\{1 : F\}$, $h_f h \in \mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$ as the product of two analytical functions in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $(\mathscr{I}_L h)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to μh, as we have $\mu h = \sum\limits_{f=1}^{F} \mu_f h_f h$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum\limits_{f=1}^{F} \mathscr{I}_L^{(f)} h_f h$. □

Proposition 8. *Let* $\Gamma \in \mathscr{D}$*,* $\mu, \nu \in \mathscr{U}(\Gamma)$*,* $\Big(\mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ *and* $\Big(\mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ *be two sequences of* Γ*-grids arbitrarily precise with respect to* μ *and* ν *respectively. Then* $\Big(\mathscr{I}_L + \mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ *is arbitrarily precise with respect to with* $\mu + \nu$*.*

Proof. $\Big(\mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ and $\Big(\mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ are arbitrarily precise with respect to μ and ν respectively, thus there exist F_1, F_2 in $\mathbb{N}^*$, $u_1, \ldots, u_{F_1}, k_1, \ldots, k_{F_2}$ in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $\mu_1, \ldots, \mu_{F_1}, \nu_1, \ldots, \nu_{F_2} \in \mathscr{U}(\Gamma)$, F_1 sequences of Γ-grids $\Big(\mathscr{I}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{I}_L^{(F_1)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\mu_1, \ldots, \mu_{F_1}$ respectively and F_2 sequences of Γ-grids $\Big(\mathscr{K}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{K}_L^{(F_2)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\nu_1, \ldots, \nu_{F_2}$ respectively such that $\mu = \sum\limits_{f=1}^{F_1} \mu_f u_f$, $\nu = \sum\limits_{f=1}^{F_2} \nu_f k_f$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum\limits_{f=1}^{F_1} \mathscr{I}_L^{(f)} u_f$, $\mathscr{K}_L = \sum\limits_{f=1}^{F_2} \mathscr{K}_L^{(f)} k_f$.

Let $F = F_1 + F_2$. Define $h_1, \ldots, h_F$ in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $\sigma_1, \ldots, \sigma_F \in \mathscr{U}(\Gamma)$ and F sequences of Γ-grids $(\mathscr{S}_L^{(1)})_{L\in\mathbb{N}^*}, \ldots, (\mathscr{S}_L^{(F)})_{L\in\mathbb{N}^*}$ as follows:

$$h_f = \begin{cases} u_f & \text{if } f \le F_1; \\ k_{f-F_1} & \text{if } f > F_1; \end{cases} \quad \sigma_f = \begin{cases} \mu_f & \text{if } f \le F_1; \\ \nu_{f-F_1} & \text{if } f > F_1; \end{cases} \quad \forall L \in \mathbb{N}^*, \mathscr{S}_L^{(f)} = \begin{cases} \mathscr{I}_L^{(f)} & \text{if } f \le F_1; \\ \mathscr{K}_L^{(f-F_1)} & \text{if } f > F_1. \end{cases}$$

Thus, for each L in $\mathbb{N}^*$, $\mathscr{I}_L + \mathscr{K}_L = \sum\limits_{f=1}^{F} \mathscr{S}_L^{(f)} h_f$ and $\mu + \nu = \sum\limits_{f=1}^{F} \sigma_f h_f$. Since $\Big(\mathscr{S}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{S}_L^{(F)}\Big)_{L\in\mathbb{N}^*}$ are strongly arbitrarily precise with respect to $\sigma_1, \ldots, \sigma_F$, $\Big(\mathscr{I}_L + \mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to $\mu + \nu$. □

Proposition 9. *Let* $\Gamma \in \mathscr{D}$*,* μ *in* $\mathscr{U}(\Gamma)$*,* $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ *be a sequence of* Γ*-grids. If* $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ *is strongly arbitrarily precise with respect to* μ*, then* $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ *is consistent with* μ*.*

Proof. See [Gerstner and Griebel, 1998, Novak and Ritter, 1997, Wasilkowski and Wozniakowski, 1995]. □

The following corollary results from an original research of the author.

Corollary 9.1. *Let $a \in \mathbb{N}$, Ω be a finite discrete set, $\Gamma = \Omega \times \mathbb{R}^a$, μ in $\mathscr{U}(\Gamma)$, $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ be a sequence of Γ-grids. If $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to μ, then $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is consistent with μ.*

Proof. $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to μ, thus there exist F in $\mathbb{N}^*$, $h_1, \dots, h_F$ in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, $\mu_1, \dots, \mu_F \in \mathscr{U}(\Gamma)$ and F sequences of Γ-grids $\Big(\mathscr{I}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \dots, \Big(\mathscr{I}_L^{(F)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\mu_1, \dots, \mu_F$ respectively such that $\mu = \sum\limits_{i=1}^{F} \mu_i h_i$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum\limits_{i=1}^{F} \mathscr{I}_L^{(i)} h_i$. Let $\boldsymbol{f}$ in $\mathscr{A}\Big(\Gamma\Big)$, we have

$$\forall L \in \mathbb{N}^*, < \mathscr{I}_L, \boldsymbol{f} >=< \sum_{i=1}^{F} \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} >= \sum_{i=1}^{F} < \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} > . \tag{3.11}$$

For each i in $\{1 : F\}$, we have

$$\forall L \in \mathbb{N}^*, < \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} >= \sum_{\gamma \in \Lambda_L^{(i)}} \boldsymbol{f}(\gamma) \pi_L^{(i)}(\gamma) h_i(\gamma) =< \mathscr{I}_L^{(i)}, \boldsymbol{f} h_i >, \tag{3.12}$$

where $\mathscr{I}_L^{(i)} = \Big\{\Lambda_L^{(i)}, \pi_L^{(i)}\Big\}$. Since $\boldsymbol{f}$ and h_i are in $\mathscr{A}\Big(\Gamma\Big)$, $\boldsymbol{f} h_i$ is also in $\mathscr{A}\Big(\Gamma\Big)$. Thus,

$$\lim_{L\to\infty} < \mathscr{I}_L^{(i)}, \boldsymbol{f} h_i >=< \mu_i, \boldsymbol{f} h_i >$$

since $\Big(\mathscr{I}_L^{(i)}\Big)_{L\in\mathbb{N}^*}$ is consistent with μ_i *cf.* Proposition 9. By substituting (3.12) in the above equation, we have

$$\lim_{L\to\infty} < \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} >=< \mu_i, \boldsymbol{f} h_i > .$$

Next, we have

$$< \mu_i, \boldsymbol{f} h_i >= \sum_{\omega\in\Omega} \int_{\mathbb{R}^a} \boldsymbol{f}(\omega, \mathbf{z}) h_i(\omega, \mathbf{z}) \mu_i(\omega, \mathbf{z}) \mathrm{d}\mathbf{z} =< \mu_i h_i, \boldsymbol{f} >,$$

thus

$$\lim_{L\to\infty} < \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} >=< \mu_i h_i, \boldsymbol{f} > . \tag{3.13}$$

By substituting (3.13) in (3.11), we have

$$\lim_{L\to\infty} < \mathscr{I}_L, \boldsymbol{f} >= \lim_{L\to\infty} \sum_{i=1}^{F} < \mathscr{I}_L^{(i)} h_i, \boldsymbol{f} >= \sum_{i=1}^{F} \lim_{L\to\infty} < \mathscr{I}_L^{(i)}, \boldsymbol{f} h_i >= \sum_{i=1}^{F} < \mu_i h_i, \boldsymbol{f} >=$$
$$< \sum_{i=1}^{F} \mu_i h_i, \boldsymbol{f} >=< \mu, \boldsymbol{f} >,$$

thus $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is consistent with μ. □

The following subsection focuses on the construction of grids on $\mathbb{R}$ [Luceno, 1999].

3.1.1 Construction of arbitrarily precise sequences of $\mathbb{R}$-grids by Gaussian quadrature

Here we recall [Luceno, 1999] the construction of grids corresponding to the Gaussian quadrature rule.

Definition 20. *Moments of a real-valued function.*

Let g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$, i in $\mathbb{N}$, the i-th moment of g is defined as

$$m_i[g] = \int\limits_{\mathbb{R}} \mathrm{z}^i g(\mathrm{z})\mathrm{dz}. \tag{3.14}$$

The Gaussian quadrature rule is used to define a sequence of grids $(\mathscr{G}_L)_{L\in\mathbb{N}^*}$ consistent with g.

Definition 21. $\mathbb{R}$*-grid corresponding to the M-point Gaussian quadrature rule.*

Let $M \in \mathbb{N}^*$, g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$ such that for each i in $\{0 : 2M - 1\}$, the i-th moment of g is finite. Let $P_1, \dots, P_M$ be polynomials computed recursively by

$$\forall i \in \{0 : M-1\}, P_{i+1}(\mathrm{z}) = (\mathrm{z} - \delta_{i+1})P_i(\mathrm{z}) - \gamma_{i+1}^2 P_{i-1}(\mathrm{z}),$$

with $P_{-1}(\mathrm{z}) = 0$, $P_0(\mathrm{z}) = 1$, $\gamma_1 = 0$ and

$$\forall i \in \{0 : M-1\}, \qquad \delta_{i+1} = \frac{\int\limits_{\mathbb{R}} \mathrm{z}P_i^2(\mathrm{z})g(\mathrm{z})\mathrm{dz}}{\int\limits_{\mathbb{R}} P_i^2(\mathrm{z})g(\mathrm{z})\mathrm{dz}}, \qquad \gamma_{i+1}^2 = \frac{\int\limits_{\mathbb{R}} P_i^2(\mathrm{z})g(\mathrm{z})\mathrm{dz}}{\int\limits_{\mathbb{R}} P_{i-1}^2(\mathrm{z})g(\mathrm{z})\mathrm{dz}}$$

computed using $\{m_i[g]\}_{1\le i\le 2M-1}$.

The $\mathbb{R}$-grid $\mathscr{G}_M = \{\Lambda_M, \pi_M\}$ corresponding to the M-point Gaussian quadrature rule with respect to g is defined by the grid nodes, which are the M distinct roots of P_M, and the grid weights, which solve the linear system below [Luceno, 1999]

$$\begin{cases} \sum\limits_{z\in\Lambda_M} \pi_M(z)P_0(z) & = 1; \\ \sum\limits_{z\in\Lambda_M} \pi_M(z)P_i(z) & = 0 \quad \forall i \in \{1, \dots, M-1\}. \end{cases} \tag{3.15}$$

Proposition 10. *Let M in $\mathbb{N}^*$, g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$ such that for each i in $\{0 : 2M-1\}$, the i-th moment of g is finite. Let $\mathscr{G}_M$ be the $\mathbb{R}$-grid $\mathscr{G}_M = \{\Lambda_M, \pi_M\}$ corresponding to the M-point Gaussian quadrature with respect to g. Then $\mathscr{G}_M$ has a degree of precision $2M-1$ with respect to g.*

Proof. See [Luceno, 1999]. □

Corollary 10.1. *[Luceno, 1999] Let g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$ such that for each i in $\mathbb{N}$, the i-th moment of g is finite. For each M in $\mathbb{N}^*$, let $\mathscr{G}_M$ be the $\mathbb{R}$-grid corresponding to the M-point Gaussian quadrature with respect to g. Then $(\mathscr{G}_M)_{M\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to g.*

Proof. For each t in $\mathbb{N}$, one can choose M_t in $\mathbb{N}^*$ such that $2M_t - 1 > t$, thus for each M in $\mathbb{N}^*$ greater than or equal to M_t, $\mathscr{G}_M$ would have a degree of precision t with respect to g *cf.* Proposition 10. Therefore, $(\mathscr{G}_M)_{M\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to g. □

Remark 1. *Let g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$ be a probability density function and $\mathscr{G}_M = \{\Lambda_M, \pi_M\}$ be the $\mathbb{R}$-grid corresponding to the M-point Gaussian quadrature with respect to g. For any f in $\mathscr{A}\Big(\mathbb{R} \to \mathbb{R}\Big)$, we have [Barrett, 1961]*

$$| < \mathscr{G}_M, f > - < g, f > | = \mathscr{O}\left(\frac{1}{M^2}\right). \tag{3.16}$$

Remark 2. *For a comparison purpose, let g in $\mathscr{F}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$ be a probability density function, $N \in \mathbb{N}$, f in $\mathscr{A}\Big(\mathbb{R} \to \mathbb{R}\Big)$. Consider a Dirac mixture distribution*

$$\widetilde{D}_N = \frac{1}{N} \sum_{z \in \Xi_N} \delta_z, \tag{3.17}$$

defined by points $\Xi_N = \{\mathrm{Z}_1, \dots, \mathrm{Z}_N\}$ independently distributed according to g.

*$< \widetilde{D}_N, f >$ can be seen as a Monte Carlo approximation to $\mathbb{E}[f(G)]$, where G is the random variable distributed according to g. The law of large numbers ensures convergence of $< \widetilde{D}_N, f >$ towards $\mathbb{E}[f(G)] =< g, f >$ at rate (*cf. *[Billingsley, 2013])*

$$\mathbb{E}\left[\Big| < \widetilde{D}_N, f > - < g, f > \Big|\right] = \mathscr{O}\left(\frac{1}{\sqrt{N}}\right). \tag{3.18}$$

*The quasi-Monte Carlo methods use the same approximation to $< g, f >$, but the elements of Ξ_N are obtained from deterministic low-discrepancy sequences. In this case, $< \widetilde{D}_N, f >$ converges towards $< g, f >$ at rate (*cf. *[Caflisch, 1998])*

$$\Big| < \widetilde{D}_N, f > - < g, f > \Big| = \mathscr{O}\left(\frac{\log N}{N}\right). \tag{3.19}$$

By comparing (3.16), (3.18) *and* (3.19), *we see that in the case of one-dimensional integration, the Gaussian quadrature method has the best convergence rate compared to the Monte-Carlo and quasi Monte-Carlo methods.*

The following subsections result from an original research of the author. They focus on the construction of grids on $\mathbb{R}^a$ for $a > 1$.

3.1.2 Construction of arbitrarily precise sequences of $\mathbb{R}^a$-grids by tensor product

Definition 22. *Tensor product grid.*

Let $\Gamma, \Gamma' \in \mathscr{D}$, $\mathscr{I} = \{\Lambda, \pi\}$ and $\mathscr{I}' = \{\Lambda', \pi'\}$ be a Γ-grid and Γ'-grid respectively. We define the tensor product grid $\mathscr{J}$ on $\Gamma \times \Gamma'$ as

$$\mathscr{J} = \left\{\Lambda \times \Lambda', \pi \otimes \pi'\right\},$$

and we note it as

$$\mathscr{J} = \mathscr{I} \otimes \mathscr{I}'.$$

Definition 23. Γ*-tensor-product grid on* Γ^N.

Let $N \in \mathbb{N}^*$, $\Gamma \in \mathscr{D}$, we say that a Γ^N-grid $\mathscr{I} = \{\Lambda^N, \pi^{(N)}\}$ is Γ-tensor-product on Γ^N if there exist $\pi_1, \pi_2, \dots \pi_N$ in $\mathscr{F}\Big(\Gamma \to \mathbb{R}\Big)$ such that

$$\forall \boldsymbol{\gamma}_{1:N} \in \Gamma^N, \pi^{(N)}\Big(\boldsymbol{\gamma}_{1:N}\Big) = \pi_1\Big(\gamma_{1:N}^{(1)}\Big)\pi_2\Big(\gamma_{1:N}^{(2)}\Big) \dots \pi_N\Big(\gamma_{1:N}^{(N)}\Big), \tag{3.20}$$

where $\boldsymbol{\gamma}_{1:N} = \begin{bmatrix}\gamma_{1:N}^{(1)} & \gamma_{1:N}^{(2)} & \dots & \gamma_{1:N}^{(N)}\end{bmatrix}$.

Proposition 11. *Let $a_1, a_2 \in \mathbb{N}$, Ω_1, Ω_2 be finite discrete sets. Define $\Gamma_1 = \Omega_1 \times \mathbb{R}^{a_1}$, $\Gamma_2 = \Omega_2 \times \mathbb{R}^{a_2}$ and let $\mu_1 \in \mathscr{U}(\Gamma_1)$, $\mu_2 \in \mathscr{U}(\Gamma_2)$.*

Let $\left(\mathscr{I}_L^{(1)}\right)_{L\in\mathbb{N}^*} = \left(\left\{\Lambda_L^{(1)}, \pi_L^{(1)}\right\}\right)_{L\in\mathbb{N}^*}$ be a sequence of Γ_1-grids strongly arbitrarily precise with respect to μ_1, $\left(\mathscr{I}_L^{(2)}\right)_{L\in\mathbb{N}^*} = \left(\left\{\Lambda_L^{(2)}, \pi_L^{(2)}\right\}\right)_{L\in\mathbb{N}^*}$ be a sequence of Γ_2-grids strongly arbitrarily precise with respect to μ_2. Then $\left(\mathscr{I}_L^{(1)} \otimes \mathscr{I}_L^{(2)}\right)_{L\in\mathbb{N}}$ is a strongly arbitrarily precise sequence of $\Gamma_1 \times \Gamma_2$-grids with respect to $\mu_1 \otimes \mu_2$.

Proof. Let us pose $\Omega = \Omega_1 \times \Omega_2$, $a = a_1 + a_2$, $\mu = \mu_1 \otimes \mu_2$ and for each L in $\mathbb{N}^*$, $\Lambda_L = \Lambda_L^{(1)} \times \Lambda_L^{(2)}$, $\pi_L = \pi_L^{(1)} \otimes \pi_L^{(2)}$, $\mathscr{I}_L = \mathscr{I}_L^{(1)} \otimes \mathscr{I}_L^{(2)}$, thus we have $\mathscr{I}_L = \{\Lambda_L, \pi_L\}$ for each L in $\mathbb{N}^*$. Let t in $\mathbb{N}$. Since $\left(\mathscr{I}_L^{(1)}\right)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to μ_1 and $\left(\mathscr{I}_L^{(2)}\right)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to μ_2, there exist $L_t^{(1)}$ and $L_t^{(2)}$ such that for all $L \in \mathbb{N}^*$ greater than or equal to $\max\left(L_t^{(1)}, L_t^{(2)}\right)$, $\left(\mathscr{I}_L^{(1)}\right)_{L\in\mathbb{N}^*}$ and $\left(\mathscr{I}_L^{(2)}\right)_{L\in\mathbb{N}^*}$ would both have a degree of precision t with respect to μ_1 and μ_2 respectively. Let us prove that for each monomial m in $\mathscr{M}_t\left(\mathbb{R}^a\right)$ and for each ω in Ω, we have

$$\forall L \in \mathbb{N}^*, L > \max\left(L_t^{(1)}, L_t^{(2)}\right) \Rightarrow \sum_{\boldsymbol{z}\in\{\mathbf{x}\in\mathbb{R}^a|(\omega,\mathbf{x})\in\Lambda\}} \mathrm{m}(\boldsymbol{z})\pi(\omega, \boldsymbol{z}) = \mu(\omega) < \mu, \mathrm{m} >, \quad (3.21)$$

which would mean that $(\mathscr{I}_L \otimes \mathscr{I}_L')_{L\in\mathbb{N}}$ is strongly arbitrarily precise with respect to $\mu_1 \otimes \mu_2$. Let L in $\mathbb{N}^*$ such that $L > \max\left(L_t^{(1)}, L_t^{(2)}\right)$, $\omega = (\omega_1, \omega_2)$ in $\Omega_1 \times \Omega_2$ and a monomial m in $\mathscr{M}_t\left(\mathbb{R}^a\right)$. By definition, there exists $(\alpha_1, \ldots, \alpha_{a_1+a_2})$ in $\left\{\mathbb{N}^a | \alpha_1 + \ldots + \alpha_a \leq t\right\}$ such that

$$\forall \mathbf{z} \in \mathbb{R}^{a_1+a_2}, \mathrm{m}(\mathbf{z}) = \mathbf{z}[1]^{\alpha_1} \ldots \mathbf{z}[a]^{\alpha_a}. \quad (3.22)$$

The above equation can be rewritten as

$$\forall \mathbf{z} \in \mathbb{R}^{a_1+a_2}, \mathrm{m}(\mathbf{z}) = \mathbf{z}[1]^{\beta_1} \ldots \mathbf{z}[a_1]^{\beta_{a_1}} \mathbf{z}[a_1+1]^{\gamma_1} \ldots \mathbf{z}[a_1+a_2]^{\gamma_{a_2}} \quad (3.23)$$

with $(\beta_1, \ldots, \beta_{a_1})$ in $\left\{\mathbb{N}^{a_1} | \beta_1 + \ldots + \beta_{a_1} \leq t\right\}$ and $(\gamma_1, \ldots, \gamma_{a_a})$ in $\left\{\mathbb{N}^{a_2} | \gamma_1 + \ldots + \gamma_{a_2} \leq t\right\}$. Therefore, we have

$$\mathrm{m} = \mathrm{m}_1 \otimes \mathrm{m}_2, \quad (3.24)$$

with monomials m_1 and m_2 in $\mathscr{M}_t\left(\mathbb{R}^{a_1}\right)$ and $\mathscr{M}_t\left(\mathbb{R}^{a_2}\right)$ defined by

$$\forall \mathbf{z} \in \mathbb{R}^{a_1}, \mathrm{m}_1(\mathbf{z}) = \mathbf{z}[1]^{\alpha_1} \ldots \mathbf{z}[a_1]^{\alpha_{a_1}}; \quad (3.25a)$$

$$\forall \mathbf{z} \in \mathbb{R}^{a_2}, \mathrm{m}_2(\mathbf{z}) = \mathbf{z}[1]^{\alpha_{a_1+1}} \ldots \mathbf{z}[a_2]^{\alpha_{a_1+a_2}}. \quad (3.25b)$$

Next, we have

$$\begin{aligned}
&\sum_{\boldsymbol{z}\in\{\mathbf{x}\in\mathbb{R}^a|(\omega,\mathbf{x})\in\Lambda_L\}} \mathrm{m}(\boldsymbol{z})\pi(\omega, \boldsymbol{z}) = \\
&\sum_{\substack{\boldsymbol{z}_1\in\left\{\mathbf{x}\in\mathbb{R}^{a_1}|(\omega_1,\mathbf{x})\in\Lambda_L^{(1)}\right\} \\ \boldsymbol{z}_2\in\left\{\mathbf{x}\in\mathbb{R}^{a_2}|(\omega_2,\mathbf{x})\in\Lambda_L^{(2)}\right\}}} \mathrm{m}_1(\boldsymbol{z}_1)\mathrm{m}_2(\boldsymbol{z}_2)\pi_L^{(1)}(\boldsymbol{z}_1, \omega_1)\pi_L^{(2)}(\boldsymbol{z}_2, \omega_2) = \\
&\sum_{\boldsymbol{z}_1\in\left\{\mathbf{x}\in\mathbb{R}^{a_1}|(\omega_1,\mathbf{x})\in\Lambda_L^{(1)}\right\}} \mathrm{m}_1(\boldsymbol{z}_1)\pi_L^{(1)}(\omega_1, \boldsymbol{z}_1) \sum_{\boldsymbol{z}_2\in\left\{\mathbf{x}\in\mathbb{R}^{a_2}|(\omega_2,\mathbf{x})\in\Lambda_L^{(2)}\right\}} \mathrm{m}_2(\boldsymbol{z}_2)\pi_L^{(2)}(\omega_2, \boldsymbol{z}_2) = \\
&\mu_1(\omega_1)\mu_2(\omega_2) < \mu_1, \mathrm{m}_1 >< \mu_2, \mathrm{m}_2 >= (\mu_1 \otimes \mu_2)(\omega) < \mu_1 \otimes \mu_2, \mathrm{m}_1 \otimes \mathrm{m}_2 >= \\
&\mu(\omega) < \mu, \mathrm{m} >, \qquad (3.26)
\end{aligned}$$

which proves (3.21). □

Corollary 11.1. *Let* $\Gamma_1, \Gamma_2 \in \mathscr{D}$*,* $\mu \in \mathscr{U}(\Gamma_1)$*,* $\nu \in \mathscr{U}(\Gamma_2)$*,* $\Big(\mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ *and* $\Big(\mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ *be two sequences of* Γ_1*-grids and* Γ_2*-grids arbitrarily precise with respect to* μ *and* ν *respectively. Then* $\Big(\mathscr{I}_L \otimes \mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ *is arbitrarily precise with respect to with* $\mu \otimes \nu$*.*

Proof. $\Big(\mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ and $\Big(\mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ are arbitrarily precise with respect to μ and ν respectively, thus there exist F_1, F_2 in $\mathbb{N}^*$, $u_1, \ldots, u_{F_1}$ in $\mathscr{A}\Big(\Gamma_1 \to \mathbb{R}\Big)$, $k_1, \ldots, k_{F_2}$ in $\mathscr{A}\Big(\Gamma_2 \to \mathbb{R}\Big)$, $\mu_1, \ldots, \mu_{F_1}$ in $\mathscr{U}(\Gamma_1)$, $\nu_1, \ldots, \nu_{F_2}$ in $\mathscr{U}(\Gamma_2)$, F_1 sequences of Γ_1-grids $\Big(\mathscr{I}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{I}_L^{(F_1)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\mu_1, \ldots, \mu_{F_1}$ respectively and F_2 sequences of Γ_2-grids $\Big(\mathscr{K}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{K}_L^{(F_2)}\Big)_{L\in\mathbb{N}^*}$ strongly arbitrarily precise with respect to $\nu_1, \ldots, \nu_{F_2}$ respectively such that $\mu = \sum_{i=1}^{F_1} \mu_i u_i$, $\nu = \sum_{j=1}^{F_2} \nu_j k_j$ and for each L in $\mathbb{N}^*$, $\mathscr{I}_L = \sum_{i=1}^{F_1} \mathscr{I}_L^{(i)} u_i$, $\mathscr{K}_L = \sum_{j=1}^{F_2} \mathscr{K}_L^{(j)} k_j$. For each L in $\mathbb{N}^*$, we have

$$\mathscr{I}_L \otimes \mathscr{K}_L = \left(\sum_{i=1}^{F_1} \mathscr{I}_L^{(i)} u_f\right) \otimes \left(\sum_{j=1}^{F_2} \mathscr{K}_L^{(j)} k_f\right) = \sum_{\substack{1\le i\le F_1\\ 1\le j\le F_2}} \mathscr{I}_L^{(i)} \otimes \mathscr{K}_L^{(j)} (u_i \otimes k_j).$$

For each (i, j) in $\{1 : F_1\}\times\{1 : F_2\}$, $u_i \otimes k_j$ is in $\mathscr{A}\Big(\Gamma_1\otimes\Gamma_2 \to \mathbb{R}\Big)$, since $u_i \in \mathscr{A}\Big(\Gamma_1 \to \mathbb{R}\Big)$ and $k_j \in \mathscr{A}\Big(\Gamma_2 \to \mathbb{R}\Big)$, and the sequence $\Big(\mathscr{I}_L^{(i)} \otimes \mathscr{K}_L^{(j)}\Big)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to $\mu_i \otimes \nu_j$ according to Proposition 11. Since

$$\mu \otimes \nu = \left(\sum_{i=1}^{F_1} \mu_i u_i\right) \otimes \left(\sum_{j=1}^{F_2} \nu_j k_j\right) = \sum_{\substack{1\le i\le F_1\\ 1\le j\le F_2}} \mu_i \otimes \nu_j (u_i \otimes k_j),$$

$\Big(\mathscr{I}_L \otimes \mathscr{K}_L\Big)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to $\mu \otimes \nu$. □

The above result allows constructing arbitrarily precise grid sequences with respect to product measures on $\mathbb{R}^a$. Arbitrarily precise grid sequences with respect to g in $\mathscr{A}\Big(\mathbb{R}^a \to \mathbb{R}\Big)$ can be obtained as follows.

Proposition 12. *Let* $a \in \mathbb{N}$*,* g *in* $\mathscr{A}\Big(\mathbb{R}^a \to \mathbb{R}\Big)$*,* $g_1, \ldots, g_a, h$ *in* $\mathscr{A}\Big(\mathbb{R} \to \mathbb{R}_+\Big)$*, such that for each* i *in* $\{1 : a\}$*,* j *in* $\mathbb{N}$*,* $m_j[g_i] < \infty$ *and*

$$g = h \cdot g_1 \otimes g_2 \otimes \ldots \otimes g_a. \tag{3.27}$$

Let $\Big(\mathscr{G}_L^{(1)}\Big)_{L\in\mathbb{N}^*}, \ldots, \Big(\mathscr{G}_L^{(a)}\Big)_{L\in\mathbb{N}^*}$ *be sequences of* $\mathbb{R}$*-grid corresponding to L-point Gaussian quadrature rules with respect to* $g_1, \ldots, g_a$ *respectively, then the sequence of* $\mathbb{R}^a$*-grids* $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ *defined by*

$$\forall L \in \mathbb{N}^*, \mathscr{I}_L = h \cdot \mathscr{G}_L^{(1)} \otimes \ldots \otimes \mathscr{G}_L^{(a)} \tag{3.28}$$

is arbitrarily precise with respect to g

Proof. It follows from Corollary 11.1 and Proposition 7. □

However, the number of grid points in a product of the same a grids grows exponentially with a. For instance, if $\mathscr{G}$ contains M distinct points, $\bigotimes_{i=1}^{a}\mathscr{G}$ would contain M^a distinct points. Thus, a direct evaluation of (3.6) would be problematic or impossible even for moderate values of a. This is why we recall the sparse grids which are better suited for high-dimensional integration.

3.1.3 Construction of arbitrarily precise sequences of $\mathbb{R}^a$-grids by Smolyak formula

Here we consider the Smolyak grids which are special case of sparse grids.

Definition 24. *Smolyak grid product.*

Let $a \in \mathbb{N}^*$, $(\mathscr{I}_l)_{1\le l\le L}$ grids on $\mathbb{R}$. The Smolyak product grid of $(\mathscr{I}_l)_{1\le l\le L}$ on $\mathbb{R}^a$ is defined as

$$\mathscr{S}_a\Big[(\mathscr{I}_l)_{1\le l\le L}\Big] = \sum_{q=L-a}^{L-1}(-1)^{L-1-q}\mathrm{C}_{a-1}^{L-1-a}\sum_{\substack{l_1,\ldots,l_a\in\mathbb{N}^*\\ l_1+\ldots+l_a\le a+q}}\mathscr{I}_{l_1}\otimes\ldots\otimes\mathscr{I}_{l_a}.$$

Proposition 13. *Let a in $\mathbb{N}^*$, $g_1,\ldots,g_a$ in $\mathscr{F}\Big(\mathbb{R}\to\mathbb{R}_+\Big)$ and $\Big(\mathscr{G}_L^{(1)}\Big)_{L\in\mathbb{N}^*},\ldots,\Big(\mathscr{G}_L^{(a)}\Big)_{L\in\mathbb{N}^*}$ be sequences of strongly arbitrarily precise $\mathbb{R}$-grids with respect to $g_1,\ldots,g_a$ respectively. Then $\Big(\mathscr{S}_a\Big[(\mathscr{G}_l)_{1\le l\le L}\Big]\Big)_{L\in\mathbb{N}^*}$ is strongly arbitrarily precise with respect to $g_1\otimes\ldots\otimes g_a$.*

Proof. See [Garcke and Griebel, 2013]. □

A sparse grid sequence arbitrarily precise with respect to g in $\mathscr{A}\Big(\mathbb{R}^a\to\mathbb{R}\Big)$ can be obtained as follows. Let $a\in\mathbb{N}^*$, g in $\mathscr{A}\Big(\mathbb{R}^a\to\mathbb{R}\Big)$, $g_1,\ldots,g_a,h$ in $\mathscr{A}\Big(\mathbb{R}\to\mathbb{R}_+\Big)$ satisfying (3.27) and such that for each i in $\{1:d\}$, j in $\mathbb{N}$, $m_j[g_i]<\infty$.

Let $\Big(\mathscr{G}_M^{(1)}\Big)_{M\in\mathbb{N}^*},\ldots,\Big(\mathscr{G}_M^{(a)}\Big)_{M\in\mathbb{N}^*}$ be sequences of $\mathbb{R}$-grids corresponding to the M-point Gaussian quadrature rules with respect to $g_1,\ldots,g_a$ respectively, then the sequence of $\mathbb{R}^a$-grids $(\mathscr{I}_M)_{M\in\mathbb{N}^*}$ defined by

$$\forall L\in\mathbb{N}^*, \mathscr{I}_M = q\cdot\mathscr{S}_a\Big[(\mathscr{G}_m)_{1\le m\le M}\Big] \tag{3.29}$$

is arbitrarily precise with respect to g according to Corollary 11.1 and Proposition 7.

Remark 3. *Let $M\in\mathbb{N}^*$, g in $\mathscr{F}\Big(\mathbb{R}\to\mathbb{R}_+\Big)$ and let for each $m\in\mathbb{N}^*, m\le M$, $\mathscr{G}_m$ the $\mathbb{R}$-grid corresponding to the M-point Gaussian quadrature with respect to g. The total number of points in $\mathscr{S}_a\Big[(\mathscr{G}_m)_{1\le m\le M}\Big]$ grows as $\mathscr{O}\left(a^M\right)$ with M and a* cf. *[Bungartz and Griebel, 2004]. Asymptotically (in M), the Smolyak grid method appears as less efficient compared to the product grid method, since one has M^a grid points in the grid product of $(\mathscr{G}_m)_{1\le m\le M}$. However, in practice, when the number of function evaluations in* (3.6) *is constrained, the sparse grids may allow achieving the same accuracy as a product grid but less points. This property is particularly important for high values of a and cases where $M\le a$. Besides, even if the asymptotic rate of convergence of the quasi-Monte Carlo method* (3.19) *is promising, the variance of its estimate may still be too high if the number of function evaluations is constrained. This is why the three methods should be taken in consideration for practical applications.*

3.1.4 Construction of arbitrarily precise sequences of $\Omega \times \mathbb{R}^a$-grids

Let Ω be a finite discrete set and a in $\mathbb{N}$, $\Gamma = \Omega \times \mathbb{R}^a$, $\mu \in \mathscr{U}(\Gamma)$. Here we consider constructing a sequence of Γ-grids arbitrarily precise with respect to μ in the case where Ω is a non-empty set, provided that constructions of sequences of $\mathbb{R}^a$-grids arbitrarily precise with respect to a measure in $\mathscr{U}(\mathbb{R}^a)$ have been exposed previously.

Let us denote $\Omega = \{\omega_1, \dots, \omega_K\}$ the elements of Ω, where $K = \mathrm{Card}(\Omega) > 0$. Define, for each i in $\{1 : K\}$, $\mu^{(1,i)} \in \mathscr{U}(\Omega)$ and $\mu^{(2,i)} \in \mathscr{U}(\mathbb{R}^a)$ as follows:

$$\forall \omega \in \Omega, \mu^{(1,i)}(\omega) = \begin{cases} 1 & \text{if } \omega = \omega_i; \\ 0 & \text{otherwise,} \end{cases} \qquad \forall \boldsymbol{z} \in \mathbb{R}^a, \mu^{(2,i)}(\boldsymbol{z}) = \mu(\omega_i, \boldsymbol{z}). \tag{3.30}$$

In this way, we have

$$\mu = \sum_{1 \le i \le K} \mu^{(1,i)} \otimes \mu^{(2,i)}. \tag{3.31}$$

Proposition 14. *Define, for each i in $\{1 : K\}$, Ω-grid $\mathscr{I}^{(1,i)} = \{\Omega, \mu^{(1,i)}\}$, thus $\mathscr{I}^{(1,i)}$ has an infinite degree of precision with respect to $\mu^{(1,i)}$. Consider, for each i in $\{1 : K\}$, a sequence of $\mathbb{R}^a$-grids $\left(\mathscr{I}_L^{(2,i)}\right)_{L \in \mathbb{N}^*}$ arbitrarily precise with respect to $\mu^{(2,i)}$. Thus, for each i in $\{1 : K\}$, $\left(\mathscr{I}^{(1,i)} \otimes \mathscr{I}_L^{(2,i)}\right)_{L \in \mathbb{N}^*}$ is arbitrarily precise with respect to $\mu^{(1,i)} \otimes \mu^{(2,i)}$ according to Corollary 11.1. Define, for each L in $\mathbb{N}^*$,*

$$\mathscr{I}_L = \sum_{1 \le i \le K} \mathscr{I}^{(1,i)} \otimes \mathscr{I}_L^{(2,i)}, \tag{3.32}$$

then $\left(\mathscr{I}_L\right)_{L \in \mathbb{N}^*}$ is arbitrarily precise with respect to μ.

Proof. The sum in the above equation is finite, it follows from Proposition 8 that $\left(\mathscr{I}_L\right)_{L \in \mathbb{N}^*}$ is arbitrarily precise with respect to μ due to (3.31). □

3.2 Markovian grid-based state estimators

This section presents the main contribution of the author in the context of the chapter. The following content results from an original research.

3.2.1 Markovian grids

Definition 25. Γ-*Markovian grid on* Γ^N.

Let $N \in \mathbb{N}^*$, $\Gamma \in \mathscr{D}$, we say that a Γ^N-grid $\mathscr{I} = \{\Lambda^N, \pi^{(N)}\}$ is Γ-Markovian on Γ^N if there exist $q_1, q_2, \dots q_{N-1}$ in $\mathscr{F}\left(\Gamma^2 \to \mathbb{R}\right)$ such that

$$\forall \boldsymbol{\gamma}_{1:N} \in \Gamma^N, \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) = q_1\left(\gamma_{1:N}^{(1)}, \gamma_{1:N}^{(2)}\right) q_2\left(\gamma_{1:N}^{(2)}, \gamma_{1:N}^{(3)}\right) \dots q_{N-1}\left(\gamma_{1:N}^{(N-1)}, \gamma_{1:N}^{(N)}\right), \tag{3.33}$$

where $\boldsymbol{\gamma}_{1:N} = \begin{bmatrix} \gamma_{1:N}^{(1)} & \gamma_{1:N}^{(2)} & \dots & \gamma_{1:N}^{(N)} \end{bmatrix}$.

Proposition 15. *Let $N \in \mathbb{N}^*$, $\Lambda = \{\gamma_1, \dots, \gamma_K\}$ be a non-empty finite discrete set of cardinal S, $\mathscr{I}^{(N)} = \{\Lambda^N, \pi^{(N)}\}$ be a Γ-Markovian grid on Γ^N and $h_1, \dots, h_n$ in $\mathscr{F}\left(\Gamma^2 \to \mathbb{R}\right)$. For each n in $\{1 : N\}$ and $\gamma \in \Lambda$, define*

$$\phi_n(\gamma) = \sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda^N, \gamma_{1:N}^{(n)} = \gamma} \pi^{(N)}(\boldsymbol{\gamma}_{1:N}). \tag{3.34}$$

Then for each n in $\{1 : N\}$, $\phi_n(\gamma)$ can be evaluated with a complexity $\mathscr{O}\left(NS^2\right)$ at each γ in Λ.

Proof. $\mathscr{J}^{(N)}$ is Γ-Markovian on Γ^N, thus there exist $q_1, q_2, \dots q_{N-1}$ in $\mathscr{F}\Big(\Gamma^2 \to \mathbb{R}\Big)$ such that

$$\forall \boldsymbol{\gamma}_{1:N} \in \Gamma^N, \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) = q_1\Big(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\Big) q_2\Big(\boldsymbol{\gamma}_{1:N}^{(2)}, \boldsymbol{\gamma}_{1:N}^{(3)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big)$$

and (3.34) becomes

$$\begin{aligned}
&\forall n \in \{1 : N\}, \boldsymbol{\gamma} \in \Lambda, \phi_n(\boldsymbol{\gamma}) = \\
&\quad \sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda^N, \boldsymbol{\gamma}_{1:N}^{(n)} = \boldsymbol{\gamma}} q_1\Big(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\Big) q_2\Big(\boldsymbol{\gamma}_{1:N}^{(2)}, \boldsymbol{\gamma}_{1:N}^{(3)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big) = \\
&\sum_{\boldsymbol{\gamma}_{1:n-1} \in \Lambda^{n-1}} q_1\Big(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\Big) q_2\Big(\boldsymbol{\gamma}_{1:N}^{(2)}, \boldsymbol{\gamma}_{1:N}^{(3)}\Big) \dots q_{n-2}\Big(\boldsymbol{\gamma}_{1:N}^{(n-2)}, \boldsymbol{\gamma}_{1:N}^{(n-1)}\Big) q_{n-1}\Big(\boldsymbol{\gamma}_{1:N}^{(n-1)}, \boldsymbol{\gamma}\Big) \times \\
&\sum_{\boldsymbol{\gamma}_{n+1:N} \in \Lambda^{N-n}} q_n\Big(\boldsymbol{\gamma}, \boldsymbol{\gamma}_{1:N}^{(n+1)}\Big) q_{n+1}\Big(\boldsymbol{\gamma}_{1:N}^{(n+1)}, \boldsymbol{\gamma}_{1:N}^{(n+2)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big). \qquad (3.35)
\end{aligned}$$

Define, for each n in $\{1 : N\}$ and $\boldsymbol{\gamma}$ in Λ,

$$\begin{aligned}
&\alpha_n(\boldsymbol{\gamma}) = \qquad (3.36a) \\
&\quad \sum_{\boldsymbol{\gamma}_{1:n-1} \in \Lambda^{n-1}} q_1\Big(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\Big) q_2\Big(\boldsymbol{\gamma}_{1:N}^{(2)}, \boldsymbol{\gamma}_{1:N}^{(3)}\Big) \dots q_{n-2}\Big(\boldsymbol{\gamma}_{1:N}^{(n-2)}, \boldsymbol{\gamma}_{1:N}^{(n-1)}\Big) q_{n-1}\Big(\boldsymbol{\gamma}_{1:N}^{(n-1)}, \boldsymbol{\gamma}\Big); \\
&\qquad (3.36b) \\
&\beta_n(\boldsymbol{\gamma}) = \sum_{\boldsymbol{\gamma}_{n+1:N} \in \Lambda^{N-n}} q_n\Big(\boldsymbol{\gamma}, \boldsymbol{\gamma}_{1:N}^{(n+1)}\Big) q_{n+1}\Big(\boldsymbol{\gamma}_{1:N}^{(n+1)}, \boldsymbol{\gamma}_{1:N}^{(n+2)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big). \quad (3.36c)
\end{aligned}$$

We have from (3.35)

$$\forall \boldsymbol{\gamma} \in \Lambda, \forall n \in \{1 : N\}, \phi_n(\boldsymbol{\gamma}) = \alpha_n(\boldsymbol{\gamma})\beta_n(\boldsymbol{\gamma}), \qquad (3.37)$$

and by evaluating (3.35) at $n = 1$ and $n = N$,

$$\forall \boldsymbol{\gamma} \in \Lambda, \alpha_1(\boldsymbol{\gamma}) = \beta_N(\boldsymbol{\gamma}) = 1. \qquad (3.38)$$

Observe that for each n in $\{1 : N-1\}$ and $\boldsymbol{\gamma}$ in Λ,

$$\begin{aligned}
&\alpha_{n+1}(\boldsymbol{\gamma}) = \\
&\sum_{\boldsymbol{\gamma}' \in \Lambda} q_n(\boldsymbol{\gamma}', \boldsymbol{\gamma}) \sum_{\boldsymbol{\gamma}_{1:n-1} \in \Lambda^{n-1}} q_1\Big(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\Big) \dots q_{n-2}\Big(\boldsymbol{\gamma}_{1:N}^{(n-2)}, \boldsymbol{\gamma}_{1:N}^{(n-1)}\Big) q_{n-1}\Big(\boldsymbol{\gamma}_{1:N}^{(n-1)}, \boldsymbol{\gamma}'\Big) = \\
&\sum_{\boldsymbol{\gamma}' \in \Lambda} q_n(\boldsymbol{\gamma}', \boldsymbol{\gamma}) \alpha_n(\boldsymbol{\gamma}'), \qquad (3.39)
\end{aligned}$$

which is a recursive equation. Thus, evaluation of $\alpha_n(\boldsymbol{\gamma})$ at $n \in \{1 : N\}$ for each $\boldsymbol{\gamma}$ in Λ requires nS summations over Λ which results in a complexity $\mathscr{O}\left(nS^2\right)$. Let us also show for each $\boldsymbol{\gamma}$ in Λ, evaluation of $\beta_n(\boldsymbol{\gamma})$ can be achieved with a complexity $\mathscr{O}\left(S^2(N-n)\right)$. For each $\boldsymbol{\gamma} \in \Lambda$,

$$\begin{aligned}
&\forall n \in \{1 : N-1\}, \forall \boldsymbol{\gamma} \in \Lambda, \beta_n(\boldsymbol{\gamma}) = \\
&\quad \sum_{\boldsymbol{\gamma}_{n+1:N} \in \Lambda^{N-n}} q_n\Big(\boldsymbol{\gamma}, \boldsymbol{\gamma}_{1:N}^{(n+1)}\Big) q_{n+1}\Big(\boldsymbol{\gamma}_{1:N}^{(n+1)}, \boldsymbol{\gamma}_{1:N}^{(n+2)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big) = \\
&\sum_{\boldsymbol{\gamma}' \in \Lambda} q_n(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \sum_{\boldsymbol{\gamma}_{n+2:N} \in \Lambda^{N-n-1}} q_{n+1}\Big(\boldsymbol{\gamma}', \boldsymbol{\gamma}_{1:N}^{(n+2)}\Big) \dots q_{N-1}\Big(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\Big) = \\
&\quad \sum_{\boldsymbol{\gamma}' \in \Lambda} q_n(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \beta_{n+1}(\boldsymbol{\gamma}'), \qquad (3.40)
\end{aligned}$$

which is a recursive equation similar to (3.39). Evaluation of $\beta_n(\boldsymbol{\gamma})$ at $n \in \{1:N\}$ for each $\boldsymbol{\gamma}$ in Λ requires $(N-n)S$ summations over Λ which results in a complexity $\mathscr{O}\left((N-n)S^2\right)$. Finally, evaluating $\alpha_n(\boldsymbol{\gamma})$ and $\beta_n(\boldsymbol{\gamma})$ at each $n \in \{1:N\}$ for each $\boldsymbol{\gamma}$ in Λ can be achieved with a complexity $\mathscr{O}\left(NS^2\right)$. As a result, evaluating $\phi_n(\boldsymbol{\gamma})$ at each $n \in \{1:N\}$ for each $\boldsymbol{\gamma}$ in Λ can be achieved with a complexity $\mathscr{O}\left(NS^2\right)$ due to (3.37). □

Remark 4. *Despite the fact that an evaluation of* (3.34) *would a priori require M^{N-1} operations, we show that an evaluation of* (3.34) *in a Markovian grid can be achieved with a complexity linear in N, which is the key point of Proposition 15. The way we evaluate* (3.34) *in the proof is similar to the well-known forward-backward algorithm. Indeed, we can see that functions $q_1, q_2, \dots q_{N-1}$ in $\mathscr{F}\Big(\Gamma^2 \to \mathbb{R}\Big)$ are not necessarily positive-valued, as it is the case in the classic version of the forward-backward algorithm.*

3.2.2 Application of Markovian grids to the Bayesian state estimation problem in POMPs

Now we consider a partially observed Markov process $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$. Let a in $\mathbb{N}^*$, N in $\mathbb{N}^*$, $\mathbf{H}_{1:N}$ be a hidden time series in $\Gamma = \mathbb{R}^a \times \Omega$ with Ω a finite-discrete set and $\mathbf{Y}_{1:N}$ observed.

Proposition 16. *Let $d, N \in \mathbb{N}^*$, $\Gamma \in \mathscr{D}$, partially observed Markov process $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ where for each n in $\{1:N\}$, $(\mathbf{H}_n, \mathbf{Y}_n) \in \Gamma \times \mathbb{R}^d$.*

Let $\mathbf{y}_{1:N} \in \mathbb{R}^{dN}$ such that $p_{\mathbf{Y}_{1:N}}(\mathbf{y}_{1:N}) \neq 0$, define $\mu_{\mathbf{y}_{1:N}} \in \mathscr{U}(\Gamma^N)$ by

$$\forall \mathbf{h}_{1:N} \in \Gamma^N, \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N}) = p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right). \tag{3.41}$$

Let $\Big(\mathscr{I}_L\Big)_{L\in\mathbb{N}^*} = \Big\{\Lambda_L^N, \pi_L^{(N)}\Big\}_{L\in\mathbb{N}^*}$ be a sequence of Γ-Markovian grids on Γ^N consistent with $\mu_{\mathbf{y}_{1:N}}$. For each n in $\{1:N\}$ and L in $\mathbb{N}^*$, define

$$\forall \boldsymbol{\gamma} \in \Lambda_L, \phi_{L,n}(\boldsymbol{\gamma}) = \sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda_L^N, \boldsymbol{\gamma}_{1:N}^{(n)} = \boldsymbol{\gamma}} \pi^{(N)}(\boldsymbol{\gamma}_{1:N}) \tag{3.42}$$

and a Γ-grid $\mathscr{P}_{L,n} = \{\Lambda_L, \upsilon_{L,n}\}$ by

$$\forall \boldsymbol{\gamma} \in \Lambda_L, \upsilon_{L,n}(\boldsymbol{\gamma}) = \frac{\phi_{L,n}(\boldsymbol{\gamma})}{\sum\limits_{\boldsymbol{\gamma}' \in \Lambda_L} \phi_{L,n}(\boldsymbol{\gamma}')}, \tag{3.43}$$

then $\Big(\mathscr{P}_{L,n}\Big)_{L\in\mathbb{N}^*}$ is consistent with $p_{\mathbf{H}_n|\mathbf{y}_{1:N}}$.

Proof. Let n in $\{1:N\}$, $\boldsymbol{f}$ in $\mathscr{A}\Big(\Gamma\Big)$, we have

$$< p_{\mathbf{H}_n|\mathbf{y}_{1:N}}, \boldsymbol{f} >= \int \boldsymbol{f}(\mathbf{h}_n) p\left(\mathbf{h}_n \left| \mathbf{y}_{1:N}\right.\right) \mathrm{d}\mathbf{h}_{1:N} = \frac{\int \boldsymbol{f}(\mathbf{h}_n) p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \mathrm{d}\mathbf{h}_{1:N}}{p_{\mathbf{Y}_{1:N}}(\mathbf{y}_{1:N})} = \frac{\int \boldsymbol{f}(\mathbf{h}_n) \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N}) \mathrm{d}\mathbf{h}_{1:N}}{\int \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N}) \mathrm{d}\mathbf{h}_{1:N}}.$$

Since $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is consistent with $\mu_{\mathbf{y}_{1:N}}$, we have

$$\lim_{L\to\infty} \sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda_L^N} \boldsymbol{f}\Big(\boldsymbol{\gamma}_{1:N}^{(n)}\Big) \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N}) = \int \boldsymbol{f}(\mathbf{h}_n) \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N}) \mathrm{d}\mathbf{h}_{1:N};$$

$$\lim_{L\to\infty} \sum_{\boldsymbol{\gamma}_{1:N} \in \Lambda_L^N} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N}) = \int \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N}) \mathrm{d}\mathbf{h}_{1:N},$$

and therefore

$$\lim_{L\to\infty} \frac{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \boldsymbol{f}\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right)\pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})}{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})} = \frac{\lim\limits_{L\to\infty} \sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \boldsymbol{f}\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right)\pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})}{\lim\limits_{L\to\infty} \sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})} =$$

$$\frac{\int \boldsymbol{f}(\mathbf{h}_n)\mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N})\mathrm{d}\mathbf{h}_{1:N}}{\int \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N})\mathrm{d}\mathbf{h}_{1:N}} =< p_{\mathbf{H}_n|\mathbf{y}_{1:N}}, \boldsymbol{f} >, \tag{3.44}$$

since $p_{\mathbf{Y}_{1:N}}(\mathbf{y}_{1:N}) = \int \mu_{\mathbf{y}_{1:N}}(\mathbf{h}_{1:N})\mathrm{d}\mathbf{h}_{1:N}$ and we suppose that $p_{\mathbf{Y}_{1:N}}(\mathbf{y}_{1:N}) \neq 0$. Let us now show that for all L in $\mathbb{N}^*$,

$$\frac{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda^N} \boldsymbol{f}\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right)\pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})}{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})} =< \mathscr{P}_{L,n}, \boldsymbol{f} > . \tag{3.45}$$

We have

$$\forall L \in \mathbb{N}^*, \frac{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \boldsymbol{f}\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right)\pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})}{\sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})} = \frac{\sum\limits_{\boldsymbol{\gamma}\in\Lambda_L} \boldsymbol{f}(\boldsymbol{\gamma}) \sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N, \boldsymbol{\gamma}_{1:N}^{(n)}=\boldsymbol{\gamma}} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})}{\sum\limits_{\boldsymbol{\gamma}'\in\Lambda_L} \sum\limits_{\boldsymbol{\gamma}_{1:N}\in\Lambda_L^N, \boldsymbol{\gamma}_{1:N}^{(n)}=\boldsymbol{\gamma}'} \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})} =$$

$$\frac{\sum\limits_{\boldsymbol{\gamma}\in\Lambda_L} \boldsymbol{f}(\boldsymbol{\gamma})\phi_{L,n}(\boldsymbol{\gamma})}{\sum\limits_{\boldsymbol{\gamma}'\in\Lambda_L} \phi_{L,n}(\boldsymbol{\gamma}')} =< \mathscr{P}_{L,n}, \boldsymbol{f} > . \tag{3.46}$$

We conclude that

$$\lim_{L\to\infty} < \mathscr{P}_{L,n}, \boldsymbol{f} >=< p_{\mathbf{H}_n|\mathbf{y}_{1:N}}, \boldsymbol{f} > \tag{3.47}$$

by substituting (3.46) in (3.44). □

Remark 5. *For each $n \in \mathbb{N}^*, n \leq N$, one can compute the quadrature weights* (3.43) *with complexity $\mathscr{O}\left(NS^2\right)$ according to Proposition* (15).

Proposition 17. *Let $d, N \in \mathbb{N}^*$, $\Gamma \in \mathscr{D}$, $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$ be a partially observed Markov process where for each n in $\{1 : N\}$, $(\mathbf{H}_n, \mathbf{Y}_n) \in \Gamma \times \mathbb{R}^d$. Let $\mathbf{y}_{1:N} \in \mathbb{R}^{dN}$ such that $p_{\mathbf{Y}_{1:N}}(\mathbf{y}_{1:N}) \neq 0$, $(\mathscr{I}_L)_{L\in\mathbb{N}^*} = (\{\Lambda_L, \pi_L\})_{L\in\mathbb{N}^*}$ be a sequence of Γ-grids arbitrarily precise with respect to $p_{\mathbf{H}_1} \in \mathscr{U}(\Gamma)$. Define $u_1, u_2, \dots u_{N-1}$ in $\mathscr{F}\left(\Lambda_L^2 \to \mathbb{R}\right)$ by*

$$\forall(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \in \Lambda_L^2, u_1(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = p_{(\mathbf{H}_1,\mathbf{H}_2,\mathbf{Y}_1,\mathbf{Y}_2)}\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}', \mathbf{y}_1, \mathbf{y}_2\right);$$
$$\forall n \in \{2 : N-1\}, \forall(\boldsymbol{\gamma}, \boldsymbol{\gamma}') \in \Lambda_L^2, u_n(\boldsymbol{\gamma}, \boldsymbol{\gamma}') = p_{(\mathbf{H}_{n+1},\mathbf{Y}_{n+1})|(\mathbf{H}_n,\mathbf{Y}_n)}\left(\boldsymbol{\gamma}', \mathbf{y}_{n+1} \middle| \boldsymbol{\gamma}, \mathbf{y}_n\right).$$

For each $L \in \mathbb{N}^*$, define Γ^N-grid $\mathscr{I}_L^{(N)} = \{\Lambda^N, \pi_L^{(N)}\}$ by

$$\forall \boldsymbol{\gamma}_{1:N} \in \Lambda_L^N, \pi_L^{(N)}(\boldsymbol{\gamma}_{1:N}) = \frac{u_1\left(\boldsymbol{\gamma}_{1:N}^{(1)}, \boldsymbol{\gamma}_{1:N}^{(2)}\right) u_2\left(\boldsymbol{\gamma}_{1:N}^{(2)}, \boldsymbol{\gamma}_{1:N}^{(3)}\right) \dots u_{N-1}\left(\boldsymbol{\gamma}_{1:N}^{(N-1)}, \boldsymbol{\gamma}_{1:N}^{(N)}\right)}{\prod\limits_{n=1}^{N} p_{\mathbf{H}_1}\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right)} \prod_{n=1}^{N} \pi_L\left(\boldsymbol{\gamma}_{1:N}^{(n)}\right), \tag{3.48}$$

then:

- *For each $L \in \mathbb{N}^*$, $\mathscr{I}_L^{(N)}$ is Γ-Markovian on Γ^N;*

- *If $u_1, u_2, \dots u_{N-1}$ are in $\mathscr{A}\Big(\Gamma^2 \to \mathbb{R}\Big)$ and $p_{\mathbf{H}_1}$ is in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, then $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to $\mu_{\mathbf{y}_{1:N}} \in \mathscr{U}(\Gamma^N)$ defined by* (3.41)*.*

Proof. Let us show that $\mathscr{I}_L^{(N)}$ is Γ-Markovian on Γ^N. Define $q_1, q_2, \dots q_{N-1}$ in $\mathscr{F}\Big(\Lambda_L^2 \to \mathbb{R}\Big)$ by

$$\forall(\boldsymbol{\gamma},\boldsymbol{\gamma}') \in \Lambda_L^2, q_1(\boldsymbol{\gamma},\boldsymbol{\gamma}') = \frac{u_1(\boldsymbol{\gamma},\boldsymbol{\gamma}')\pi_L\Big(\boldsymbol{\gamma}\Big)\pi_L\Big(\boldsymbol{\gamma}'\Big)}{p_{\mathbf{H}_1}\Big(\boldsymbol{\gamma}\Big)p_{\mathbf{H}_1}\Big(\boldsymbol{\gamma}'\Big)}; \tag{3.49a}$$

$$\forall n \in \{2 : N-1\}, \forall(\boldsymbol{\gamma},\boldsymbol{\gamma}') \in \Lambda_L^2, q_n(\boldsymbol{\gamma},\boldsymbol{\gamma}') = \frac{u_n(\boldsymbol{\gamma},\boldsymbol{\gamma}')\pi_L\Big(\boldsymbol{\gamma}'\Big)}{p_{\mathbf{H}_1}\Big(\boldsymbol{\gamma}'\Big)}. \tag{3.49b}$$

Thus, $\pi_L^{(N)}(\boldsymbol{\gamma}_{1:N})$ verifies (3.33), therefore $\mathscr{I}_L^{(N)}$ is Γ-Markovian on Γ^N.

Let us now show that $\Big(\mathscr{I}_L^{(N)}\Big)_{L\in\mathbb{N}^*}$ is consistent with $\mu_{\mathbf{y}_{1:N}}$ defined by (3.41) under condition that $u_1, u_2, \dots u_{N-1}$ are in $\mathscr{A}\Big(\Gamma^2 \to \mathbb{R}\Big)$ and $p_{\mathbf{H}_1}$ is in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$. The sequence of Γ-tensor-product-grids $\Big(\bigotimes_{n=1}^N \mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ on Γ^N is arbitrarily precise with respect to the product measure $\bigotimes_{n=1}^N p_{\mathbf{H}_1}$ according to Corollary 11.1, since $(\mathscr{I}_L)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to $p_{\mathbf{H}_1}$. By applying Proposition 7 to $\Big(\bigotimes_{n=1}^N \mathscr{I}_L\Big)_{L\in\mathbb{N}^*}$ with
$h \in \mathscr{A}\Big(\Gamma^N \to \mathbb{R}\Big)$ defined by

$$\forall \mathbf{h}_{1:N} \in \Gamma^N, h(\mathbf{h}_{1:N}) = \frac{u_1\Big(\mathbf{h}_{1:N}^{(1)}, \mathbf{h}_{1:N}^{(2)}\Big)u_2\Big(\mathbf{h}_{1:N}^{(2)}, \mathbf{h}_{1:N}^{(3)}\Big)\dots u_{N-1}\Big(\mathbf{h}_{1:N}^{(N-1)}, \mathbf{h}_{1:N}^{(N)}\Big)}{\prod_{n=1}^{N} p_{\mathbf{H}_1}\Big(\mathbf{h}_{1:N}^{(n)}\Big)},$$

$\Big(\mathscr{I}_L^{(N)}\Big)_{L\in\mathbb{N}^*}$ is arbitrarily precise with respect to $\mu_{\mathbf{y}_{1:N}}$, since we have

$$\forall \mathbf{h}_{1:N} \in \Gamma^N, \prod_{n=1}^{N-1} u_n(\mathbf{h}_{1:N}) = p\left(\mathbf{h}_{1:2}, \mathbf{y}_{1:2}\right) \prod_{n=2}^{N-1} p\left(\mathbf{h}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{h}_n, \mathbf{y}_n \right.\right) = p\left(\mathbf{h}_{1:N}, \mathbf{y}_{1:N}\right) \tag{3.50}$$

by Markovianity of $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$. □

As a result, given a stationary POMP $(\mathbf{H}_{1:N}, \mathbf{Y}_{1:N})$, the Markovian-grid based method for Bayesian inference runs as follows:

- *Step 1 (preparatory)*: consider a sequence of Γ-grids $(\mathscr{I}_L)_{L\in\mathbb{N}^*} = (\{\Lambda_L, \pi_L\})_{L\in\mathbb{N}^*}$ arbitrarily precise with respect to $p_{\mathbf{H}_1}$. Such sequences can be constructed by using methods from Sections 3.1.1-3.1.4;

- *Step 2*: on receiving an observation $\mathbf{y}_{1:N}$, compute values $q_n(\boldsymbol{\gamma},\boldsymbol{\gamma}')$ for each n in $\{1 : N-1\}$ and $(\boldsymbol{\gamma},\boldsymbol{\gamma}')$ in Λ_L^2 by using (3.49);

- *Step 3*: compute $\alpha_n(\boldsymbol{\gamma}), \beta_n(\boldsymbol{\gamma})$ for each n in $\{1 : N\}$ and $\boldsymbol{\gamma} \in \Lambda_L$ by using recursive formulas

  $$\forall \boldsymbol{\gamma} \in \Lambda_L, \alpha_{n+1}(\boldsymbol{\gamma}) = \sum_{\boldsymbol{\gamma}\prime \in \Lambda_L} q_n(\boldsymbol{\gamma}\prime, \boldsymbol{\gamma})\alpha_n(\boldsymbol{\gamma}\prime), \qquad \beta_n(\boldsymbol{\gamma}) = \sum_{\boldsymbol{\gamma}\prime \in \Lambda_L} q_n(\boldsymbol{\gamma}, \boldsymbol{\gamma}\prime)\beta_{n+1}(\boldsymbol{\gamma}\prime)$$

  and initialization

  $$\forall \boldsymbol{\gamma} \in \Lambda_L, \alpha_1(\boldsymbol{\gamma}) = \beta_N(\boldsymbol{\gamma}) = 1. \tag{3.51}$$

Provided that $u_1, u_2, \dots u_{N-1}$ are in $\mathscr{A}\Big(\Gamma^2 \to \mathbb{R}\Big)$ and $p_{\mathbf{H}_1}$ is in $\mathscr{A}\Big(\Gamma \to \mathbb{R}\Big)$, we have

$$\lim_{L\to\infty} \frac{\sum\limits_{\boldsymbol{\gamma}\in\Lambda_L} \boldsymbol{f}(\boldsymbol{\gamma})\alpha_n(\boldsymbol{\gamma})\beta_n(\boldsymbol{\gamma})}{\sum\limits_{\boldsymbol{\gamma}'\in\Lambda_L} \alpha_n(\boldsymbol{\gamma}')\beta_n(\boldsymbol{\gamma}')} =< p_{\mathbf{H}_n|\mathbf{y}_{1:N}}, \boldsymbol{f} > \tag{3.52}$$

for any $\boldsymbol{f}$ in $\mathscr{A}\Big(\Gamma\Big)$.

3.3 Filtering in the multi-asset volatility model

We consider the application of the Markovian Grid-Based Filter (MGF) to the state estimation in a Hidden Markov Model (HMM).

We consider an example of a stochastic volatility model in the multi-asset framework [Gouriéroux et al., 2009]. Let $\mathbf{Y}_n \in \mathbb{R}^2$ denote the log-returns of two correlated assets. We assume that

$$\mathbf{Y}_n \sim \mathscr{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_n\right), \tag{3.53}$$

where $\boldsymbol{\Sigma}_n \in \mathbb{R}^{2\times 2}$ is the dynamic covariance of $\mathbf{Y}_n$. We assume that $\boldsymbol{\Sigma}_n$ follows a Wishart autoregressive process [Gouriéroux et al., 2009] and we set for our example

$$\boldsymbol{\Sigma}_n = \mathbf{X}_n\mathbf{X}_n^\top + \boldsymbol{Q}; \tag{3.54a}$$

$$\mathbf{X}_{n+1} = \boldsymbol{A}\mathbf{X}_n + \boldsymbol{D}\mathbf{U}_n, \tag{3.54b}$$

where $\mathbf{X}_n \in \mathbb{R}^2$, $\boldsymbol{Q}$, $\boldsymbol{A}$ and $\boldsymbol{D}$ are fixed in $\mathbb{R}^{2\times 2}$, $\boldsymbol{Q}$ is positive definite, $\{\mathbf{U}_n\}_{n\geq 1}$ is a Gaussian white noise process in $\mathbb{R}^2$ and $\mathbf{X}_0 = 0$.

In the simulation study, we apply our algorithm to approximate $\{\mathbf{X}_n\}_{n\geq 1}$ by $\{\mathrm{R}_n\}_{n\geq 1}$. Next, we estimate $\mathbf{X}_n$ given $\mathbf{Y}_{1:n}$ by the MGF, then we compute filtered estimates of $\boldsymbol{\Sigma}_n$.

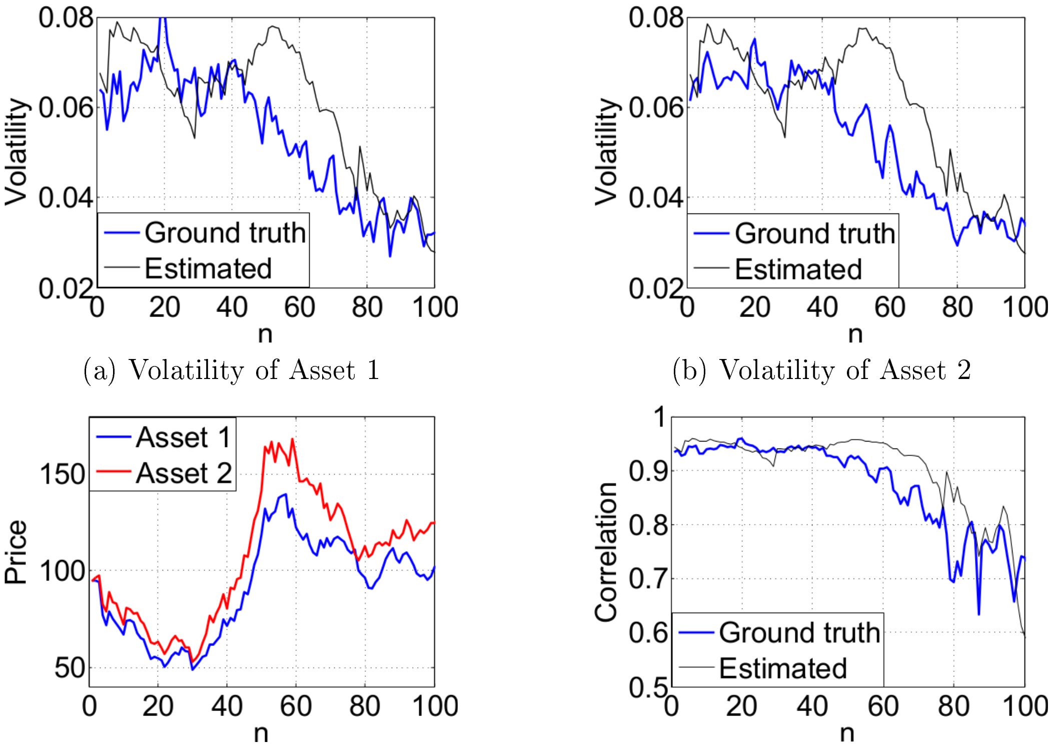
Figure 3.1 illustrates an example of the realization of the multivariate stochastic volatility process and posterior estimation of the volatilities and correlations which we obtain by using the MGF. The parameters of the multivariate stochastic volatility model are in Table 3.1

$\boldsymbol{Q}$	$\boldsymbol{A}$	$\boldsymbol{D}\boldsymbol{D}^\top$
$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \times 10^{-4}$	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \times 10^{-5}$

Table 3.1: The parameters of the volatility model {(3.53), (3.54)}

3.4 Conclusion

We proposed a novel state estimator(MGSE) for general POMP with hybrid state space. We applied it to the problem of Bayesian inference in POMPs. Experiments on the multivariate stochastic volatility model show that the method proposed is suitable for high-dimensional state spaces and may realize some speedups compared to the existing approaches.



(a) Volatility of Asset 1

(b) Volatility of Asset 2

(c) Simulated prices

(d) Correlations of log-returns

Figure 3.1: A realization of the multivariate stochastic volatility process {(3.53), (3.54)}. In figures (a), (b) and (d), the black line plots the filtering estimates of the volatilities and correlations.

Chapter 4

Bayesian state estimation in partially observed Markov processes with discrete state space

The Hidden Markov Model (HMM) [Cappé et al., 2005, Gobet and Maire, 2005, Potin et al., 2006b, Rabiner, 1989, Caron et al., 2006, Potin et al., 2006a, Benhamou et al., 2010] is an important tool in the modern modeling of various types of problems and is an active topic of research activity. This model is extensively reviewed in the literature [Bhar and Hamori, 2006, Mamon and Elliott, 2007, Koski, 2001, Vidyasagar, 2014]. Let $N \in \mathbb{N}^*, d' \in \mathbb{N}^*$, we consider hidden random sequence $\{\mathrm{R}_1, \ldots, \mathrm{R}_N\} = \mathrm{R}_{1:N}$, where for each n in $\{1 : N\}$, R_n is in a finite set $\Omega = \{1 : K\}$ and an observed sequence $\mathbf{Y}_{1:N}$, where for each n in $\{1 : N\}$, $\mathbf{Y}_n$ is in $\mathbb{R}^{d'}$.

If the pair $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is a classic HMM, then $\mathrm{R}_{1:N}$ is a Markov chain. The Pairwise Markov Model (PMM) extends HMMs by only assuming that $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is Markovian [Pieczynski, 2003]. Since the hidden process $\mathrm{R}_{1:N}$ is not necessarily Markovian in PMMs, the latter are strictly more general than HMMs [Lanchantin et al., 2011]. In a stationary and time-reversible PMM, $\mathrm{R}_{1:N}$ is Markovian if and only if the conditional dependencies in the PMM verify specific conditions [Lanchantin et al., 2011]. Indeed, the classic Bayesian estimation algorithms, used in HMMs, such as the Baum-Welch algorithm and the Viterbi algorithm apply in PMMs as well, thanks to the fact that $\mathrm{R}_{1:N}$ is Markovian given $\mathbf{Y}_{1:N}$. Let us note that PMMs have been shown to be more efficient than HMMs in the context of unsupervised image segmentation [Derrode and Pieczynski, 2004].

Next, the Triplet Markov Model (TMM) [Pieczynski et al., 2003] extends PMM by adding a discrete-valued latent process $\mathrm{U}_{1:N} = \mathrm{U}_{1:N}$, where each U_n takes its value in a finite set $\{\lambda_1, \ldots, \lambda_M\}$. In such a model, $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}, \mathrm{U}_{1:N})$ is Markovian. Despite that none of processes $\mathrm{R}_{1:N}$, $\mathbf{Y}_{1:N}$, $(\mathrm{R}_{1:N}, \mathrm{U}_{1:N})$, $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$, $(\mathbf{Y}_{1:N}, \mathrm{U}_{1:N})$ is necessarily Markovian, the Baum-Welch algorithm (but not the Viterbi algorithm) applies in TMMs [Lanchantin et al., 2011]. Let us remark that a sub-class of TMMs is shown to be efficient in image processing in [Lanchantin et al., 2011], where it substantially outperforms HMMs. Further researches demonstrate that TMMs allow a semi-Markovian modeling of $\mathrm{R}_{1:N}$ [Lapuyade-Lahorgue and Pieczynski, 2012], which is a valuable result since the hidden semi-Markov models are particularly well-suited for a scope of applications [Yu, 2016, Barbu and Limnios, 2016]. Besides, the *bivariate hidden Markov chains* [Ephraim and Mark, 2015, Sun et al., 2016], which are similar to a sub-class of TMMs, do also provide a framework for efficient data processing. For these reasons, we believe that researches on TMMs may have a considerable impact. TMMs apply in the context of signal processing [Lapuyade-Lahorgue and Pieczynski, 2012, Cam et al., 2008], image processing [Bricq et al., 2006] and canceling non-stationary noise [Boudaren et al., 2011].

In this chapter, we consider HMMs, PMMs and TMMs with discrete state space. The object of the chapter consists in exploring whether using PMMs and TMMs instead of HMMs is meaningful for practical applications. This is done through simulation-based comparisons among several variants of PMMs and TMMs with respect to classic HMMs. Specifically, we consider Gaussian and gamma observation distributions in order to quantify the impact of skewness and excess kurtosis of the latter on the estimation accuracy.

In the next section we present HMMs, PMMs and TMMs. Exact Bayesian inference algorithms for these models are detailed in Section 4.2. Section 4.3 contains a contribution of the author, which is an extensive performance comparisons across the estimators corresponding to HMMs, PMMs and TMMs with discrete state space. Section 4.4 contains another contribution of the author, which is a novel modeling of financial time series with discrete-space PMMs, as well as an application to real-world data with an analysis of the results and discussion.

The section is mainly a compilation of authors' papers [Gorynin et al., 2017c, Gorynin et al., 2017d].

4.1 Hidden, pairwise and triplet Markov Models with discrete state space

The idea of hidden and pairwise Markov models is to describe the probability distribution of $\mathbf{Y}_{1:N}$ by using a hidden time series $\mathrm{R}_{1:N}$, where for each n in $\{1:N\}$, R_n is in $\Omega = \{1:K\}$. Specifically, one defines the probability distribution $p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right)$ of the pair $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$. In this case, we have

$$p\left(\mathbf{y}_{1:N}\right) = \sum_{\mathrm{r}_{1:N} \in \Omega^N} p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right).$$

Both hidden and pairwise Markov models are used to define $p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right)$. In this section we recall the definition and statistical properties of these models.

Definition 26. *HMM*

The pair $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$ is a HMM if it verifies

$$p\left(\mathrm{r}_{1:N}, \mathrm{y}_{1:N}\right) = p\left(\mathrm{r}_1\right) \prod_{n=1}^{N-1} p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n\right.\right) \prod_{n=1}^{N} p\left(\mathrm{y}_n \left| \mathrm{r}_n\right.\right). \tag{4.1}$$

Any HMM has the following properties:

- (P1): $\mathrm{R}_{1:N}$ is a Markov chain;
- (P2): $\mathbf{Y}_{1:N}$ are independent conditional on $\mathrm{R}_{1:N}$;
- (P3): For each n in $\{1:N\}$, $p\left(\mathbf{y}_n \left| \mathrm{r}_{1:N}\right.\right) = p\left(\mathbf{y}_n \left| \mathrm{r}_n\right.\right)$.

Definition 27. *PMM*

The pair $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$ is a pairwise Markov model if its distribution is of the following form:

$$p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathrm{r}_1, \mathbf{y}_1\right) p\left(\mathrm{r}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathbf{y}_1\right.\right) \ldots p\left(\mathrm{r}_N, \mathbf{y}_N \left| \mathrm{r}_{N-1}, \mathbf{y}_{N-1}\right.\right), \tag{4.2}$$

which means that $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$ is Markovian.

The HMM distribution is

$$p\left(\mathrm{r}_{1:N}, \mathbf{y}_{1:N}\right) = p\left(\mathrm{r}_1\right) p\left(\mathbf{y}_1 \left|\mathrm{r}_1\right.\right) p\left(\mathrm{r}_2 \left|\mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left|\mathrm{r}_2\right.\right) \dots p\left(\mathrm{r}_N \left|\mathrm{r}_{N-1}\right.\right) p\left(\mathbf{y}_N \left|\mathrm{r}_N\right.\right), \tag{4.3}$$

and $p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left|\mathrm{r}_n, \mathbf{y}_n\right.\right)$ from (4.2) can be written as

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left|\mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left|\mathrm{r}_n, \mathbf{y}_n\right.\right) p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right.\right) : \tag{4.4}$$

we see that a PMM is an HMM if and only if for each n in $\{1 : N-1\}$,

$$p\left(\mathrm{r}_{n+1} \left|\mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathrm{r}_{n+1} \left|\mathrm{r}_n\right.\right); \tag{4.5a}$$

$$p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n+1}\right.\right). \tag{4.5b}$$

This highlights the stronger assumptions which are made implicitly when a real-world system is modeled by HMM whereas the same system could possibly be represented as a PMM.

Let us consider stationary PMMs for which $p\left(\mathrm{r}_n, \mathbf{y}_n, \mathrm{r}_{n+1}, \mathbf{y}_{n+1}\right)$ does not depend on n. Thus, the whole distribution is defined by $p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right)$. In addition, we assume that the model is time-reversible, which means that for each ω_i, ω_j in Ω and $\mathbf{y}$, $\mathbf{y}'$ in $\mathbb{R}$,

$$p\left(\mathrm{r}_1 = \omega_i, \mathbf{y}_1 = \mathbf{y}, \mathrm{r}_2 = \omega_j, \mathbf{y}_2 = \mathbf{y}'\right) = p\left(\mathrm{r}_2 = \omega_i, \mathbf{y}_2 = \mathbf{y}, \mathrm{r}_1 = \omega_j, \mathbf{y}_1 = \mathbf{y}'\right). \tag{4.6}$$

The following Proposition results from the general result shown in [Lanchantin et al., 2011]:

Proposition 18. *Let* $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$ *be a stationary time-reversible PMM. The following conditions are equivalent:*

- $\mathrm{R}_{1:N}$ *is a Markov chain;*
- *for each* n *in* $\{1 : N-1\}$, $p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n+1}, \mathrm{r}_n\right.\right) = p\left(\mathbf{y}_{n+1} \left|\mathrm{r}_{n+1}\right.\right)$*;*
- *for each* n *in* $\{1 : N\}$, $p\left(\mathbf{y}_n \left|\mathrm{r}_{1:N}\right.\right) = p\left(\mathbf{y}_n \left|\mathrm{r}_n\right.\right)$*.*

Thus, in a stationary time-reversible PMM $\left(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N}\right)$, $\mathrm{R}_{1:N}$ is Markovian if and only if

$$p\left(\mathbf{y}_2 \left|\mathrm{r}_1, \mathrm{r}_2\right.\right) = p\left(\mathbf{y}_2 \left|\mathrm{r}_2\right.\right), \tag{4.7}$$

which is equivalent to

$$p\left(\mathbf{y}_1 \left|\mathrm{r}_1, \mathrm{r}_2\right.\right) = p\left(\mathbf{y}_1 \left|\mathrm{r}_1\right.\right). \tag{4.8}$$

Let us consider the following sub-models of the PMM.

— The Hidden Markov Model With Conditionally Independent Noise (HMM-IN), which is the classic HMM. The related transition kernel $p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \left|\mathrm{r}_n, \mathbf{y}_n\right.\right)$ is

$$p\left(\mathrm{r}_2, \mathbf{y}_2 \left|\mathrm{r}_1, \mathbf{y}_1\right.\right) = p\left(\mathrm{r}_2 \left|\mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left|\mathrm{r}_2\right.\right) \tag{4.9}$$

and $p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right)$ verifies

$$p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right) = p\left(\mathrm{r}_1, \mathrm{r}_2\right) p\left(\mathbf{y}_1 \left|\mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left|\mathrm{r}_2\right.\right). \tag{4.10}$$

— The Hidden Markov Model With Conditionally Correlated Noise (HMM-CN), where $\mathrm{R}_{1:N}$ is Markovian, observation variables $\mathbf{Y}_{1:N}$ are correlated given $\mathrm{R}_{1:N}$, and which is not an HMM-IN (see Figure 4.1). The related transition kernel is

$$p\left(\mathrm{r}_2, \mathbf{y}_2 \left|\mathrm{r}_1, \mathbf{y}_1\right.\right) = p\left(\mathrm{r}_2 \left|\mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left|\mathrm{r}_2, \mathbf{y}_1\right.\right). \tag{4.11}$$

— The Pairwise Markov Model With Conditionally Independent Noise (PMM-IN), where $\mathrm{R}_{1:N}$ is not Markovian, observation variables $\mathbf{Y}_{1:N}$ are independent given $\mathrm{R}_{1:N}$, and which is not an HMM-IN. In PMM-IN, we have $p\left(\mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{r}_2, \mathbf{y}_1\right.\right) = p\left(\mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{r}_2\right.\right)$ and

$$p\left(\mathrm{r}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathbf{y}_1\right.\right) = p\left(\mathrm{r}_2 \left| \mathrm{r}_1, \mathbf{y}_1\right.\right) p\left(\mathbf{y}_2 \left| \mathrm{r}_2, \mathrm{r}_1\right.\right); \tag{4.12}$$

$$p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right) = p\left(\mathrm{r}_1, \mathrm{r}_2\right) p\left(\mathbf{y}_1 \left| \mathrm{r}_1, \mathrm{r}_2\right.\right) p\left(\mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{r}_2\right.\right). \tag{4.13}$$

— The Pairwise Markov Model With Conditionally Correlated Noise (PMM-CN), where $\mathrm{R}_{1:N}$ is not Markovian and observation variables $\mathbf{Y}_{1:N}$ are correlated given $\mathrm{R}_{1:N}$, which is neither HMM-IN, PMM-IN or HMM-CN (see Figure 4.1). The related transition kernel is of the general form

$$p\left(\mathrm{r}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathbf{y}_1\right.\right) = p\left(\mathrm{r}_2 \left| \mathrm{r}_1, \mathbf{y}_1\right.\right) p\left(\mathbf{y}_2 \left| \mathrm{r}_2, \mathrm{r}_1, \mathbf{y}_1\right.\right). \tag{4.14}$$

The whole distribution of a PMM-CN can be derived from $p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right)$. We consider the latter in the following form:

$$p\left(\mathrm{r}_1, \mathbf{y}_1, \mathrm{r}_2, \mathbf{y}_2\right) = p\left(\mathrm{r}_1, \mathrm{r}_2\right) p\left(\mathbf{y}_1, \mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{r}_2\right.\right). \tag{4.15}$$

The dependency graphs of the four sub-models of PMM are presented in Figure 4.2.

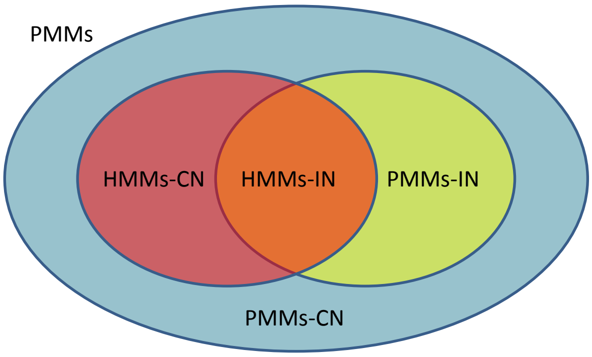


Figure 4.1: Venn diagram for various sub-models of PMM. The area contained by all of the three circles represents PMM. PMM-CN is represented by the rock blue color, PMM-IN and HMM-CN is represented by rose and yellow respectively. The orange color represents HMM-IN.

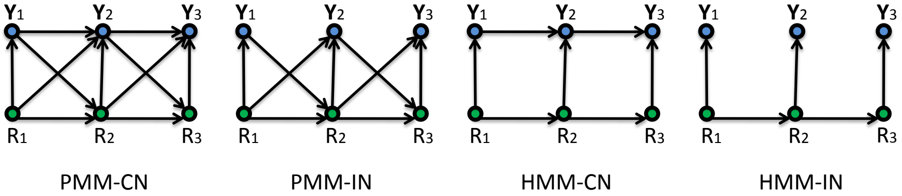


Figure 4.2: Dependency graphs of PMM-CN, PMM-IN, HMM-CN and HMM-IN.

The TMM makes use of an additional discrete-valued process $\mathrm{U}_{1:N}$, where each U_n takes its value in a finite set $\Lambda = \{\lambda_1, \ldots, \lambda_M\}$.

Definition 28. *TMM*

The triplet $(\mathrm{R}_{1:N}, \mathrm{U}_{1:N}, \mathbf{Y}_{1:N})$ is a TMM if its distribution is of the following form:

$$p\left(\mathrm{r}_{1:N}, \mathrm{u}_{1:N}, \mathbf{y}_{1:N}\right) = \\ p\left(\mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1\right) p\left(\mathrm{r}_2, \mathrm{u}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1\right.\right) \ldots p\left(\mathrm{r}_N, \mathrm{u}_N, \mathbf{y}_N \left| \mathrm{r}_{N-1}, \mathrm{u}_{N-1}, \mathbf{y}_{N-1}\right.\right), \tag{4.16}$$

which means that $(\mathrm{R}_{1:N}, \mathrm{U}_{1:N}, \mathbf{Y}_{1:N})$ is Markovian.

TMMs have a high potential of modeling; specifically, $\mathrm{U}_{1:N}$ can be multivariate in a way that each sequence $\mathrm{U}^{(i)}_{1:N}$ in $\mathrm{U}_{1:N} = [\mathrm{U}^{(1)}_{1:N}; \dots ; \mathrm{U}^{(s)}_{1:N}]^\top$ would map a separate property. For example, the non-stationary hidden semi-Markov models can be seen as a TMM $(\mathrm{R}_{1:N}, \mathrm{U}^{(1)}_{1:N}, \mathrm{U}^{(2)}_{1:N}, \mathbf{Y}_{1:N})$ in which $\mathrm{U}^{(1)}_{1:N}$ models the semi-Markovianity and $\mathrm{U}^{(2)}_{1:N}$ stands for the non-stationarity [Lapuyade-Lahorgue and Pieczynski, 2012].

4.2 Exact Bayesian state estimation

In this section, we recall exact Bayesian state estimation algorithms for the hidden, pairwise and triplet Markov models, known as the forward-backward algorithm. We begin by presenting the PMM version of the algorithm, and we detail how HMM and TMM versions can be derived from it.

The PMM forward-backward algorithm allows computing $p\left(\mathrm{r}_n = \omega \,|\mathbf{y}_{1:N}\right)$ for each n in $\{1 : N\}$ and ω in Ω. Let us consider the following *forward* and *backward* probabilities, defined in a PMM by $\alpha_n(\mathrm{r}_n) = p\left(\mathrm{r}_n, \mathbf{y}_{1:n}\right)$ and $\beta_n(\mathrm{r}_n) = p\left(\mathbf{y}_{n+1:N} \,|\mathrm{r}_n, \mathbf{y}_n\right)$. The following recursions allow computing $\alpha_n(\mathrm{r}_n)$ and $\beta_n(\mathrm{r}_n)$ for any r_n:

$$\alpha_1(\mathrm{r}_1) = p\left(\mathrm{r}_1, \mathbf{y}_1\right); \tag{4.17a}$$

$$\alpha_1(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n \in \Omega} p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \,|\mathrm{r}_n, \mathbf{y}_n\right) \alpha_n(\mathrm{r}_n); \tag{4.17b}$$

$$\beta_N(\mathrm{r}_N) = 1; \tag{4.17c}$$

$$\beta_n(\mathrm{r}_n) = \sum_{\mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \,|\mathrm{r}_n, \mathbf{y}_n\right) \beta_{n+1}(\mathrm{r}_{n+1}). \tag{4.17d}$$

Then $p\left(\mathrm{r}_n = \omega \,|\mathbf{y}_{1:N}\right)$ is computed by:

$$p\left(\mathrm{r}_n = \omega \,|\mathbf{y}_{1:N}\right) = \frac{\beta_n(\mathrm{r}_n)\alpha_n(\mathrm{r}_n)}{\sum\limits_{\mathrm{r}^*_n \in \Omega} \beta_n(\mathrm{r}^*_n)\alpha_n(\mathrm{r}^*_n)}. \tag{4.18}$$

The complexity of this algorithm is linear in N.

The HMM and TMM versions of the forward-backward algorithm can be derived as follows.

— The HMM version is derived under conditions (4.5a) and (4.5b), in which case

$$p\left(\mathrm{r}_{n+1}, \mathbf{y}_{n+1} \,|\mathrm{r}_n, \mathbf{y}_n\right) = p\left(\mathrm{r}_{n+1} \,|\mathrm{r}_n\right) p\left(\mathbf{y}_{n+1} \,|\mathbf{y}_n\right).$$

— The TMM version is derived by considering the hidden process $\mathrm{V}_{1:N}$, where for each n in $\{1 : N\}$, $\mathrm{V}_n = (\mathrm{R}_n, \mathrm{U}_n)$ in $\Omega \times \Lambda$. Thus, $(\mathrm{V}_{1:N}, \mathbf{Y}_{1:N})$ is a PMM and it is possible to apply the PMM version of the forward-backward algorithm to compute $p\left(\mathrm{r}_n, \mathrm{u}_n \,|\mathbf{y}_{1:N}\right)$ for each $(\mathrm{r}_n, \mathrm{u}_n)$ in $\Omega \times \Lambda$. Finally, one has

$$p\left(\mathrm{r}_n \,|\mathbf{y}_{1:N}\right) = \sum_{\mathrm{u}_n \in \Lambda} p\left(\mathrm{r}_n, \mathrm{u}_n \,|\mathbf{y}_{1:N}\right). \tag{4.19}$$

The Maximum Posterior Mode (MPM) estimator is defined as

$$\forall n \in 1 : N, \widehat{\mathrm{r}}_n = \arg\max_{\omega \in \Omega} p\left(\mathrm{r}_n = \omega \,|\mathbf{y}_{1:N}\right). \tag{4.20}$$

We see that the MPM estimator is computable in HMMs and PMMs as well as in TMMs, despite the fact that $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ can be non Markovian in TMMs.

4.3 Performance comparison across PMM estimators

Here we present different experiments comparing PMM-CN, PMM-IN, HMM-CN and HMM-IN from Section 4.1, in the case of Gaussian and gamma observation distributions.

We consider the case where $\Omega = \{\omega_1, \omega_2\}$ and for each n in $\{1 : N\}$, $\mathbf{Y}_n$ is one-dimensional.

4.3.1 Gaussian PMM estimators

We parameterize PMM-CN by $\epsilon \in [0, 0.5]$ and $\rho \in [0, 1]$ as follows:

$$p\left(\mathrm{r}_1, \mathrm{r}_2\right) = \begin{cases} 0.5 - \epsilon & \text{if } \mathrm{r}_1 = \mathrm{r}_2; \\ \epsilon & \text{if } \mathrm{r}_1 \neq \mathrm{r}_2; \end{cases} \tag{4.21a}$$

$$p\left(\mathrm{y}_1, \mathrm{y}_2 \left| \mathrm{r}_1, \mathrm{r}_2 \right.\right) = \mathcal{N}\left(\begin{bmatrix} \mathrm{y}_1 \\ \mathrm{y}_2 \end{bmatrix}; \begin{bmatrix} \mu_1(\mathrm{r}_1, \mathrm{r}_2) \\ \mu_2(\mathrm{r}_1, \mathrm{r}_2) \end{bmatrix}, \begin{bmatrix} \sigma_1^2(\mathrm{r}_1, \mathrm{r}_2) & \rho\sigma_1(\mathrm{r}_1, \mathrm{r}_2)\sigma_2(\mathrm{r}_1, \mathrm{r}_2) \\ \rho\sigma_1(\mathrm{r}_1, \mathrm{r}_2)\sigma_2(\mathrm{r}_1, \mathrm{r}_2) & \sigma_2^2(\mathrm{r}_1, \mathrm{r}_2) \end{bmatrix} \right). \tag{4.21b}$$

The coefficients ϵ and ρ depend on the experimental setting; the values of the remaining parameters per each pair $(\mathrm{r}_1, \mathrm{r}_2)$ are fixed and presented in Table 4.1.

$(\mathrm{r}_1, \mathrm{r}_2)$	$\mu_1(\mathrm{r}_1, \mathrm{r}_2)$	$\mu_2(\mathrm{r}_1, \mathrm{r}_2)$	$\sigma_1(\mathrm{r}_1, \mathrm{r}_2)$	$\sigma_2(\mathrm{r}_1, \mathrm{r}_2)$
(ω_1, ω_1)	-5	-5	14	14
(ω_1, ω_2)	-3	3	7	9
(ω_2, ω_1)	3	-3	9	7
(ω_2, ω_2)	5	5	20	20

Table 4.1: Mean and variance parameters of Gaussian distributions in (4.21b).

Let us specify the sampling procedure corresponding to Gaussian PMM-CN (4.21). We begin by sampling $(\mathrm{r}_1, \mathrm{r}_2)$ from (4.21a), then we sample $(\mathrm{y}_1, \mathrm{y}_2)$ given $(\mathrm{r}_1, \mathrm{r}_2)$ from (4.21b). Next, given $(\mathrm{r}_n, \mathbf{y}_n)$ for $n \geq 2$, we sample $(\mathrm{r}_{n+1}, \mathrm{y}_{n+1})$ as follows. Firstly, we sample r_{n+1} from $p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathrm{y}_n \right.\right)$, where

$$p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathrm{y}_n \right.\right) = \frac{p\left(\mathrm{r}_n, \mathrm{r}_{n+1}\right) p\left(\mathrm{y}_n \left| \mathrm{r}_n, \mathrm{r}_{n+1} \right.\right)}{\sum\limits_{\mathrm{r}^*_{n+1} \in \Omega} p\left(\mathrm{r}_n, \mathrm{r}^*_{n+1}\right) p\left(\mathrm{y}_n \left| \mathrm{r}_n, \mathrm{r}^*_{n+1} \right.\right)}, \tag{4.22}$$

with

$$p\left(\mathrm{y}_n \left| \mathrm{r}_n, \mathrm{r}_{n+1} \right.\right) = \mathcal{N}\Big(\mathrm{y}_n; \mu_1(\mathrm{r}_n, \mathrm{r}_{n+1}), \sigma_1^2(\mathrm{r}_n, \mathrm{r}_{n+1})\Big). \tag{4.23}$$

Secondly, we sample y_{n+1} from $p\left(\mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathrm{y}_n \right.\right)$, where

$$p\left(\mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathrm{y}_n \right.\right) = \\ \mathcal{N}\left(\mathrm{y}_{n+1}; \mu_2(\mathrm{r}_n, \mathrm{r}_{n+1}) + \rho\frac{\sigma_2(\mathrm{r}_n, \mathrm{r}_{n+1})}{\sigma_1(\mathrm{r}_n, \mathrm{r}_{n+1})}\Big(\mathrm{y}_n - \mu_1(\mathrm{r}_n, \mathrm{r}_{n+1})\Big), \sigma_2^2(\mathrm{r}_n, \mathrm{r}_{n+1})(1 - \rho^2)\right). \tag{4.24}$$

The experimental setting consists in sampling $(\mathrm{r}_1, \mathrm{y}_1, \ldots, \mathrm{r}_N, \mathrm{y}_N)$ from a given PMM-CN, then estimating $\mathrm{r}_{1:N}$ from $\mathrm{y}_{1:N}$ by four MPM estimators corresponding to the original PMM-CN and its approximations which are PMM-IN, HMM-CN and HMM-IN. We define the related parameters as follows.

– In a Gaussian HMM-IN, one has $p\left(\mathrm{y}_1 \,|\mathrm{r}_1, \mathrm{r}_2\right) = p\left(\mathrm{y}_1 \,|\mathrm{r}_1\right)$ and $p\left(\mathrm{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1\right) = p\left(\mathrm{y}_2 \,|\mathrm{r}_2\right)$, thus

$$p\left(\mathrm{y}_1 \,|\mathrm{r}_1\right) = \mathscr{N}\Big(\mathrm{y}_1; \mu_1^{\texttt{(HMM-IN)}}(\mathrm{r}_1), \sigma_1^{2\texttt{(HMM-IN)}}(\mathrm{r}_1)\Big); \tag{4.25}$$

$$p\left(\mathrm{y}_2 \,|\mathrm{r}_2\right) = \mathscr{N}\Big(\mathrm{y}_2; \mu_2^{\texttt{(HMM-IN)}}(\mathrm{r}_2), \sigma_2^{2\texttt{(HMM-IN)}}(\mathrm{r}_2)\Big). \tag{4.26}$$

Given a Gaussian PMM-CN, we define the parameters of the corresponding HMM-IN by adapting the general principle of moment-matching as follows:

$$\mu_1^{\texttt{(HMM-IN)}}(\mathrm{r}_1) = \sum_{\mathrm{r}_2 \in \Omega} \mu_1(\mathrm{r}_1, \mathrm{r}_2) p\left(\mathrm{r}_2 \,|\mathrm{r}_1\right); \tag{4.27a}$$

$$\sigma_1^{2\texttt{(HMM-IN)}}(\mathrm{r}_1) = \sum_{\mathrm{r}_2 \in \Omega} \Big(\sigma_1^2(\mathrm{r}_1, \mathrm{r}_2) + \big(\mu_1(\mathrm{r}_1, \mathrm{r}_2) - \mu_1^{\texttt{(HMM-IN)}}(\mathrm{r}_1)^2\big)\Big) p\left(\mathrm{r}_2 \,|\mathrm{r}_1\right). \tag{4.27b}$$

By the stationarity assumption, we have for any ω in Ω,

$$\mu_2^{\texttt{(HMM-IN)}}(\omega) = \mu_1^{\texttt{(HMM-IN)}}(\omega); \tag{4.28a}$$

$$\sigma_2^{2\texttt{(HMM-IN)}}(\omega) = \sigma_1^{2\texttt{(HMM-IN)}}(\omega). \tag{4.28b}$$

– In a Gaussian HMM-CN, one has $p\left(\mathrm{y}_1 \,|\mathrm{r}_1, \mathrm{r}_2\right) = p\left(\mathrm{y}_1 \,|\mathrm{r}_1\right)$ and $p\left(\mathrm{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1\right) = p\left(\mathrm{y}_2 \,|\mathrm{r}_2, \mathrm{y}_1\right)$. Thus, we consider the same distribution $p\left(\mathrm{y}_1 \,|\mathrm{r}_1\right)$ as in the case of HMM-IN defined by (4.27). Regarding $p\left(\mathrm{y}_2 \,|\mathrm{r}_2, \mathrm{y}_1\right)$, we have

$$p\left(\mathrm{y}_2 \,|\mathrm{r}_2, \mathrm{y}_1\right) = \\ \mathscr{N}\left(\mathrm{y}_2; \mu_1^{\texttt{(HMM-IN)}}(\mathrm{r}_2) + \rho \frac{\sigma_2^{\texttt{(HMM-IN)}}(\mathrm{r}_2)}{\sigma_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2)} \Big(\mathrm{y}_n - \mu_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2)\Big), \sigma_2^{2\texttt{(HMM-IN)}}(\mathrm{r}_2)(1-\rho^2)\right), \tag{4.29}$$

where $\{\mu_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2), \sigma_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2)\}_{\mathrm{r}_2 \in \Omega}$ are computed by using the principle of moment matching:

$$\mu_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2) = \sum_{\mathrm{r}_1 \in \Omega} \mu_1(\mathrm{r}_1, \mathrm{r}_2) p\left(\mathrm{r}_1 \,|\mathrm{r}_2\right); \tag{4.30a}$$

$$\sigma_{1|2}^{2\texttt{(HMM-CN)}}(\mathrm{r}_2) = \sum_{\mathrm{r}_1 \in \Omega} \Big(\sigma_1^2(\mathrm{r}_1, \mathrm{r}_2) + \big(\mu_1(\mathrm{r}_1, \mathrm{r}_2) - \mu_{1|2}^{\texttt{(HMM-CN)}}(\mathrm{r}_2)^2\big)\Big) p\left(\mathrm{r}_1 \,|\mathrm{r}_2\right). \tag{4.30b}$$

– Finally, a Gaussian PMM-IN approximation of PMM-CN verifies

$$p\left(\mathrm{y}_1 \,|\mathrm{r}_1, \mathrm{r}_2\right) = \mathscr{N}\Big(\mathrm{y}_1; \mu_1(\mathrm{r}_1, \mathrm{r}_2), \sigma_1^2(\mathrm{r}_1, \mathrm{r}_2)\Big); \tag{4.31a}$$

$$p\left(\mathrm{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2\right) = \mathscr{N}\Big(\mathrm{y}_2; \mu_2(\mathrm{r}_1, \mathrm{r}_2), \sigma_2^2(\mathrm{r}_1, \mathrm{r}_2)\Big), \tag{4.31b}$$

since $p\left(\mathrm{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1\right) = p\left(\mathrm{y}_2 \,|\mathrm{r}_1, \mathrm{r}_2\right)$, which is equivalent to set $\rho = 0$.

The PMM-CN estimator is statistically optimal in terms of the classification rate and we consider its accuracy as a reference. The aim of the experiments is to study if the misclassification rate is sensitive to the choice of approximation PMM-IN, HMM-CN or HMM-IN, and up to which extent. We apprehend this sensitivity through the relative error rate, defined as follows:

$$\tau^{\texttt{(model)}} = \frac{L\Big(\mathrm{r}_{1:N}, \widehat{\mathrm{r}_{1:N}}^{\texttt{(model)}}\Big) - L\Big(\mathrm{r}_{1:N}, \widehat{\mathrm{r}_{1:N}}^{\texttt{(PMM-CN)}}\Big)}{L\Big(\mathrm{r}_{1:N}, \widehat{\mathrm{r}_{1:N}}^{\texttt{(PMM-CN)}}\Big)}, \tag{4.32a}$$

$$L\Big(\mathrm{r}_{1:N}, \widehat{\mathrm{r}_{1:N}}^{\texttt{(model)}}\Big) = \frac{1}{N} \sum_{n=1}^{N} \delta\Big(\widehat{\mathrm{r}}_n^{\texttt{(model)}} \neq \mathrm{r}_n\Big), \tag{4.32b}$$

where $\delta(.)$ is the indicator function and $\widehat{\mathrm{r}_{1:N}}^{(\texttt{model})}$ is the state estimate computed by using the Bayesian-optimal MPM state estimator related to the corresponding model. For example, a relative error rate of 100% means that the reference model decreases the misclassification percentage by a half when compared to the proposal one. We report in Tables 4.3 and 4.4 relative error rates for various values of ϵ and ρ. We also report in Table 4.2 the corresponding statistically optimal loss function values. That is to illustrate that the chosen parameter set actually represents a considerable noise level. Figures 4.3 and 4.4 present more exhaustive results regarding the relative error rate of the HMM-IN.

ϵ \ ρ	0.00	0.35	0.70	0.90
0.05	0.20	0.24	0.25	0.24
0.15	0.28	0.29	0.27	0.23
0.20	0.29	0.29	0.25	0.21
0.35	0.26	0.23	0.17	0.12

Table 4.2: Error rate (4.32b) of model (4.21) for varying ϵ and ρ. Sample size is 1000 and the results are averaged over 100 experiments.

ϵ	HMM-IN	HMM-CN	PMM-IN
0.05	41%	13%	38%
0.15	47%	34%	25%
0.20	56%	38%	24%
0.35	58%	31%	37%
Avg	**51%**	**29%**	**31%**

Table 4.3: Relative error rates (4.32a) of the three Gaussian PMM sub-models for varying ϵ with $\rho = 0.75$. Sample size is 1000 and the results are averaged over 100 experiments.

ρ	HMM-IN	HMM-CN	PMM-IN
0.00	13%	13%	0%
0.35	18%	14%	6%
0.70	38%	26%	23%
0.90	69%	52%	44%
Avg	**35%**	**26%**	**18%**

Table 4.4: Relative error rates (4.32a) of the three Gaussian PMM sub-models for varying ρ with $\epsilon = 0.125$. Sample size is 1000 and the results are averaged over 100 experiments.

Regarding results presented in Tables 4.3 and 4.4, we notice that the HMM-IN approximation seems to be the least accurate, while PMM-IN and HMM-CN have both fairly the same degree of performance. Regarding Figures 4.3 and 4.4, we observe that $\tau^{(\texttt{HMM-IN})} < 20\%$ only if $\rho < 0.4$ and $\epsilon < 0.15$, $\tau^{(\texttt{HMM-IN})} < 50\%$ only if $\rho < 0.6$, and $\tau^{(\texttt{HMM-IN})} < 80\%$ only if $\rho < 0.85$. For extreme values of ϵ in a neighborhood of 0.5 and for ρ in a neighborhood of 1, $\tau^{(\texttt{HMM-IN})}$ diverges.

We notice that both features of PMM-CN, *i.e.* $p\left(\mathrm{r}_{n+1} \middle| \mathrm{r}_n, \mathbf{y}_n\right) \neq p\left(\mathrm{r}_{n+1} \middle| \mathrm{r}_n\right)$ and $p\left(\mathbf{y}_{n+1} \middle| \mathrm{r}_{n+1}, \mathrm{r}_n, \mathbf{y}_n\right) \neq p\left(\mathbf{y}_{n+1} \middle| \mathrm{r}_{n+1}, \mathrm{r}_n\right)$ contribute independently to improving its accuracy over the simpler models. For these reasons, PMM-CN may decrease the misclassification rate of HMM-IN by a half in several settings.

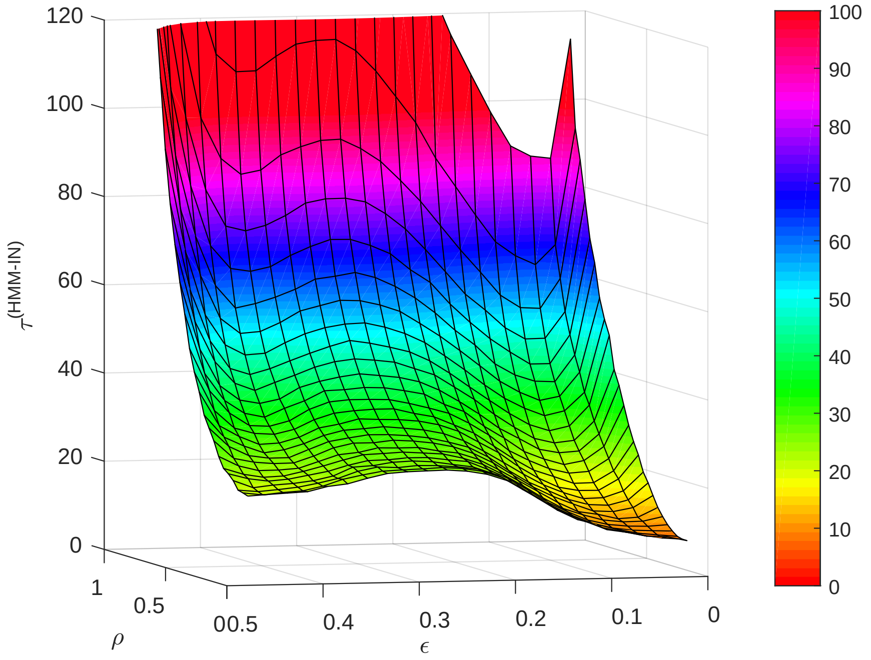


Figure 4.3: Relative error rate surface plot for Gaussian HMM-IN (4.32a) in function of (ϵ, ρ). Sample size is 1000 and the results are averaged over 100 experiments.

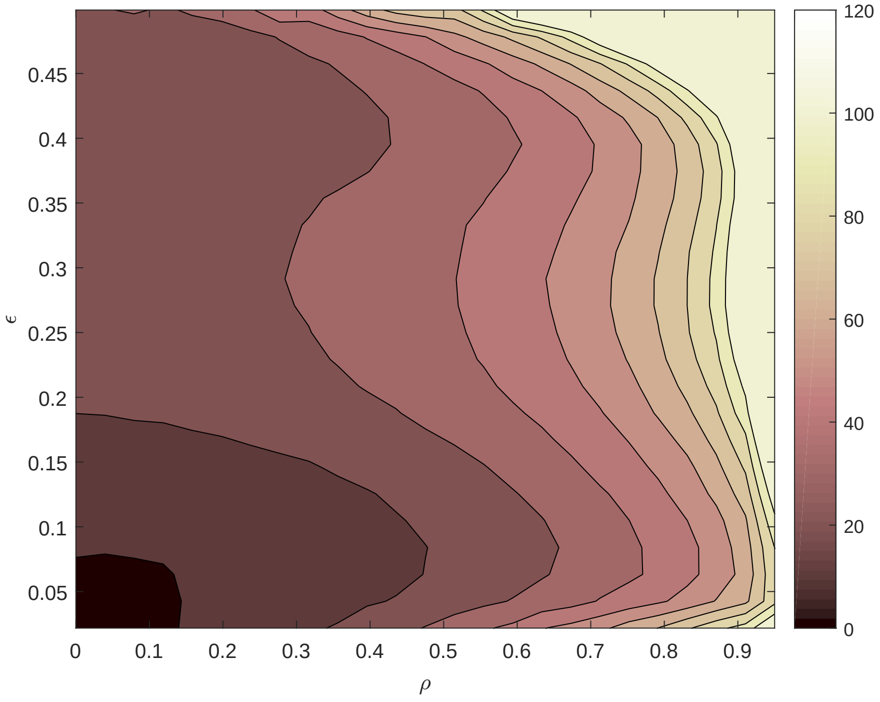


Figure 4.4: Relative error rate (4.32a) contour plot for Gaussian HMM-IN in function of (ϵ, ρ). The contour lines refer to relative error rates of 10%, 20%, ..., 90% and 100%. Sample size is 1000 and the results are averaged over 100 experiments.

4.3.2 Gamma PMM estimators

Here we introduce a class of non-Gaussian pairwise Markov models in order to study whether previous findings generalize to a broader class of observation distributions. We consider hidden Markov models with exponential noise [Lethanh and Adey, 2013] and we extend them to hidden Markov models with gamma noise. Then we introduce PMM-CN with gamma noise and we conduct similar experiments as in the previous subsection.

For shape parameter k in $\mathbb{R}_+^*$ and scale parameter θ in $\mathbb{R}_+^*$, let us note with $\gamma(k,\theta)$ the corresponding gamma distribution. Its probability density function is

$$\gamma(y;k,\theta) = \frac{y^{k-1}\exp\left(-\frac{y}{\theta}\right)}{\Gamma(k)\theta^k}\mathbb{1}_{y>0}, \tag{4.33}$$

where $\delta(.)$ is the indicator function and Γ is the gamma function:

$$\Gamma(k) = \int\limits_0^{+\infty} t^{k-1}\exp(-t)\,\mathrm{d}t. \tag{4.34}$$

The exponential distribution

$$\mathscr{E}(y;k,\lambda) = \lambda\exp(-\lambda y)\,\mathbb{1}_{y>0} \tag{4.35}$$

is a special case of the gamma distribution, corresponding to $k=1$ and $\theta = \frac{1}{\lambda}$. For large values of k, $\gamma(k,\theta)$ is well approximated by Gaussian distribution $\mathcal{N}(k\theta, k\theta^2)$; if k is close to zero, the gamma distribution is highly asymmetric.

Let $\rho \in [0,1]$, we consider stationary gamma-autoregressive process [Gourieroux and Jasiak, 2006] defined by

$$p(\mathrm{y}_1) = \gamma(\mathrm{y}_1;k,\theta); \tag{4.36a}$$

$$p(\mathrm{y}_{n+1}|\mathrm{y}_n) = \bar{\gamma}\left(\mathrm{y}_{n+1};k,\frac{\rho}{\theta(1-\rho)}\mathrm{y}_n,\theta(1-\rho)\right), \tag{4.36b}$$

where, for β in $\mathbb{R}_+^*$, $\bar{\gamma}(y;k,\beta,\theta)$ is non-central gamma distribution

$$\bar{\gamma}(y;k,\beta,\theta) = \sum_{t=0}^{+\infty}\frac{\beta^t y^{k+t-1}\exp\left(-\frac{y}{\theta}\right)}{t!\Gamma(k+t)\theta^{k+t}\exp(\beta)}\mathbb{1}_{y>0}. \tag{4.37}$$

The mean and variance of $\bar{\gamma}(k,\beta,\theta)$ are $k\theta+\beta\theta$ and $k\theta^2+2\beta\theta^2$ respectively; besides, we have $\bar{\gamma}(k,0,\theta)=\gamma(k,\theta)$.

Hence, let $\sigma_1,\sigma_2:\Omega^2\to\mathbb{R}_+^*$,$\mu_1,\mu_2:\Omega^2\to\mathbb{R}$, $\rho\in[0,1[$, we define gamma PMM-CN as follows:

$$p(\mathrm{y}_1|\mathrm{r}_1,\mathrm{r}_2) = \gamma(\mathrm{y}_1-\mu_1(\mathrm{r}_1,\mathrm{r}_2);k,\theta_1(\mathrm{r}_1,\mathrm{r}_2)); \tag{4.38a}$$

$$p(\mathrm{y}_{n+1}|\mathbf{y}_n,\mathrm{r}_n,\mathrm{r}_{n+1}) = \bar{\gamma}\left(\mathrm{y}_{n+1}-\mu_2(\mathrm{r}_1,\mathrm{r}_2);k,\frac{\rho\Big(\mathrm{y}_n-\mu_1(\mathrm{r}_1,\mathrm{r}_2)\Big)}{\theta_1(\mathrm{r}_n,\mathrm{r}_{n+1})(1-\rho)},\theta_2(\mathrm{r}_n,\mathrm{r}_{n+1})(1-\rho)\right), \tag{4.38b}$$

where $\theta_1(\mathrm{r}_n,\mathrm{r}_{n+1}) = \frac{\sigma_1(\mathrm{r}_n,\mathrm{r}_{n+1})}{\sqrt{k}}$, $\theta_2(\mathrm{r}_n,\mathrm{r}_{n+1}) = \frac{\sigma_2(\mathrm{r}_n,\mathrm{r}_{n+1})}{\sqrt{k}}$.

This model is consistent with the definition of the autoregressive gamma process in the same way as the Gaussian PMM is consistent with the autoregressive Gaussian process. Moreover, this model generalizes exponential hidden Markov models: in gamma HMMs, $\rho=0$, $\sigma_1(\mathrm{r}_n,\mathrm{r}_{n+1}),\mu_1(\mathrm{r}_n,\mathrm{r}_{n+1})$ depend only on r_n, $\sigma_2(\mathrm{r}_n,\mathrm{r}_{n+1}),\mu_2(\mathrm{r}_n,\mathrm{r}_{n+1})$ depend only on r_{n+1} and exponential HMMs verify additionally $k=1$. Moreover, any gamma PMM can be approximated by gamma HMM-IN, gamma HMM-CN and gamma PMM-IN by using the corresponding formulas (4.27)-(4.28), (4.30) and (4.31).

Similarly to the previous subsection, we report in Tables 4.6 and 4.7 relative error rates of the three sub-models of gamma PMM and we report in Table 4.5 the corresponding statistically optimal loss function values. We consider the case of exponential models *i.e.* $k=1$. The values of $\sigma_1(\mathrm{r}_n,\mathrm{r}_{n+1}),\mu_1(\mathrm{r}_n,\mathrm{r}_{n+1}),\sigma_2(\mathrm{r}_n,\mathrm{r}_{n+1}),\mu_2(\mathrm{r}_n,\mathrm{r}_{n+1})$ are the same as previously and given in Table 4.1.

ρ ϵ	0.00	0.35	0.70	0.90
0.05	0.34	0.38	0.40	0.38
0.15	0.41	0.43	0.42	0.39
0.20	0.43	0.43	0.42	0.38
0.35	0.43	0.42	0.38	0.31

Table 4.5: Error rate (4.32b) of gamma PMM (4.38) for varying ϵ and ρ. Sample size is 1000 and the results are averaged over 100 experiments.

ϵ	HMM-IN	HMM-CN	PMM-IN
0.05	79%	75%	16%
0.15	130%	115%	23%
0.20	171%	125%	30%
0.35	186%	125%	63%
Avg	**142%**	**29%**	**33%**

Table 4.6: Relative error rates (4.32a) of the three gamma PMM sub-models for varying ϵ and $\rho = 0.75$, $k = 1$. Sample size is 1000 and the results are averaged over 100 experiments.

ρ	HMM-IN	HMM-CN	PMM-IN
0.00	118%	118%	0%
0.35	114%	112%	4%
0.70	113%	109%	17%
0.90	129%	108%	41%
Avg	**119%**	**112%**	**16%**

Table 4.7: Relative error rates (4.32a) of the three gamma PMM sub-models for varying ρ and $\epsilon = 0.125$, $k = 1$. Sample size is 1000 and the results are averaged over 100 experiments.

This simulation study shows that non-Gaussian PMMs allow achieving substantial gains in accuracy, as well as the Gaussian ones. Moreover, PMMs seem outperform HMMs even more when the observation distributions are asymmetric. In order to validate this finding, we consider a fixed pair (ϵ, ρ) and we gradually increase the value of the shape parameter k from 0.1 to 10. We report the corresponding relative error rates of gamma HMM-IN with respect to gamma PMM-CN in Figure 4.5. When $k = 1$, the corresponding relative error rate is 118% and can be found in Table 4.7. When $k = 10$, the corresponding gamma distribution is close to the normal distribution, and the corresponding relative error rate can be found in Table 4.4.

4.3.3 TMM estimators

We considered previously three extensions of the classic HMM-IN. Here we propose two other ones, which are based on TMMs: the Simplified Triplet Markov Model (STMM) and the Triplet Markov Model With Independent Noise (TMM-IN). They are defined as follows.

— STMM is a stationary time-reversible TMM, whose distribution is defined by

$$p\left(\mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1, \mathrm{r}_2, \mathrm{u}_2, \mathbf{y}_2\right) = p\left(\mathrm{u}_1, \mathrm{u}_2\right) p\left(\mathbf{y}_1 \left| \mathrm{u}_1 \right.\right) p\left(\mathrm{r}_1 \left| \mathrm{u}_1 \right.\right) p\left(\mathbf{y}_2 \left| \mathrm{u}_2 \right.\right) p\left(\mathrm{r}_2 \left| \mathrm{u}_2 \right.\right). \tag{4.39}$$

The corresponding transition kernel is

$$p\left(\mathrm{r}_2, \mathrm{u}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1 \right.\right) = p\left(\mathrm{u}_2 \left| \mathrm{u}_1 \right.\right) p\left(\mathbf{y}_2 \left| \mathrm{u}_2 \right.\right) p\left(\mathrm{r}_2 \left| \mathrm{u}_2 \right.\right). \tag{4.40}$$

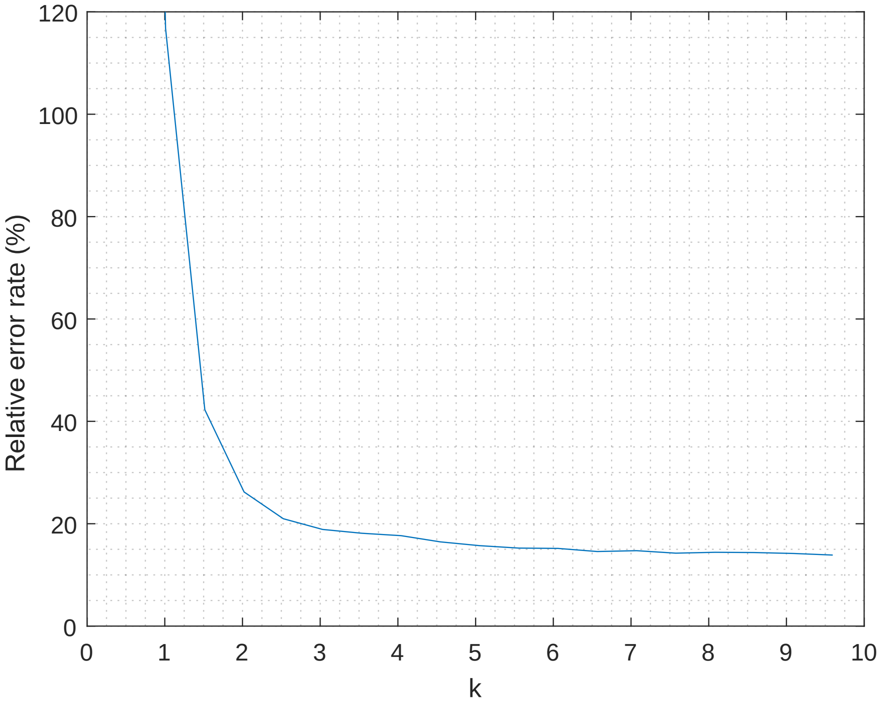


Figure 4.5: Relative error rate (4.32a) of gamma HMM-IN with respect to gamma PMM-CN, for $\epsilon = 0.125$ and $\rho = 0$, in function of the shape parameter k. Sample size is 1000 and the results are averaged over 100 experiments.

$(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ is not Markovian in STMM, so STMM is not a PMM. In fact, one can see an STMM as a hidden Markov model with $\mathrm{U}_{1:N}$ hidden and $(\mathrm{R}_{1:N}, \mathbf{Y}_{1:N})$ observed, and it is well-known that the observed process is not Markovian in such a model.

— TMM-IN is an extension of the STMM on the one hand, and an extension of the classic HMM-IN on the other hand. Specifically, let $\mathrm{V}_{1:N} = (\mathrm{R}_{1:N}, \mathrm{U}_{1:N})$, then we assume that $(\mathrm{V}_{1:N}, \mathbf{Y}_{1:N})$ is a classic HMM-IN and this is why we denote it by TMM-IN. Thus, the distribution of a stationary TMM-IN is given by

$$p\left(\mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1, \mathrm{r}_2, \mathrm{u}_2, \mathbf{y}_2\right) = p\left(\mathrm{u}_1, \mathrm{u}_2, \mathrm{r}_1, \mathrm{r}_2\right) p\left(\mathbf{y}_1 \left| \mathrm{u}_1, \mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left| \mathrm{u}_2, \mathrm{r}_2\right.\right), \tag{4.41}$$

and the corresponding transition kernel is

$$p\left(\mathrm{r}_2, \mathrm{u}_2, \mathbf{y}_2 \left| \mathrm{r}_1, \mathrm{u}_1, \mathbf{y}_1\right.\right) = p\left(\mathrm{u}_2, \mathrm{r}_2 \left| \mathrm{u}_1, \mathrm{r}_1\right.\right) p\left(\mathbf{y}_2 \left| \mathrm{u}_2, \mathrm{r}_2\right.\right). \tag{4.42}$$

The dependency graphs of STMM and TMM-IN are given in Figure 4.6.

We simulate data according to an STMM and we recover $\mathrm{R}_{1:N}$ from $\mathrm{Y}_{1:N}$ with the STMM on the one hand, and with an HMM-IN on the other hand.

Let $\Omega = \{\omega_1, \omega_2\}$ and $\Lambda = \{\lambda_1, \lambda_2\}$, we define the following STMM, whose distribution (4.39) is specified as follows:

$$p\left(\mathrm{u}_1, \mathrm{u}_2\right) = \begin{cases} 0.49 & \text{if } \mathrm{u}_1 = \mathrm{u}_2; \\ 0.01 & \text{if } \mathrm{u}_1 \neq \mathrm{u}_2; \end{cases} \tag{4.43a}$$

$$p\left(\mathrm{r}_1 \left| \mathrm{u}_1\right.\right) = \begin{cases} 0.7 & \text{if } \mathrm{r}_1 = \mathrm{u}_1; \\ 0.3 & \text{if } \mathrm{r}_1 \neq \mathrm{u}_1; \end{cases} \tag{4.43b}$$

$$p\left(\mathrm{y}_1 \left| \mathrm{u}_1\right.\right) = \mathcal{N}(\mathrm{y}_1; \mu_{\mathrm{u}}(\mathrm{u}_1), \sigma^2), \tag{4.43c}$$

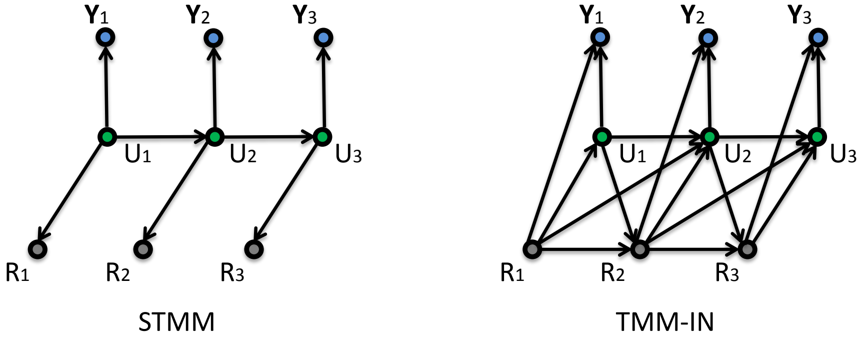


Figure 4.6: Dependency graphs of STMM and TMM-IN.

where $\mu_{\mathrm{u}}(\lambda_1) = -1$ and $\mu_{\mathrm{u}}(\lambda_2) = 1$. We define the distribution $p\left(\mathrm{r}_1, \mathrm{y}_1, \mathrm{r}_2, \mathrm{y}_2\right)$ in (4.10) of the HMM-IN approximation to (4.39) as follows:

$$p\left(\mathrm{r}_1, \mathrm{r}_2\right) = \sum_{\mathrm{u}_1, \mathrm{u}_2 \in \Lambda} p\left(\mathrm{u}_1, \mathrm{u}_2\right) p\left(\mathrm{r}_1 \left| \mathrm{u}_1\right.\right) p\left(\mathrm{r}_2 \left| \mathrm{u}_2\right.\right); \tag{4.44a}$$

$$p\left(\mathrm{y}_1 \left| \mathrm{r}_1\right.\right) = \mathscr{N}\Big(\mathrm{y}_1; \mu_{\mathrm{r}}(\mathrm{r}_1), \sigma_{\mathrm{r}}^2(\mathrm{r}_1)\Big), \tag{4.44b}$$

where the parameters of $p\left(\mathrm{y}_1 \left| \mathrm{r}_1\right.\right)$ are computed by the principle of moment-matching:

$$\mu_{\mathrm{r}}(\mathrm{r}_1) = \sum_{\mathrm{u}_1 \in \Lambda} p\left(\mathrm{u}_1 \left| \mathrm{r}_1\right.\right) \mu_{\mathrm{u}}(\mathrm{u}_1); \tag{4.45a}$$

$$\sigma_{\mathrm{r}}^2(\mathrm{r}_1) = \sigma^2 + \sum_{\mathrm{u}_1 \in \Lambda} \Big(\mu_{\mathrm{u}}(\mathrm{u}_1) - \mu_{\mathrm{r}}(\mathrm{r}_1)\Big)^2 p\left(\mathrm{u}_1 \left| \mathrm{r}_1\right.\right). \tag{4.45b}$$

We computing misclassification rates for STMM and its HMM-IN approximation for various values of σ. We observe that HMM-IN approximation appears to be fairly suboptimal for several values of σ.

Next, we compare TMM-IN (4.41) with the classic HMM-IN and its three extensions. The first one is known as the mixture-HMM [Paul, 1991] and is a classic hidden Markov model where $\mathrm{R}_{1:N}$ is Markovian, observation variables $\mathrm{Y}_{1:N}$ are independent given $\mathrm{R}_{1:N}$ and the observation density is represented by a mixture of Gaussian distributions. We denote it as mixture-HMM-IN. The two others are obtained from HMM-IN and mixture-HMM-IN by considering Markovianity of order 2, *cf. e.g.* [Vidyasagar, 2014]. They are denoted by HMM-IN-2 and mixture-HMM-IN-2 respectively. We set $\Omega = \{\omega_1, \omega_2\}$ and $\Lambda = \{\lambda_1, \lambda_2\}$.

– The mixture-HMM-IN is a TMM-IN sub-model in which $\mathrm{R}_{1:N}$ and $\mathrm{U}_{1:N}$ are independent and variables $\mathrm{U}_{1:N}$ are independent too. The corresponding distribution of $(\mathrm{R}_{1:N}, \mathrm{U}_{1:N}, \mathrm{Y}_{1:N})$ is the following:

$$\begin{aligned} &p\left(\mathrm{r}_{1:N}, \mathrm{u}_{1:N}, \mathrm{y}_{1:N}\right) = \\ &\quad p\left(\mathrm{r}_1\right) p\left(\mathrm{r}_2 \left| \mathrm{r}_1\right.\right) \dots p\left(\mathrm{r}_N \left| \mathrm{r}_{N-1}\right.\right) p\left(\mathrm{u}_1\right) p\left(\mathrm{u}_2\right) \dots p\left(\mathrm{u}_N\right) p\left(\mathbf{y}_1 \left| \mathrm{r}_1, \mathrm{u}_1\right.\right) \dots p\left(\mathbf{y}_N \left| \mathrm{r}_N, \mathrm{u}_N\right.\right). \end{aligned} \tag{4.46}$$

– In an HMM-IN-2, one has

$$p\left(\mathrm{r}_{1:N}, \mathrm{y}_{1:N}\right) = p\left(\mathrm{r}_1, \mathrm{r}_2\right) p\left(\mathrm{r}_3 \left| \mathrm{r}_1, \mathrm{r}_2\right.\right) \dots p\left(\mathrm{r}_N \left| \mathrm{r}_{N-1}, \mathrm{r}_{N-2}\right.\right) p\left(\mathrm{y}_1 \left| \mathrm{r}_1\right.\right) \dots p\left(\mathrm{y}_N \left| \mathrm{r}_N\right.\right). \tag{4.47}$$

Thus, an HMM-IN-2 is technically a TMM-IN where $\Lambda = \Omega$ and for each n in $\{1 : N\}$, $\mathrm{U}_n = \mathrm{X}_{n-1}$ and $p\left(\mathrm{y}_n \left| \mathrm{r}_n, \mathrm{u}_n\right.\right) = p\left(\mathrm{y}_n \left| \mathrm{r}_n\right.\right)$. Notice that if K is the number of elements in Ω, K' is that of Λ, then the dimension of the hidden space of TMM is KK'. Thus,

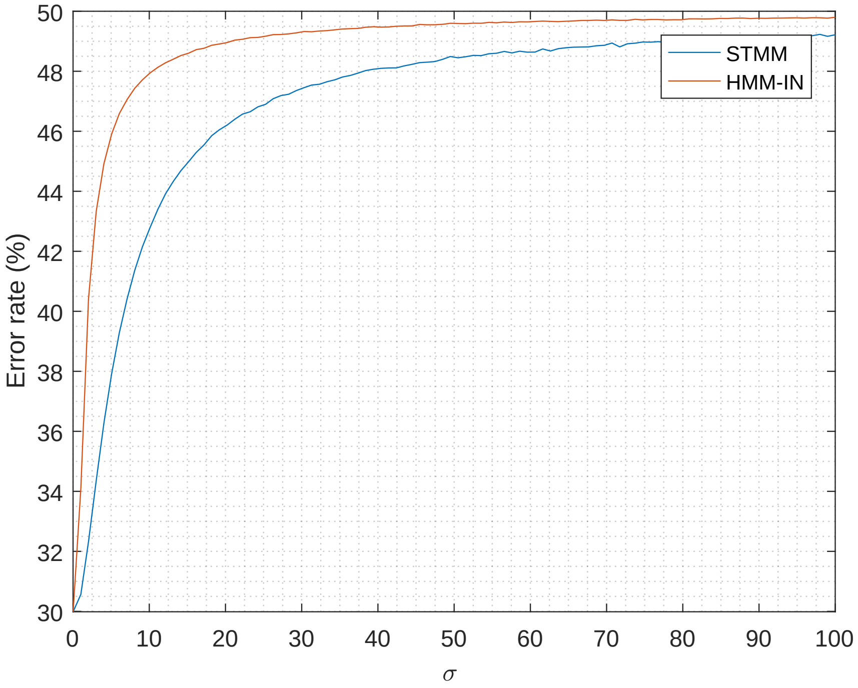


Figure 4.7: Misclassification rates of STMM and HMM-IN for various values of σ in (4.43c). Sample size is 1000 and the results are averaged over 100 experiments.

HMM-IN-2 is equivalent to TMM-IN in terms of the dimension of the hidden space if $K = K'$. Therefore, we find meaningful to compare TMMs-IN with HMMs-IN as well as with HMMs-IN-2 in our experiments.

The state distribution $p\left(\mathrm{r}_1, \mathrm{u}_1, \mathrm{r}_2, \mathrm{u}_2\right)$ within TMMs-IN considered for the experiments is given in Table 4.8.

$(\mathrm{r}_2, \mathrm{u}_2)$ $(\mathrm{r}_1, \mathrm{u}_1)$	(ω_1, λ_1)	(ω_1, λ_2)	(ω_2, λ_1)	(ω_2, λ_2)
(ω_1, λ_1)	0.22	0.01	0.01	0.01
(ω_1, λ_2)	0.01	0.22	0.01	0.01
(ω_2, λ_1)	0.01	0.01	0.22	0.01
(ω_2, λ_2)	0.01	0.01	0.01	0.22

Table 4.8: Probability values $\{p(\mathrm{x}_1, \mathrm{u}_1, \mathrm{x}_2, \mathrm{u}_2) | \mathrm{x}_1, \mathrm{x}_2 \in \Omega, \mathrm{u}_1, \mathrm{u}_2 \in \Lambda\}$.

Regarding the observation space, we set

$$p\left(\mathrm{y}_1 \left| \mathrm{u}_1, \mathrm{r}_1\right.\right) = \mathcal{N}\left(\mu(\mathrm{u}_1, \mathrm{r}_1), 1\right). \tag{4.48}$$

Let us consider three following cases of positioning of $\mu(\mathrm{u}_1, \mathrm{r}_1)$:

1. : $\mu(\omega_1, \lambda_1) < \mu(\omega_1, \lambda_2) < \mu(\omega_2, \lambda_2) < \mu(\omega_2, \lambda_1)$;
2. : $\mu(\omega_1, \lambda_1) < \mu(\omega_2, \lambda_1) < \mu(\omega_2, \lambda_2) < \mu(\omega_1, \lambda_2)$;
3. : $\mu(\omega_1, \lambda_1) < \mu(\omega_2, \lambda_1) < \mu(\omega_1, \lambda_2) < \mu(\omega_2, \lambda_2)$.

Given the symmetries of $p\left(\mathrm{r}_1, \mathrm{u}_1, \mathrm{r}_2, \mathrm{u}_2\right)$, these cases are exhaustive regarding the problem of estimation of $\mathrm{R}_{1:N}$ from $\mathrm{Y}_{1:N}$. We consider sampling $(\mathrm{R}_{1:N}, \mathrm{Y}_{1:N})$ from TMM-IN and estimating $R_{1:N}$ by HMM-IN, HMM-IN-2, mixture-HMM-IN, mixture-HMM-IN-2 and TMM-IN estimators. In our experiments, we consider simulated samples of size 1000 and we average results over 100 independent identical experiments.

— In Case 1, all the five estimators yield pretty much the same result.

— In Case 2, we consider $\Delta > 0$ and set

$$\begin{cases} \mu(\omega_1, \lambda_1) = -2\Delta; \\ \mu(\omega_2, \lambda_1) = -\Delta; \\ \mu(\omega_2, \lambda_2) = \Delta; \\ \mu(\omega_1, \lambda_2) = 2\Delta. \end{cases} \tag{4.49}$$

Figure 4.8 presents error rates of the five estimators. We observe that the non-mixture classic models are asymptotically sub-optimal.

— In Case 3, we consider $\Delta > 0$ and set

$$\begin{cases} \mu(\omega_1, \lambda_1) = -2\Delta; \\ \mu(\omega_2, \lambda_1) = -\Delta; \\ \mu(\omega_1, \lambda_2) = \Delta; \\ \mu(\omega_2, \lambda_2) = 2\Delta. \end{cases} \tag{4.50}$$

Figure 4.9 presents error rates of the five estimators in Case 3. We observe that the non-mixture classic models diverge. Moreover, we see that the TMM-IN estimator may be significantly more accurate than that of the classic models. The gap we observe between classic mixture-based and TMM-IN estimators may be due to taking the Markovianity of $(R_{1:N}, U_{1:N})$ into account. We pointed out that extending the state space of classic models may not result in improving the accuracy of the corresponding estimators.

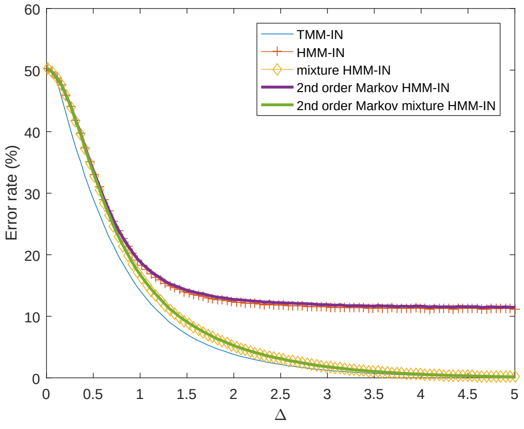


Figure 4.8: Performances comparison between the TMM-IN estimator and its approximations given by the classic models in Case 2, for various Δ in (4.49). Sample size is 1000 and the results are averaged over 100 experiments.

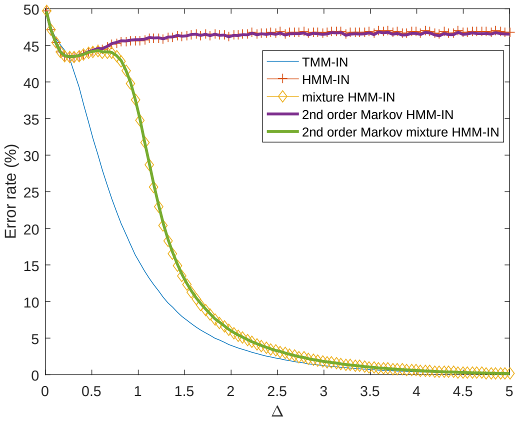


Figure 4.9: Performances comparison between the TMM-IN estimator and its approximations given by the classic models in Case 3, for various Δ in (4.50). Sample size is 1000 and the results are averaged over 100 experiments.

4.3.4 Conclusions

We compared the accuracy of the MPM estimators based on the classic HMM and its extensions which are the PMM and the TMM. PMM and TMM frameworks allowed to achieve substantial improvements of the estimation accuracy. Such improvements were particularly visible when the observation distribution was heavily autocorrelated and/or if the hidden chain was far from being Markovian.

As it is known [Derrode and Pieczynski, 2004], the parameter estimation in the models considered is quite robust, thus the present results confirm the suitability of PMM and TMM frameworks for real-world applications involving unsupervised learning.

4.4 Stock forecasting with PMMs

In this section, we investigate an application of PMMs to stock market prediction.

Universally acknowledged features of financial time series include volatility clustering, autocorrelation in returns and the Asymmetric Volatility Phenomenon (AVP). A well-established methodology consists in using a mathematical model to describe available data and to project it into the future. The Autoregressive Integrated Moving Average (ARIMA) and the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) models are popular among practitioners. These models are reviewed in [Montgomery et al., 1990]. In recent years, there was an increasing interest in the regime-switching models, reviewed *e.g.* in [Mamon and Elliott, 2014]. In financial markets, these models allow identifying *bull* and *bear* alternating regimes. A *bull* state is characterized by positive expectation of log-returns and low volatility, while a *bear* state is driven by negative expected log-returns and high volatility. Let also mention the technical analysis which provides a range of approaches for market prediction [Blanchet-Scalliet et al., 2007].

The HMMs provide a suitable framework for modeling regime-switching. An important example of such framework is available in *e.g.* [Hassan and Nath, 2005]. These models use a hidden sequence of the same length as the sequence of observed log-returns. The HMMs are known to be robust and straightforward to implement. However, the HMMs do not take the following potential features of stock dynamics into account:

- (F1): log-returns may be correlated given the state variables;
- (F2): the future state and current log-return may not be independent given the current state.

PMMs are able to include both features (F1) and (F2) in the HMMs for the same processing cost. The purpose of this section is to introduce a modeling of financial time series with the PMMs with discrete state space. Specifically, we investigate if the PMMs can allow improving forecasting performance and if both features (F1) and (F2) should be taken into account.

Figure 4.10 presents dependency graphs of the HMM and PMM.

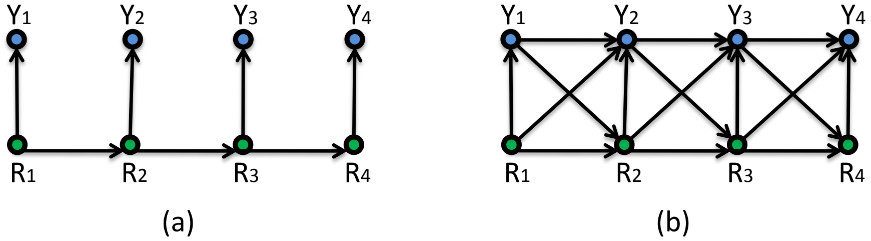


Figure 4.10: Dependency graphs of the HMM (a) and PMM (b).

Consider the following decomposition of $p\left(\mathrm{r}_{n+1}, \mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{y}_n \right.\right)$, for n in $\{1 : N-1\}$:

$$p\left(\mathrm{r}_{n+1}, \mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{y}_n \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathrm{y}_n \right.\right) p\left(\mathrm{y}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathrm{y}_n \right.\right).$$

From the above equation, we see that a PMM is an HMM if, and only if, for each n in $\{1:N-1\}$:

$$p\left(\mathrm{y}_{n+1}\,|\mathrm{r}_n,\mathrm{r}_{n+1},\mathrm{y}_n\right) = p\left(\mathrm{y}_{n+1}\,|\mathrm{r}_{n+1}\right); \tag{4.51a}$$

$$p\left(\mathrm{r}_{n+1}\,|\mathrm{r}_n,\mathrm{y}_n\right) = p\left(\mathrm{r}_{n+1}\,|\mathrm{r}_n\right). \tag{4.51b}$$

We also consider two subclasses of PMMs where only one of the constraints (4.51a)-(4.51b) is relaxed.

Definition 29. *Pairwise Markov models-F1 and pairwise Markov models-F2*

PMM ($\mathrm{R}_{1:N}$,$\mathrm{Y}_{1:N}$) is called PMM-F1 (PMM-F2) if it verifies, for each n in $\{1:N-1\}$, (4.51b) $\big($(4.51a)$\big)$ respectively.

Figure 4.11 presents dependency graphs of PMM-F1 and PMM-F2. In practice, one should specify the families of distributions to which $p\left(\mathrm{y}_n\,|\mathrm{r}_n\right)$, $p\left(\mathrm{r}_{n+1}\,|\mathrm{r}_n,\mathrm{y}_n\right)$ and $p\left(\mathrm{y}_{n+1}\,|\mathrm{r}_n,\mathrm{r}_{n+1},\mathrm{y}_n\right)$ belong to.

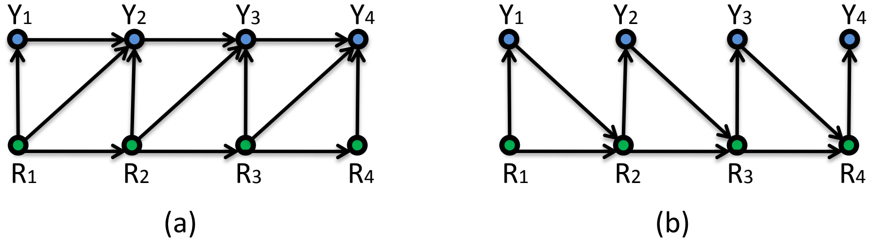


Figure 4.11: Dependency graphs of PMM-F1 (a) and PMM-F2 (b).

Let us introduce a pairwise Markov modeling of asset log-returns. Specifically, we explain how the PMMs allow modeling features (F1) and (F2). We also outline various types of PMM data processing, such as the state estimation, forecasting and parameter inference.

Let S_n be the stock price at time n, $n \in \mathbb{N}$. The log-return Y_n at time $n > 0$ is defined by

$$\mathrm{Y}_n = \log(S_n) - \log(S_{n-1}). \tag{4.52}$$

In the classic Black-Scholes model, the log-returns $\mathrm{Y}_{1:N}$ are assumed to be normally distributed and to have the same mean μ and standard deviation σ. In other words, we have, for each $n > 0$,

$$\mathrm{Y}_n = \mu + \sigma \mathrm{U}_n,$$

where $\{\mathrm{U}_n\}_{n>0}$ are zero-mean, unit-variance independent Gaussian random variables, also known as the standard Gaussian white noise. μ and σ are known as the average return (or drift) and the volatility of the stock.

The HMM allows extending the classic Black-Scholes model by making μ and σ dependent on hidden variables. Let $\mathrm{R}_{1:N}$ be a Markov chain, then let

$$\mathrm{Y}_n = \mu(\mathrm{r}_n) + \sigma(\mathrm{r}_n)\mathrm{U}_n, \tag{4.53}$$

with $\{\mathrm{U}_n\}_{1\leq n\leq N}$ standard Gaussian white noise variables. The parameters of this model include the initial state distribution, Markov chain transition matrix $p\left(\mathrm{r}_{n+1} = \omega'\,|\mathrm{r}_n = \omega\right)$ for each $\omega, \omega' \in \Omega$ and the values of the drift and volatility per state $\{\mu(\omega), \sigma(\omega)\}_{\omega\in\Omega}$. For example, if ω_1 is associated with the *bear* market state and ω_2 with the *bull* state, one would expect $\mu(\omega_1) < 0 < \mu(\omega_2)$ and $\sigma(\omega_1) > \sigma(\omega_2)$. The Hidden Markov modeling of $\mathrm{Y}_{1:N}$ is given by (P1)-(P3) and

$$\forall n, 1 \leq n \leq N, p\left(\mathrm{y}_n \,|\mathrm{r}_n\right) = \mathcal{N}\left(\mathrm{y}_n; \mu(\mathrm{r}_n), \sigma(\mathrm{r}_n)^2\right). \tag{4.54}$$

The PMMs provide a more flexible framework than that of HMMs. In order to fulfill the requirement (F1), we define a first-order autoregressive model of $\mathrm{Y}_{1:N}$ given $\mathrm{R}_{1:N}$. We set

$$\mathrm{U}_{n+1} = \rho(\mathrm{R}_n, \mathrm{R}_{n+1})\mathrm{U}_n + \sqrt{1 - \rho(\mathrm{R}_n, \mathrm{R}_{n+1})^2}\mathrm{V}_{n+1}, \tag{4.55}$$

where $n > 0$, $\mathrm{U}_1, \{\mathrm{V}_n\}_{n>0}$ are standard Gaussian white noise variables and for each $\omega, \omega' \in \Omega, |\rho(\omega, \omega')| < 1$.

As regards the feature (F2), we make R_{n+1} dependent on Y_n given R_n by using the concept of the logistic function. Specifically, in the case where Ω contains only two elements $\{\omega_1, \omega_2\}$, we set

$$p\left(\mathrm{r}_{n+1} = \omega_1 \,|\mathrm{r}_n, \mathrm{u}_n\right) = \frac{1}{1 + e^{-a(\mathrm{r}_n) - b(\mathrm{r}_n)\mathrm{u}_n}}, \tag{4.56}$$

where for each $\omega \in \Omega$, $a(\omega) \in \mathbb{R}, b(\omega) \in \mathbb{R}$.

Finally, we combine (4.53), (4.55) and (4.56) to define a pairwise Markov modeling of $\mathrm{Y}_{1:N}$:

$$p\left(\mathrm{y}_1 \,|\mathrm{r}_1\right) = \mathcal{N}\Big(\mathrm{y}_1; \mu(\mathrm{r}_1), \sigma^2(\mathrm{r}_1)\Big); \tag{4.57a}$$

$$p\left(\mathrm{r}_{n+1} = \omega_1 \,|\mathrm{r}_n, \mathrm{y}_n\right) = \frac{1}{1 + e^{-a(\mathrm{r}_n) - \frac{b(\mathrm{r}_n)}{\sigma(\mathrm{r}_n)}(\mathrm{y}_n - \mu(\mathrm{r}_n))}}; \tag{4.57b}$$

$$p\left(\mathrm{y}_{n+1} \,|\mathrm{r}_n, \mathrm{r}_{n+1}, \mathrm{y}_n\right) = \mathcal{N}\Big(\mathrm{y}_{n+1}; \mu(\mathrm{r}_{n+1}) + \frac{\rho(\mathrm{r}_n, \mathrm{r}_{n+1})\sigma(\mathrm{r}_{n+1})}{\sigma(\mathrm{r}_n)}\left(\mathrm{y}_n - \mu(\mathrm{r}_n)\right), \sigma(\mathrm{r}_{n+1})^2(1 - \rho(\mathrm{r}_n, \mathrm{r}_{n+1})^2)\Big). \tag{4.57c}$$

The parameters of this model are

$$\boldsymbol{\theta} = \{\pi(\omega), \mu(\omega), \sigma(\omega), a(\omega), b(\omega), \rho(\omega, \omega')\}_{\omega, \omega' \in \Omega}, \tag{4.58}$$

where $\pi(\omega) = \mathbb{P}\left[\mathrm{R}_n = \omega\right]$ for each $\omega \in \Omega$. This model is presented for $\Omega = \{\omega_1, \omega_2\}$, but one can consider a more general definition by using the multinomial logistic function, as explained in [Böhning, 1992].

Processing of incoming data $\{\mathrm{Y}_n\}_{n>0}$ in a PMM involves determining $p\left(\mathrm{r}_n \,|\mathrm{y}_{1:n}\right)$. The filtering distribution is given by

$$p\left(\mathrm{r}_n \,|\mathrm{y}_{1:n}\right) = \frac{\alpha_n(\mathrm{r}_n)}{\sum\limits_{\mathrm{r}_n \in \Omega} \alpha_n(\mathrm{r}_n)}, \tag{4.59}$$

where for all n in $\mathbb{N}$ and r_n in Ω, $\alpha_n(\mathrm{r}_n)$ is computed as detailed in Section 4.2.

Forecasting consists in computing $p\left(\mathrm{y}_{n+1:n+p} \,|\mathrm{y}_{1:n}\right)$ for $p > 0$. An important case of forecasting is the one-step-ahead forecasting, for which $p = 1$. In this case, it is also particularly important to forecast Z_{n+1}, where

$$\mathrm{Z}_{n+1} = \begin{cases} 1 & \text{if } \mathrm{Y}_{n+1} < 0; \\ 2 & \text{otherwise.} \end{cases} \tag{4.60}$$

Z_{n+1} represents the direction of the stock price change during the day $n + 1$. The anticipated price change at $n + 1$ given the information available at n is defined by

$$\widehat{z}_{n+1|n} = \begin{cases} 1 & \text{if } \mathbb{P}\left[\mathrm{Y}_{n+1} < 0 | \mathrm{y}_{1:n}\right] > 0.5; \\ 2 & \text{otherwise.} \end{cases} \tag{4.61}$$

Algorithm 2. *One-step-ahead forecasting in PMMs*

Let $n > 0$,

- Compute $p\left(\mathrm{r}_n \,|\mathrm{y}_{1:n}\right)$ *cf.* (4.59);
- Compute $p\left(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathrm{y}_{1:n}\right)$:

$$p\left(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathrm{y}_{1:n}\right) = p\left(\mathrm{r}_n \,|\mathrm{y}_{1:n}\right) p\left(\mathrm{r}_{n+1} \,|\mathrm{r}_n, \mathrm{y}_n\right);$$

- Compute, for each r_n, r_{n+1} in Ω, $\widehat{m}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})$ and $\widehat{s^2}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})$:

$$\begin{aligned}\widehat{m}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) &= \mu(\mathrm{r}_{n+1}) + \frac{\rho(\mathrm{r}_n, \mathrm{r}_{n+1})\sigma(\mathrm{r}_{n+1})}{\sigma(\mathrm{r}_n)}\left(\mathrm{y}_n - \mu(\mathrm{r}_n)\right)\\ \widehat{s^2}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) &= (1 - \rho(\mathrm{r}_n, \mathrm{r}_{n+1})^2)\sigma^2(\mathrm{r}_{n+1});\end{aligned}$$

- The predictive distribution $p\left(\mathrm{y}_{n+1} \,|\mathrm{y}_{1:n}\right)$ is a mixture of normal densities:

$$p\left(\mathrm{y}_{n+1} \,|\mathrm{y}_{1:n}\right) = \sum_{\mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathrm{y}_{1:n}\right) \mathcal{N}\left(\mathrm{y}_{n+1}; \widehat{m}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}), \widehat{s^2}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})\right)$$

  Compute the one-step-ahead forecast

$$\widehat{\mathrm{y}}_{n+1|n} \stackrel{\text{def}}{=\!=} \mathbb{E}\left[\mathrm{Y}_{n+1} \,|\mathrm{y}_{1:n}\right]$$

  as the mean of the mixture, that is

$$\widehat{\mathrm{y}}_{n+1|n} = \sum_{\mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathrm{y}_{1:n}\right) \widehat{m}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}); \tag{4.62}$$

- Let Φ denote the normal cumulative distribution function, compute $p\left(\mathrm{y}_{n+1} < 0 \,|\mathrm{y}_{1:n}\right)$ by

$$\sum_{\mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathrm{y}_{1:n}\right) \times \Phi\left(-\frac{\widehat{m}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})}{\widehat{s}_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})}\right).$$

(The algorithm ends here)

Contrary to the one-step-ahead forecasting, there is no apparent closed-form expression for $p\left(\mathrm{y}_{n+1:n+p} \,|\mathrm{y}_{1:n}\right)$ in the case of multistep forecasting in PMMs.

Let $N > 0$, $\mathrm{Y}_{1:N}$ be an observed time series of log-returns. The next step is the PMM parameter estimation, whose goal is to infer the parameter vector $\boldsymbol{\theta}$ (4.58) from the observed data $\mathrm{Y}_{1:N}$.

The Expectation-Maximization (EM) and the Iterative Conditional Estimation (ICE) are well-known parameter estimation algorithms. These algorithms are well suited for both HMMs and PMMs, and the details may be found in [Derrode and Pieczynski, 2004].

Alternatively, $\boldsymbol{\theta}$ can be estimated by using the principle of Empirical Risk Minimization (ERM). Several methods for proving consistency of such estimators are provided in *e.g.* [Lugosi and Zeger, 1995]. Let us recall the general idea of the ERM. Assume a training set $(\mathrm{x}_{1:N}, \mathrm{y}_{1:N})$ in $(\mathscr{X} \times \mathscr{Y})^N$, a prediction function $h : \mathscr{X} \to \mathscr{Y}$ and a loss function $L : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}_+$. The empirical risk associated with the prediction function h is defined as

$$\widehat{R}(h) = \frac{1}{N}\sum_{n=1}^{N} L(h(\mathrm{x}_n), \mathrm{y}_n).$$

Thus, the idea of the ERM is to find a function h for which the risk is minimal.

Regarding the context of forecasting, we have $\mathrm{x}_n = \mathrm{y}_{1:n}$ and $h(\mathrm{x}_n) = \widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n})$, where $\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n})$ is computed from $\boldsymbol{\theta}$ and $\mathrm{y}_{1:n}$ by (4.62). We consider the following loss functions:

$$L_1(\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}), \mathrm{y}_{n+1}) = |\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}) - \mathrm{y}_{n+1}|,$$
$$L_2(\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}), \mathrm{y}_{n+1}) = (\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}) - \mathrm{y}_{n+1})^2.$$

The associated risk functions are

$$\widehat{R}_1(\boldsymbol{\theta}) = \frac{1}{N-1}\sum_{n=1}^{N-1} |\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}) - \mathrm{y}_{n+1}|, \tag{4.64a}$$

$$\widehat{R}_2(\boldsymbol{\theta}) = \frac{1}{N-1}\sum_{n=1}^{N-1} (\widehat{\mathrm{y}}^{\boldsymbol{\theta}}_{n+1|n}(\mathrm{y}_{1:n}) - \mathrm{y}_{n+1})^2. \tag{4.64b}$$

Let $\lambda > 0$, the following risk function realizes a trade-off between $\widehat{R}_1(\boldsymbol{\theta})$ and $\widehat{R}_2(\boldsymbol{\theta})$:

$$\widehat{R}(\boldsymbol{\theta}; \lambda) = \lambda \widehat{R}_1(\boldsymbol{\theta}) + \widehat{R}_2(\boldsymbol{\theta}). \tag{4.65}$$

We estimate $\boldsymbol{\theta}$ by minimizing (4.65) for various values of λ. There is no closed expression known for the corresponding update equations and we solve the optimization problem by the Particle Swarm Optimization (PSO). PSO methods [Poli et al., 2007] are non-convex global optimization algorithms.

Let us present our methodology to compare the efficiency of PMM with that of HMM on historical stock quotes. Given a data set $\mathscr{H} = \{\mathrm{y}_1, .., \mathrm{y}_M\}$ with successive daily log-returns of an asset $\mathscr{E}$, we split $\mathscr{H}$ into two juxtaposed sets as follows: $\mathscr{H}_{\text{training}} = \{\mathrm{y}_1, .., \mathrm{y}_N\}$ and $\mathscr{H}_{\text{test}} = \{\mathrm{y}_{N+1}, .., \mathrm{y}_M\}$. The first set is used to estimate the parameter $\boldsymbol{\theta}$ by minimizing (4.65) for a given λ, while the second set only serves to assess the efficiency of each model considered. The models are compared in terms of the outcome produced by the following trading system. At the beginning of each day $n+1$, $N \leq n < M$, the system buys asset $\mathscr{E}$ only if the one-day-ahead forecast (4.61) produced by the model is positive, *i.e.* if $\widehat{z}_{n+1|n} = 2$, and sells the asset at the end of the day. In the case of a negative forecast, the system avoids any trading operations on $\mathscr{E}$. Next, we compute the absolute return of the system on $\mathscr{H}_{\text{test}}$ and compare it with that of the asset. Let us recall that the absolute return of $\mathscr{E}$ relative to date N is defined as

$$\tau(n; N) = \frac{S_n - S_N}{S_N}, \tag{4.66}$$

for $n \geq N$. Equivalently, $\tau(n; N)$ can be written as a function of the log-returns:

$$\tau(n; N) = \exp\left(\sum_{t=N+1}^{n} \mathrm{y}_t\right) - 1.$$

Thus, the absolute return of the trading system considered can be written as

$$\tau^*(n; N) = \exp\left(\sum_{t=N}^{n-1} \mathrm{y}_{t+1}\delta(\widehat{z}_{t+1|t} = 2)\right) - 1. \tag{4.67}$$

We apply this methodology to Cliffs Natural Resources Stock prices (NYSE:CLF). Stock quotes are taken from the Yahoo! database and correspond to the business days from 01/02/1990 to 12/13/1993 for $\mathscr{H}_{\text{training}}$ and from 12/14/1993 to 09/29/1994 for $\mathscr{H}_{\text{test}}$. In this configuration, the size of $\mathscr{H}_{\text{training}}$ is $N = 1000$, the size of $\mathscr{H}_{\text{test}}$ is 200 and the total size of the data set $\mathscr{H}$ is $M = 1200$. In every experiment, the state space consists of only two elements. Figures 4.12 and 4.13 display the values of risks $\widehat{R}_1(\boldsymbol{\theta})$ and $\widehat{R}_2(\boldsymbol{\theta})$ *cf.* (4.64) for $\boldsymbol{\theta}$ minimizing (4.65), in function of λ. Absolute returns generated by four models on the test set are given in Table 4.9 for various values of λ. Figure 4.14 displays the returns produced per each model in function of time with $\lambda = 0$.

	$\lambda = 10^{-3}$	$\lambda = 10^{-2}$	$\lambda = 1$	$\lambda = 10^{2}$	$\lambda = 10^{3}$
HMM	17%	13%	10%	10%	10%
PMM-F1	16%	14%	11%	9%	9%
PMM-F2	21%	20%	19%	14%	16%
PMM	21%	20%	19%	14%	16%

Table 4.9: Absolute returns (4.67) of HMM, PMM-F1, PMM-F2 and PMM-based trading systems on NYSE:CLF historical prices. The returns are related to the period from 12/14/1993 to 09/29/1994.

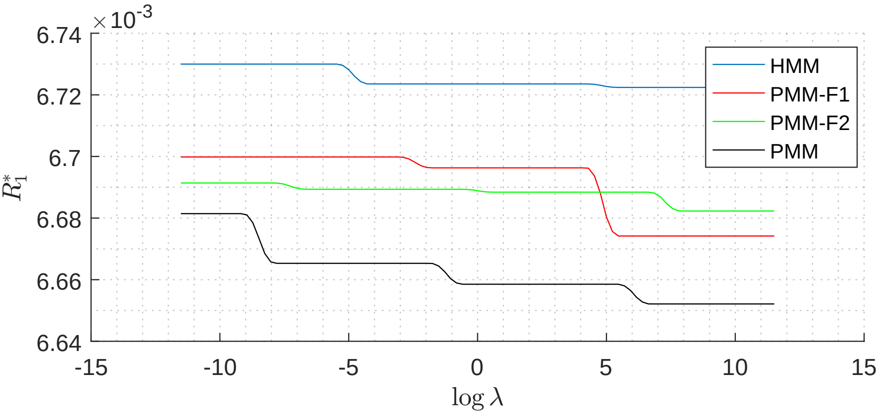


Figure 4.12: Values $\widehat{R}_1^*(\lambda) = \widehat{R}_1(\boldsymbol{\theta})$ in function of λ, where $\boldsymbol{\theta}$ minimizes (4.65).

Let us make several brief observations.

Figures 4.12 and 4.13 are consistent with the definition of $\boldsymbol{\theta}$ as the minimum of (4.65). When λ increases, $\widehat{R}_1^*(\lambda) = \widehat{R}_1(\boldsymbol{\theta})$ decreases and $\widehat{R}_2^*(\lambda) = \widehat{R}_2(\boldsymbol{\theta})$ increases, and vice versa, and this holds for the four models.

Progressive inclusion of features (F1) and (F2) in the HMM improves both risk values computed on $\mathscr{H}_{\text{training}}$, as expected, independently of the value of λ.

We can see from Figure 4.14, that PMM-F1 implies a more risk-adverse trading strategy than that of HMM, and the related generated return increases almost monotonically. However, PMM-F1 may not be well suited for a *bull* market. PMM-F2 and HMM appear to be better suited for *bull* dynamics, while PMM-F2 seems to be less vulnerable than HMM to abrupt drops of asset value.

As a discussion, we proposed a meaningful parameterization of PMM for modeling financial time series. The results show that both features (F1) and (F2) can be captured by PMMs, which was expected. Another interesting point is that these features seem to be present in real-world data, and thus PMMs provide a better forecast. One can intuitively understand why using the feature (F1) should improve forecasting, while (F2) is more difficult to interpret. Suppose for example that during the *bull* state, the return Y_n appears to be excessively negative compared to the average return of the *bull* market. In this case, the current state may become fairly uncertain in an HMM. The PMM incorporates (F2) by using the distribution $p\left(\mathrm{r}_{n+1} \,|\mathrm{r}_n, \mathrm{y}_n\right)$ which allows to decide to which extent Y_n should affect the expectation of R_{n+1}.

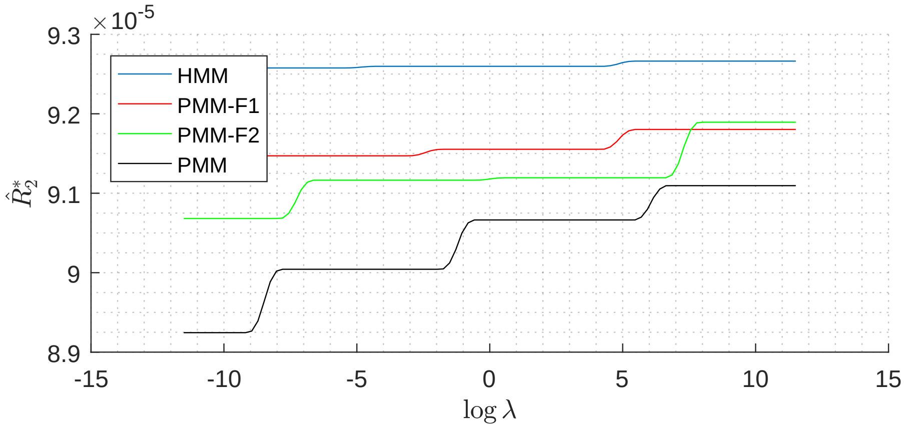


Figure 4.13: Values $\widehat{R}_2^*(\lambda) = \widehat{R}_2(\boldsymbol{\theta})$ in function of λ, where $\boldsymbol{\theta}$ minimizes (4.65).

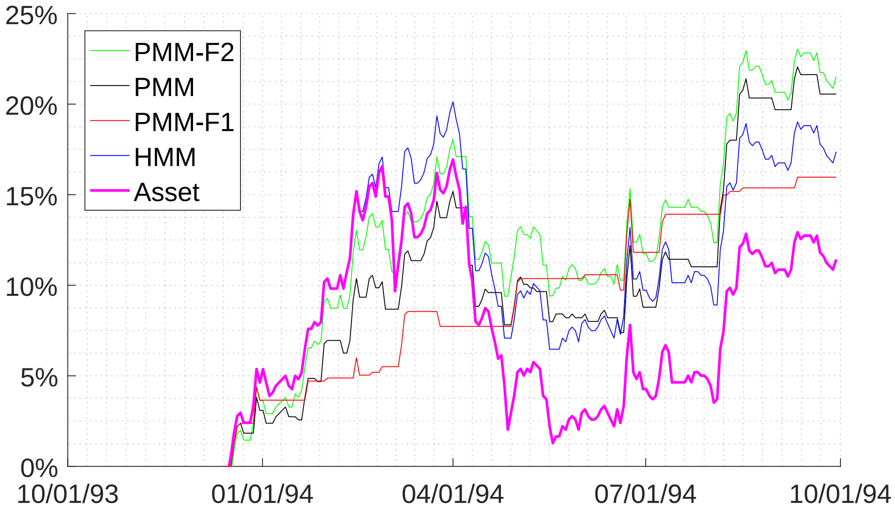


Figure 4.14: Absolute returns (4.67) from 12/14/1993 generated by PMM-based trading systems on NYSE:CLF historical data. PMM models are estimated on the data from 01/02/1990 to 12/13/1993 by minimizing (4.65) with $\lambda = 0$. Four charts (from top to bottom) relate to the four models. The last chart is the absolute return of the asset (4.66).

Table 4.9 indicates that the outcome produced by each model is sensitive to the value of λ. In general, such a parameter should be chosen by a cross-validation procedure accordingly to the application considered.

Our experiments indicate that a more complex structure of PMMs may allow identifying better suited regimes for specific application. We believe that the presented way of use of the flexibility of PMM will allow overcoming principal constraints of HMMs.

This study has several limitations. Firstly, we assume only two regimes in our models. Next, the Gaussian mixture density and non-Gaussian heavy tailed observation distributions could be considered as well. We only consider closing price per day, while daily opening, low and high prices are also available as well. Finally, our study concerns only one period of stock prices and only one stock was used in the experiment.

4.5 Conclusion

We compared the accuracy of MPM estimators based on the classic HMM and its extensions which are the PMM and the TMM. PMM and TMM frameworks allowed to achieve substantial improvements of the estimation accuracy. Such improvements were particularly visible when the observation distribution was heavily autocorrelated and/or when the hidden chain was far from being Markovian.

We also introduced a pairwise Markov model of financial time series, obtained by incorporating features such as the correlation of log-returns given the state variables and dependence of the future state upon the current log-return given the current state. The results show that both of these features contribute to improving the performance of the model in applications related to stock forecasting.

Chapter 5

Bayesian state estimation in partially observed Markov processes with hybrid state space

This chapter is devoted to the Bayesian inference in partially observed Markov processes (POMPs) with hybrid state space. By Bayesian inference we mean estimating the filtering and smoothing distributions. The Conditionally Gaussian Linear State-Space Model (CGLSSM)[Cappé et al., 2005] is an important model which belongs to the class of Partially Observable Markov Process (POMP)s with hybrid state space. This model is also known as the Switching Linear Dynamical System (SLDS) [Costa et al., 2006].

The two sections of the chapter are devoted to the corresponding contributions of the author. The first one presents a novel Bayesian inference algorithm for the SLDS, and more generally, for the Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM) [Abbassi et al., 2015]. The second one presents a novel algorithm for filtering in switching systems, with an emphasis that these systems may be non-linear and/or non-Gaussian.

The section is mainly a compilation of authors' papers [Gorynin and Pieczynski, 2017a, Gorynin et al., 2016c, Gorynin and Pieczynski, 2017b].

5.1 Bayesian smoothing in conditionally linear POMPs with hybrid state space

The concept of the SLDS [Costa et al., 2006] is presented in different fields, such as econometrics [Kim, 1994], finance [Azzouzi and Nabney, 1999], tracking [Weiss et al., 2004], speech recognition [Mesot and Barber, 2007], pattern recognition [Pavlovic et al., 2001], among others [Ristic et al., 2004]. These systems are also known as jump Markov models (processes), switching conditional linear Gaussian state-space models, interacting multiple models. There is no exact Bayesian filtering or smoothing algorithm tractable in the general SLDS context [Lerner, 2002]. Previous research on smoothed inference in SLDSs includes the most popular Kim method [Kim and Nelson, 1999], simulation-based algorithms [Doucet et al., 2001, Fong et al., 2002, Särkkä et al., 2012, Carter and Kohn, 1996], recent smoothed inference by expectation correction [Barber, 2006] and various deterministic approximations [Zoeter and Heskes, 2006]. Simulation-based methods intrinsically use Monte-Carlo integration in the state space. Thus, the accuracy of such approaches depends on the number of simulated particles. Besides, if the number of simulated particles is insufficient for the state space dimension, these estimators would have high variance, while achieving an acceptable variance would mean for them a high processing cost. Indeed, it is possible to bypass the need of numerical integration by assuming a conditional independence [Kim, 1994] and the effect induced by such assumption is insignificant [Barber, 2006]. We also note the Rao-Blackwellised particle filters [Murphy and Russell, 2001] which are designed to replace the problem of sampling in continuous state space by an explicit integration [Barber, 2006]. The algorithm is illustrated through an application to the problem of trend estimation.

In this section, we introduce an approach of fast smoothing in the Stationary Conditionally Gaussian Pairwise Markov Switching Model (SCGPMSM) [Abbassi et al., 2015]. The interest of the new method is that it uses Bayesian assimilation to obtain a smoothed estimate so the forward and backward passes can run independently. The main idea is to use the classic Switching Kalman Filter (SKF) twice: firstly, as usual, and a second time applied to time-reversed dynamics of the system. Then our smoothed solution is obtained by using standard Gaussian conditioning formulas to combine the two distributions computed by the SKF. We discovered that the resulting algorithm performs as well as the particle smoother both in terms of the mean squared error and regime misclassification rate. It also allowed substantial gains in processing cost when compared to the particle smoother. The main results are presented in the SCGPMSM framework rather than in that of the classic SLDS. Indeed, formally, SCGPMSMs are switching linear models which extend the classic SLDSs (see Figure 5.1 and Figure 5.2). Our decision to use the SCGPMSM framework is exclusively motivated by its suitability for presentation of our algorithms and its potential to enhance them with a greater degree of generality. We first present the SLDSs and SCGPMSMs, as well as the SKF. Next, we describe the novel Reverse Switching Kalman Filter (RSKF) and the proposed Bayesian assimilation of estimates of SKF and RSKF.

Let $\Omega = \{1 : K\}$, in an SLDS, we have:

$$p\left(\mathbf{x}_1 \left| \mathrm{r}_1 \right.\right) = \mathscr{N}\left(\mathbf{x}_1; \boldsymbol{m}_1(\mathrm{r}_1), \boldsymbol{\Sigma}_1(\mathrm{r}_1)\right); \tag{5.1a}$$

$$\forall n \in \{1 : (N-1)\}, \mathbf{X}_{n+1} = \boldsymbol{T}_{n+1}(\mathrm{R}_{n+1})\mathbf{X}_n + \mathbf{a}_{n+1}(\mathrm{R}_{n+1}) + \boldsymbol{Q}_{n+1}(\mathrm{R}_{n+1})\mathbf{U}_{n+1}; \tag{5.1b}$$

$$\forall n \in \{1 : N\}, \mathbf{Y}_n = \boldsymbol{H}_n(\mathrm{R}_n)\mathbf{X}_n + \boldsymbol{b}_n(\mathrm{R}_n) + \boldsymbol{S}_n(\mathrm{R}_n)\mathbf{V}_n; \tag{5.1c}$$

$$\forall n \in \{1 : (N-1)\}, \forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega, p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{x}_n, \mathbf{y}_n \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n \right.\right). \tag{5.1d}$$

Here, for each n in $\{1 : N\}$, the value of R_n determines the data generating process used to create $(\mathbf{X}_n, \mathbf{Y}_n)$. The dependency graph of SLDSs is given in Figure 5.1.

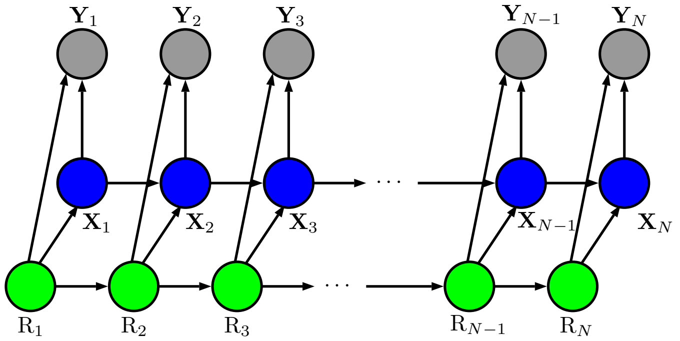


Figure 5.1: Dependency graph of classic SLDSs (5.1).

If at time $n+1$, the value of R_{n+1} is different from that of R_n, we say that the system has switched at $n+1$.

It is also noteworthy that in an SLDS, $((\mathrm{R}_n, \mathbf{X}_n), \mathbf{Y}_n)_{1 \leq n \leq N}$ is a hidden Markov chain with $\mathbf{Y}_{1:N}$ observed. Thus, SLDSs can be seen as hidden Markov models with hybrid state space: continuous-valued $\mathbf{X}_{1:N}$ and discrete-valued $\mathrm{R}_{1:N}$. To summarize, in an SLDS, $\mathrm{R}_{1:N}$, $(\mathrm{R}_n, \mathbf{X}_n)_{1 \leq n \leq N}$ and $(\mathrm{R}_n, \mathbf{X}_n, \mathbf{Y}_n)_{1 \leq n \leq N}$ are Markov processes.

It is noticed [Abbassi et al., 2015] that in general, system (5.1) is not stationary, but may be asymptotically stationary. In this case, the stationary asymptote of (5.1) is of form (5.2).

We consider SCGPMSMs [Abbassi et al., 2015], where $(\mathrm{R}_n, \mathbf{X}_n, \mathbf{Y}_n)_{1 \leq n \leq N}$ and $\mathrm{R}_{1:N}$ are stationary Markovian and

$$\forall n \in \{1:(N-1)\}, p\left(\mathbf{z}_n, \mathbf{z}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1} \right.\right) = \\ \mathscr{N}\left(\begin{bmatrix} \mathbf{z}_n \\ \mathbf{z}_{n+1} \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n) \\ \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n+1}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n) & \boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_n, \mathrm{r}_{n+1}) \\ \boldsymbol{\Gamma}_{\mathbf{Z}_2\mathbf{Z}_1}(\mathrm{r}_n, \mathrm{r}_{n+1}) & \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{n+1}) \end{bmatrix}\right), \tag{5.2}$$

with

$$\forall n \in \{1:N\}, \mathbf{Z}_n = \begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix}.$$

The direct dynamics of SCGPMSM are defined as

$$\forall n \in \{1:(N-1)\}, \forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega,$$

$$\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{\Gamma}_{\mathbf{Z}_2\mathbf{Z}_1}(\mathrm{r}_{n+1}, \mathrm{r}_n)\boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n)^{-1}; \tag{5.3a}$$

$$\boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n+1}) - \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n); \tag{5.3b}$$

$$\boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{n+1}) - \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})\boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_n, \mathrm{r}_{n+1}). \tag{5.3c}$$

The reversal dynamics of SCGPMSM are defined as

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega,$$

$$\boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) = \boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_{n-1}, \mathrm{r}_n)\boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n)^{-1}; \tag{5.4a}$$

$$\boldsymbol{L}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) = \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n-1}) - \boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n)\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n); \tag{5.4b}$$

$$\boldsymbol{Q}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) = \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{n-1}) - \boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n)\boldsymbol{\Gamma}_{\mathbf{Z}_2\mathbf{Z}_1}(\mathrm{r}_n, \mathrm{r}_{n-1}). \tag{5.4c}$$

SCGPMSMs include stationary SLDSs (5.1) and allso allows incorparating complementary conditional dependencies. Their dependency graph is given in Figure 5.2.

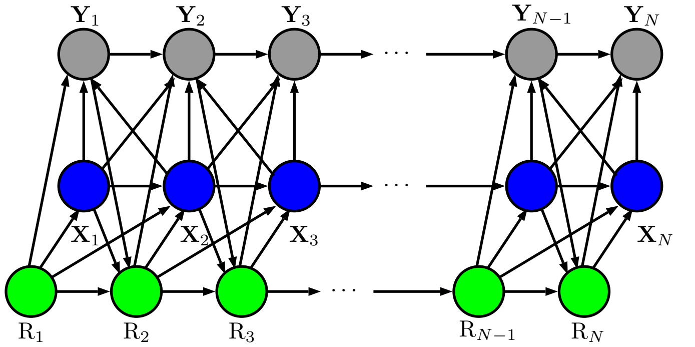


Figure 5.2: Dependency graph of SCGPMSMs (5.2).

The direct dynamics (5.3) of an SCGPMSM are such that

$$\forall n \in \{1:(N-1)\}, p\left(\mathbf{z}_{n+1} \left| \mathbf{z}_n, \mathrm{r}_n, \mathrm{r}_{n+1} \right.\right) = \\ \mathscr{N}\left(\mathbf{z}_{n+1}; \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})\mathbf{z}_n + \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}), \boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1})\right). \tag{5.5}$$

These dynamics define the SCGPMSM, since

$$p\left(\mathrm{r}_{1:N}, \mathbf{z}_{1:N}\right) = p\left(\mathrm{r}_{1:N}\right) p\left(\mathbf{z}_{1:N} \left| \mathrm{r}_{1:N} \right.\right); \\ p\left(\mathbf{z}_{1:N} \left| \mathrm{r}_{1:N} \right.\right) = p\left(\mathbf{z}_1 \left| \mathrm{r}_{1:N} \right.\right) p\left(\mathbf{z}_2 \left| \mathrm{r}_{1:N}, \mathbf{z}_1 \right.\right) \dots p\left(\mathbf{z}_N \left| \mathrm{r}_{1:N}, \mathbf{z}_{N-1} \right.\right) = \\ p\left(\mathbf{z}_1 \left| \mathrm{r}_1 \right.\right) p\left(\mathbf{z}_2 \left| \mathrm{r}_1, \mathrm{r}_2, \mathbf{z}_1 \right.\right) \dots p\left(\mathbf{z}_N \left| \mathrm{r}_{N-1}, \mathrm{r}_N, \mathbf{z}_{N-1} \right.\right).$$

by Markovianity of $(\mathrm{R}_n, \mathbf{Z}_n)_{1 \leq n \leq N}$ and $p\left(\mathbf{z}_1 \left| \mathrm{r}_1 \right.\right) = \mathscr{N}\left(\mathbf{z}_1; \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_1), \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_1)\right)$.

Indeed, the same SCGPMSM can be also defined by using the reversal dynamics, since

$$\begin{aligned}&p\left(\mathbf{z}_{1:N}\left|\mathrm{r}_{1:N}\right.\right)=p\left(\mathbf{z}_{N}\left|\mathrm{r}_{1:N}\right.\right)p\left(\mathbf{z}_{N-1}\left|\mathrm{r}_{1:N},\mathbf{z}_{N}\right.\right)\dots p\left(\mathbf{z}_{1}\left|\mathrm{r}_{1:N},\mathbf{z}_{2}\right.\right)=\\&\quad p\left(\mathbf{z}_{N}\left|\mathrm{r}_{N}\right.\right)p\left(\mathbf{z}_{N-1}\left|\mathrm{r}_{N-1},\mathrm{r}_{N},\mathbf{z}_{N}\right.\right)\dots p\left(\mathbf{z}_{1}\left|\mathrm{r}_{2},\mathrm{r}_{1},\mathbf{z}_{2}\right.\right);\\&p\left(\mathbf{z}_{N}\left|\mathrm{r}_{N}\right.\right)=\mathcal{N}\left(\mathbf{z}_{N};\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{N}),\boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{N})\right),\end{aligned}$$

and the reversal dynamics (5.4) are such that

$$\begin{aligned}&\forall n\in\{2:N\},p\left(\mathbf{z}_{n-1}\left|\mathbf{z}_{n},\mathrm{r}_{n-1},\mathrm{r}_{n}\right.\right)=\\&\quad\mathcal{N}\left(\mathbf{z}_{n-1};\boldsymbol{F}^{*}(\mathrm{r}_{n-1},\mathrm{r}_{n})\mathbf{z}_{n+1}+\boldsymbol{L}^{*}(\mathrm{r}_{n-1},\mathrm{r}_{n}),\boldsymbol{Q}^{*}(\mathrm{r}_{n-1},\mathrm{r}_{n})\right).\end{aligned}\tag{5.6}$$

The reversal dynamics are used in the smoothed inference of SCGPMSM, specifically in the backward pass.

5.1.1 Approximate Bayesian state estimation

Let us consider an SCGPMSM and let $A \xlongequal[\text{approx.}]{\text{def}} B$ mean that A is computed in a way to approximate the value of B. The SKF allows to compute:

$$\forall n\in\{1:N\},\forall\mathrm{r}_{n}\in\Omega,\qquad \pi_{n}(\mathrm{r}_{n})\xlongequal[\text{approx.}]{\text{def}}p\left(\mathrm{r}_{n}\left|\mathbf{y}_{1:n}\right.\right);\tag{5.7a}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \pi_{n+1}(\mathrm{r}_{n},\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}p\left(\mathrm{r}_{n},\mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right);\tag{5.7b}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \pi_{n+1}(\mathrm{r}_{n}|\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}p\left(\mathrm{r}_{n}\left|\mathrm{r}_{n+1},\mathbf{y}_{1:n+1}\right.\right);\tag{5.7c}$$

$$\forall n\in\{1:N\},\forall\mathrm{r}_{n}\in\Omega,\qquad \widehat{\mathbf{x}}_{n|n}(\mathrm{r}_{n})\xlongequal[\text{approx.}]{\text{def}}\mathbb{E}\left[\mathbf{X}_{n}\left|\mathbf{y}_{1:n},\mathrm{r}_{n}\right.\right];\tag{5.7d}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_{n},\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}\mathbb{E}\left[\mathbf{X}_{n+1}\left|\mathbf{y}_{1:n},\mathrm{r}_{n},\mathrm{r}_{n+1}\right.\right];\tag{5.7e}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \widehat{\mathbf{z}}_{n+1|n}(\mathrm{r}_{n},\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}\mathbb{E}\left[\mathbf{Z}_{n+1}\left|\mathbf{y}_{1:n},\mathrm{r}_{n},\mathrm{r}_{n+1}\right.\right];\tag{5.7f}$$

$$\forall n\in\{1:N\},\forall\mathrm{r}_{n}\in\Omega,\qquad \widehat{\mathbf{\Sigma}}_{n|n}(\mathrm{r}_{n})\xlongequal[\text{approx.}]{\text{def}}\mathrm{Var}\left[\mathbf{X}_{n}\left|\mathbf{y}_{1:n},\mathrm{r}_{n}\right.\right];\tag{5.7g}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \widehat{\mathbf{\Sigma}}_{n+1|n}(\mathrm{r}_{n},\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}\mathrm{Var}\left[\mathbf{X}_{n+1}\left|\mathbf{y}_{1:n},\mathrm{r}_{n},\mathrm{r}_{n+1}\right.\right];\tag{5.7h}$$

$$\forall n\in\{1:(N-1)\},\forall\mathrm{r}_{n},\mathrm{r}_{n+1}\in\Omega,\qquad \widehat{\mathbf{\Gamma}}_{n+1|n}(\mathrm{r}_{n},\mathrm{r}_{n+1})\xlongequal[\text{approx.}]{\text{def}}\mathrm{Var}\left[\mathbf{Z}_{n+1}\left|\mathbf{y}_{1:n},\mathrm{r}_{n},\mathrm{r}_{n+1}\right.\right].\tag{5.7i}$$

For each n in $\{1:N\}$, the SKF uses the following assumption

$$p\left(\mathbf{x}_{n}\left|\mathbf{y}_{1:n},\mathrm{r}_{n}\right.\right)=\mathcal{N}\left(\mathbf{x}_{n};\widehat{\mathbf{x}}_{n|n}(\mathrm{r}_{n}),\widehat{\mathbf{\Sigma}}_{n|n}(\mathrm{r}_{n})\right).$$

Indeed, the SKF was originally designed for SLDSs of form (5.1). Here we present a slightly enhanced version of the original SKF which is applicable to the SCGPMSM.

Algorithm 3. *Switching Kalman filter*

Initialization: for each r_1 in Ω,

$$\pi_1(\mathrm{r}_1) = \frac{p(\mathrm{r}_1)\,\mathcal{N}(\mathbf{y}_1; \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_1), \boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}_1))}{\sum\limits_{\mathrm{r}_1 \in \Omega} p(\mathrm{r}'_1)\,\mathcal{N}(\mathbf{y}_1; \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}'_1), \boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}'_1))};$$

$$\widehat{\mathbf{x}}_{1|1}(\mathrm{r}_1) = \boldsymbol{\mu}_{\mathbf{X}}(\mathrm{r}_1) + \boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_1)\boldsymbol{\Gamma}_{\mathbf{Y}}^{-1}(\mathrm{r}_1)(\mathbf{y}_1 - \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_1));$$

$$\widehat{\boldsymbol{\Sigma}}_{1|1}(\mathrm{r}_1) = \boldsymbol{\Gamma}_{\mathbf{X}}(\mathrm{r}_1) - \boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_1)\boldsymbol{\Gamma}_{\mathbf{Y}}^{-1}(\mathrm{r}_1)\boldsymbol{\Gamma}_{\mathbf{YX}}(\mathrm{r}_1),$$

where $\boldsymbol{\mu}_{\mathbf{X}}(\mathrm{r}_1) \in \mathbb{R}^d$, $\boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_1) \in \mathbb{R}^{d'}$, $\boldsymbol{\Gamma}_{\mathbf{X}}(\mathrm{r}_1) \in \mathbb{R}^{d\times d}$, $\boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_1) \in \mathbb{R}^{d\times d'}$, $\boldsymbol{\Gamma}_{\mathbf{YX}}(\mathrm{r}_1) \in \mathbb{R}^{d'\times d}$, $\boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}_1) \in \mathbb{R}^{d'\times d'}$ are defined from

$$\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_1) = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{X}}(\mathrm{r}_1) \\ \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_1) \end{bmatrix}, \qquad \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_1) = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathbf{X}}(\mathrm{r}_1) & \boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_1) \\ \boldsymbol{\Gamma}_{\mathbf{YX}}(\mathrm{r}_1) & \boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}_1) \end{bmatrix}.$$

Recursion: compute $\{\pi_{n+1}(\mathrm{r}_{n+1}), \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1}), \widehat{\boldsymbol{\Sigma}}_{n+1|n+1}(\mathrm{r}_{n+1})\}_{\mathrm{r}_{n+1}\in\Omega}$ from $\{\pi_n(\mathrm{r}_n), \widehat{\mathbf{x}}_{n|n}(\mathrm{r}_n), \widehat{\boldsymbol{\Sigma}}_{n|n}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ for each n in $\{1:(N-1)\}$. Let r_n,r_{n+1} in Ω,
a)*time update*:

$$\widehat{\mathbf{z}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) \begin{bmatrix} \widehat{\mathbf{x}}_{n|n}(\mathrm{r}_n) \\ \mathbf{y}_n \end{bmatrix} + \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \begin{bmatrix} \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) \\ \widehat{\mathbf{y}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) \end{bmatrix}; \tag{5.8a}$$

$$\widehat{\boldsymbol{\Gamma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}_{n|n}(\mathrm{r}_n) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) & \widehat{\boldsymbol{C}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) \\ \widehat{\boldsymbol{C}}_{n+1|n}^\top(\mathrm{r}_n, \mathrm{r}_{n+1}) & \widehat{\boldsymbol{S}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) \end{bmatrix}, \tag{5.8b}$$

where $\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})$, $\boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1})$, $\boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1})$ are given by (5.3).
b)*measurement update*

$$\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) + \widehat{\boldsymbol{C}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})\widehat{\boldsymbol{S}}_{n+1|n}^{-1}(\mathrm{r}_n, \mathrm{r}_{n+1})(\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})); \tag{5.9a}$$

$$\widehat{\boldsymbol{\Sigma}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \widehat{\boldsymbol{\Sigma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) - \widehat{\boldsymbol{C}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})\widehat{\boldsymbol{S}}_{n+1|n}^{-1}\widehat{\boldsymbol{C}}_{n+1|n}^\top(\mathrm{r}_n, \mathrm{r}_{n+1}); \tag{5.9b}$$

Next, let $c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} p(\mathbf{y}_{n+1}\,|\mathbf{y}_{1:n}, \mathrm{r}_n, \mathrm{r}_{n+1})$, we have

$$c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \mathcal{N}\left(\mathbf{y}_{n+1}; \widehat{\mathbf{y}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}), \widehat{\boldsymbol{S}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})\right); \tag{5.10}$$

— Update the posterior distribution of the discrete state:

$$\forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega, \pi_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \frac{\pi_n(\mathrm{r}_n)p(\mathrm{r}_{n+1}\,|\mathrm{r}_n)\,c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})}{\sum\limits_{\mathrm{r}'_n, \mathrm{r}'_{n+1}\in\Omega} \pi_n(\mathrm{r}'_n)p(\mathrm{r}'_{n+1}\,|\mathrm{r}'_n)\,c_{n+1}(\mathrm{r}'_n, \mathrm{r}'_{n+1})}; \tag{5.11a}$$

$$\forall \mathrm{r}_{n+1} \in \Omega, \pi_{n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n\in\Omega} \pi_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}). \tag{5.11b}$$

— Compute, for each r_{n+1} in Ω, $\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1})$ and $\widehat{\boldsymbol{\Sigma}}_{n+1|n+1}(\mathrm{r}_{n+1})$:

$$\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n\in\Omega} \pi_{n+1}(\mathrm{r}_n|\mathrm{r}_{n+1})\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}); \tag{5.12a}$$

$$\widehat{\boldsymbol{\Sigma}}_{n+1|n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n\in\Omega} \pi_{n+1}(\mathrm{r}_n|\mathrm{r}_{n+1})\widehat{\boldsymbol{\Sigma}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) + \sum_{\mathrm{r}_n\in\Omega} \pi_{n+1}(\mathrm{r}_n|\mathrm{r}_{n+1}) \times \left(\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) - \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1})\right)\left(\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) - \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1})\right)^\top, \tag{5.12b}$$

with

$$\pi_{n+1}(\mathrm{r}_n|\mathrm{r}_{n+1}) = \frac{\pi_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})}{\pi_{n+1}(\mathrm{r}_{n+1})}. \tag{5.13}$$

(The algorithm ends here)

Let us now introduce another method of smoothing in SCGPMSMs we propose. The main particularity of the new method is that it is based on Bayesian assimilation. We first introduce the reverse switching Kalman filter used to process $\mathbf{Y}_{1:N}$ in the reverse order by using the reversal dynamics.

By analogy with the SKF, let us define the RSKF that is used to compute:

$$\forall n \in \{1:N\}, \forall \mathrm{r}_n \in \Omega, \qquad \pi_n^*(\mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} p\left(\mathrm{r}_n \left| \mathbf{y}_{n:N} \right.\right); \tag{5.14a}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \qquad \pi_{n-1}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} p\left(\mathrm{r}_{n-1}, \mathrm{r}_n \left| \mathbf{y}_{n-1:N} \right.\right); \tag{5.14b}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \qquad \pi_{n-1}^*(\mathrm{r}_n|\mathrm{r}_{n-1}) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathbf{y}_{n-1:N} \right.\right); \tag{5.14c}$$

$$\forall n \in \{1:N\}, \forall \mathrm{r}_n \in \Omega, \qquad \widehat{\mathbf{x}}_{n|n}^*(\mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_{n:N}, \mathrm{r}_n \right.\right]; \tag{5.14d}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \quad \widehat{\mathbf{x}}_{n-1|n}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathbf{y}_{n:N}, \mathrm{r}_{n-1}, \mathrm{r}_n \right.\right]; \tag{5.14e}$$

$$\tag{5.14f}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \qquad \widehat{\mathbf{z}}_{n-1|n}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathbb{E}\left[\mathbf{Z}_{n-1} \left| \mathbf{y}_{n:N}, \mathrm{r}_{n-1}, \mathrm{r}_n \right.\right]; \tag{5.14g}$$

$$\forall n \in \{1:N\}, \forall \mathrm{r}_n \in \Omega, \qquad \widehat{\mathbf{\Sigma}}_{n|n}^*(\mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathrm{Var}\left[\mathbf{X}_n \left| \mathbf{y}_{n:N}, \mathrm{r}_n \right.\right]; \tag{5.14h}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \quad \widehat{\mathbf{\Sigma}}_{n-1|n}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathrm{Var}\left[\mathbf{X}_{n-1} \left| \mathbf{y}_{n:N}, \mathrm{r}_{n-1}, \mathrm{r}_n \right.\right]; \tag{5.14i}$$

$$\forall n \in \{2:N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \quad \widehat{\mathbf{\Gamma}}_{n-1|n}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} \mathrm{Var}\left[\mathbf{Z}_{n-1} \left| \mathbf{y}_{n:N}, \mathrm{r}_{n-1}, \mathrm{r}_n \right.\right]. \tag{5.14j}$$

under assumption that

$$\forall n \in \{1:N\}, p\left(\mathbf{x}_n \left| \mathbf{y}_{n:N}, \mathrm{r}_n \right.\right) = \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}^*(\mathrm{r}_n), \widehat{\mathbf{\Sigma}}_{n|n}^*(\mathrm{r}_n)\right). \tag{5.15}$$

The RSKF runs as follows:

Algorithm 4. *Reverse switching Kalman filter*

Initialization: for each r_N in Ω,

$$\pi_N^*(\mathrm{r}_N) = \frac{p\left(\mathrm{r}_N\right) \mathcal{N}\left(\mathbf{y}_N; \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_N), \boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}_N)\right)}{\sum\limits_{\mathrm{r}'_N=1}^{K} p\left(\mathrm{r}'_N\right) \mathcal{N}\left(\mathbf{y}_N; \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}'_N), \boldsymbol{\Gamma}_{\mathbf{Y}}(\mathrm{r}'_N)\right)};$$

$$\widehat{\mathbf{x}}_{N|N}^*(\mathrm{r}_N) = \boldsymbol{\mu}_{\mathbf{X}}(\mathrm{r}_N) + \boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_N)\boldsymbol{\Gamma}_{\mathbf{Y}}^{-1}(\mathrm{r}_N)(\mathbf{y}_N - \boldsymbol{\mu}_{\mathbf{Y}}(\mathrm{r}_N));$$

$$\widehat{\mathbf{\Sigma}}_{N|N}^*(\mathrm{r}_N) = \boldsymbol{\Gamma}_{\mathbf{X}} - \boldsymbol{\Gamma}_{\mathbf{XY}}(\mathrm{r}_N)\boldsymbol{\Gamma}_{\mathbf{Y}}^{-1}(\mathrm{r}_N)\boldsymbol{\Gamma}_{\mathbf{YX}}(\mathrm{r}_N).$$

Recursion: compute $\{\pi^*_{n-1}(\mathrm{r}_{n-1}), \widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}), \widehat{\mathbf{\Sigma}}^*_{n-1|n-1}(\mathrm{r}_{n-1})\}_{\mathrm{r}_{n-1}\in\Omega}$ from $\{\pi^*_n(\mathrm{r}_n), \widehat{\mathbf{x}}^*_{n|n}(\mathrm{r}_n), \widehat{\mathbf{\Sigma}}^*_{n|n}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ for each n in $\{2:N\}$. Let r_{n-1}, r_n in Ω,
a)*time update*:

$$\widehat{\mathbf{z}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \begin{bmatrix} \widehat{\mathbf{x}}^*_{n|n}(\mathrm{r}_n) \\ \mathbf{y}_n \end{bmatrix} + \boldsymbol{L}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) = \begin{bmatrix} \widehat{\mathbf{x}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \\ \widehat{\mathbf{y}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \end{bmatrix}; \quad (5.16a)$$

$$\widehat{\mathbf{\Gamma}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) \begin{bmatrix} \widehat{\mathbf{\Sigma}}^*_{n|n}(\mathrm{r}_n) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \boldsymbol{F}^*(\mathrm{r}_{n-1}, \mathrm{r}_n)^\top + \boldsymbol{Q}^*(\mathrm{r}_{n-1}, \mathrm{r}_n) =$$
$$\begin{bmatrix} \widehat{\mathbf{\Sigma}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) & \widehat{\boldsymbol{C}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \\ \widehat{\boldsymbol{C}}^{*\top}_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) & \widehat{\boldsymbol{S}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \end{bmatrix}. \quad (5.16b)$$

b)*measurement update*

$$\widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \widehat{\mathbf{x}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) +$$
$$\widehat{\boldsymbol{C}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \widehat{\boldsymbol{S}}^{*-1}_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n)(\mathbf{y}_{n-1} - \widehat{\mathbf{y}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n)); \quad (5.17a)$$

$$\widehat{\mathbf{\Sigma}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \widehat{\mathbf{\Sigma}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) - \widehat{\boldsymbol{C}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n) \widehat{\boldsymbol{S}}^{*-1}_{n-1|n} \widehat{\boldsymbol{C}}^{*\top}_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n); \quad (5.17b)$$

Next, let $c^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=}} p\left(\mathbf{y}_{n-1} \left| \mathbf{y}_{n..N}, \mathrm{r}_{n-1}, \mathrm{r}_n \right.\right)$, we have

$$c^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \mathcal{N}\left(\mathbf{y}_{n-1}; \widehat{\mathbf{y}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n), \widehat{\boldsymbol{S}}^*_{n-1|n}(\mathrm{r}_{n-1}, \mathrm{r}_n)\right). \quad (5.18)$$

— Update the posterior distribution of the discrete state:

$$\forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, \pi^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) = \frac{\pi^*_n(\mathrm{r}_n) p\left(\mathrm{r}_{n-1} \left| \mathrm{r}_n \right.\right) c^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n)}{\sum\limits_{\mathrm{r}'_n, \mathrm{r}'_{n+1} \in \Omega} \pi^*_n(\mathrm{r}'_n) p\left(\mathrm{r}'_{n-1} \left| \mathrm{r}'_n \right.\right) c^*_{n-1}(\mathrm{r}'_{n-1}, \mathrm{r}'_n)}; \quad (5.19a)$$

$$\forall \mathrm{r}_{n-1} \in \Omega, \pi^*_{n-1}(\mathrm{r}_{n-1}) = \sum_{\mathrm{r}_n \in \Omega} \pi^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n). \quad (5.19b)$$

— Compute, for each r_{n-1} in Ω, $\widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1})$ and $\widehat{\mathbf{\Sigma}}^*_{n-1|n-1}(\mathrm{r}_{n-1})$:

$$\widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}) = \sum_{\mathrm{r}_n=1}^{K} \pi^*_{n-1}(\mathrm{r}_n|\mathrm{r}_{n-1}) \widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n); \quad (5.20a)$$

$$\widehat{\mathbf{\Sigma}}_{n-1|n-1}(\mathrm{r}_{n-1}) = \sum_{\mathrm{r}_n=1}^{K} \pi^*_{n-1}(\mathrm{r}_n|\mathrm{r}_{n-1}) \widehat{\mathbf{\Sigma}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) + \sum_{\mathrm{r}_n=1}^{K} \pi^*_{n-1}(\mathrm{r}_n|\mathrm{r}_{n-1}) \times \quad (5.20b)$$
$$(\widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) - \widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}))(\widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n) - \widehat{\mathbf{x}}^*_{n-1|n-1}(\mathrm{r}_{n-1}))^\top, \quad (5.20c)$$

with

$$\pi^*_{n-1}(\mathrm{r}_n|\mathrm{r}_{n-1}) = \frac{\pi^*_{n-1}(\mathrm{r}_{n-1}, \mathrm{r}_n)}{\pi^*_{n-1}(\mathrm{r}_{n-1})}. \quad (5.21)$$

(The algorithm ends here)

Our idea to set up a Bayesian-assimilation-based smoothed inference is the following. First, we use the estimates of $\{p\left(\mathrm{r}_{n-1}\left|\mathbf{y}_{1:n-1}\right.\right)\}_{\mathrm{r}_{n-1}\in\Omega}$ and $\{p\left(\mathrm{r}_{n+1}\left|\mathbf{y}_{n+1:N}\right.\right)\}_{\mathrm{r}_{n+1}\in\Omega}$ computed by SKF and RSKF to compute estimates of $\{p\left(\mathrm{r}_{n}\left|\mathbf{y}_{1:N}\right.\right)\}_{\mathrm{r}_{n}\in\Omega}$ at each n in $\{2:N-1\}$, as illustrated in Figure 5.3.

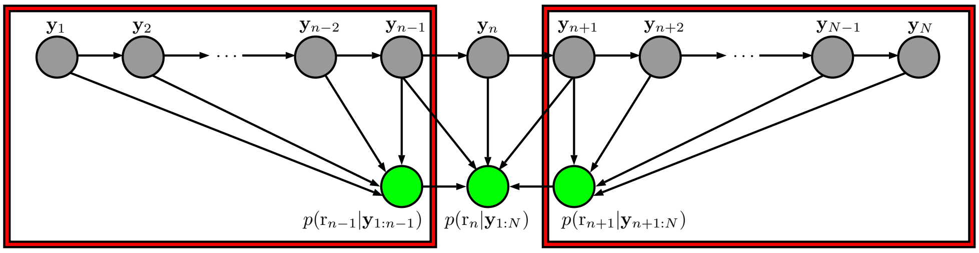


Figure 5.3: Bayesian-assimilation-based smoothed inference of the discrete state.

To this end we consider the following conditional distribution crucial for Bayesian assimilation

$$\forall n\in\{2:(N-1)\},\forall \mathrm{r}_{n-1},\mathrm{r}_n,\mathrm{r}_{n+1}\in\Omega,$$
$$p\left(\mathrm{r}_n\left|\mathrm{r}_{n-1},\mathrm{r}_{n+1},\mathbf{y}_{n-1:n+1}\right.\right)=\frac{p\left(\mathrm{r}_{n-1},\mathrm{r}_n,\mathrm{r}_{n+1}\right)p\left(\mathbf{y}_{n-1:n+1}\left|\mathrm{r}_{n-1},\mathrm{r}_n,\mathrm{r}_{n+1}\right.\right)}{\sum\limits_{\mathrm{r}'_n\in\Omega}p\left(\mathrm{r}_{n-1},\mathrm{r}'_n,\mathrm{r}_{n+1}\right)p\left(\mathbf{y}_{n-1:n+1}\left|\mathrm{r}_{n-1},\mathrm{r}'_n,\mathrm{r}_{n+1}\right.\right)},\quad(5.22)$$

illustrated in Figure 5.4.

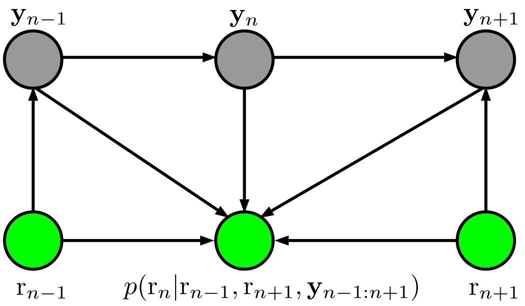


Figure 5.4: Evaluation conditional distribution in (5.22).

Let us define for each n in $\{1:N\}$, r_n in Ω,

$$\pi_{n|N}(\mathrm{r}_n)\overset{\text{def}}{\underset{\text{approx.}}{=\!=}}p\left(\mathrm{r}_n\left|\mathbf{y}_{1:N}\right.\right)$$

computed by the algorithm below.

Algorithm 5. *Smoothed inference of the discrete state*

- Compute, for all n in $\{1:N\}$, r_n in Ω

$$\pi_n(\mathrm{r}_n)\overset{\text{def}}{\underset{\text{approx.}}{=\!=}}p\left(\mathrm{r}_n\left|\mathbf{y}_{1:n}\right.\right),\qquad\qquad \pi^*_n(\mathrm{r}_n)\overset{\text{def}}{\underset{\text{approx.}}{=\!=}}p\left(\mathrm{r}_n\left|\mathbf{y}_{n:N}\right.\right)$$

by the SKF and RSKF.

- Let

$$\forall \mathrm{r}_1\in\Omega,\pi_{1|N}(\mathrm{r}_1)=\pi^*_1(\mathrm{r}_1),\qquad\qquad \forall \mathrm{r}_N\in\Omega,\pi_{N|N}(\mathrm{r}_N)=\pi_N(\mathrm{r}_N);$$

For each n in $\{2:N-1\}$, r_n in Ω, $\pi_{n|N}(\mathrm{r}_n)$ is computed as follows:

$$\pi_{n|N}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right) \pi_{n-1}(\mathrm{r}_{n-1})\pi^*_{n+1}(\mathrm{r}_{n+1}). \tag{5.23}$$

Note that $\{\pi_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ are computed using $\{\pi_{n-1}(\mathrm{r}_{n-1})\}_{\mathrm{r}_{n-1}\in\Omega}$, $\{\pi^*_{n+1}(\mathrm{r}_{n+1})\}_{\mathrm{r}_{n+1}\in\Omega}$ only and $\mathbf{y}_{n-1:n+1}$. In other words, $\{\pi_{n|N}\}_{n\in 1:N}$ are computed independently from each other.

(The algorithm ends here)

Let us justify formula (5.23).

Justification: Four our Bayesian assimilation technique, we assume the following:

$$\forall n \in \{1 : (N-1)\}, \forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega, p\left(\mathrm{r}_n \left| \mathrm{r}_{n+1}, \mathbf{x}_{n+1}, \mathbf{y}_{1:n+1} \right.\right) = p\left(\mathrm{r}_n \left| \mathrm{r}_{n+1}, \mathbf{y}_{1:n+1} \right.\right); \tag{5.24a}$$

$$\forall n \in \{2 : N\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n \in \Omega, p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathbf{x}_{n-1}, \mathbf{y}_{n-1:N} \right.\right) = p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathbf{y}_{n-1:N} \right.\right); \tag{5.24b}$$

$$\begin{aligned}&\forall n \in \{2 : N-1\}, \forall \mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega, p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right) = \\ &\quad p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right).\end{aligned} \tag{5.24c}$$

Assumption (5.24a) is a classic one which can be found in the literature on smoothing in SLDSs [Kim and Nelson, 1999], while assumptions (5.24b) and (5.24c) are similar to (5.24a). Let n in $2 : (N-1)$, r_n in Ω. Observe that

$$\begin{aligned}&p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} \int p\left(\mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1} \left| \mathbf{y}_{1:N} \right.\right) \mathrm{d}\mathbf{x}_{n-1}\mathrm{d}\mathbf{x}_{n+1}; \\ &p\left(\mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1} \left| \mathbf{y}_{1:N} \right.\right) = \\ &\quad p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \mathbf{y}_{1:N} \right.\right) p\left(\mathbf{x}_{n-1}, \mathbf{x}_{n+1} \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{1:N} \right.\right) p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \left| \mathbf{y}_{1:N} \right.\right).\end{aligned} \tag{5.25a}$$

In the above formula,

$$\int \ldots \mathrm{d}\mathbf{x}_{n-1}\mathrm{d}\mathbf{x}_{n+1}$$

means

$$\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \ldots \mathrm{d}\mathbf{x}_{n-1}\mathrm{d}\mathbf{x}_{n+1}.$$

Similarly,

$$\int \ldots \mathrm{d}\mathbf{x}_n$$

means

$$\int_{\mathbb{R}^d} \ldots \mathrm{d}\mathbf{x}_n$$

for the rest of the report, and so on.

One has the following from the Markovianity of $(\mathbf{X}_n, \mathrm{R}_n, \mathbf{Y}_n)_{1\leq n\leq N}$,

$\forall \mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega,$

$$p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \mathbf{y}_{1:N} \right.\right) = p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right). \tag{5.26}$$

Assumption (5.24c) allows approximating $p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right)$ as follows *cf.* (5.25):

$$p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right) p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \left| \mathbf{y}_{1:N} \right.\right).$$

Next, observe that

$$\begin{aligned} p\left(\mathrm{r}_{n-1},\mathrm{r}_{n+1}\middle|\mathbf{y}_{1..N}\right) &= \sum_{\mathrm{r}'_n\in\Omega} p\left(\mathrm{r}_{n-1},\mathrm{r}'_n,\mathrm{r}_{n+1}\middle|\mathbf{y}_{1:N}\right) = \\ &= \sum_{\mathrm{r}'_n\in\Omega} p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}'_n,\mathbf{y}_{1..N}\right) p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}'_n,\mathbf{y}_{1:N}\right) p\left(\mathrm{r}'_n\middle|\mathbf{y}_{1:N}\right), \end{aligned} \tag{5.27}$$

thus

$$\begin{aligned} &p\left(\mathrm{r}_n\middle|\mathbf{y}_{1:N}\right) = \\ &\sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n\middle|\mathrm{r}_{n-1},\mathrm{r}_{n+1},\mathbf{y}_{n-1:n+1}\right) \sum_{\mathrm{r}'_n\in\Omega} p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}'_n,\mathbf{y}_{1:N}\right) p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}'_n,\mathbf{y}_{1:N}\right) p\left(\mathrm{r}'_n\middle|\mathbf{y}_{1:N}\right). \end{aligned} \tag{5.28}$$

On the one hand, assuming (5.24a) results in

$$\forall n\in 2:N, \forall \mathrm{r}_{n-1},\mathrm{r}_n\in\Omega, p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right) = p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{y}_{1:n}\right), \tag{5.29}$$

since

$$\begin{aligned} &\forall n\in 2:N, \forall \mathrm{r}_{n-1},\mathrm{r}_n\in\Omega, p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right) = \int p\left(\mathrm{r}_{n-1},\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{x}_n,\mathbf{y}_{1:N}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{x}_n,\mathbf{y}_{1:n}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{y}_{1:n}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}_n,\mathbf{y}_{1:n}\right). \end{aligned}$$

On the other hand, assuming (5.24b) results in

$$\forall n\in\{1:N-1\}, \forall \mathrm{r}_n,\mathrm{r}_{n+1}\in\Omega, p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{y}_{1..N}\right) = p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{y}_{n:N}\right), \tag{5.30}$$

since

$$\begin{aligned} &\forall n\in\{1:N-1\}, \forall \mathrm{r}_n,\mathrm{r}_{n+1}\in\Omega, p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{y}_{1..N}\right) = \int p\left(\mathrm{r}_{n+1},\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{x}_n,\mathbf{y}_{1:N}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{x}_n,\mathbf{y}_{n:N}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = \\ &\int p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{y}_{n:N}\right) p\left(\mathbf{x}_n\middle|\mathrm{r}_n,\mathbf{y}_{1:N}\right)\mathrm{d}\mathbf{x}_n = p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}_n,\mathbf{y}_{n:N}\right). \end{aligned}$$

Thus, by substituting (5.29) and (5.30) in (5.28), we have

$$\begin{aligned} &p\left(\mathrm{r}_n\middle|\mathbf{y}_{1:N}\right) = \\ &\sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n\middle|\mathrm{r}_{n-1},\mathrm{r}_{n+1},\mathbf{y}_{n-1:n+1}\right) \sum_{\mathrm{r}'_n\in\Omega} p\left(\mathrm{r}_{n-1}\middle|\mathrm{r}'_n,\mathbf{y}_{1:n}\right) p\left(\mathrm{r}_{n+1}\middle|\mathrm{r}'_n,\mathbf{y}_{n:N}\right) p\left(\mathrm{r}'_n\middle|\mathbf{y}_{1:N}\right). \end{aligned} \tag{5.31}$$

We see that our Bayesian-assimilation-based smoothing solution

$$\forall n\in\{2:N-1\}, \mathrm{r}_n\in\Omega, \pi_{n|N}(\mathrm{r}_n) \overset{\text{def}}{\underset{\text{approx.}}{=\!=\!=}} p\left(\mathrm{r}_n\middle|\mathbf{y}_{1:N}\right)$$

can be expressed from the outputs of SKF and RSKF

$$\pi_n(\mathrm{r}_{n-1}|\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} p\left(\mathrm{r}_{n-1}\,|\mathrm{r}_n, \mathbf{y}_{1:n}\right);$$

$$\pi_n(\mathrm{r}_{n+1}|\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} p\left(\mathrm{r}_{n+1}\,|\mathrm{r}_n, \mathbf{y}_{n:N}\right),$$

as follows:

$$\forall n \in \{2:N-1\}, \mathrm{r}_n \in \Omega, \pi_{n|N}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n\,|\mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1}\right) \sum_{\mathrm{r}'_n\in\Omega} \pi_n(\mathrm{r}_{n-1}|\mathrm{r}'_n)\pi^*_n(\mathrm{r}_{n+1}|\mathrm{r}'_n)\pi_{n|N}(\mathrm{r}'_n). \tag{5.32}$$

Note that the above equation defines $\{\pi_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ as the solution of a linear system. Specifically, suppose that $\pi_{n|N}$ is a column vector whose consecutive elements are $\pi_{n|N}(\omega_1), \ldots, \pi_{n|N}(\omega_K)$. Thus, $\pi_{n|N}$ verifies

$$\pi_{n|N} = \boldsymbol{A}_n \pi_{n|N}, \tag{5.33}$$

where $\boldsymbol{A}_n$ is defined as follows:

$$\forall 1 \leq i,j \leq M, \boldsymbol{A}_n(i,j) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\omega_i\,|\mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1}\right) \pi_n(\mathrm{r}_{n-1}|\omega_j)\pi^*_n(\mathrm{r}_{n+1}|\omega_j).$$

Thus, $\pi_{n|N}$ is invariant with respect to multiplication by $\boldsymbol{A}_n$ and therefore it can be approximated iteratively. Let us initialize this recursion by dropping conditional dependencies on r'_n in (5.32):

$$\forall n \in \{2:N-1\}, \mathrm{r}_n \in \Omega,$$
$$\pi^{(0)}_{n|N}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n\,|\mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1}\right) \sum_{\mathrm{r}'_n\in\Omega} \pi_{n-1}(\mathrm{r}_{n-1})\pi^*_{n+1}(\mathrm{r}_{n+1})\pi_{n|N}(\mathrm{r}'_n) =$$
$$\sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} p\left(\mathrm{r}_n\,|\mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1}\right) \pi_{n-1}(\mathrm{r}_{n-1})\pi^*_{n+1}(\mathrm{r}_{n+1}). \tag{5.34}$$

$\pi_{n|N}$ is therefore can be approximated by iterating

$$\pi^{(i+1)}_{n|N} = \boldsymbol{A}_n \pi^{(i)}_{n|N}. \tag{5.35}$$

However, in practice, iterating (5.35) does not seem to affect initialization (5.34) significantly. That is why we suggest using closed-form formula (5.34), given in (5.23), as the smoothed estimate of the discrete state. □

Next, we use estimates of $\left\{\mathbb{E}\left[\mathbf{X}_{n-1}\,|\mathbf{y}_{1:n-1}, \mathrm{r}_{n-1}\right]\right\}_{\mathrm{r}_{n-1}\in\Omega}$, $\left\{\mathbb{E}\left[\mathbf{X}_{n+1}\,|\mathbf{y}_{n+1:N}, \mathrm{r}_{n+1}\right]\right\}_{\mathrm{r}_{n+1}\in\Omega}$ to compute estimates of $\left\{\mathbb{E}\left[\mathbf{X}_n\,|\mathbf{y}_{1:N}, \mathrm{r}_n\right]\right\}_{\mathrm{r}_n\in\Omega}$.
To this purpose, let us define $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\gamma}$ dependent on $\mathrm{r}_{n-1:n+1}$ such that

$$\mathbb{E}\left[\mathbf{X}_n\,|\mathbf{X}_{n-1}, \mathbf{X}_{n+1}, \mathrm{r}_{n-1:n+1}, \mathbf{y}_{n-1:n+1}\right] = \boldsymbol{\alpha}_1(\mathrm{r}_{n-1:n+1})\mathbf{X}_{n-1} + \boldsymbol{\alpha}_2(\mathrm{r}_{n-1:n+1})\mathbf{X}_{n+1} + $$
$$+ \boldsymbol{\beta}_1(\mathrm{r}_{n-1:n+1})\mathbf{y}_{n-1} + \boldsymbol{\beta}_2(\mathrm{r}_{n-1:n+1})\mathbf{y}_n + \boldsymbol{\beta}_3(\mathrm{r}_{n-1:n+1})\mathbf{y}_{n+1} + \boldsymbol{\gamma}(\mathrm{r}_{n-1:n+1}), \tag{5.36}$$

as illustrated in Figure 5.5.
For each n in $\{2:N-1\}$, r_n in Ω, we define

$$\widehat{\mathbf{x}}^-_{n-1|n}(\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} \mathbb{E}\left[\mathbf{X}_{n-1}\,|\mathrm{r}_n, \mathbf{y}_{1..n}\right];$$

$$\widehat{\mathbf{x}}^+_{n+1|n}(\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} \mathbb{E}\left[\mathbf{X}_{n+1}\,|\mathrm{r}_n, \mathbf{y}_{n..N}\right];$$

$$\forall \mathrm{r}_{n-1}, \mathrm{r}_{n+1} \in \Omega, \pi_{n|N}(\mathrm{r}_{n-1}, \mathrm{r}_{n+1}|\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1}\,|\mathrm{r}_n, \mathbf{y}_{1:N}\right);$$

$$\widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n) \xlongequal[\text{approx.}]{\text{def}} \mathbb{E}\left[\mathbf{X}_n\,|\mathrm{r}_n, \mathbf{y}_{1..N}\right].$$

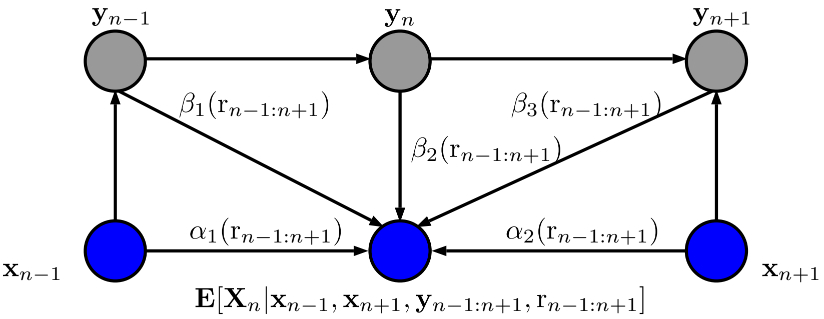

$\mathbf{y}_{n-1}$
$\mathbf{y}_n$
$\mathbf{y}_{n+1}$
$\beta_1(\mathrm{r}_{n-1:n+1})$
$\beta_3(\mathrm{r}_{n-1:n+1})$
$\beta_2(\mathrm{r}_{n-1:n+1})$
$\alpha_1(\mathrm{r}_{n-1:n+1})$
$\alpha_2(\mathrm{r}_{n-1:n+1})$
$\mathbf{x}_{n-1}$
$\mathbf{x}_{n+1}$
$\mathbf{E}[\mathbf{X}_n|\mathbf{x}_{n-1},\mathbf{x}_{n+1},\mathbf{y}_{n-1:n+1},\mathrm{r}_{n-1:n+1}]$

Figure 5.5: $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\gamma}$ in (5.36).

These quantities are computed as follows:

$$\widehat{\mathbf{x}}^-_{n-1|n}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n-1}\in\Omega} \widehat{\mathbf{x}}_{n-1|n-1}(\mathrm{r}_{n-1})\pi_n(\mathrm{r}_{n-1}|\mathrm{r}_n); \tag{5.37a}$$

$$\widehat{\mathbf{x}}^+_{n+1|n}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n+1}\in\Omega} \widehat{\mathbf{x}}^*_{n+1|n+1}(\mathrm{r}_{n+1})\pi^*_n(\mathrm{r}_{n+1}|\mathrm{r}_n); \tag{5.37b}$$

$$\pi_{n|N}(\mathrm{r}_{n-1},\mathrm{r}_{n+1}|\mathrm{r}_n) = \frac{p\left(\mathrm{r}_n\left|\mathrm{r}_{n-1},\mathrm{r}_{n+1},\mathbf{y}_{n-1:n+1}\right.\right)}{\pi_{n|N}(\mathrm{r}_n)} \sum_{\mathrm{r}'_n\in\Omega} \pi_n(\mathrm{r}_{n-1}|\mathrm{r}'_n)\pi^*_n(\mathrm{r}_{n+1}|\mathrm{r}'_n)\pi_{n|N}(\mathrm{r}'_n), \tag{5.37c}$$

where

- $\{\widehat{\mathbf{x}}_{n-1|n-1}(\mathrm{r}_{n-1}), \pi_n(\mathrm{r}_{n-1}|\mathrm{r}_n)\}_{\mathrm{r}_{n-1},\mathrm{r}_n\in\Omega}$ are computed by the SKF;
- $\{\widehat{\mathbf{x}}^*_{n+1|n+1}(\mathrm{r}_{n+1}), \pi^*_n(\mathrm{r}_{n+1}|\mathrm{r}_n)\}_{\mathrm{r}_n,\mathrm{r}_{n+1}\in\Omega}$ are computed by the RSKF;
- $\{\pi_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ are computed by (5.23).

$\{\widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n\in\Omega}$ are computed as follows:

$$\widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n) = \sum_{\mathrm{r}_{n-1},\mathrm{r}_{n+1}\in\Omega} \pi_{n|N}(\mathrm{r}_{n-1},\mathrm{r}_{n+1}|\mathrm{r}_n)\Big(\boldsymbol{\alpha}_1(\mathrm{r}_{n-1:n+1})\widehat{\mathbf{x}}^-_{n-1|n}(\mathrm{r}_n) + \boldsymbol{\alpha}_2(\mathrm{r}_{n-1:n+1})\widehat{\mathbf{x}}^+_{n+1|n}(\mathrm{r}_n) +$$
$$\boldsymbol{\beta}_1(\mathrm{r}_{n-1:n+1})\mathbf{y}_{n-1} + \boldsymbol{\beta}_2(\mathrm{r}_{n-1:n+1})\mathbf{y}_n + \boldsymbol{\beta}_3(\mathrm{r}_{n-1:n+1})\mathbf{y}_{n+1} + \boldsymbol{\gamma}(\mathrm{r}_{n-1:n+1})\Big), \tag{5.38}$$

which is illustrated in Figure 5.6.

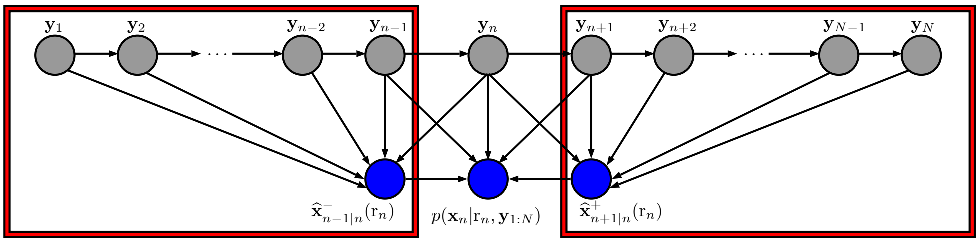


Figure 5.6: Bayesian-assimilation-based smoothed inference of the continuous state.

Let us justify formula (5.38).

Justification: First, observe that we have the following from the law of total expectation:

$$\mathbb{E}\left[\mathrm{X}_n\left|\mathrm{r}_n,\mathbf{y}_{1:N}\right.\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbf{X}_n\left|\mathbf{X}_{n-1},\mathbf{X}_{n+1},\mathrm{r}_{n-1:n+1},\mathbf{y}_{1:N}\right.\right]\left|\mathrm{r}_n,\mathbf{y}_{1:N}\right.\right].$$

We have the following from the Markovianity of $(\mathbf{X}_n,\mathrm{R}_n,\mathbf{Y}_n)_{1\le n\le N}$,

$$p\left(\mathbf{x}_n\left|\mathbf{x}_{n-1},\mathbf{x}_{n+1},\mathrm{r}_{n-1:n+1},\mathbf{y}_{1:N}\right.\right) = p\left(\mathbf{x}_n\left|\mathbf{x}_{n-1},\mathbf{x}_{n+1},\mathrm{r}_{n-1:n+1},\mathbf{y}_{n-1:n+1}\right.\right).$$

Thus,

$$\mathbb{E}\left[\mathrm{X}_n\left|\mathrm{r}_n,\mathbf{y}_{1:N}\right.\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbf{X}_n\left|\mathbf{X}_{n-1},\mathbf{X}_{n+1},\mathrm{r}_{n-1:n+1},\mathbf{y}_{n-1:n+1}\right.\right]\left|\mathrm{r}_n,\mathbf{y}_{1:N}\right.\right].$$

Next, by using $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\gamma}$ defined in (5.36), we have

$$\mathbb{E}\left[\mathrm{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right] = \sum_{\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right) \Big(\boldsymbol{\alpha}_1(\mathrm{r}_{n-1:n+1}) \mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right] + \boldsymbol{\alpha}_2(\mathrm{r}_{n-1:n+1}) \mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right] + \boldsymbol{\beta}_1(\mathrm{r}_{n-1:n+1}) \mathbf{y}_{n-1} + \boldsymbol{\beta}_2(\mathrm{r}_{n-1:n+1}) \mathbf{y}_n + \boldsymbol{\beta}_3(\mathrm{r}_{n-1:n+1}) \mathbf{y}_{n+1} + \boldsymbol{\gamma}(\mathrm{r}_{n-1:n+1}) \Big). \tag{5.39}$$

Regarding $\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]$, we have

$$\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_{n-1}, \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right] \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right] = \sum_{\mathrm{r}_{n-1} \in \Omega} p\left(\mathrm{r}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right) \mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_{n-1}, \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]. \tag{5.40}$$

Similarly, we have for $\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]$:

$$\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1..N} \right.\right] \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right] = \sum_{\mathrm{r}_{n+1} \in \Omega} p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right) \mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1..N} \right.\right]. \tag{5.41}$$

Recall that for each r_{n-1},r_n,r_{n+1} in Ω,

- $p\left(\mathrm{r}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right) = p\left(\mathrm{r}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..n} \right.\right)$ under assumption (5.24a);
- $p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right) = p\left(\mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{n..N} \right.\right)$ under assumption (5.24b);
- We have under assumptions (5.24a)-(5.24c)

$$p\left(\mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1} \left| \mathbf{y}_{1:N} \right.\right) = p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right) \sum_{\mathrm{r}'_n \in \Omega} p\left(\mathrm{r}_{n-1} \left| \mathrm{r}'_n, \mathbf{y}_{1:n} \right.\right) p\left(\mathrm{r}_{n+1} \left| \mathrm{r}'_n, \mathbf{y}_{n:N} \right.\right) p\left(\mathrm{r}'_n \left| \mathbf{y}_{1:N} \right.\right).$$

Indeed, $\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_{n-1}, \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]$ and $\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1..N} \right.\right]$ are not easily accessible without further approximation. We propose to approximate them by $\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_{n-1}, \mathbf{y}_{1..n} \right.\right]$ and $\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_{n+1}, \mathbf{y}_{n..N} \right.\right]$ respectively. Thus,

- $\mathbb{E}\left[\mathbf{X}_{n-1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]$ is approximated by $\widehat{\mathbf{x}}^-_{n-1|n}(\mathrm{r}_n)$ defined in (5.37a) *cf.* (5.40);
- $\mathbb{E}\left[\mathbf{X}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1..N} \right.\right]$ is approximated by $\widehat{\mathbf{x}}^+_{n+1|n}(\mathrm{r}_n)$ defined in (5.37b) *cf.* (5.41);
- For all $\mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1}$, $p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right)$ is approximated by $\pi_{n|N}(\mathrm{r}_{n-1}, \mathrm{r}_{n+1}|\mathrm{r}_n)$ defined in (5.37c) since we have under assumptions (5.24a)-(5.24c):

$$p\left(\mathrm{r}_{n-1}, \mathrm{r}_{n+1} \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right) = \frac{p\left(\mathrm{r}_{n-1}, \mathrm{r}_n, \mathrm{r}_{n+1} \left| \mathbf{y}_{1:N} \right.\right)}{p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right)} = \frac{p\left(\mathrm{r}_n \left| \mathrm{r}_{n-1}, \mathrm{r}_{n+1}, \mathbf{y}_{n-1:n+1} \right.\right)}{p\left(\mathrm{r}_n \left| \mathbf{y}_{1:N} \right.\right)} \sum_{\mathrm{r}'_n \in \Omega} p\left(\mathrm{r}_{n-1} \left| \mathrm{r}'_n, \mathbf{y}_{1:n} \right.\right) p\left(\mathrm{r}_{n+1} \left| \mathrm{r}'_n, \mathbf{y}_{n:N} \right.\right) p\left(\mathrm{r}'_n \left| \mathbf{y}_{1:N} \right.\right);$$

- Finally, $\mathbb{E}\left[\mathrm{X}_n \left| \mathrm{r}_n, \mathbf{y}_{1:N} \right.\right]$ is approximated by $\widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n)$ defined in (5.38) *cf.* (5.39).

□

The whole algorithm runs as follows:

Algorithm 6. *Smoothed inference by Bayesian assimilation*

1. Run Algorithms 3 and 4 to obtain RSKF and SKF outputs (5.7), (5.14). SKF and RSKF may run in parallel;
2. Run Algorithms 5 to obtain smoothed estimates of the discrete state $\{\pi_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n \in \Omega}$;
3. For each n in $2 : N-1$, r_n in Ω, compute smoothed estimates of the continuous state $\{\widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n)\}_{\mathrm{r}_n \in \Omega}$ by (5.37)-(5.38);
4. $\mathbb{E}\left[\mathbf{X}_n \left| \mathbf{y}_{1:N} \right.\right]$ is then approximated by $\sum_{\mathrm{r}_n \in \Omega} \widehat{\mathbf{x}}_{n|N}(\mathrm{r}_n) \pi_{n|N}(\mathrm{r}_n)$.

(The algorithm ends here)

5.1.2 Applications to trend estimation

Here we illustrate our smoothing algorithm applied to the problem of trend estimation in financial time series. We will consider a classic model without switching and then extend it by incorporating a switching process. Let $N \in \mathbb{N}^*$ be a sample size, the classic Local Trend Model (LTM) [Tsay, 2005] reads:

$$\mathrm{X}_1 \sim \mathcal{N}(m_1, \Sigma_1); \tag{5.42a}$$

$$\forall n \in \{1:(N-1)\}, \mathrm{X}_{n+1} = \phi \mathrm{X}_n + q\mathrm{U}_{n+1} + a; \tag{5.42b}$$

$$\forall n \in \{1:N\}, \mathrm{Y}_n = \mathrm{X}_n + \sigma \mathrm{V}_n, \tag{5.42c}$$

where ϕ, q, a and σ are fixed parameters in $\mathbb{R}$, $|\phi| < 1$, $\mathrm{X}_{1:N} \in \mathbb{R}$, $\mathrm{Y}_{1:N} \in \mathbb{R}$, $\mathrm{U}_{2:N}, \mathrm{V}_{1:N}$ are zero-mean unit-variance Gaussian white noise in $\mathbb{R}$. The terms involved in this model have the following meaning.

- $\mathrm{Y}_{1:N}$ are log-returns computed from the price chart of an asset and $\mathrm{X}_{1:N}$ is their underlying trend. $\mathrm{X}_{1:N}$ is supposed to be estimated from $\mathrm{Y}_{1:N}$;
- σ is the standard deviation of price movements which are irrelevant to the underlying trend. In other words, σ quantifies the market noise;
- ϕ is the persistence of trend in time. In practice, it is common to consider that $\phi \approx 1$.
- a can be seen as the intercept in linear regression equation (5.42b). This parameter is related to the ergodic mean of $\{\mathrm{Y}_n\}_{\in \mathbb{N}}$ as follows:

$$\lim_{n\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathrm{Y}_n = \frac{a}{1-\phi}.$$

- q is known as the conditional variance of the trend. It determines how flexible the trend is;
- m_1 and Σ_1 are chosen in a way that $\mathrm{X}_{1:N}$ would be stationary. Specifically, one has

$$m_1 = \frac{a}{1-\phi}, \qquad \Sigma_1 = \frac{q^2}{1-\phi^2}.$$

We consider extending this model by making a dependent on a stationary Markov chain $\mathrm{R}_{1:N}$ in $\Omega = \{\omega_1, \omega_2\}$:

$$\forall n \in \{1:(N-1)\}, \mathrm{X}_{n+1} = \phi \mathrm{X}_n + q\mathrm{U}_{n+1} + a(\mathrm{R}_{n+1}); \tag{5.43a}$$

$$\forall n \in \{1:N\}, \mathrm{Y}_n = \mathrm{X}_n + \sigma \mathrm{V}_n, \tag{5.43b}$$

with the same assumptions as for (5.42). Thus, we obtain a Local Switching Trend Model (LSTM). We suppose that Markov chain $\mathrm{R}_{1:N}$ is stationary and verifies

$$\forall \omega \in \Omega, p(\mathrm{r}_1 = \omega) = 0.5; \tag{5.44a}$$

$$\forall n \in \{1:(N-1)\}, p(\mathrm{r}_{n+1} \neq \omega \,|\mathrm{r}_n = \omega) = \delta. \tag{5.44b}$$

Here, both δ and ϕ specify the persistence of the trend.

The SCGPMSM form (5.2) of (5.43) is computed as follows. Define, for each $\mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega$,

$$\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \begin{bmatrix} \phi & 0 \\ \phi & 0 \end{bmatrix}, \qquad \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \begin{bmatrix} a(\mathrm{r}_{n+1}) \\ a(\mathrm{r}_{n+1}) \end{bmatrix}, \boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \begin{bmatrix} q^2 & q^2 \\ q^2 & \sigma^2 \end{bmatrix}. \tag{5.45}$$

Thus, SLDS (5.43) verifies

$$\forall n \in \{1:(N-1)\}, p(\mathbf{z}_{n+1} \,|\mathbf{z}_n, \mathrm{r}_n, \mathrm{r}_{n+1}) = \mathcal{N}(\mathbf{z}_{n+1}; \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})\mathbf{z}_n + \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}), \boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1})),$$

with $\forall n \in \{1:N\}$, $\mathbf{z}_n = \begin{bmatrix} \mathrm{x}_n \\ \mathrm{y}_n \end{bmatrix}$. The SCGPMSM parameters (5.2) $\{\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n), \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n)\}_{\mathrm{r}_n \in \Omega}$ of (5.43) verify

$$\forall \mathrm{r}_{n+1} \in \Omega, \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n \in \Omega} p(\mathrm{r}_n \,|\mathrm{r}_{n+1}) \Big(\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n) + \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) \Big); \tag{5.46a}$$

$$\begin{aligned} &\forall \mathrm{r}_{n+1} \in \Omega, \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{n+1}) + \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n+1}) \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_{n+1})^\top = \\ &\quad \sum_{\mathrm{r}_n \in \Omega} p(\mathrm{r}_n \,|\mathrm{r}_{n+1}) \Big(\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_{n+1}) \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1}) \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n) \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \\ &\quad \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) \boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n)^\top \boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1}) \boldsymbol{L}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \boldsymbol{Q}(\mathrm{r}_n, \mathrm{r}_{n+1}) \Big). \end{aligned} \tag{5.46b}$$

The above equation can be solved in $\{\boldsymbol{\mu}_{\mathbf{Z}}(\mathrm{r}_n), \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n)\}_{\mathrm{r}_n \in \Omega}$ analytically for parameters (5.45). In the general case, one usually uses iterative techniques to find an approximate solution. Finally, regarding $\{\boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_n, \mathrm{r}_{n+1}), \boldsymbol{\Gamma}_{\mathbf{Z}_2\mathbf{Z}_1}(\mathrm{r}_n, \mathrm{r}_{n+1})\}_{\mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega}$ in (5.2), we have

$$\forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \Omega, \boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{\Gamma}_{\mathbf{Z}}(\mathrm{r}_n)\boldsymbol{F}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top, \quad \boldsymbol{\Gamma}_{\mathbf{Z}_2\mathbf{Z}_1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \boldsymbol{\Gamma}_{\mathbf{Z}_1\mathbf{Z}_2}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top.$$

In order to find realistic parameter values for (5.43), we estimated model (5.42) from the daily price chart of S&P 500 stock market index between 04-Jul-2014 and 02-Jun-2016. We found

$$\sigma = 0.0090, \qquad q = 3 \cdot 10^{-4};$$
$$\phi = 0.9900, \qquad a = 1.172 \cdot 10^{-6}.$$

Therefore, we considered the following cases for the parameters of (5.43):

- Regarding $a(\omega_1)$ and $a(\omega_2)$: the low-spread case - $a(\omega_1) = -2.5 \cdot 10^{-4}$, $a(\omega_2) = 2.5 \cdot 10^{-4}$) - and the high-spread case - $a(\omega_1) = -5 \cdot 10^{-4}$, $a(\omega_2) = 5 \cdot 10^{-4}$;

- Regarding σ: the case of low market noise - $\sigma = 0.01$ - and the case of high market noise - $\sigma = 0.05$;

- Regarding δ: the case of low-persistent trend - $\delta = 0.1$ - and the case of highly persistent trend - $\delta = 0.01$;

- Regarding q: the case of low conditional variance of the trend - $q = 10^{-4}$ - and the case of high conditional variance - $q = 10^{-3}$;

- Regarding ϕ, we fix its value at 0.99. We observed that when low values of ϕ make the LSTM behave as a classic hidden Markov model with discrete state space.

Thus, we consider 16 different experiment settings in total, generated by combining the aforementioned cases. We perform the following experiment 100 times per each of these settings. First, we generate sample $\{\mathrm{x}_{1:N}, \mathrm{y}_{1:N}\}$ from (5.43) with the parameters as in Table 5.1 with $N = 1000$. Next, we recover the smoothed trend estimates from $\mathrm{y}_{1:N}$ by applying the proposed method (Algorithm 6) and the Particle Smoother (PS). The PS we use is given in Section 1.4.2. We use 2000 particles in the particle smoother. Finally, we compute the Mean Squared Error (MSE) and the Mean Misclassification Error (MME) defined by

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\widehat{\mathrm{x}}_n - \mathrm{x}_n)^2 \tag{5.47}$$

and

$$\mathrm{MME} = \frac{1}{N}\sum_{n=1}^{N}(\widehat{\mathrm{r}}_n \neq \mathrm{r}_n) \tag{5.48}$$

respectively.

We report in Tables 5.1 and 5.2 the average MSE and MME respectively over these experiments.

The simulation study indicates that the proposed method estimates nearly optimally the continuous state as well as the discrete state. Compared to the particle smoother, our method executes 20 times faster on average. An example of estimating a hidden trajectory $\mathrm{x}_{1:N}$ by our smoother is presented in Figure 5.7.

Next, we compare the trend estimates of the S&P 500 stock market index (SPX) between 04-Jul-2014 and 02-Jun-2016, produced by LTM (5.42) and LSTM (5.43). The historical data were taken from the Yahoo database and is displayed in Figure 5.8. The working sample $\mathrm{y}_{1:N}$ of log-returns contains $N = 500$ observations. The sample mean is $1.1720 \cdot 10^{-4}$ and its standard deviation is 0.0091.

We estimate the parameters of LTM (5.42) as follows. First, we set $\phi = 0.99$ and we estimate a by the method of moments applied to the ergodic mean of log-returns in the LTM, that is by equating it with the empirical ergodic mean of $\mathrm{y}_{1:N}$:

$$\frac{a}{1-\phi} = \frac{1}{N}\sum_{n=1}^{N}\mathrm{y}_n \Rightarrow a = (1-\phi)\frac{1}{N}\sum_{n=1}^{N}\mathrm{y}_n.$$

# setting	$a(\omega_1)$	$a(\omega_2)$	σ	δ	q	ϕ	Algorithm 6	PS
1	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.01	10^{-4}	0.99	$2.1 \cdot 10^{-6}$	$1.9 \cdot 10^{-6}$
2	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.01	10^{-3}	0.99	$5.5 \cdot 10^{-6}$	$5.4 \cdot 10^{-6}$
3	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.10	10^{-4}	0.99	$3.2 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$
4	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.10	10^{-3}	0.99	$5.9 \cdot 10^{-6}$	$5.8 \cdot 10^{-6}$
5	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.01	10^{-4}	0.99	$2.7 \cdot 10^{-5}$	$2.7 \cdot 10^{-5}$
6	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.01	10^{-3}	0.99	$3.5 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$
7	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.10	10^{-4}	0.99	$1.8 \cdot 10^{-5}$	$1.7 \cdot 10^{-5}$
8	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.10	10^{-3}	0.99	$3.1 \cdot 10^{-5}$	$3.0 \cdot 10^{-5}$
9	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.01	10^{-4}	0.99	$2.0 \cdot 10^{-6}$	$1.9 \cdot 10^{-6}$
10	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.01	10^{-3}	0.99	$6.0 \cdot 10^{-6}$	$5.8 \cdot 10^{-6}$
11	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.10	10^{-4}	0.99	$5.3 \cdot 10^{-6}$	$5.2 \cdot 10^{-6}$
12	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.10	10^{-3}	0.99	$7.1 \cdot 10^{-6}$	$6.9 \cdot 10^{-6}$
13	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.01	10^{-4}	0.99	$4.1 \cdot 10^{-5}$	$4.0 \cdot 10^{-5}$
14	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.01	10^{-3}	0.99	$4.6 \cdot 10^{-5}$	$4.5 \cdot 10^{-5}$
15	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.10	10^{-4}	0.99	$3.7 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$
16	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.10	10^{-3}	0.99	$4.5 \cdot 10^{-5}$	$4.3 \cdot 10^{-5}$

Table 5.1: Comparison of mean squared error (5.47) of smoothing with various parameters of local switching trend model (5.43) by the Bayesian-assimilation based approach (Algorithm 6) and the PS .

# setting	$a(\omega_1)$	$a(\omega_2)$	σ	δ	q	ϕ	Algorithm 6	PS
1	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.01	10^{-4}	0.99	0.06	0.05
2	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.01	10^{-3}	0.99	0.12	0.11
3	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.10	10^{-4}	0.99	0.31	0.30
4	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.01	0.10	10^{-3}	0.99	0.38	0.36
5	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.01	10^{-4}	0.99	0.16	0.16
6	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.01	10^{-3}	0.99	0.19	0.19
7	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.10	10^{-4}	0.99	0.44	0.43
8	$-2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	0.05	0.10	10^{-3}	0.99	0.46	0.44
9	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.01	10^{-4}	0.99	0.02	0.02
10	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.01	10^{-3}	0.99	0.06	0.05
11	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.10	10^{-4}	0.99	0.24	0.23
12	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.01	0.10	10^{-3}	0.99	0.31	0.29
13	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.01	10^{-4}	0.99	0.11	0.10
14	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.01	10^{-3}	0.99	0.13	0.12
15	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.10	10^{-4}	0.99	0.39	0.38
16	$-5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	0.05	0.10	10^{-3}	0.99	0.41	0.39

Table 5.2: Comparison of mean misclassification error (5.48) of smoothing with various parameters of local switching trend model (5.43) by the Bayesian-assimilation based approach (Algorithm 6) and the PS .

We make vary parameter σ depending on our thoughts of the level of market noise. Finally, parameter q is chosen in a way that

$$\frac{1}{N}\sum_{n=1}^{N}(\mathrm{y}_n - \widehat{\mathrm{x}}_n)^2 = \sigma^2, \tag{5.49}$$

where $\widehat{\mathrm{x}}_{1:N}$ are smoothed estimates of the trend in the LTM computed with the Kalman smoother. Equation (5.49) is general and ensures that the level of the market noise in the model is consistent with the data and smoothed estimates computed using this model.

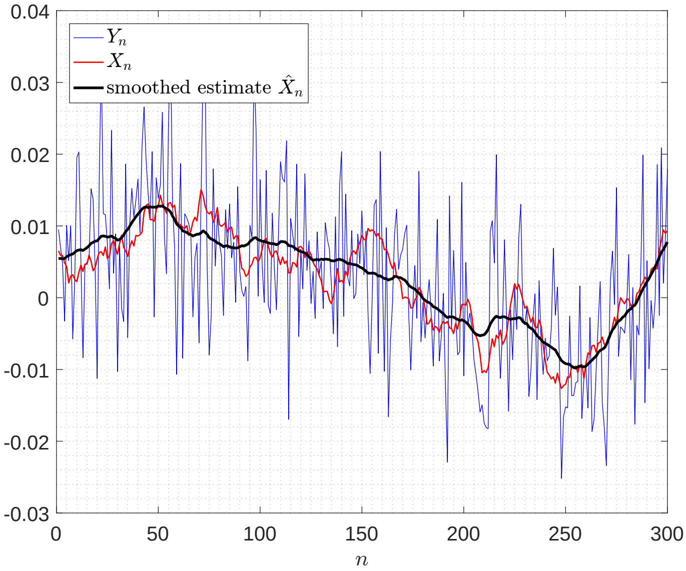
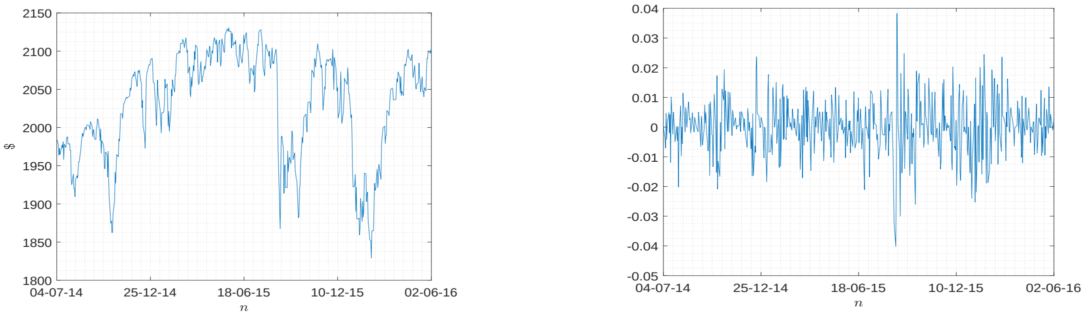


Figure 5.7: Example of estimating a hidden trajectory in setting #1 from Table 5.1.



Price chart

Log-returns

Figure 5.8: Price chart and corresponding log-returns of the S&P 500 stock market index between 04-Jul-2014 and 02-Jun-2016.

Regarding the parameters of LSTM (5.43), we set $\phi = 0.99, \delta = 0.01$, $a(\omega_1) = m - 2s$, $a(\omega_2) = m + 2s$, where

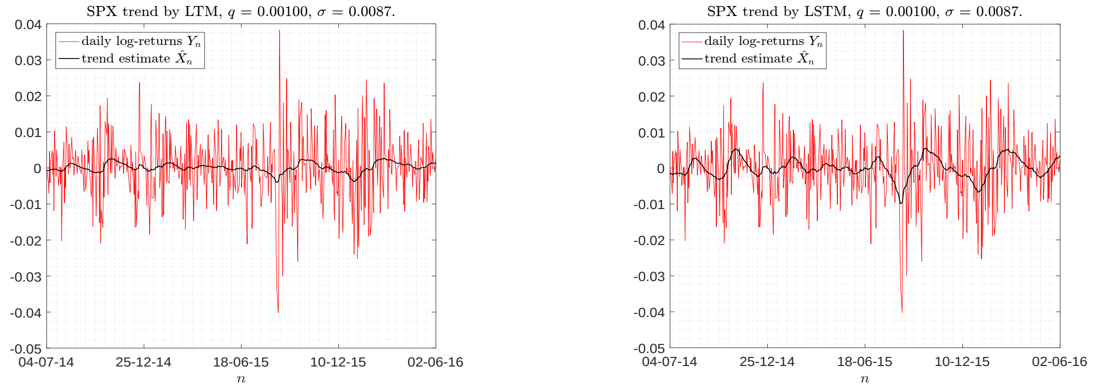
$$m = \frac{1}{N}\sum_{n=1}^{N} \mathrm{y}_n, \qquad s = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (\mathrm{y}_n - m)^2}.$$

Similarly, we make vary σ and parameter q is chosen such that (5.49) is satisfied. We consider three cases:

- The case of low market noise: $\sigma = 0.0077$. In this case, we find $q = 3 \cdot 10^{-3}$ for both LTM and LSTM. The two models produced nearly the same trend estimates.
- The case of moderate market noise: $\sigma = 0.0087$. In this case, we find $q = 10^{-3}$ for both LTM and LSTM. This case is illustrated in Figure 5.9;
- The case of high market noise: $\sigma = 0.0091$. In this case, we find $q = 10^{-4}$ for the LTM and $q = 6 \cdot 10^{-5}$ for the LSTM. This case is illustrated in Figure 5.10.

Let us discuss these results.

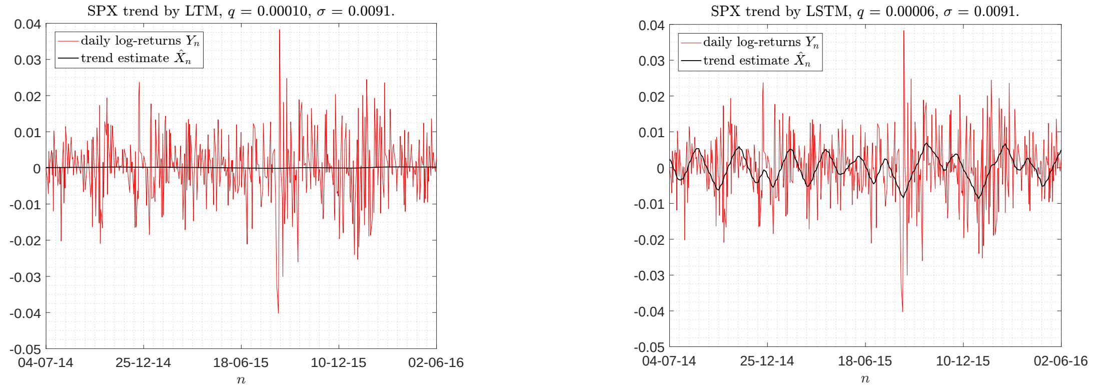
- We see that in the case of low market noise, LTM and LSTM produce nearly the same trend. However, this trend may be of a limited use, since it changes direction too frequently and appears to be affected by some kind of noise. Thus, the low market noise assumption may be erroneous.



LTM

LSTM

Figure 5.9: S&P 500 index (SPX) trend estimates by LTM (5.42) and LSTM (5.43) assuming $\sigma = 0.0087$.



LTM

LSTM

Figure 5.10: S&P 500 index (SPX) trend estimates by LTM (5.42) and LSTM (5.43) assuming $\sigma = 0.0091$.

- Regarding the case of moderate market noise, both models have produced trends in which one can identify distinct moves (upwards and downwards). However, these moves seem to be better presented by the switching model. An increased flexibility of the LSTM compared to that of the LTM possibly allowed to find a better suited trend. We also note that the moderate market noise assumption seems to be appropriate for trend estimation.
- Finally, under the assumption of high market noise, the LTM trend degenerates to a constant function while the LSTM trend seems to overfit the working sample. Indeed, this value of market noise is close to the standard deviation of the sample. Thus, this kind of behavior was expected from the LTM. However, we see that using a switching model may involve risks of overfitting and therefore additional control measures should be considered.

In this study, we used a control parameter σ quantifying the market noise level. However, let us notice that a recent CGPMSM-based unsupervised smoothing technique [Zheng et al., 2016] may allow recovering the trend without considering the LSTM explicitly.

5.2 Bayesian filtering in non-linear non-Gaussian POMPs with hybrid state space

Markov-switching dynamical models (see, *e.g.*, [Olteanu et al., 2004, Wu et al., 2004, Doucet et al., 2001, Logothetis and Krishnamurthy, 1999, Olteanu and Rynkiewicz, 2007, Chen and Liu, 2000, Li and Jilkov, 2005, Andrieu et al., 2003a, Blanchet-Scalliet, 2001, Caron et al., 2007]) allow modeling situations where the dynamics of the system depend upon unknown exogenous discrete-valued factors *cf.* Fig. 5.11. Bayesian inference in these systems is usually dealt with switching filters [Wu et al., 2004, Fu et al., 2010, Logothetis and Krishnamurthy, 1999, Zhao and Liu, 2012, Gao et al., 2012, Toledo-Moreo et al., 2007, Pieczynski, 2011a, Jilkov and Li, 2004, Liao and Chen, 2006, Togneri et al., 2001] or sequential Monte-Carlo methods [Doucet et al., 2001, Andrieu et al., 2003a, Driessen and Boers, 2004, Chen and Liu, 2000, Doucet et al., 2000]. The simulation-based filters are asymptotically optimal, but can be computationally intensive. The usual switching filters are derived from the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or Gauss-Hermite Filter (GHF). EKF, UKF, GHF and their variants are discussed in [Afshari et al., 2017].

Indeed, the EKF, UKF and GHF evaluate local integrals by using a Gaussian approximation in the *joint* state-observation space. However, this *joint* Gaussian approximation does not lead to satisfactory results in many important applications. For example, in the Stochastic Volatility (SV) model [Jacquier et al., 2002], the observed and hidden variables are uncorrelated but dependent. The recent Conditional Gauss-Hermite Filter (CGHF), [Singer, 2015] uses a weaker assumption and is proven to be efficient in the cases where a specific form of the observation equation must be taken into account *cf.* [Singer, 2015, Zoeter et al., 2004].

Sequential Monte Carlo and quasi-Monte Carlo [Niederreiter, 2010] methods are important simulation-based methodologies to solve the filtering problem. Among them, the Particle Filter (PF), [Doucet and Johansen, 2009] is a well-known stochastic algorithm. The Gaussian Particle Filter (GPF), [Kotecha and Djuric, 2003] is a modification of the PF which avoids resampling and allows parallel processing. The CGHF is an "accelerated" version of the GPF [Zoeter, 2007, Singer, 2015, Zoeter et al., 2006, Zoeter et al., 2004, Nikolaev et al., 2014], where one uses the Gaussian quadrature to evaluate local integrals. Compared to the Monte Carlo integration, the Gaussian quadrature has a better convergence rate *cf. e.g.* [Luceno, 1999].

The novelty of the work presented in this section consists in extending the CGHF to handle Markov-switching dynamics. In fact, the CGHF is applicable only for recovering continuous variables, while our extension Switching Conditional Gauss-Hermite Filter (SCGHF) allows recovering both continuous and discrete states simultaneously. In other words, we introduce a switching version of the CGHF. This chapter also extends the conference paper [Gorynin et al., 2016c] whose scope was limited to specific volatility models; the general algorithm we introduce here is applicable to any switching system.

We first recall the current approaches to solve the filtering problem. Next, we expose the algorithm we propose. We provide an empirical comparison of the proposed algorithm with the switching Kalman filter [Wu et al., 2004] and the particle filter [Gordon, 1997] in the context of the Markov-switching stochastic volatility model.

5.2.1 Filtering in non-linear non-Gaussian systems under the Gaussian conditional density assumption

Here we recall three general approaches to the problem of non-linear non-Gaussian filtering: the Gaussian Filter (GF) and the Conditional Gaussian Filter (CGF).

Let us consider the following general form of non-linear non-Gaussian systems:

$$\mathbf{X}_{n+1} = \boldsymbol{f}_{n+1}(\mathbf{X}_n, \mathbf{U}_{n+1}) \qquad \text{for } n \in \mathbb{N}^*, n < N; \tag{5.50a}$$

$$p\left(\mathbf{y}_n \left| \mathbf{x}_n \right.\right) \propto h_n(\mathbf{y}_n, \mathbf{x}_n) \qquad \text{for } n \in \mathbb{N}^*, n \leq N, \tag{5.50b}$$

with Markovian continuous states $\mathbf{X}_{1:N}$ in $\mathbb{R}^d$ and observations $\mathbf{Y}_{1:N}$ in $\mathbb{R}^{d'}$ which are independent given $\mathbf{X}_{1:N}$. Variables $\mathbf{U}_{1:N}$ are independent zero-mean unit-variance Gaussian vectors in $\mathbb{R}^q$. For each $n \in \mathbb{N}^*, n < N$, function $\boldsymbol{f}_{n+1} : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ in (5.50a) determines the time evolution of the system. Equation (5.50b) means that for each n in $\{1 : N\}$, the Probability Density Function (pdf) of $\mathbf{Y}_n$ given $\mathbf{X}_n$ is available analytically.

Note that even if $\mathbf{U}_{1:N}$ are Gaussian, the transition density $p\left(\mathbf{x}_{n+1} \left| \mathbf{x}_n \right.\right)$ is not Gaussian unless function $\boldsymbol{f}_{n+1} : (\mathbf{x}_n, \mathbf{u}_{n+1}) \to \mathbf{x}_{n+1}$ is linear in $\mathbf{u}_{n+1}$. Since there is no assumptions on linearity of $\boldsymbol{f}_{n+1}$, system (5.50) is non-linear non-Gaussian in general.

We consider the filtering problem, *i.e.* recursive estimation of $p\left(\mathbf{x}_n \left| \mathbf{y}_{1:n} \right.\right)$ for consecutive natural n.

The GF generalizes the unscented Kalman and Gauss-Hermite Kalman filters. The main idea is to assume that the following one-step predicting density is Gaussian:

$$\forall n \in \mathbb{N}, p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_{n+1} \\ \mathbf{y}_{n+1}\end{bmatrix}; \begin{bmatrix}\widehat{\mathbf{x}}_{n+1|n} \\ \widehat{\mathbf{y}}_{n+1|n}\end{bmatrix}, \begin{bmatrix}\mathbf{P}^{\mathbf{xx}}_{n+1|n} & \mathbf{P}^{\mathbf{xy}}_{n+1|n} \\ \mathbf{P}^{\mathbf{yx}}_{n+1|n} & \mathbf{P}^{\mathbf{yy}}_{n+1|n}\end{bmatrix}\right), \tag{5.51}$$

where $\widehat{\mathbf{x}}_{n+1|n} \in \mathbb{R}^{d}$, $\widehat{\mathbf{y}}_{n+1|n} \in \mathbb{R}^{d'}$, $\mathbf{P}^{\mathbf{xx}}_{n+1|n} \in \mathbb{R}^{d \times d}$, $\mathbf{P}^{\mathbf{xy}}_{n+1|n} \in \mathbb{R}^{d \times d'}$, $\mathbf{P}^{\mathbf{yx}}_{n+1|n} \in \mathbb{R}^{d' \times d}$ and $\mathbf{P}^{\mathbf{yy}}_{n+1|n} \in \mathbb{R}^{d' \times d'}$.

The above assumption means that the GF proceeds as if $p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1} \left| \mathbf{y}_{1:n} \right.\right)$ was Gaussian, even if it is actually not. Indeed, it also implies that

$$\forall n \in \mathbb{N}^{*}, p_{n|n}\left(\mathbf{x}_{n}\right) = p\left(\mathbf{x}_{n} \left| \mathbf{y}_{1:n} \right.\right) = \mathcal{N}\left(\mathbf{x}_{n}; \widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n}\right); \tag{5.52a}$$

$$\forall n \in \mathbb{N}, p_{n+1|n}\left(\mathbf{x}_{n+1}\right) = p\left(\mathbf{x}_{n+1} \left| \mathbf{y}_{1:n} \right.\right) = \mathcal{N}\left(\mathbf{x}_{n+1}; \widehat{\mathbf{x}}_{n+1|n}, \widehat{\mathbf{\Gamma}}_{n+1|n}\right), \tag{5.52b}$$

where $\widehat{\mathbf{\Gamma}}_{n+1|n} = \mathbf{P}^{\mathbf{xx}}_{n+1|n}$. $(\widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n})$ are obtained as the parameters of the conditional Gaussian distribution of $\mathbf{X}_n$ given $\mathbf{Y}_{1:n}$ from (5.51):

$$\widehat{\mathbf{x}}_{n|n} = \widehat{\mathbf{x}}_{n|n-1} + \mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1}(\mathbf{y}_n - \widehat{\mathbf{y}}_{n|n-1}); \tag{5.53a}$$

$$\widehat{\mathbf{\Gamma}}_{n|n} = \widehat{\mathbf{\Gamma}}_{n|n-1} - \mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1}\mathbf{P}^{\mathbf{yx}}_{n|n-1}. \tag{5.53b}$$

Then GF computes $\widehat{\mathbf{x}}_{n+1|n+1}$ and $\widehat{\mathbf{\Gamma}}_{n+1|n+1}$ from $\widehat{\mathbf{x}}_{n|n}$, $\widehat{\mathbf{\Gamma}}_{n|n}$ and $\mathbf{y}_{n+1}$ as follows:

— *Time update*

$$\widehat{\mathbf{x}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) p_{n|n}\left(\mathbf{x}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1}; \tag{5.54a}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})^{\top} p_{n|n}\left(\mathbf{x}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1} - \widehat{\mathbf{x}}_{n+1|n}\widehat{\mathbf{x}}^{\top}_{n+1|n}. \tag{5.54b}$$

— *Measurement update*

$$\widehat{\mathbf{y}}_{n+1|n} = \int \mathbf{y}_{n+1} h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1} \mathrm{d}\mathbf{y}_{n+1}; \tag{5.55a}$$

$$\mathbf{P}^{\mathbf{xy}}_{n+1|n} = \int (\mathbf{x}_{n+1} - \widehat{\mathbf{x}}_{n+1|n})(\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|n})^{\top} h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1} \mathrm{d}\mathbf{y}_{n+1}. \tag{5.55b}$$

$$\mathbf{P}^{\mathbf{yy}}_{n+1|n} = \int \mathbf{y}_{n+1}\mathbf{y}^{\top}_{n+1} h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1} \mathrm{d}\mathbf{y}_{n+1} - \widehat{\mathbf{y}}_{n+1|n}\widehat{\mathbf{y}}^{\top}_{n+1|n}. \tag{5.55c}$$

$\widehat{\mathbf{x}}_{n+1|n+1}$ and $\widehat{\mathbf{\Gamma}}_{n+1|n+1}$ are then obtained from by applying (5.53) to $\widehat{\mathbf{x}}_{n+1|n}$, $\widehat{\mathbf{y}}_{n+1|n}$, $\widehat{\mathbf{\Gamma}}_{n+1|n}$, $\mathbf{P}^{\mathbf{xy}}_{n+1|n}$, $\mathbf{P}^{\mathbf{yy}}_{n+1|n}$, and $\mathbf{P}^{\mathbf{yx}}_{n+1|n} = (\mathbf{P}^{\mathbf{xy}}_{n+1|n})^{\top}$.

Let $\mathbf{z} = [\mathbf{x}_n, \mathbf{u}_{n+1}]$; the integrals in (5.54) are of the form

$$\int \boldsymbol{g}(\mathbf{z})\omega(\mathbf{z})\mathrm{d}\mathbf{z}, \tag{5.56}$$

where $\omega(\mathbf{z}) = p_{n|n}\left(\mathbf{x}_n\right) p\left(\mathbf{u}_{n+1}\right)$. Since $p_{n|n}\left(\mathbf{x}_n\right)$ and $p\left(\mathbf{u}_{n+1}\right)$ are Gaussian, we see that $\omega(\mathbf{z})$ is Gaussian too.

Similarly, by setting $\mathbf{z} = [\mathbf{y}_{n+1}, \mathbf{x}_{n+1}]$ and $\omega(\mathbf{z}) = h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}\right)$, the integrals in (5.55) are of the form (5.56) too. Thus, thanks to the GF, the filtering problem is reduced to evaluating integrals of the form (5.56) with Gaussian probability density function $\omega(\mathbf{z})$. In general, one cannot compute exactly (5.56). Approximate computing methodologies for such integrals is known as the Gaussian weighted integration methods and can be dealt, for example, with the Gauss-Hermite quadrature, as detailed in Appendix B. Other approaches include the Monte-Carlo integration [Kotecha and Djuric, 2003], quasi-Monte Carlo integration [Morokoff and Caflisch, 1995], spherical-radial integration rules [Monahan and Genz, 1997], unscented transform [Zoeter et al., 2004], sparse grids [Jia et al., 2012] and many others [Miller III and Rice, 1983, Lu and Darmofal, 2004, Gorynin et al., 2016b].

One can evaluate integrals in (5.54) exactly when function $\boldsymbol{f}_{n+1} : (\mathbf{x}_n, \mathbf{u}_n) \to \mathbf{x}_{n+1}$ is linear in $\mathbf{u}_n$. Similarly, integrals in (5.55) can be evaluated exactly when $h_n : (\mathbf{y}_n, \mathbf{x}_n) \to \mathbb{R}_+$ is the Gaussian probability density of $\mathbf{y}_n$ with constant variance and mean linear in $\mathbf{x}_n$. When both conditions are met, system (5.50) is known as linear Gaussian system, in which the Kalman filter allows computing the optimal filtering solution.

The Gaussian filters, which include *e.g.* the unscented and Gauss-Hermite Kalman filters, have demonstrated their suitability for a wide scope of application where the observation noise (5.50b) is additive and Gaussian.

The main drawbacks of the GF come from its fundamental approximation (5.51). Let us note the following:

- If the observation noise is heavy-tailed or affected by large outliers, then approximation (5.51) may cause a divergence of the filter, since it lacks the high-order moments of $\mathbf{Y}_n$ given $\mathbf{X}_n$ - see, *e.g.*, [Roth et al., 2013].
- The GF would always yield $\mathbf{P}^{\mathbf{xy}}_{n|n-1}\left(\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right)^{-1} = \mathbf{0}$ if the observation noise has an infinite variance, unless the state posterior variance is infinite too *cf.* (5.53). In this case, the GF never updates the measurement and therefore fails to extract any information from the observed data.
- If the observation noise is multiplicative (for example, if $h_n(\mathbf{y}_n, \mathbf{x}_n)$ is symmetric in $\mathbf{y}_n$ and only the variance of $\mathbf{Y}_n$ given $\mathbf{X}_n$ depends on $\mathbf{X}_n$, as it is the case in the stochastic volatility models), then the GF always obtains $\mathbf{P}^{\mathbf{xy}}_{n+1|n} = \mathbf{0}$ *cf.* (5.55b). In this case, the GF does not extract any information from the observed data.

The recent CGF [Singer, 2015] has been designed to be able to take a specific form of the observation equation into account and overcome the outlined drawbacks of GF. The idea was to assume (5.52), which is a consequence of (5.51), without assuming (5.51) itself. In this sense, the CGF requires a strictly weaker assumption than the original GF. In the CGF, the filtering and one-step predicting densities are assumed Gaussian:

$$p_{n|n}(\mathbf{x}_n) = p(\mathbf{x}_n | \mathbf{y}_{1:n}) = \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}, \widehat{\mathbf{\Gamma}}_{n|n}\right); \tag{5.57a}$$

$$p_{n+1|n}(\mathbf{x}_{n+1}) = p(\mathbf{x}_{n+1} | \mathbf{y}_{1:n}) = \mathcal{N}\left(\mathbf{x}_{n+1}; \widehat{\mathbf{x}}_{n+1|n}, \widehat{\mathbf{\Gamma}}_{n+1|n}\right). \tag{5.57b}$$

It means that the CGF proceeds as if these densities were Gaussian, even if they are actually not.

The classic Bayesian equations allow deriving the CGF solution. The CGF computes $\widehat{\mathbf{x}}_{n+1|n+1}$ and $\widehat{\mathbf{\Gamma}}_{n+1|n+1}$ from $\widehat{\mathbf{x}}_{n|n}$, $\widehat{\mathbf{\Gamma}}_{n|n}$ and $\mathbf{y}_{n+1}$:

— *Time update*

$$\widehat{\mathbf{x}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) p_{n|n}(\mathbf{x}_n) p(\mathbf{u}_{n+1}) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1}; \tag{5.58a}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n} = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1}) \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathbf{u}_{n+1})^\top p_{n|n}(\mathbf{x}_n) p(\mathbf{u}_{n+1}) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1} - \widehat{\mathbf{x}}_{n+1|n} \widehat{\mathbf{x}}^\top_{n+1|n}. \tag{5.58b}$$

— *Measurement update*

$$c_{n+1} = \int h_{n+1}(\mathbf{y}_{n+1}; \mathbf{x}_{n+1}) p_{n+1|n}(\mathbf{x}_{n+1}) \mathrm{d}\mathbf{x}_{n+1}; \tag{5.59a}$$

$$\widehat{\mathbf{x}}_{n+1|n+1} = \int \mathbf{x}_{n+1} \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}(\mathbf{x}_{n+1})}{c_{n+1}} \mathrm{d}\mathbf{x}_{n+1}; \tag{5.59b}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n+1} = \int \mathbf{x}_{n+1} \mathbf{x}^\top_{n+1} \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}) p_{n+1|n}(\mathbf{x}_{n+1})}{c_{n+1}} \mathrm{d}\mathbf{x}_{n+1} - \widehat{\mathbf{x}}_{n+1|n+1} \widehat{\mathbf{x}}^\top_{n+1|n+1}. \tag{5.59c}$$

In general, one cannot compute integrals in (5.58) exactly unless $\boldsymbol{f}_{n+1} : (\mathbf{x}_n, \mathbf{u}_n) \to \mathbf{x}_{n+1}$ is linear in $\mathbf{u}_n$. Similarly, one cannot compute integrals in (5.59) exactly unless the function $h_n : (\mathbf{y}_n, \mathbf{x}_n) \to \mathbb{R}_+$ is the Gaussian probability density of $\mathbf{y}_n$ with constant variance and mean linear in $\mathbf{x}_n$. The CGHF implements the Gauss-Hermite quadrature technique specified in Appendix B to compute these integrals in the case where the exact solution is unavailable.

To summarize, the PF (*cf.* Section 1.4.1) is an asymptotically optimal (in M) method of filtering which makes no assumption on the form of the conditional density. However, the computational load of PF may be too heavy to ensure an acceptable variance of the state estimate. The GF and CGF are based on nested simplifying assumptions in order to reduce the problem of filtering to the problem of computing Gaussian-weighted integrals. The GF makes a strong assumption (5.51) on the form of the joint state-observation predictive density, which may be inappropriate for several important applications. The CGF makes a strictly weaker assumption which concerns only the predictive state density. Theoretical and empirical evidence presented in [Singer, 2015] demonstrates that CGF overcomes the outlined drawbacks of the GF. All the three methods approximate $p\left(\mathbf{x}_n \mid \mathbf{y}_{1:n}\right)$ with the same complexity $\mathscr{O}\left(Mn\right)$, where M is the number of simulated particles for the PF or the total number of integration nodes used by the GF or CGF *cf.* Appendix B.

5.2.2 Filtering in switching non-linear non-Gaussian systems under the Gaussian conditional density assumption

Here we present the main contribution of this section, which consists on extending the CGF to the switching systems. These systems may be seen as hidden Markov models of type $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N}), \mathbf{Y}_{1:N})$, where $(\mathbf{X}_{1:N}, \mathrm{R}_{1:N})$ is Markovian and hidden, while $\mathbf{Y}_{1:N}$ is observed. The state variables in switching systems are of two types: continuous-valued ones, denoted by $\mathbf{X}_{1:N}$, and discrete-valued ones, denoted by $\mathrm{R}_{1:N}$. It is also assumed that $\mathrm{R}_{1:N}$ is a Markov chain, while $\mathbf{X}_{1:N}$ is Markovian given $\mathrm{R}_{1:N}$.

We consider the general form of switching systems:

$$\mathbf{X}_{n+1} = \boldsymbol{f}_{n+1}(\mathbf{X}_n, \mathrm{R}_n, \mathrm{R}_{n+1}, \mathbf{U}_{n+1}) \qquad \text{for } n = 1:N-1; \tag{5.60a}$$

$$p\left(\mathbf{y}_n \mid \mathbf{x}_n, \mathrm{r}_n\right) \propto h_n(\mathbf{y}_n, \mathbf{x}_n, \mathrm{r}_n) \qquad \text{for } n = 1:N, \tag{5.60b}$$

where $\mathrm{R}_{1:N}$ is a Markov chain in $\Omega = \{1:K\}$, $K \in \mathbb{N}^*$. We suppose that the dependency graph of this model is that of Fig. 5.11.

Let us announce the assumptions involved and derive the corresponding integral equations. Let us assume K Gaussian filtering densities and K^2 Gaussian predicting densities:

$$\forall \mathrm{r}_n \in \{1:K\}, \quad p_{n|n}\left(\mathbf{x}_n|\mathrm{r}_n\right) = p\left(\mathbf{x}_n \mid \mathrm{r}_n, \mathbf{y}_{1:n}\right) = \mathscr{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}(\mathrm{r}_n), \widehat{\boldsymbol{\Gamma}}_{n|n}(\mathrm{r}_n)\right); \tag{5.61a}$$

$$\forall \mathrm{r}_n, \mathrm{r}_{n+1} \in \{1:K\}, \quad p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right) = p\left(\mathbf{x}_{n+1} \mid \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1:n}\right) = \mathscr{N}\left(\mathbf{x}_{n+1}; \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}), \widehat{\boldsymbol{\Gamma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})\right). \tag{5.61b}$$

There are K^2 Gaussian predicting densities since the one-step ahead prediction is based on both current r_n and future r_{n+1} possible values of the discrete state.

The Switching Conditional Gaussian Filter (SCGF) computes, for each r_{n+1} in $\{1:K\}$, $\widehat{\mathbf{x}}_{n+1|n+1}\ (\mathrm{r}_{n+1})$, $\widehat{\boldsymbol{\Gamma}}_{n+1|n+1}\ (\mathrm{r}_{n+1})$ and $p\left(\mathrm{r}_{n+1} \mid \mathbf{y}_{1:n+1}\right)$ by using $\widehat{\mathbf{x}}_{n|n}(\mathrm{r}_n), \widehat{\boldsymbol{\Gamma}}_{n|n}(\mathrm{r}_n)$, $p\left(\mathrm{r}_n \mid \mathbf{y}_{1:n}\right)$, $\mathbf{y}_{n+1}$ as follows.

For each r_n and r_{n+1} in Ω,

— *Time update*:

$$\widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{u}_{n+1}) p_{n|n}\left(\mathbf{x}_n|\mathrm{r}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1}; \tag{5.62a}$$

$$\widehat{\boldsymbol{\Gamma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \int \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{u}_{n+1}) \boldsymbol{f}_{n+1}(\mathbf{x}_n, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{u}_{n+1})^\top p_{n|n}\left(\mathbf{x}_n|\mathrm{r}_n\right) p\left(\mathbf{u}_{n+1}\right) \mathrm{d}\mathbf{x}_n \mathrm{d}\mathbf{u}_{n+1} - \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) \widehat{\mathbf{x}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top. \tag{5.62b}$$

The *measurement update* consists of multiple steps:

a) for each r_n and r_{n+1} in Ω:

$$c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \int h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}, \mathrm{r}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1}; \tag{5.63a}$$

$$\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \int \mathbf{x}_{n+1} \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}, \mathrm{r}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right)}{c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})} \mathrm{d}\mathbf{x}_{n+1}; \tag{5.63b}$$

$$\widehat{\mathbf{\Gamma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \int \mathbf{x}_{n+1}\mathbf{x}_{n+1}^\top \frac{h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}, \mathrm{r}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right)}{c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})} \mathrm{d}\mathbf{x}_{n+1} - \tag{5.63c}$$

$$\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top. \tag{5.63d}$$

b) update the posterior distribution of the discrete state:

$$p\left(\mathrm{r}_n, \mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right) \propto p\left(\mathrm{r}_n\left|\mathbf{y}_{1:n}\right.\right) p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n\right.\right) c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}); \tag{5.64a}$$

$$p\left(\mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right) = \sum_{\mathrm{r}_n=1}^{K} p\left(\mathrm{r}_n, \mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right). \tag{5.64b}$$

c) derive, for each r_{n+1} in $\{1 : K\}$, $\widehat{\mathbf{x}}_{n+1|n+1}\left(\mathrm{r}_{n+1}\right)$ and $\widehat{\mathbf{\Gamma}}_{n+1|n+1}\left(\mathrm{r}_{n+1}\right)$:

$$\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1}) = \sum_{\mathrm{r}_n=1}^{K} p\left(\mathrm{r}_n\left|\mathrm{r}_{n+1}, \mathbf{y}_{1:n+1}\right.\right) \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}); \tag{5.65a}$$

$$\begin{gathered}\widehat{\mathbf{\Gamma}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) = \\ \sum_{\mathrm{r}_n=1}^{K} \left(\widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})^\top + \widehat{\mathbf{\Gamma}}_{n+1|n}(\mathrm{r}_n, \mathrm{r}_{n+1})\right) p\left(\mathrm{r}_n\left|\mathrm{r}_{n+1}, \mathbf{y}_{1:n+1}\right.\right) - \\ \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1}) \widehat{\mathbf{x}}_{n+1|n+1}(\mathrm{r}_{n+1})^\top;\end{gathered} \tag{5.65b}$$

$$p\left(\mathrm{r}_n\left|\mathrm{r}_{n+1}, \mathbf{y}_{1:n+1}\right.\right) = \frac{p\left(\mathrm{r}_n, \mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right)}{p\left(\mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right)}. \tag{5.65c}$$

d) the state estimates at the current iteration are:

$$\widehat{\mathbf{x}}_{n+1|n+1} = \sum_{\mathrm{r}_{n+1}=1}^{K} \widehat{\mathrm{x}}_{n+1|n+1}(\mathrm{r}_{n+1}) p\left(\mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right); \tag{5.66a}$$

$$\widehat{\mathrm{r}}_{n+1|n+1} = \underset{\mathrm{r}_{n+1}\in\{1:K\}}{\arg\max}\; p\left(\mathrm{r}_{n+1}\left|\mathbf{y}_{1:n+1}\right.\right). \tag{5.66b}$$

Equation (5.64a) and the idea to use the Gaussian quadrature to evaluate $c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1})$ is the key part of this work. It allows updating the posterior distribution of the discrete states. In order to justify the equation, let us decompose $p\left(\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{y}_{1:n}\right.\right)$ as follows:

$$p\left(\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{y}_{1:n}\right.\right) = p\left(\mathbf{y}_{n+1}\left|\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1:n}\right.\right) p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n, \mathbf{y}_{1:n}\right.\right) p\left(\mathrm{r}_n\left|\mathbf{y}_{1:n}\right.\right),$$

where

$$p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n, \mathbf{y}_{1:n}\right.\right) = p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n\right.\right).$$

cf. Fig. 5.11. It also follows from Fig. 5.11 that

$$p\left(\mathbf{y}_{n+1}\left|\mathbf{x}_{n+1}, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1:n}\right.\right) = p\left(\mathbf{y}_{n+1}\left|\mathbf{x}_{n+1}, \mathrm{r}_{n+1}\right.\right),$$

thus

$$\begin{aligned}p\left(\mathbf{y}_{n+1}\left|\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1:n}\right.\right) &= \int p\left(\mathbf{y}_{n+1}\left|\mathbf{x}_{n+1}, \mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{1:n}\right.\right) p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1} \\ &\propto \int h_{n+1}(\mathbf{y}_{n+1}, \mathbf{x}_{n+1}, \mathrm{r}_{n+1}) p_{n+1|n}\left(\mathbf{x}_{n+1}|\mathrm{r}_n, \mathrm{r}_{n+1}\right) \mathrm{d}\mathbf{x}_{n+1} = c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}).\end{aligned}$$

Therefore,

$$p\left(\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{n+1}\left|\mathbf{y}_{1:n}\right.\right) \propto c_{n+1}(\mathrm{r}_n, \mathrm{r}_{n+1}) p\left(\mathrm{r}_{n+1}\left|\mathrm{r}_n\right.\right) p\left(\mathrm{r}_n\left|\mathbf{y}_{1:n}\right.\right),$$

and $p(\mathrm{r}_n, \mathrm{r}_{n+1} \,|\mathbf{y}_{1:n+1}) \propto p(\mathrm{r}_n, \mathrm{r}_{n+1}, \mathbf{y}_{n+1} \,|\mathbf{y}_{1:n})$ which proves (5.64a).

Equation (5.65) is a classic moment-matching combination of the previous per-model estimates given new observation.

Finally, we define the SCGHF as an approximation of the SCGF which uses the Gaussian quadrature to evaluate (5.62) and (5.63) *cf.* Appendix B. Our algorithm approximates $p(\mathrm{r}_n, \mathbf{x}_n \,|\mathbf{y}_{1:n})$ with complexity $\mathcal{O}\left(K^2 M n\right)$, where M is the total number of integration nodes.

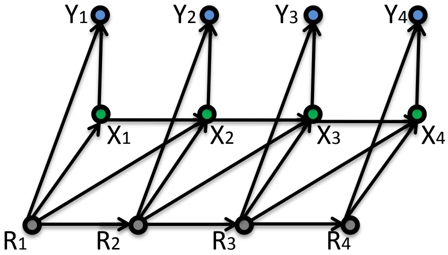


Figure 5.11: The dependency graph of Markov-switching system (5.60). Here, $(\mathrm{R}_n)_{n\in\mathbb{N}^*}$, $(\mathrm{R}_n, \mathbf{X}_n)_{n\in\mathbb{N}^*}$ and $(\mathrm{R}_n, \mathbf{X}_n, \mathbf{Y}_n)_{n\in\mathbb{N}^*}$ are Markovian.

5.2.3 Applications to switching volatility estimation

Here we compare the performance of the proposed method with the classic particle filter and the switching Kalman filter [Wu et al., 2004]. Let us consider the Markov Switching Stochastic Volatility (MSSV) model [Carvalho and Lopes, 2007]:

$$\mathrm{X}_{n+1} = \alpha_{\mathrm{R}_{n+1}} + \phi \mathrm{X}_n + \sigma \mathrm{U}_{n+1}; \quad \mathrm{X}_0 = \mathrm{x}_0; \tag{5.67a}$$

$$\mathrm{Y}_n = \exp(\mathrm{X}_n \backslash 2)\mathrm{V}_n, \tag{5.67b}$$

where for each $n \in \mathbb{N}$, $\mathrm{X}_n \in \mathbb{R}$, $\mathrm{Y}_n \in \mathbb{R}$, $\{\mathrm{R}_n\}_{n\geq 1}$ is a stationary Markov chain in $\{1 : K\}$, $\{\mathrm{U}_n\}_{n\geq 1}$, $\{\mathrm{V}_n\}_{n\geq 1}$ are i.i.d standard Gaussian variables and $(\alpha_1, \ldots, \alpha_K, \phi, \sigma, \mathrm{x}_0)$ are fixed parameters. The initial state distribution $p(\mathrm{r}_1)$ is then the eigenvector of the corresponding Markov transition matrix. When $K = 2$, the Markov chain is defined by $p_{1|1} = p(\mathrm{r}_{n+1} = 1 \,|\mathrm{r}_n = 1)$ and $p_{2|2} = p(\mathrm{r}_{n+1} = 2 \,|\mathrm{r}_n = 2)$. Realistic parameter values for $K = 2$ which we use in the experiments can be found *e.g.* in [Carvalho and Lopes, 2007].

Let $\mathrm{Z}_n = \log \mathrm{Y}_n^2$, then

$$\mathrm{Z}_n = \mathrm{X}_n + \log \mathrm{V}_n^2. \tag{5.68}$$

The model of $(\mathrm{X}_n, \mathrm{Z}_n)$ is switching linear and non-Gaussian. Thus, one can estimate the hidden variables by the switching Kalman filter. Indeed, it means approximating the distribution of $\log \mathrm{V}_n^2$ by $\mathcal{N}\left(\mathbb{E}[\log \mathrm{V}_n^2], \mathrm{Var}[\log \mathrm{V}_n^2]\right)$.

As an experiment, we perform the following experiment 100 times per each test. We begin by sampling $\{\mathrm{x}_n, \mathrm{y}_n\}_{n\in\mathbb{N}^*, n\leq N}$ from (5.67) with the parameters as in Table 5.3 with $N = 1000$. Next, we recover the state estimates from $\{\mathrm{y}_n\}_{1\leq n\leq N}$ by the SKF, the proposed method - SCGHF and the PF. The PF algorithm we use is given in Section 1.4.1. We use grids with a total of 9 integration nodes in SCGHF and 2000 particles in the particle filter. Finally, we compute the Relative Mean Squared Error (RMSE) and the MME defined by

$$\mathrm{RMSE} = \frac{1}{N} \sum_{n=1}^{N} \frac{\left(\widehat{\mathrm{x}}_{n|n}(\mathrm{y}_{1:n}) - \mathrm{x}_n\right)^2}{\mathrm{Var}(\mathbf{X}_n)} \tag{5.69}$$

and

$$\mathrm{MME} = \frac{1}{N} \sum_{n=1}^{N} \left(\widehat{\mathrm{r}}_{n|n}(\mathrm{y}_{1:n}) \neq \mathrm{r}_n\right). \tag{5.70}$$

We report in Table 5.4 the average RMSE over these experiments, and Table 5.5 presents the corresponding standard deviations. In Tables 5.4 and 5.5, PF ($M_{\min}$) refers to the PF algorithm which uses the minimal number of particle to obtain a quasi-optimal solution. We determined $M_{\min}$ by hand for each experiment, and we found $M_{\min} = 100$ for Test 1, $M_{\min} = 120$ for Test 2, $M_{\min} = 200$ for Test 3 and $M_{\min} = 50$ for Test 4, *cf.* Fig. 5.12. The related processing time

	Test 1	Test 2	Test 3	Test 4
α_1	-2.500	-1.500	-0.500	-2.500
α_2	-1.000	-0.600	-0.200	-1.000
ϕ	0.500	0.500	0.500	0.500
σ	0.100	0.100	0.100	0.100
$p_{1\|1}$	0.990	0.990	0.990	0.500
$p_{2\|2}$	0.985	0.985	0.985	0.500
x_0	-3.500	-2.100	-0.700	-3.500

Table 5.3: MSSV parameters per each test.

values are given in Table 5.6. However, these time values are supplied on an indicative basis only, since the processing time depends on the PC system configuration, processor type and settings, PF implementation and compilation details, software specifications and so on.

The simulation study indicates that the accuracy of our method is improved compared to the SKF and is nearly optimal. Note that (5.68) is linear, so the switching versions of EKF, UKF and Quadrature Kalman Filter (QKF) would produce the same result as the classic SKF in this example. That is due to the same joint state-space Gaussian approximation (5.51) involved in these approaches. Fig. 5.13 suggests that the PF should use at least 500 particles to obtain satisfactory results. Regarding the computational load, our method used only an equivalent of 40 particles and thus realized a substantial speedup. Besides, the performance of the SCGHF was nearly optimal with only 3 integration nodes per dimension. A more extensive study, has shown that the autoregressive parameters and the Markov chain transition probabilities in (5.67) did not affect the performance of the SCGHF compared to the PF. As a general conclusion of this study, we observed that there is no notable difference between the asymptotic solution of the PF and the output of the SCGHF.

An example of recovering a hidden trajectory by the SCGHF and the SKF is presented in Fig. 5.14. Fig. 5.15 is related to Fig. 5.14 and presents a comparative density plot with profiles of $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ estimated by the SKF, SCGHF and PF at $n = 580$. Indeed, the PF does not provide any analytic expression of the underlying distribution approximating $p(\mathbf{x}_n|\mathbf{y}_{1:n})$, as opposed to the SKF and SCGHF. Thus, we used a kernel smoothing technique to estimate the approximating distribution of the PF from the locations of the particles.

	SKF		SCGHF		PF		PF ($M_{\min}$)	
	RMSE	MME	RMSE	MME	RMSE	MME	RMSE	MME
Test 1	0.1565	0.0630	0.0909	0.0402	0.0917	0.0405	0.1046	0.0437
Test 2	0.3166	0.1118	0.1987	0.0729	0.2008	0.0735	0.2165	0.0775
Test 3	0.7876	0.2808	0.6411	0.2137	0.6481	0.2160	0.6801	0.2274
Test 4	0.8217	0.3624	0.7394	0.3326	0.7408	0.3334	0.7712	0.3479

Table 5.4: The RMSE and MME statistics for the SKF, SCGHF and PF with different MSSV parameters from Table 5.3. PF ($M_{\min}$) refers to the PF algorithm which uses the minimal number of particle to obtain a quasi-optimal solution. We use $M_{\min} = 100$ for Test 1, $M_{\min} = 120$ for Test 2, $M_{\min} = 200$ for Test 3 and $M_{\min} = 50$ for Test 4.

	SKF		SCGHF		PF	
	std[RMSE]	std[MME]	std[RMSE]	std[MME]	std[RMSE]	std[MME]
Test 1	0.0206	0.0085	0.0130	0.0060	0.0132	0.0060
Test 2	0.0402	0.0154	0.0243	0.0095	0.0247	0.0096
Test 3	0.0518	0.0296	0.0488	0.0247	0.0510	0.0254
Test 4	0.0148	0.0064	0.0138	0.0063	0.0140	0.0065

Table 5.5: Standard deviations of the RMSE and MME statistics estimated for the SKF, SCGHF and PF with different MSSV parameters from Table 5.3.

	Test 1	Test 2	Test 3	Test 4
SKF	0.05	0.05	0.05	0.05
SCGHF	0.08	0.08	0.08	0.08
PF	1.12	1.12	1.12	1.12
PF (M_{min})	0.35	0.37	0.42	0.32

Table 5.6: Processing times (in seconds) for SKF, SCGHF and PF required to process a trajectory of length $N = 1000$ in the framework of the MSSV model, per each test. PF (M_{min}) refers to the PF algorithm which uses the minimal number of particle to obtain a quasi-optimal solution. We use $M_{\text{min}} = 100$ for Test 1, $M_{\text{min}} = 120$ for Test 2, $M_{\text{min}} = 200$ for Test 3 and $M_{\text{min}} = 50$ for Test 4.

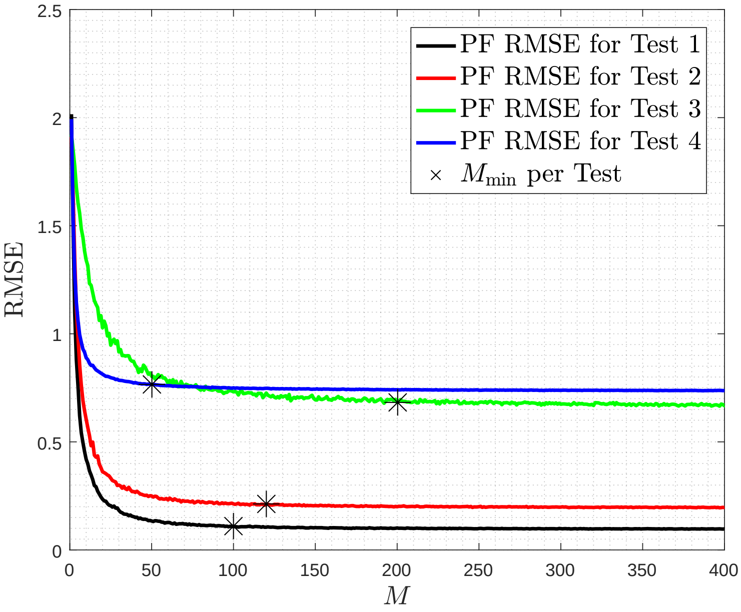


Figure 5.12: The RMSE of the PF in the MSSV model compared in function of number of particles M, and minimum numbers of particles that would result in a nearly optimal solution which are $M_{\text{min}} = 100$ for Test 1, $M_{\text{min}} = 120$ for Test 2, $M_{\text{min}} = 200$ for Test 3 and $M_{\text{min}} = 50$ for Test 4.

5.3 Conclusion

In the first section, we proposed an original algorithm for smoothing in stationary SLDSs, and, more generally, in CGPMSMs. The algorithm is based on two filters which run independently in the direct and reverse order. The outputs of these filters are combined by using the dynamics of the system. The algorithm is fast and appears as an interesting alternative to the particle smoother methods. Comparison with the results produced by the particle smoother show that the approximation error of our method is negligibly small.

In the second section, we introduced and tested a novel general deterministic method of filtering in switching systems. A simulation study confirmed that the new algorithm has an improved accuracy and robustness compared to the classic approach. Mean squared error measures of the proposed method are practically optimal, while its computational load is low when compared to the particle filter. The algorithm is applicable to a large class of switching models which involve regime changes, strong non-linearity and non-Gaussian distributions.

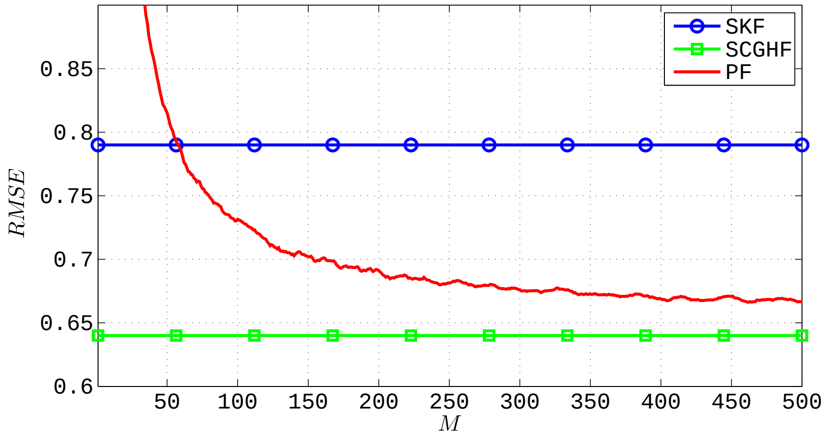


Figure 5.13: The RMSE of the PF in the MSSV model compared to that of the SKF and SCGHF in function of the number of particles. MSSV parameters are those from Test 3 in Table 5.3.

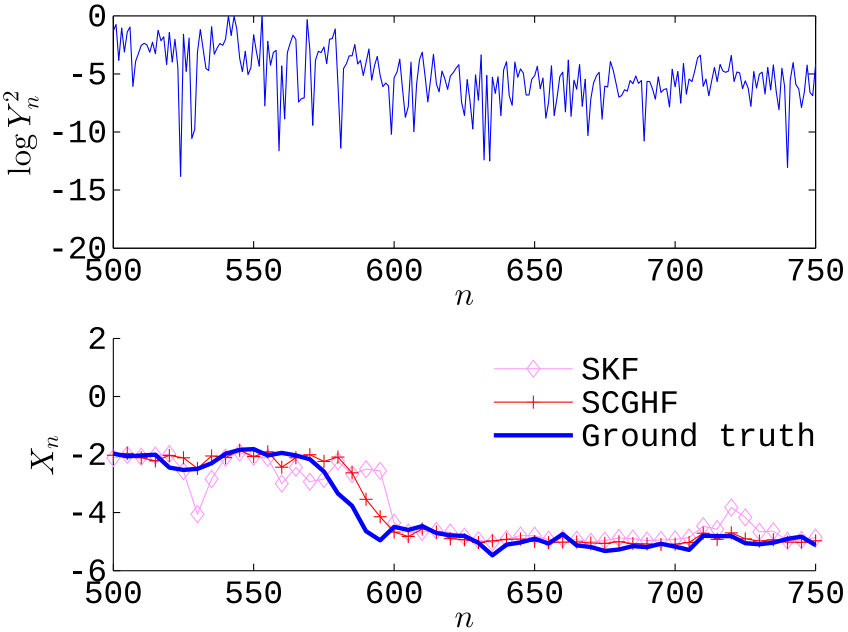


Figure 5.14: Example of a state estimation in the MSSV model with the SCGHF and SKF. MSSV parameters are those from Test 3 in Table 5.3. The ground truth trajectory switches at $n = 575$.

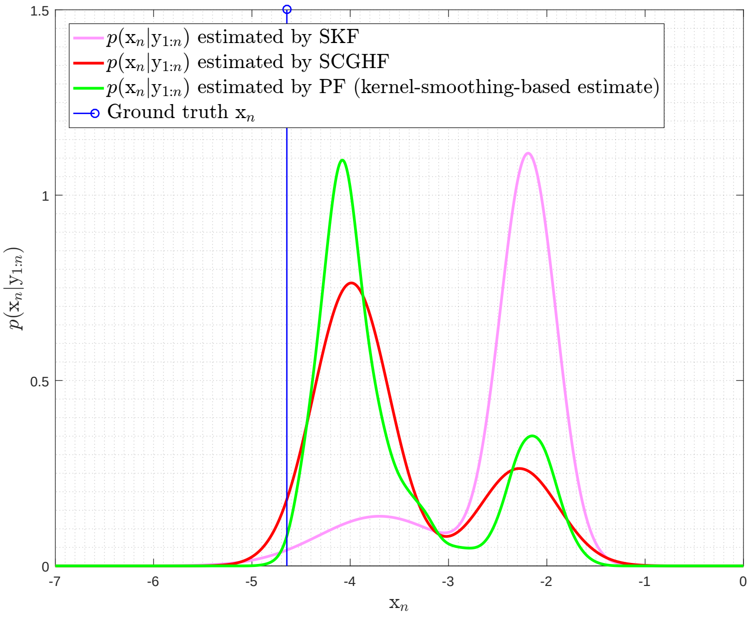


Figure 5.15: Comparative plot with profiles of $p(\mathrm{x}_n|\mathrm{y}_{1:n})$ for $n = 580$ estimated by the SKF, SCGHF and PF, related to the trajectory from Fig. 5.14.

Chapter 6

Conclusion

This report is concerned with the Partially Observable Markov Process (POMP) framework which is widely used in a range of important applications. The sequential Monte-Carlo methods are common approaches of Bayesian state estimation in POMPs. They are asymptotically optimal, but may need a considerable computational cost. The report is devoted to the alternative methods of state estimation in POMPs, developed by the author in order to allow an accurate state estimation for a lower computational cost.

Chapter 2 explores the Conditionally Gaussian Observed Markov Switching Model (CGOMSM). This model allows a practical implementation of the corresponding exact filter and smoother. The chapter also details the principle of Learned Conditionally Gaussian Observed Markov Switching Model Filter (LCGOMSMF) and Learned Conditionally Gaussian Observed Markov Switching Model Smoother (LCGOMSMS). The Expectation-Maximization (EM) algorithm, derived by the author for the CGOMSM framework, allows approximating an arbitrary stationary Markovian process by a CGOMSM, which is used in both LCGOMSMF and LCGOMSMS.

Chapter 3 introduces a general-purpose computational technique for Bayesian state estimation in POMPs, called Markovian Grid-Based State Estimator (MGSE). It is based on Markov-like properties of the grid weight function. The author provides a construction allowing obtaining a convergent solution.

Chapter 4 contains an extensive comparison among Maximum Posterior Mode (MPM) estimators based on the classic Hidden Markov Model (HMM) and its extensions which are the Pairwise Markov Model (PMM) and the Triplet Markov Model (TMM). PMM and TMM frameworks allowed to achieve substantial improvements of the estimation accuracy. Such improvements were particularly visible when the observation distribution was heavily autocorrelated and/or if the hidden chain was far from being Markovian. The author also contributed in defining an PMM-based modeling of assets' log-returns and backtesting it.

Chapter 5 introduces a novel smoothing technique for the Switching Linear Dynamical System (SLDS) and, more generally, for the Conditionally Gaussian Pairwise Markov Switching Model (CGPMSM). This technique provided interesting results on a real-world data example. The chapter also introduces an extension of the Conditional Gaussian Filter (CGF) to the hybrid-state POMPs proposed by the author. The main reason for considering such an extension is that the classic Gaussian Filter (GF) approach has several important drawbacks.

The accuracy of the proposed methods has been compared with that of the sequential Monte-Carlo methods and has been shown to be competitive. The pertinence and suitability of the research for real-world applications has been confirmed by an extensive experiment-driven study. We also notice that the case of a high-dimensional state space should not be a problem for the methods proposed, while this case may be problematic for a range of sequential Monte-Carlo methods.

Appendix A

Matrix characterization of conditional independence in Gaussian vectors

Here we recall a classic result which provides conditional distributions in a Gaussian vector. We derive from it a matrix formula characterizing conditional independence of Gaussian variables which make part of a Gaussian vector.

Proposition 19. *Let $a, b \in \mathbb{N}^*$, $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian vector in $\mathbb{R}^{a+b}$ partitioned as follows*

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}, \tag{A.1}$$

where $\mathbf{X}_1 \in \mathbb{R}^a$, $\mathbf{X}_2 \in \mathbb{R}^b$, and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ accordingly partitioned as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Gamma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Gamma}_2 \end{bmatrix}, \tag{A.2}$$

where $\boldsymbol{\mu}_1 \in \mathbb{R}^a$, $\boldsymbol{\mu}_2 \in \mathbb{R}^b$, $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{a\times a}$, $\boldsymbol{\Gamma}_2 \in \mathbb{R}^{b\times b}$ and $\boldsymbol{\Sigma}_{12} \in \mathbb{R}^{a\times b}$. Then the distribution of $\mathbf{X}_1$ given $\mathbf{X}_2 = \mathbf{x}_2$ is Gaussian with mean vector $\boldsymbol{\mu}_{1|2}$ and covariance $\boldsymbol{\Gamma}_{1|2}$ defined as follows:

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\Sigma}_{12}\boldsymbol{\Gamma}_2^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2), \qquad \boldsymbol{\Gamma}_{1|2} = \boldsymbol{\Gamma}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{12}^\top. \tag{A.3}$$

The following lemma characterizes the conditional independence of Gaussian variables within a Gaussian vector.

Lemma 1. *Let $a, b, c \in \mathbb{N}^*$, $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a Gaussian vector in $\mathbb{R}^{a+b+c}$ partitioned as follows*

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix}, \tag{A.4}$$

where $\mathbf{X}_1 \in \mathbb{R}^a$, $\mathbf{X}_2 \in \mathbb{R}^b$, $\mathbf{X}_3 \in \mathbb{R}^c$, and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ accordingly partitioned as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Gamma}_1 & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13} \\ \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Gamma}_2 & \boldsymbol{\Sigma}_{23} \\ \boldsymbol{\Sigma}_{13}^\top & \boldsymbol{\Sigma}_{23}^\top & \boldsymbol{\Gamma}_3 \end{bmatrix}, \tag{A.5}$$

where $\boldsymbol{\mu}_1 \in \mathbb{R}^a$, $\boldsymbol{\mu}_2 \in \mathbb{R}^b$, $\boldsymbol{\mu}_3 \in \mathbb{R}^c$, $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{a\times a}$, $\boldsymbol{\Gamma}_2 \in \mathbb{R}^{b\times b}$, $\boldsymbol{\Gamma}_3 \in \mathbb{R}^{c\times c}$, $\boldsymbol{\Sigma}_{12} \in \mathbb{R}^{a\times b}$, $\boldsymbol{\Sigma}_{13} \in \mathbb{R}^{a\times c}$ and $\boldsymbol{\Sigma}_{23} \in \mathbb{R}^{b\times c}$. Then $\mathbf{X}_1$ and $\mathbf{X}_3$ are independent given $\mathbf{X}_2$ if and only if

$$\boldsymbol{\Sigma}_{13} = \boldsymbol{\Sigma}_{12}^\top\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{23}. \tag{A.6}$$

Proof. The distribution $p\left(\mathbf{x}_1, \mathbf{x}_3 \,|\mathbf{x}_2\right)$ is Gaussian and *cf.* Proposition 19, and its covariance matrix is

$$\begin{bmatrix} \boldsymbol{\Gamma}_1 & \boldsymbol{\Sigma}_{13} \\ \boldsymbol{\Sigma}_{13}^\top & \boldsymbol{\Gamma}_3 \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{23}^\top \end{bmatrix} \boldsymbol{\Gamma}_2^{-1} \begin{bmatrix} \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_{23} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_{13} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{23} \\ \boldsymbol{\Sigma}_{13}^\top - \boldsymbol{\Sigma}_{23}^\top\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Gamma}_3 - \boldsymbol{\Sigma}_{23}^\top\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{23} \end{bmatrix} \tag{A.7}$$

□

$\mathbf{X}_1$ and $\mathbf{X}_3$ are independent given $\mathbf{X}_2$ if and only if the matrix in the Right Hand Side Term (RHS) of the above equation is block-diagonal, *i.e.* $\boldsymbol{\Sigma}_{13} - \boldsymbol{\Sigma}_{12}^\top\boldsymbol{\Gamma}_2^{-1}\boldsymbol{\Sigma}_{23} = \mathbf{0}$.

Appendix B

Constructing multivariate Gauss-Hermite quadrature

Here we present the construction of one-dimensional and multidimensional Gauss-Hermite quadrature rules. The Gauss-Hermite quadrature which is an algorithm of approximation of the Gaussian-weighted integral, *i.e.* an integral of the form

$$\int \boldsymbol{g}(\mathbf{z})\omega(\mathbf{z})\mathrm{d}\mathbf{z}, \tag{B.1}$$

with Gaussian probability density function $\omega(\mathbf{z})$.

Let us first consider the case of one-dimensional Gaussian-weighted integral, where $\omega(\mathrm{z})$ is the standard normal distribution:

$$\omega(\mathrm{z}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\mathrm{z}^2}{2}\right).$$

An N-point Gauss-Hermite quadrature rule is an approximation to (B.1) of the form

$$I \approx \sum_{q=1}^{N} \pi_q g(\xi_q), \tag{B.2}$$

where points $(\xi_q)_{1\leq q\leq N}$ (integration nodes) and weights $(\pi_q)_{1\leq q\leq N}$ are such that (B.2) is exact if g is a polynomial up to the $(2N-1)^{\mathrm{th}}$ order.

In order to compute the parameters of the N-point Gauss-Hermite quadrature, one uses the first $(2N-1)$ moments of $\omega(\mathrm{z})$:

$$\forall i \in \mathbb{N}, i \leq N-1, n_i = \int \mathrm{z}^i \omega(\mathrm{z})\mathrm{d}\mathrm{z} = \left\{ \begin{array}{cc} (i-1)!! & \text{for even } i \\ 0 & \text{for odd } i \end{array} \right. .$$

where $i!!$ denotes the double factorial, *i.e.* the product of all numbers from i to 1 that have the same parity as i.

Next, one defines the following polynomial recursion $\{P_i\}_{i=1}^{N}$:

$$P_{i+1}(\mathrm{z}) = (\mathrm{z} - \delta_{i+1})P_i(\mathrm{z}) - \gamma_{i+1}^2 P_{i-1}(\mathrm{z}) \text{ for } i \geq 0,$$

where $\forall \mathrm{z}, P_{-1}(\mathrm{z}) = 0$, $P_0(\mathrm{z}) = 1$, $\gamma_1 = 0$ and

$$\delta_{i+1} = \frac{\mathbb{E}[\mathrm{z}P_i^2(\mathrm{z})]}{\mathbb{E}[P_i^2(\mathrm{z})]}; \gamma_{i+1}^2 = \frac{\mathbb{E}[P_i^2(\mathrm{z})]}{\mathbb{E}[P_{i-1}^2(\mathrm{z})]}.$$

The quadrature nodes are the roots of P_N and the quadrature weights are the solution of the linear system

$$\sum_{q=1}^{N} \pi_q P_i(\xi_q) = \left\{ \begin{array}{cc} 1 & \text{for } i = 0 \\ 0 & \text{for } i \in \{1, \ldots, N-1\} \end{array} \right. . \tag{B.3}$$

Now we consider the case of d-dimensional Gaussian-weighted integral:

$$\boldsymbol{J} = \int \boldsymbol{g}(\mathbf{z}) \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{\mathbf{z}^\top \mathbf{z}}{2}\right) \mathrm{d}\mathbf{z}, \tag{B.4}$$

where function $\boldsymbol{g}$ is weighted by standard normal distribution in $\mathbb{R}^d$. Since this distribution is the product of one-dimensional standard normal distributions, we can approximate (B.4) by successively applying (B.2) and obtain

$$\boldsymbol{J} \approx \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} \cdots \sum_{q_d=1}^{N} \pi_{q_1}\pi_{q_2}\ldots\pi_{q_d}\boldsymbol{g}([\xi_1,\xi_2,\ldots,\xi_d]^\top). \tag{B.5}$$

Here, the total number of grid points is N^d.

Finally, consider the general case of d-dimensional Gaussian-weighted integral:

$$\boldsymbol{K} = \int \boldsymbol{g}(\mathbf{z})\frac{1}{(2\pi)^{d/2}\det(\boldsymbol{\Sigma})^{1/2}}\exp\left(-\frac{(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})}{2}\right)\mathrm{d}\mathbf{z}. \tag{B.6}$$

where function $\boldsymbol{g}$ is weighted by a normal distribution in $\mathbb{R}^d$ with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$.

The technique of stochastic decoupling defines a linear transformation allowing to write (B.6) in the form (B.4). Let us denote by $\boldsymbol{C}$ a Cholesky decomposition of $\boldsymbol{\Sigma}$, *i.e.* a matrix that verifies

$$\boldsymbol{C}\boldsymbol{C}^\top = \boldsymbol{\Sigma}. \tag{B.7}$$

Such a matrix exists and is invertible provided that $\boldsymbol{\Sigma}$ is positive definite. Consider the following linear transformation:

$$\mathbf{v} = \boldsymbol{C}^{-1}(\mathbf{z}-\boldsymbol{\mu}), \tag{B.8}$$

thus

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{C}\mathbf{v}. \tag{B.9}$$

The Jacobian of (B.9) is $\nabla_{\mathbf{v}}(\mathbf{z}) = \boldsymbol{C}$ and we have from (B.7) $\det(\boldsymbol{\Sigma})^{1/2} = \det\boldsymbol{C} = \det\nabla_{\mathbf{v}}(\mathbf{z})$, so we obtain by substituting (B.9) into (B.6):

$$\boldsymbol{K} = \int \boldsymbol{g}(\boldsymbol{\mu}+\boldsymbol{C}\mathbf{v})\frac{1}{(2\pi)^{d/2}}\exp\left(-\frac{\mathbf{v}^\top\mathbf{v}}{2}\right)\mathrm{d}\mathbf{v}. \tag{B.10}$$

Thus, we obtain the Gauss-Hermite quadrature rule for (B.6) by applying (B.5) to (B.10):

$$\boldsymbol{K} \approx \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} \cdots \sum_{q_d=1}^{N} \pi_{q_1}\pi_{q_2}\ldots\pi_{q_d}\boldsymbol{g}(\boldsymbol{\mu}+\boldsymbol{C}[\xi_1,\xi_2,\ldots,\xi_d]^\top). \tag{B.11}$$

As a result, we see that once the integration nodes $(\xi_q)_{1\leq q\leq N}$ and weights $(\pi_q)_{1\leq q\leq N}$ are known, one can approximate (B.6) by (B.11). The complexity of computation of a Cholesky decomposition of $\boldsymbol{\Sigma}$ is $\mathcal{O}\left(d^3\right)$, while the complexity of evaluating of (B.11) is $\mathcal{O}\left(N^d\right)$.

Appendix C

Proof of the EM algorithm for the CGOMSM

This Annex is concerned with the derivation of the EM Algorithm presented in Section 2.3. We begin with a recall of on the weighted least squares regression. Next, we use these results in the derivation of the EM algorithm for the CGOMSM.

C.1 Weighted least squares regression

Let $N \in \mathbb{N}^*, d \in \mathbb{N}, d' \in \mathbb{N}$, $\boldsymbol{x}_{1:N}$, $\boldsymbol{y}_{1:N}$, $\pi_{1:N}$ sequences taking values in $\mathbb{R}^d$, $\mathbb{R}^{d'}$ and $\mathbb{R}_+$ respectively. The weighted least squares regression consists in finding parameters $(\mathscr{A}, \mathscr{B}, \mathscr{R})$ in $\left(\mathbb{R}^{d' \times d}, \mathbb{R}^{d'}, \mathcal{S}_{++}^{d'}\right)$ which maximize

$$\sum_{n=1}^{N} \pi_n \log \mathscr{N}(\boldsymbol{y}_n; \mathscr{A}\boldsymbol{x}_n + \mathscr{B},\ \mathscr{R}).$$

Proposition 20 establishes a closed-form expression of $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{R}$ in function of $\boldsymbol{x}_{1:N}$, $\boldsymbol{y}_{1:N}$ and $\pi_{1:N}$.

Lemma 2. *Define function $\Psi_d : \mathcal{S}_{++}^d \to \mathbb{R}$ by:*

$$\Psi_d(\boldsymbol{M}) = \log |\boldsymbol{M}| - \operatorname{tr}(\boldsymbol{M}).$$

Then,

$$\operatorname{argmax}_{\ \boldsymbol{M} \in \mathcal{S}_{++}^d} \left[\Psi_d(\boldsymbol{M})\right] = \boldsymbol{I}_d, \tag{C.1}$$

where $\boldsymbol{I}_d$ is the identity matrix in $\mathbb{R}^{d \times d}$.

Proof. In the case where $d = 1$, function Ψ_1 is defined by

$$\forall x \in \mathbb{R}_+^*,\ \Psi_1(x) = \log(x) - x$$

and attains its maximum at $x = 1$. By induction, suppose that we have proven (C.1) for some $d \in \mathbb{N}^*$, let us prove it for $d+1$. Let $\boldsymbol{M}$ be a matrix in $\mathcal{S}_{++}^{d+1}$. It can be represented in a block-wise form as follows:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{12}^\top & M_{22} \end{bmatrix},$$

where $\boldsymbol{M}_{11} \in \mathcal{S}_{++}^d$, $M_{22} \in \mathbb{R}_+^*$, $\boldsymbol{M}_{12} \in \mathbb{R}^{d \times 1}$. Define $\boldsymbol{S}$ in $\mathcal{S}_{++}^{d+1}$ by:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_{11} & \boldsymbol{S}_{12} \\ \boldsymbol{S}_{12}^\top & S_{22} \end{bmatrix} = \operatorname{argmax}_{\ \boldsymbol{M}_{11}, \boldsymbol{M}_{12}, M_{22}, \boldsymbol{M} \in \mathcal{S}_{++}^{d+1}} \left[\Psi_{d+1}(\boldsymbol{M})\right].$$

The block determinant formula for the matrix $\boldsymbol{M}$ yields:

$$|\boldsymbol{M}| = |\boldsymbol{M}_{11}| \left| M_{22} - \boldsymbol{M}_{12} \boldsymbol{M}_{11}^{\boldsymbol{-1}} \boldsymbol{M}_{12}^\top \right|.$$

Then,

$$\mathit{\Psi}_{d+1}\left(\boldsymbol{M}\right) = \log\left|\boldsymbol{M}_{11}\right| - \operatorname{tr}\left(\boldsymbol{M}_{11}\right) + \log\left|M_{22} - \boldsymbol{M}_{12}\boldsymbol{M}_{11}^{\boldsymbol{-1}}\boldsymbol{M}_{12}^{\top}\right| - \operatorname{tr}\left(M_{22}\right).$$

Since $\boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}$, then $\boldsymbol{M}_{\boldsymbol{11}}^{\boldsymbol{-1}} \in \mathscr{S}_{++}^{d}$ and

$$M_{22} \geq \boldsymbol{M}_{22} - \boldsymbol{M}_{12}\boldsymbol{M}_{11}^{\boldsymbol{-1}}\boldsymbol{M}_{12}^{\top},$$

with equality if and only if $\boldsymbol{M}_{12} = \boldsymbol{0}$. Thus, $\boldsymbol{S}_{12} = \boldsymbol{0}$. Next, we have

$$\begin{aligned}&\operatorname{argmax}_{\ \boldsymbol{M}_{11}, \boldsymbol{M}_{12}=\boldsymbol{0}, M_{22}, \boldsymbol{M} \in \mathscr{S}_{++}^{q+1}}\left[\mathit{\Psi}_{d+1}\left(\boldsymbol{M}\right)\right] = \\ &\quad \operatorname{argmax}_{\ \boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}, M_{22} \in \mathbb{R}_{+}^{*}}\left[\left(\log\left|\boldsymbol{M}_{11}\right| - \operatorname{tr}\left(\boldsymbol{M}_{11}\right)\right) + \left(\log\left|M_{22}\right| - \operatorname{tr}(M_{22})\right)\right].\end{aligned}$$

Since we have:

$$\begin{aligned}&\max_{\ \boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}, M_{22} \in \mathbb{R}_{+}^{*}}\left[\left(\log\left|\boldsymbol{M}_{11}\right| - \operatorname{tr}\left(\boldsymbol{M}_{11}\right)\right) + \left(\log\left|M_{22}\right| - \operatorname{tr}\left(M_{22}\right)\right)\right] = \\ &\quad \max_{\ \boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}}\left[\log\left|\boldsymbol{M}_{11}\right| - \operatorname{tr}\left(\boldsymbol{M}_{11}\right)\right] + \max_{\ M_{22} \in \mathbb{R}_{+}^{*}}\left[\log\left|M_{22}\right| - \operatorname{tr}\left(M_{22}\right)\right] = \\ &\quad \max_{\ \boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}}\left[\mathit{\Psi}_{d}\left(\boldsymbol{M}_{11}\right)\right] + \max_{\ M_{22} \in \mathbb{R}_{+}^{*}}\left[\mathit{\Psi}_{1}\left(M_{22}\right)\right],\end{aligned}$$

then, by induction,

$$\boldsymbol{S}_{11} = \operatorname{argmax}_{\ \boldsymbol{M}_{11} \in \mathscr{S}_{++}^{d}}\left[\mathit{\Psi}_{d}\left(\boldsymbol{M}_{11}\right)\right] = \boldsymbol{I}_{d}$$

and

$$S_{22} = \operatorname{argmax}_{\ M_{22} \in \mathbb{R}_{+}^{*}}\left[\mathit{\Psi}_{1}\left(M_{22}\right)\right] = 1.$$

Therefore, $\boldsymbol{S} = \boldsymbol{I}_{d+1}$ what proves the induction hypothesis.

□

Proposition 20. *Suppose that* $\mathscr{W} = \sum_{n=1}^{N}\pi_n > 0$.

$$f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right) = \sum_{n=1}^{N}\pi_n \log\mathscr{N}\left(\boldsymbol{y}_n; \mathscr{A}\boldsymbol{x}_n + \mathscr{B},\ \mathscr{R}\right).$$

Then the maximum of f *with respect to* $\mathscr{A}$**,** $\mathscr{B}$ *and* $\mathscr{R}$ *is given by:*

$$\begin{bmatrix} \mathscr{A}_0 & \mathscr{B}_0 \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{x}_n^{\top} & \sum_{n=1}^{N}\pi_n\boldsymbol{y}_n \end{bmatrix}\begin{bmatrix} \sum_{n=1}^{N}\pi_n\boldsymbol{x}_n\boldsymbol{x}_n^{\top} & \sum_{n=1}^{N}\pi_n\boldsymbol{x}_n \\ \sum_{n=1}^{N}\pi_n\boldsymbol{x}_n^{\top} & \mathscr{W} \end{bmatrix}^{-1};$$

$$\mathscr{R}_{\boldsymbol{0}} = \frac{1}{\mathscr{W}}\left(\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{y}_n^{\top} - \mathscr{A}_0\sum_{n=1}^{N}\pi_n\boldsymbol{x}_n\boldsymbol{y}_n^{\top} - \mathscr{B}_0\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n^{\top}\right).$$

Proof. Observe that

$$\begin{aligned}&f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right) = \\ &\quad -\sum_{n=1}^{N}\pi_n\frac{d'}{2}\left(\log 2\pi\right) - \frac{1}{2}\sum_{n=1}^{N}\pi_n\log\left|\mathscr{R}\right| - \frac{1}{2}\sum_{n=1}^{N}\pi_n(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B})^{\top}\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right),\end{aligned}$$

and

$$\begin{aligned}&\sum_{n=1}^{N}\pi_n(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B})^{\top}\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right) = \\ &\quad \sum_{n=1}^{N}\pi_n\operatorname{tr}\left[\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right)^{\top}\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right)\right] = \\ &\quad \sum_{n=1}^{N}\pi_n\operatorname{tr}\left[\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right)\left(\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}\right)^{\top}\right].\end{aligned}$$

Let us define, $\widetilde{\mathscr{A}} \in \mathbb{R}^{d' \times (d+1)}$ by $\widetilde{\mathscr{A}} = \begin{bmatrix} \mathscr{A} & \mathscr{B} \end{bmatrix}$. Let $\widetilde{\boldsymbol{x}}_n = \begin{bmatrix} \boldsymbol{x}_n \\ 1 \end{bmatrix}$ for each n, then

$$\sum_{n=1}^{N} \pi_n (\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B})^{\top} \mathscr{R}^{-1} (\boldsymbol{y}_n - \mathscr{A}\boldsymbol{x}_n - \mathscr{B}) = \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}\, \widetilde{\boldsymbol{x}}_n \right) \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}} \widetilde{\boldsymbol{x}}_n \right)^{\top} \right].$$

and

$$f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right) = -\frac{\mathscr{W}}{2} \left(d' \log 2\pi + \log|\mathscr{R}| + \frac{1}{\mathscr{W}} \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}\, \widetilde{\boldsymbol{x}}_n \right) \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}\, \widetilde{\boldsymbol{x}}_n \right)^{\top} \right] \right).$$

Recall that we can solve the optimization problem

$$f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right) \to \max$$

by considering the double optimization:

$$\max_{\mathscr{A} \in \mathbb{R}^{d' \times d}, \mathscr{B} \in \mathbb{R}^{d'}, \mathscr{R} \in \mathscr{S}_{++}^{d'}} f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right) = \max_{\mathscr{R} \in \mathscr{S}_{++}^{d'}} \left[\max_{\widetilde{\mathscr{A}} \in \mathbb{R}^{d' \times (d+1)}} f_{\mathscr{R}}\left(\widetilde{\mathscr{A}} \right) \right],$$

where $f_{\mathscr{R}}\left(\widetilde{\mathscr{A}}\right) \in \mathscr{F}\left(\mathbb{R}^{d' \times (d+1)} \to \mathbb{R}\right)$ is defined by $f_{\mathscr{R}}\left(\widetilde{\mathscr{A}}\right) = f\left(\mathscr{A}, \mathscr{B}, \mathscr{R}\right)$.

Let us first prove that $f_{\mathscr{R}}$ is concave. By the affine map invariance property of the concave functions, it is equivalent to proving that the real function h

$$h(t) = f_{\mathscr{R}}(\boldsymbol{U} + t\boldsymbol{V})$$

is concave in t for any $\boldsymbol{U}$, $\boldsymbol{V}$ in $\mathbb{R}^{d' \times (d+1)}$.

We have

$$\begin{aligned} h(t) = & -\frac{\mathscr{W}}{2} \left(d' \log 2\pi + \log|\mathscr{R}| \right) - \frac{1}{2} \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} (\boldsymbol{y}_n \boldsymbol{y}_n^{\top} - 2\boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{U}^{\top} + \boldsymbol{U} \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{U}^{\top}) \right] \\ & - t \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} (\boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{V}^{\top} - \boldsymbol{U} \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{V}^{\top}) \right] - \frac{t^2}{2} \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{V}^{\top} \right]. \end{aligned}$$

We see that h is a second order polynomial and thus h is concave if and only if

$$\sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{V}^{\top} \right] \geq 0.$$

Observe that we have:

$$\sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} \boldsymbol{V}^{\top} \right] = \sum_{n=1}^{N} \pi_n \mathrm{tr} \left[\widetilde{\boldsymbol{x}}_n \boldsymbol{V}^{\top} \mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \right] = \sum_{n=1}^{N} \pi_n (\boldsymbol{V} \widetilde{\boldsymbol{x}}_n)^{\top} \mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n.$$

Since $\mathscr{R}^{-1} \in \mathscr{S}_{++}^{q}$, for any n, $(\boldsymbol{V}\widetilde{\boldsymbol{x}}_n)^{\top} \mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \geq 0$ and $\pi_n \geq 0$ so

$$\sum_{n=1}^{N} \pi_n (\boldsymbol{V} \widetilde{\boldsymbol{x}}_n)^{\top} \mathscr{R}^{-1} \boldsymbol{V} \widetilde{\boldsymbol{x}}_n \geq 0,$$

therefore $f_{\mathscr{R}}$ is concave and we can find its global maximum by solving $\frac{\partial f_{\mathscr{R}}}{\partial \widetilde{\mathscr{A}}} = 0$.
The differentiation of $f_{\mathscr{R}}$ with respect to $\widetilde{\mathscr{A}}$ yields:

$$\frac{\partial f_{\mathscr{R}}}{\partial \widetilde{\mathscr{A}}} = \mathscr{R}^{-1} \sum_{n=1}^{N} \pi_n \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}\, \widetilde{\boldsymbol{x}}_n \right) \widetilde{\boldsymbol{x}}_n^{\top}.$$

Since $\mathscr{R} \in \mathscr{S}_{++}^{q}$, $\frac{\partial f_{\mathscr{R}}}{\partial \widetilde{\mathscr{A}}} \left(\widetilde{\mathscr{A}_{\mathbf{0}}} \right) = 0$ is equivalent to

$$\sum_{n=1}^{N} \pi_n \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}_{\mathbf{0}}}\, \widetilde{\boldsymbol{x}}_n \right) \widetilde{\boldsymbol{x}}_n^{\top} = 0,$$

which, in turn, is equivalent to

$$\sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} = \widetilde{\mathscr{A}}_{\mathbf{0}} \sum_{n=1}^{N} \pi_n \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top}.$$

Moreover,

$$\sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} = \sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \left[\begin{array}{cc} \widetilde{\boldsymbol{x}}_n^{\top} & 1 \end{array} \right] = \left[\begin{array}{cc} \sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} & \sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \end{array} \right]$$

and

$$\widetilde{\mathscr{A}}_{\mathbf{0}} \sum_{n=1}^{N} \pi_n \; \widetilde{\boldsymbol{x}}_n \widetilde{\boldsymbol{x}}_n^{\top} {=} \widetilde{\mathscr{A}}_{\mathbf{0}} \left[\begin{array}{cc} \sum_{n=1}^{N} \pi_n \boldsymbol{x}_n \widetilde{\boldsymbol{x}}_n^{\top} & \sum_{n=1}^{N} \pi_n \boldsymbol{x}_n \\ \sum_{n=1}^{N} \pi_n \widetilde{\boldsymbol{x}}_n^{\top} & \mathscr{W} \end{array} \right].$$

Therefore,

$$\widetilde{\mathscr{A}}_{\mathbf{0}} = \left[\begin{array}{cc} \mathscr{A}_0 & \mathscr{B}_0 \end{array} \right] = \left[\begin{array}{cc} \sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \widetilde{\boldsymbol{x}}_n^{\top} & \sum_{n=1}^{N} \pi_n \boldsymbol{y}_n \end{array} \right] \left[\begin{array}{cc} \sum_{n=1}^{N} \pi_n \boldsymbol{x}_n \widetilde{\boldsymbol{x}}_n^{\top} & \sum_{n=1}^{N} \pi_n \boldsymbol{x}_n \\ \sum_{n=1}^{N} \pi_n \widetilde{\boldsymbol{x}}_n^{\top} & \mathscr{W} \end{array} \right]^{-1},$$

and $\widetilde{\mathscr{A}}_{\mathbf{0}}$ is a global maximum of $f_{\mathscr{R}}$.
Let us define function $g \in \mathscr{F}\left(\mathcal{S}_{++}^{d} \to \mathbb{R}\right)$ by:

$$g\left(\mathscr{R}\right) = \max_{\widetilde{\mathscr{A}} \in \mathbb{R}^{d' \times (d+1)}} f_{\mathscr{R}}\left(\widetilde{\mathscr{A}}\right).$$

We have

$$g\left(\mathscr{R}\right) = f\left(\mathscr{A}_0, \mathscr{B}_0, \mathscr{R}\right) = -\frac{\mathscr{W}}{2}\left(d'\text{log}2\pi + \log\left|\mathscr{R}\right| + \frac{1}{\mathscr{W}}\sum_{n=1}^{N} \pi_n \text{tr}\left[\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)^{\top}\right]\right).$$

Next,

$$\frac{1}{\mathscr{W}}\sum_{n=1}^{N} \pi_n \text{tr}\left[\mathscr{R}^{-1}\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)^{\top}\right] = \text{tr}\left[\mathscr{R}^{-1}\frac{1}{\mathscr{W}}\sum_{n=1}^{N} \pi_n \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)^{\top}\right].$$

Define

$$\mathscr{R}_{\mathbf{0}} = \frac{1}{\mathscr{W}}\sum_{n=1}^{N} \pi_n \left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\ \widetilde{\boldsymbol{x}}_n\right)^{\top}.$$

We then have

$$g\left(\mathscr{R}\right) = -\frac{\mathscr{W}}{2}\left(d\text{log}2\pi + \log\left|\mathscr{R}\right| + \text{tr}\left[\mathscr{R}^{-1}\mathscr{R}_{\mathbf{0}}\right]\right).$$

Moreover,

$$\begin{gathered} g\left(\mathscr{R}\right) = -\frac{\mathscr{W}}{2}\left(d'\text{log}2\pi - \log\left|\mathscr{R}^{-1}\right| + \text{tr}\left[\mathscr{R}^{-1}\mathscr{R}_{\mathbf{0}}\right]\right) = \\ -\frac{\mathscr{W}}{2}\left(d'\text{log}2\pi - \Psi_{d'}\left(\mathscr{R}^{-1}\mathscr{R}_{\mathbf{0}}\right) + \log\left|\mathscr{R}_{\mathbf{0}}\right|\right). \end{gathered}$$

Since $\mathscr{R}_{\mathbf{0}}$ is independent from $\mathscr{R}$, the optimization of g with respect to $\mathscr{R}$ is equivalent to maximizing $\Psi_{d'}\left(\mathscr{R}^{-1}\mathscr{R}_{\mathbf{0}}\right)$. Thus, we conclude from Lemma 2 that the unique maximum of g is $\mathscr{R}_{\mathbf{0}}$. Finally,

$$\mathscr{R}_{\mathbf{0}} = \text{argmax}_{\mathscr{R} \in \mathcal{S}_{++}^{d'}} \max_{\mathscr{A} \in \mathbb{R}^{d' \times d}, \mathscr{B} \in \mathbb{R}^{d'}} f_{\mathscr{R}}\left(\mathscr{A}, \mathscr{B}\right)$$

and $\left[\begin{array}{cc}\mathscr{A}_0 & \mathscr{B}_0\end{array}\right] = \operatorname{argmax}_{\mathscr{A}\in\mathbb{R}^{d'\times d},\mathscr{B}\in\mathbb{R}^{d'}} f_{\mathscr{R}_{\mathbf{0}}}(\mathscr{A},\mathscr{B})$, which allows to accomplish the proof. In addition, we have

$$\mathscr{R}_{\mathbf{0}} = \frac{1}{\mathscr{W}}\sum_{n=1}^{N}\pi_n\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\,\widetilde{\boldsymbol{x}}_n\right)\left(\boldsymbol{y}_n - \widetilde{\mathscr{A}}_{\mathbf{0}}\,\widetilde{\boldsymbol{x}}_n\right)^{\top} =$$
$$\frac{1}{\mathscr{W}}\left(\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{y}_n^{\top} - \widetilde{\mathscr{A}}_{\mathbf{0}}\sum_{n=1}^{N}\pi_n\widetilde{\boldsymbol{x}}_n\boldsymbol{y}_n^{\top} - \sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\widetilde{\boldsymbol{x}}_n^{\top}\widetilde{\mathscr{A}}_{\mathbf{0}}^{\top} + \widetilde{\mathscr{A}}_{\mathbf{0}}\sum_{n=1}^{N}\pi_n\widetilde{\boldsymbol{x}}_n\widetilde{\boldsymbol{x}}_n^{\top}\widetilde{\mathbf{A}}_{\mathbf{0}}^{\top}\right) =$$
$$\frac{1}{\mathscr{W}}\left(\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{y}_n^{\top} - \widetilde{\mathscr{A}}_{\mathbf{0}}\sum_{n=1}^{N}\pi_n\widetilde{\boldsymbol{x}}_n\boldsymbol{y}_n^{\top}\right) = \frac{1}{\mathscr{W}}\left(\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{y}_n^{\top} - \left[\begin{array}{cc}\mathscr{A}_0 & \mathscr{B}_0\end{array}\right]\sum_{n=1}^{N}\pi_n\left[\begin{array}{c}\boldsymbol{x}_n\\ 1\end{array}\right]\boldsymbol{y}_n^{\top}\right) =$$
$$\frac{1}{\mathscr{W}}\left(\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n\boldsymbol{y}_n^{\top} - \mathscr{A}_0\sum_{n=1}^{N}\pi_n\boldsymbol{x}_n\boldsymbol{y}_n^{\top} - \mathscr{B}_0\sum_{n=1}^{N}\pi_n\boldsymbol{y}_n^{\top}\right).$$

□

C.2 The EM algorithm for the CGOMSM

Here we suppose that we are given a training sample $(\mathbf{x}_{1:N},\mathbf{y}_{1:N})$. The object of this section is to present a derivation of the EM algorithm applied to estimate the CGOMSM parameters from $(\mathbf{x}_{1:N},\mathbf{y}_{1:N})$. We recall that the CGOMSM model of triplet $(\mathbf{X}_{1:N},\mathbf{Y}_{1:N},\mathrm{R}_{1:N})$ in $\mathbb{R}^d\times\mathbb{R}^{d'}\times\Omega$ is parameterized by $\boldsymbol{\theta}$, where

$$\boldsymbol{\theta} = \left\{\boldsymbol{\mu}_i^{(\boldsymbol{\theta})},\boldsymbol{\Gamma}_i^{(\boldsymbol{\theta})},p_{ij}^{(\boldsymbol{\theta})},\boldsymbol{A}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{B}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{C}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{D}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{F}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{H}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{\Pi}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{\Lambda}_{ij}^{(q)}\,|1\leq i,j\leq K\right\}, \tag{C.2}$$

and $\Omega=\{1:K\}$. Here, for simplicity, we use another parameterization of the CGOMSM, which is:

$$\boldsymbol{\theta} = \left\{p_{j|i}^{\boldsymbol{\theta}},\boldsymbol{A}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{B}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{C}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{D}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{F}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{H}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{\Pi}_{ij}^{(\boldsymbol{\theta})},\boldsymbol{\Lambda}_{ij}^{(q)}\,|1\leq i,j\leq K\right\}, \tag{C.3}$$

where for each i,j in Ω, $p_{j|i}^{\boldsymbol{\theta}}$ in (C.3) is defined by

$$p_{j|i}^{\boldsymbol{\theta}} = \frac{p_{i,j}^{\boldsymbol{\theta}}}{\sum_{j'=1}^{K}p_{i,j'}^{\boldsymbol{\theta}}}.$$

$\boldsymbol{\theta}$ in (C.3) parameterizes only the transitions $p(\mathbf{x}_{n+1},\mathbf{y}_{n+1},\mathrm{r}_{n+1}\mid\mathbf{x}_n,\mathbf{y}_n,\mathrm{r}_n)$ for $n\in\{1:N-1\}$ and is estimated by using the EM algorithm. Once (C.3) is estimated, (C.2) is chosen consistently with (C.3) knowing the fact that $(\mathbf{X}_{1:N},\mathbf{Y}_{1:N},\mathrm{R}_{1:N})$ is stationary.

The EM algorithm is an iterative method to find parameter estimates of a statistical model, where the model depends on unobserved latent variables. The EM iteration alternates between performing an Expectation step of the EM algorithm (E-step), which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and an Maximization step of the EM algorithm (M-step), which computes parameters maximizing the expected log-likelihood found on the E-step.

Let q denote the iteration count of the EM algorithm. The parameter value at the q-th iteration is denoted by $\boldsymbol{\theta}^{(q)}$, where we set for simplicity:

$$\boldsymbol{\theta}^{(q)} = \left\{p_{j|i}^{(q)},\boldsymbol{A}_{ij}^{(q)},\boldsymbol{B}_{ij}^{(q)},\boldsymbol{C}_{ij}^{(q)},\boldsymbol{D}_{ij}^{(q)},\boldsymbol{F}_{ij}^{(q)},\boldsymbol{H}_{ij}^{(q)},\boldsymbol{\Pi}_{ij}^{(q)},\boldsymbol{\Lambda}_{ij}^{(q)}\,|1\leq i,j\leq K\right\},$$

At the E-step, we compute

$$\mathfrak{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right) = \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{X}_{1:N},\mathbf{Y}_{1:N},\mathrm{R}_{1:N}\right)|\mathbf{X}_{1:N}=\mathbf{x}_{1:N},\mathbf{Y}_{1:N}=\mathbf{y}_{1:N}\right].$$

In the case of the CGOMSM, it involves computing the posterior distribution of the hidden states $\mathrm{R}_{1:N}$ conditional on the input data $(\mathbf{x}_{1:N},\mathbf{y}_{1:N})$ and the current parameter value $\boldsymbol{\theta}^{(q)}$. Specifically, (2.48) computes

$$\psi_n^{(q)}(i,j) = \mathbb{P}_{\boldsymbol{\theta}^{(q)}}\left[\mathrm{R}_n=i,\mathrm{R}_{n+1}=j|\mathbf{X}_{1:N}=\mathbf{x}_{1:N},\mathbf{Y}_{1:N}=\mathbf{y}_{1:N}\right],$$

in a CGOMSM, which appear when computing $\mathfrak{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$.

At the M-step, the new parameter estimate $\boldsymbol{\theta}^{(q+1)}$ is computed as follows:

$$\boldsymbol{\theta}^{(q+1)} = \text{argmax}_{\ \boldsymbol{\theta}} \left[\mathfrak{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)\right].$$

We accomplish this step by using the formulas (2.43)-(2.44) from Chapter 2. The point of what follows is to clarify computing and maximization of $\mathfrak{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$.

We have:

$$p\left(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}, \mathrm{r}_{1:N}\right) = p\left(\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1\right) \prod_{n=1}^{N-1} p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}, \mathrm{r}_{n+1} \mid \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n\right),$$

from the Markovianity of $\left(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N}, \mathrm{R}_{1:N}\right)$. Next,

$$p\left(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}, \mathrm{r}_{n+1} \mid \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n\right) = p\left(\mathrm{r}_{n+1} \mid \mathrm{r}_n\right) p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n, \mathbf{y}_{n+1}, \mathrm{r}_{n+1}\right) p\left(\mathbf{y}_{n+1} \mid \mathbf{y}_n, \mathrm{r}_n, \mathrm{r}_{n+1}\right),$$

in a CGOMSM, with

$$p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n = i, \mathbf{y}_{n+1}, \mathrm{r}_{n+1} = j\right) = \mathcal{N}\left(\mathbf{x}_{n+1}; \boldsymbol{A}_{ij}\mathbf{x}_n + \boldsymbol{B}_{ij}\mathbf{y}_n + \boldsymbol{C}_{ij}\mathbf{y}_{n+1} + \boldsymbol{F}_{ij},\ \boldsymbol{\Pi}_{ij}(\boldsymbol{\Pi}_{ij})^\top\right);$$

$$p\left(\mathbf{y}_{n+1} \mid \mathbf{y}_n, \mathrm{r}_n = i, , \mathrm{r}_{n+1} = j\right) = \mathcal{N}\left(\mathbf{y}_{n+1}; \boldsymbol{D}_{ij}\mathbf{y}_n + \boldsymbol{H}_{ij},\ \boldsymbol{\Lambda}_{ij}(\boldsymbol{\Lambda}_{ij})^\top\right).$$

Let us compute $\mathfrak{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$. An expression for $\log p\left(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}, \mathrm{r}_{1:N}\right)$ is:

$$\begin{aligned}
&\log p\left(\mathbf{x}_{1:N}, \mathbf{y}_{1:N}, \mathrm{r}_{1:N}\right) = \\
&\quad \log p\left(\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1\right) + \sum_{n=1}^{N-1} \log p(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}, \mathrm{r}_{n+1} | \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n) = \log p\left(\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1\right) + \\
&\quad \sum_{n=1}^{N-1} \log p(\mathrm{r}_{n+1} | \mathrm{r}_n) + \sum_{n=1}^{N-1} \log p(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n, \mathbf{y}_{n+1}, \mathrm{r}_{n+1}) + \sum_{n=1}^{N-1} \log p(\mathbf{y}_{n+1} | \mathbf{y}_n, \mathrm{r}_n, \mathrm{r}_{n+1}).
\end{aligned}$$

We have

$$\begin{aligned}
\mathfrak{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) &= \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1\right) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right] + \\
&\sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathrm{r}_{n+1} | \mathrm{r}_n) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right] + \\
&\sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n, \mathbf{y}_{n+1}, \mathrm{r}_{n+1}) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right] + \\
&\sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{y}_{n+1} | \mathbf{y}_n, \mathrm{r}_n, \mathrm{r}_{n+1}) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right] = \\
&\quad \mathfrak{Q}_0\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) + \mathfrak{Q}_1\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) + \mathfrak{Q}_2\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) + \mathfrak{Q}_3\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right),
\end{aligned}$$

where

$$\begin{aligned}
\mathfrak{Q}_0\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) &= \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}_1, \mathbf{y}_1, \mathrm{r}_1\right) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\right]; \\
\mathfrak{Q}_1\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) &= \sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathrm{r}_{n+1} | \mathrm{r}_n) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right]; \\
\mathfrak{Q}_2\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) &= \sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_{n+1} | \mathbf{x}_n, \mathbf{y}_n, \mathrm{r}_n, \mathbf{y}_{n+1}, \mathrm{r}_{n+1}) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right]; \\
\mathfrak{Q}_3\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) &= \sum_{n=1}^{N-1} \mathbb{E}_{\boldsymbol{\theta}^{(q)}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{y}_{n+1} | \mathbf{y}_n, \mathrm{r}_n, \mathrm{r}_{n+1}) | \mathbf{X}_{1:N} = \mathbf{x}_{1:N}, \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\ \right].
\end{aligned}$$

Next, we have

$$\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}p_{\boldsymbol{\theta}}(\mathrm{r}_{n+1}=j|\mathrm{r}_n=i)=\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}p_{j|i}^{\boldsymbol{\theta}}\ ;$$

$$\mathfrak{Q}_2\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}p_{\boldsymbol{\theta}}(\mathbf{x}_{n+1}|\mathbf{x}_n,\mathbf{y}_n,\mathrm{r}_n=i,\mathbf{y}_{n+1},\mathrm{r}_{n+1}=j)\ =$$

$$\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}\mathscr{N}\left(\mathbf{x}_{n+1};\boldsymbol{A}_{ij}^{\boldsymbol{\theta}}\mathbf{x}_n+\boldsymbol{B}_{ij}^{\boldsymbol{\theta}}\mathbf{y}_n+\boldsymbol{C}_{ij}^{\boldsymbol{\theta}}\mathbf{y}_{n+1}+\boldsymbol{F}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right);$$

$$\mathfrak{Q}_3\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}p_{\boldsymbol{\theta}}(\mathbf{y}_{n+1}|\mathbf{y}_n,\mathrm{r}_n=i,\mathrm{r}_{n+1}=j)\ =$$

$$\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}\mathscr{N}\left(\mathbf{y}_{n+1};\boldsymbol{D}_{ij}^{\boldsymbol{\theta}}\mathbf{y}_n+\boldsymbol{H}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right).$$

The M-step consists of maximization of $\mathfrak{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta_q}\right)$ with respect to $\boldsymbol{\theta}$.

We assume that $\mathfrak{Q}_0\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ does not contribute significantly to the value of $\mathfrak{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$, thus we drop this first term from the equation. The remaining component are $\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$, $\mathfrak{Q}_2\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$, $\mathfrak{Q}_3\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ and they can be maximized independently, which leads to maximizing their sum and therefore maximizing $\mathfrak{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$.

— Maximizing $\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$:
Let us recall that

$$\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\mathrm{log}p_{j|i}^{\boldsymbol{\theta}}\ .$$

The above expression can be developed to

$$\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{i\in\Omega}\sum_{j\in\Omega}\sum_{n=1}^{N-1}\psi_n^{(q)}(i,j)\mathrm{log}p_{j|i}^{\boldsymbol{\theta}}\ =\sum_{i=1}^{K}\mathfrak{T}(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}),$$

where for each $i\in\Omega$, $\mathfrak{T}\left(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\sum_{j=1}^{K}\sum_{n=1}^{N-1}\psi_n^{(q)}(i,j)\mathrm{log}p_{j|i}^{\boldsymbol{\theta}}$. Since for each $i\in\Omega$, $p_{j|i}^{\boldsymbol{\theta}}$ are linked only by equation

$$\forall i\in\Omega,\sum_{j=1}^{K}p_{j|i}^{\boldsymbol{\theta}}=1,$$

$\mathfrak{Q}_1\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ can be maximized by maximizing independently each $\mathfrak{T}(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)})$ constrained to $\sum_{j=1}^{K}p_{j|i}^{\boldsymbol{\theta}}=1$, and for each value of i.
Consider for each $i\in\Omega$,

$$\mathscr{L}\left(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)=\mathfrak{T}\left(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)-\lambda\left(\sum_{j=1}^{K}p_{j|i}^{\boldsymbol{\theta}}-1\right),$$

where λ is a Lagrange multiplier. The differentiation of $\mathscr{L}\left(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ with respect to $p_{j|i}^{\boldsymbol{\theta}}$ yields

$$\frac{\partial\mathscr{L}\left(i,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)}{\partial p_{j|i}^{\boldsymbol{\theta}}}=\sum_{n=1}^{N-1}\psi_n^{(q)}(i,j)\frac{1}{p_{j|i}^{\boldsymbol{\theta}}}-\lambda.$$

We note by $\left\{p_{j|i}^{(q+1)}\right\}_{j=1}^{K}$ the values of $\left\{p_{j|i}^{\boldsymbol{\theta}}\right\}_{j=1}^{K}$ which maximize $\mathfrak{T}(i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ constrained to

$$\sum_{j=1}^{K} p_{j|i}^{\boldsymbol{\theta}} = 1.$$

From $\left.\frac{\partial \mathscr{L}\left(i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)}{\partial p_{j|i}^{\boldsymbol{\theta}}}\right|_{p_{j|i}^{(q+1)}} = 0$ for each $j \in \Omega$, we have

$$p_{j|i}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \psi_n^{(q)}(i,j)}{\lambda}$$

Since $\sum_{j=1}^{K} p_{j|i}^{(q+1)} = 1$, we have $\lambda = \sum_{j=1}^{K} \sum_{n=1}^{N-1} \psi_n^{(q)}(i,j)$. Therefore, the M-step formula for $p_{j|i}^{(q)}$ is:

$$p_{j|i}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \psi_n^{(q)}(i,j)}{\sum_{j=1}^{K} \sum_{n=1}^{N-1} \psi_n^{(q)}(i,j)}.$$

— Maximizing $\mathfrak{Q}_2\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$

Let us recall that

$$\mathfrak{Q}_2\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = \sum_{n=1}^{N-1} \sum_{i,j \in \Omega} \psi_n^{(q)}(i,j) \log \mathscr{N}\left(\mathbf{x}_{n+1}; \boldsymbol{A}_{ij}^{\boldsymbol{\theta}} \mathbf{x}_n + \boldsymbol{B}_{ij}^{\boldsymbol{\theta}} \mathbf{y}_n + \boldsymbol{C}_{ij}^{\boldsymbol{\theta}} \mathbf{y}_{n+1} + \boldsymbol{F}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right).$$

Let us consider, for each i, j, $1 \leq i, j \leq K$,

$$p_{ij}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \psi_n^{(q)}(i,j)}{N-1}.$$

Define, for each i, j, $1 \leq i, j \leq K$,

$$\mathfrak{U}\left(i, j, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = \sum_{n=1}^{N-1} \frac{\psi_n^{(q)}(i,j)}{p_{ij}^{(q+1)}} \log \mathscr{N}\left(\mathbf{x}_{n+1}; \boldsymbol{A}_{ij}^{\boldsymbol{\theta}} \mathbf{x}_n + \boldsymbol{B}_{ij}^{\boldsymbol{\theta}} \mathbf{y}_n + \boldsymbol{C}_{ij}^{\boldsymbol{\theta}} \mathbf{y}_{n+1} + \boldsymbol{F}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Pi}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right),$$

Thus,

$$\mathfrak{Q}_2\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right) = \sum_{i,j \in \Omega} \mathfrak{U}\left(i, j, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$$

Each term $\mathfrak{U}\left(i, j, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$ from the above equation can be maximized independently from each other.

The maximization of $\sum_{n=1}^{N} \pi_n \log \mathscr{N}(\mathbf{y}_n; \mathscr{A}\mathbf{x}_n + \mathscr{B},\ \mathscr{R})$ for any $\boldsymbol{x}_{1:N}$, $\boldsymbol{y}_{1:N}$, $\pi_{1:N}$ is the object of Proposition 20.

Thus, we can set:

$$\forall n \in \{1 : N-1\}, \qquad \begin{bmatrix} \mathbf{x}_n \\ \mathbf{y}_n \\ \mathbf{y}_{n+1} \end{bmatrix} \rightarrow \boldsymbol{x}_n, \qquad \boldsymbol{x}_{n+1} \rightarrow \boldsymbol{y}_n, \qquad \frac{\psi_n^{(q)}(i,j)}{p_{ij}^{(q+1)}} \rightarrow \pi_n$$

thus $\mathscr{W} = \sum_{n=1}^{N} \pi_n = (N-1)$ and we find from the Proposition 20 that the values $\boldsymbol{A}_{ij}^{(q+1)}$, $\boldsymbol{B}_{ij}^{(q+1)}$, $\boldsymbol{C}_{ij}^{(q+1)}$, $\boldsymbol{F}_{ij}^{(q+1)}$, $\boldsymbol{C}_{ij}^{(q+1)}$, $\boldsymbol{\Pi}_{ij}^{(q+1)}$ which maximize $\mathfrak{U}\left(i, j, \boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}\right)$ are given by:

$$\left[\boldsymbol{A}_{ij}^{(q+1)} \boldsymbol{B}_{ij}^{(q+1)} \boldsymbol{C}_{ij}^{(q+1)} \boldsymbol{F}_{ij}^{(q+1)}\right] =$$

$$\begin{bmatrix} \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_{n+1} \mathbf{x}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_{n+1} \mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_{n+1} \mathbf{y}_{n+1}^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_{n+1} \end{bmatrix} \times$$

$$\begin{bmatrix} \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_n \mathbf{x}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_n \mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_n \mathbf{y}_{n+1}^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_n \\ \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_n \mathbf{x}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_n \mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_n \mathbf{y}_{n+1}^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_n \\ \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_{n+1} \mathbf{x}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_{n+1} \mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_{n+1} \mathbf{y}_{n+1}^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_{n+1} \\ \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{x}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}} \sum_{n=1}^{N} \psi_n^{(q)}(i,j) \mathbf{y}_{n+1}^{\top} & N-1 \end{bmatrix}^{-1}$$

$$\boldsymbol{\Pi}_{ij}^{(q+1)}\left(\boldsymbol{\Pi}_{ij}^{(q+1)}\right)^{\top} =$$

$$\frac{1}{N-1}\left(\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{x}_{n+1}\mathbf{x}_{n+1}^{\top} - \left[\boldsymbol{A}_{ij}^{(q+1)}\ \boldsymbol{B}_{ij}^{(q+1)}\ \boldsymbol{C}_{ij}^{(q+1)}\right]\begin{bmatrix}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{x}_n\mathbf{x}_{n+1}^{\top}\\ \frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_n\mathbf{x}_{n+1}^{\top}\\ \frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_{n+1}\mathbf{x}_{n+1}^{\top}\end{bmatrix} - \boldsymbol{F}_{ij}^{(q+1)}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{x}_{n+1}\right).$$

— Maximizing $\mathfrak{Q}_3\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$
Let us recall that

$$\mathfrak{Q}_3\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right) = \sum_{n=1}^{N-1}\sum_{i,j\in\Omega}\psi_n^{(q)}(i,j)\log\mathscr{N}\left(\mathbf{y}_{n+1};\boldsymbol{D}_{ij}^{\boldsymbol{\theta}}\mathbf{y}_n+\boldsymbol{H}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right).$$

Let us consider, for each i,j, $1\le i,j\le K$,
$\mathfrak{V}\left(i,j,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right) = \sum_{n=1}^{N-1}\psi_n^{(q)}(i,j)\log\mathscr{N}\left(\mathbf{y}_{n+1};\boldsymbol{D}_{ij}^{\boldsymbol{\theta}}\mathbf{y}_n+\boldsymbol{H}_{ij}^{\boldsymbol{\theta}},\ \boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\left(\boldsymbol{\Lambda}_{ij}^{\boldsymbol{\theta}}\right)^{\top}\right)$. Next,

$$\mathfrak{Q}_3\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right) = \sum_{i,j\in\Omega}\mathfrak{V}\left(i,j,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right),$$

and each term $\mathfrak{V}\left(i,j,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ can be maximized independently from each other.
Let us set then

$$\forall n\in\{1:N-1\}, \mathbf{y}_n\to\boldsymbol{x}_n,\ \mathbf{y}_{n+1}\to\boldsymbol{y}_n, \frac{\psi_n^{(q)}(i,j)}{p_{ij}^{(q+1)}}\to\pi_n$$

thus $\mathscr{W}=\sum_{n=1}^{N}\pi_n=(N-1)$ and we find from the Proposition 20 that the values $\boldsymbol{D}_{ij}^{(q+1)}$, $\boldsymbol{H}_{ij}^{(q+1)}$, $\boldsymbol{\Lambda}_{ij}^{(q+1)}$ which maximize $\mathfrak{V}\left(i,j,\boldsymbol{\theta},\boldsymbol{\theta}^{(q)}\right)$ are given by:

$$\begin{aligned}&\left[\begin{array}{cc}\boldsymbol{D}_{ij}^{(q+1)} & \boldsymbol{H}_{ij}^{(q+1)}\end{array}\right] =\\ &\left[\begin{array}{cc}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_{n+1}\mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_{n+1}\end{array}\right]\times\\ &\left[\begin{array}{cc}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_n\mathbf{y}_n^{\top} & \frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_n\\ \frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_n^{\top} & N-1\end{array}\right]^{-1};\end{aligned}$$

$$\boldsymbol{\Lambda}_{ij}^{(q+1)}\left(\boldsymbol{\Lambda}_{ij}^{(q+1)}\right)^{\top} =$$

$$\frac{1}{N-1}\left(\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_{n+1}\mathbf{y}_{n+1}^{\top} - \boldsymbol{D}_{ij}^{(q+1)}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_n\mathbf{y}_{n+1}^{\top} - \boldsymbol{H}_{ij}^{(q+1)}\frac{1}{p_{ij}^{(q+1)}}\sum_{n=1}^{N}\psi_n^{(q)}(i,j)\mathbf{y}_{n+1}^{\top}\right).$$

As a result, we derived closed-form M-step update formulas for $p_{j|i}^{(q+1)}$, $\boldsymbol{A}_{ij}^{(q+1)}$, $\boldsymbol{B}_{ij}^{(q+1)}$, $\boldsymbol{C}_{ij}^{(q+1)}$, $\boldsymbol{F}_{ij}^{(q+1)}$, $\boldsymbol{C}_{ij}^{(q+1)}$, $\boldsymbol{\Pi}_{ij}^{(q+1)}$, $\boldsymbol{D}_{ij}^{(q+1)}$, $\boldsymbol{H}_{ij}^{(q+1)}$ and $\boldsymbol{\Lambda}_{ij}^{(q+1)}$.

Bibliography

[Abbassi et al., 2015] Abbassi, N., Benboudjema, D., Derrode, S., and Pieczynski, W. (2015). Optimal Filter Approximations in Conditionally Gaussian Pairwise Markov Switching Models. *IEEE Transactions on Automatic Control*, 60(4):1104–1109.

[Abbassi et al., 2011] Abbassi, N., Benboudjema, D., and Pieczynski, W. (2011). Kalman filtering approximations in triplet Markov Gaussian switching models. In *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*, pages 77–80.

[Ades and Van Leeuwen, 2015] Ades, M. and Van Leeuwen, P. (2015). The equivalent-weights particle filter in a high-dimensional system. *Quarterly Journal of the Royal Meteorological Society*, 141(687):484–503.

[Afshari et al., 2017] Afshari, H., Gadsden, S., and Habibi, S. (2017). Gaussian Filters for Parameter and State Estimation: A General Review of Theory and Recent Trends. *Signal Processing*, 135:218 – 238.

[Andrieu et al., 2003a] Andrieu, C., Davy, M., and Doucet, A. (2003a). Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, 51(7):1762–1770.

[Andrieu and Doucet, 2002] Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):827–836.

[Andrieu et al., 2010] Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

[Andrieu et al., 2003b] Andrieu, C., Freitas, N., Doucet, A., and Jordan, M. (2003b). An Introduction to MCMC for Machine Learning. *Machine learning*, 50(1-2):5–43.

[Azzouzi and Nabney, 1999] Azzouzi, M. and Nabney, I. (1999). Modelling financial time series with switching state space models. In *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, pages 240–249, New York City, USA.

[Banga et al., 1992] Banga, C., Ghorbel, F., and Pieczynski, W. (1992). Unsupervised Bayesian classifier applied to the segmentation of retina image. In *Proceedings of the 14th Annual International Conference on the IEEE Engineering in Medicine and Biology Society*, pages 1847–1848, Paris, France.

[Barber, 2006] Barber, D. (2006). Expectation correction for smoothed inference in switching linear dynamical systems. *The Journal of Machine Learning Research*, 7:2515–2540.

[Barbu and Limnios, 2016] Barbu, S. and Limnios, N., editors (2016). *Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications.* Springer-Verlag, New York.

[Barrett, 1961] Barrett, W. (1961). Convergence properties of Gaussian quadrature formulae. *The Computer Journal*, 3(4):272–277.

[Benhamou et al., 2010] Benhamou, E., Gobet, E., and Miri, M. (2010). Time dependent Heston model. *SIAM Journal on Financial Mathematics*, 1(1):289–325.

[Beskos et al., 2017] Beskos, A., Crisan, D., Jasra, A., Kamatani, K., and Zhou, Y. (2017). A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability*, 49(1):24–48.

[Bhar and Hamori, 2006] Bhar, R. and Hamori, S. (2006). *Hidden Markov Models: Applications to Financial Economics*, volume 40. Springer Science & Business Media.

[Billingsley, 2013] Billingsley, P. (2013). *Convergence of probability measures*. John Wiley & Sons.

[Blanchet-Scalliet, 2001] Blanchet-Scalliet, C. (2001). *Processus à sauts et risque de défaut*. PhD thesis, Université d'Evry-Val d'Essonne.

[Blanchet-Scalliet et al., 2007] Blanchet-Scalliet, C., Diop, A., Gibson, R., Talay, D., and Tanré, E. (2007). Technical Analysis Compared to Mathematical Models Based Methods under Parameters Mis-specification. *Journal of Banking & Finance*, 31(5):1351–1373.

[Böhning, 1992] Böhning, D. (1992). Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44(1):197–200.

[Boudaren et al., 2011] Boudaren, M., Pieczynski, W., and Monfrini, E. (2011). Unsupervised segmentation of non stationary data hidden with non stationary noise. In *Proceedings of the International Workshop on Systems, Signal Processing and their Applications(WOSSPA)*, pages 255–258, Tipaza, Algeria.

[Bricq et al., 2006] Bricq, S., Collet, C., and Armspach, J. (2006). Triplet Markov chain for 3D MRI brain segmentation using a probabilistic atlas. In *Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 386–389, Arlington, VA, USA.

[Briers et al., 2010] Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing Algorithms for State–Space Models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89.

[Bungartz and Griebel, 2004] Bungartz, H.-J. and Griebel, M. (2004). Sparse grids. *Acta numerica*, 13(1):147–269.

[Caflisch, 1998] Caflisch, R. E. (1998). Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49.

[Cam et al., 2008] Cam, S. L., Salzenstein, F., and Collet, C. (2008). Fuzzy pairwise Markov chain to segment correlated noisy data. *Signal Processing*, 88(10):2526 – 2541.

[Cappé et al., 2005] Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer-Verlag.

[Caron et al., 2007] Caron, F., Davy, M., Duflos, E., and Vanheeghe, P. (2007). Particle Filtering for Multisensor Data Fusion with Switching Observation Models: Application to Land Vehicle Positioning. *IEEE transactions on Signal Processing*, 55(6):2703–2719.

[Caron et al., 2006] Caron, F., Duflos, E., Pomorski, D., and Vanheeghe, P. (2006). GPS/IMU Data Fusion using Multisensor Kalman Filtering: Introduction of Contextual Apects. *Information fusion*, 7(2):221–230.

[Carpenter et al., 1999] Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

[Carter and Kohn, 1996] Carter, C. and Kohn, R. (1996). Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83(3):589–601.

[Carvalho and Lopes, 2007] Carvalho, C. and Lopes, H. (2007). Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51(9):4526–4542.

[Celeux and Govaert, 1992] Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332.

[Centeno and Salido, 2009] Centeno, M. and Salido, R. (2009). Estimation of Asymmetric Stochastic Volatility Models For Stock Exchange Index Returns. *International advances in economic research*, 15(1):71–87.

[Chen and Liu, 2000] Chen, R. and Liu, J. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508.

[Cornebise et al., 2008] Cornebise, J., Moulines, E., and Olsson, J. (2008). Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480.

[Costa et al., 2006] Costa, O., Fragoso, M., and Marques, R. (2006). *Discrete-Time Markov Jump Linear Systems*. Springer Science & Business Media.

[Da-Silva et al., 2011] Da-Silva, C., Migon, H., and Correia, L. (2011). Dynamic Bayesian beta models . *Computational Statistics & Data Analysis*, 55(6):2074 – 2089.

[Del Moral and Jacod, 2001] Del Moral, P. and Jacod, L. (2001). Interacting particle filtering with discrete observations. In *Sequential Monte Carlo methods in practice*, pages 43–75. Springer.

[Delmas, 1995] Delmas, J. (1995). Relations entre les algorithmes d'estimation iteratives EM et ICE avec exemples d'application. In *Proceedings of the 15 Colloque sur le traitement du signal et des images (GRETSI)*, Juan les Pins, France.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[Derrode and Pieczynski, 2004] Derrode, S. and Pieczynski, W. (2004). Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9):2477–2489.

[Derrode and Pieczynski, 2013] Derrode, S. and Pieczynski, W. (2013). Exact Fast Computation of Optimal Filter in Gaussian Switching Linear Systems. *IEEE Signal Processing Letters*, 20(7):701–704.

[Dickey and Fuller, 1979] Dickey, D. and Fuller, W. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431.

[Douc and Cappe, 2005] Douc, R. and Cappe, O. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 64–69, Zagreb, Croatia.

[Douc and Matias, 2001] Douc, R. and Matias, C. (2001). Asymptotics of the maximum likelihood estimator for general hidden Markov models. *Bernoulli*, 7(3):381–420.

[Douc et al., 2011] Douc, R., Moulines, E., Olsson, L., and Van Handel, R. (2011). Consistency of the maximum likelihood estimator for general hidden Markov models. *the Annals of Statistics*, 39(1):474–513.

[Douc et al., 2004] Douc, R., Moulines, E., and Rydén, T. (2004). Asymptotic properties of the maximum likelihood estimator in autoregressive models with Markov regime. *The Annals of statistics*, 32(5):2254–2304.

[Doucet et al., 2000] Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

[Doucet et al., 2001] Doucet, A., Gordon, N., and Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624.

[Doucet and Johansen, 2009] Doucet, A. and Johansen, A. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

[Doucet and Johansen, 2011] Doucet, A. and Johansen, A. (2011). *A tutorial on particle filtering and smoothing: Fifteen years later*. Eds. London, U.K., Oxford Univ. Press.

[Driessen and Boers, 2004] Driessen, H. and Boers, Y. (2004). An efficient particle filter for jump Markov nonlinear systems. In *Proceedings of the Target Tracking 2004: Algorithms and Applications*, pages 19–22, Brighton, UK. IET.

[Durham, 2006] Durham, G. (2006). Monte Carlo methods for estimating, smoothing, and filtering one-and two-factor stochastic volatility models. *Journal of Econometrics*, 133(1):273–305.

[Durham, 2007] Durham, G. (2007). SV mixture models with application to S&P 500 index returns. *Journal of Financial Economics*, 85(3):822–856.

[Ephraim and Mark, 2015] Ephraim, Y. and Mark, B. (2015). Causal Recursive Parameter Estimation for Discrete-Time Hidden Bivariate Markov Chains. *IEEE Trans. Signal Processing*, 63(8):2108–2117.

[Farmer and Toda, 2017] Farmer, L. and Toda, A. (2017). Discretizing nonlinear, non-Gaussian Markov processes with exact conditional moments. *Quantitative Economics*, 8(2):651–683.

[Fong et al., 2002] Fong, W., Godsill, S., Doucet, A., and West, M. (2002). Monte Carlo smoothing with application to audio signal enhancement. *IEEE transactions on signal processing*, 50(2):438–449.

[Fu et al., 2010] Fu, X., Jia, Y., Du, J., and Yu, F. (2010). New interacting multiple model algorithms for the tracking of the manoeuvring target. *IET Control Theory & Applications*, 4(10):2184–2194.

[Gao et al., 2012] Gao, L., Xing, J., Ma, Z., Sha, J., and Meng, X. (2012). Improved IMM algorithm for nonlinear maneuvering target tracking. *Procedia Engineering*, 29:4117–4123.

[Garcke and Griebel, 2013] Garcke, J. and Griebel, M. (2013). *Sparse Grids and Applications*. Springer-Verlag Berlin Heidelberg.

[Gerber and Chopin, 2015] Gerber, M. and Chopin, N. (2015). Sequential quasi Monte Carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579.

[Gerstner and Griebel, 1998] Gerstner, T. and Griebel, M. (1998). Numerical integration using sparse grids. *Numerical algorithms*, 18(3):209–232.

[Geweke, 1989] Geweke, J. (1989). Bayesian Inference in Econometric Models using Monte Carlo Integration. *Econometrica: Journal of the Econometric Society*, 57(6):1317–1339.

[Ghahramani and Hinton, 2000] Ghahramani, Z. and Hinton, G. (2000). Variational Learning for Switching State-Space Models. *Neural computation*, 12(4):831–864.

[Gobet and Maire, 2005] Gobet, E. and Maire, S. (2005). Sequential Control Variates for Functionals of Markov Processes. *SIAM Journal on Numerical Analysis*, 43(3):1256–1275.

[Gordon, 1997] Gordon, N. (1997). A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):353–358.

[Gorynin et al., 2016a] Gorynin, I., Crelier, L., Gangloff, H., Monfini, E., and Pieczynski, W. (2016a). Performance comparison across hidden, pairwise and triplet Markov models' estimators. In *Proceedings of the 5th International Conference on Applied and Computational Mathematics (ICACM)*, Mallorca, Spain.

[Gorynin et al., 2015] Gorynin, I., Derrode, S., Monfrini, E., and Pieczynski, W. (2015). Exact fast smoothing in switching models with application to stochastic volatility. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pages 924–928, Nice, France.

[Gorynin et al., 2017a] Gorynin, I., Derrode, S., Monfrini, E., and Pieczynski, W. (2017a). Fast Filtering in Switching Approximations of Nonlinear Markov Systems With Applications to Stochastic Volatility. *IEEE Transactions on Automatic Control*, 62(2):853–862.

[Gorynin et al., 2017b] Gorynin, I., Derrode, S., Monfrini, E., and Pieczynski, W. (2017b). Fast smoothing in switching approximations of non-linear and non-Gaussian models. *Computational Statistics & Data Analysis*, 114:38 – 46.

[Gorynin et al., 2017c] Gorynin, I., H.Gangloff, Monfrini, E., and Pieczynski, W. (2017c). Assessing the segmentation performance of pairwise and triplet Markov models. *submitted to Signal Processing*.

[Gorynin et al., 2016b] Gorynin, I., Monfrini, E., and Pieczynski, W. (2016b). Fast filtering with new sparse transition Markov chains. In *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP)*, Palma de Mallorca, Spain.

[Gorynin et al., 2016c] Gorynin, I., Monfrini, E., and Pieczynski, W. (2016c). Unsupervised learning of Markov-switching stochastic volatility with an application to market data. In *Proceedings of the 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Vietri sul Marne, Salerno, Italy.

[Gorynin et al., 2017d] Gorynin, I., Monfrini, E., and Pieczynski, W. (2017d). Pairwise Markov Models for Stock Index Forecasting. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Kos Island, Greece.

[Gorynin and Pieczynski, 2017a] Gorynin, I. and Pieczynski, W. (2017a). Bayesian-Assimilation-Based Smoothing in Switching Systems with Application to Financial Trend Analysis. *submitted to Journal of Machine Learning Research.*

[Gorynin and Pieczynski, 2017b] Gorynin, I. and Pieczynski, W. (2017b). Switching conditional Gauss-Hermite filter with application to jump volatility model. *submitted to Signal Processing.*

[Gospodinov and Lkhagvasuren, 2014] Gospodinov, N. and Lkhagvasuren, D. (2014). A Moment-Matching Method For Approximating Vector Autoregressive Processes By Finite-State Markov Chains. *Journal of Applied Econometrics*, 29(5):843–859.

[Gourieroux and Jasiak, 2006] Gourieroux, C. and Jasiak, J. (2006). Autoregressive gamma processes. *Journal of Forecasting*, 25(2):129–152.

[Gouriéroux et al., 2009] Gouriéroux, C., Jasiak, J., and Sufana, R. (2009). The Wishart autoregressive process of multivariate stochastic volatility. *Journal of Econometrics*, 150(2):167–181.

[Gunning, 1965] Gunning, R. (1965). *Analytic Functions of Several Complex Variables.* AMS Chelsea Publishing.

[Harvey and Luati, 2014] Harvey, A. and Luati, A. (2014). Filtering with heavy tails. *Journal of the American Statistical Association*, 109(507):1112–1122.

[Hassan and Nath, 2005] Hassan, M. and Nath, B. (2005). Stock market forecasting using hidden Markov model: a new approach. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 192–196, Wroclaw, Poland. IEEE.

[Jacquier et al., 1994] Jacquier, E., Polson, N., and Rossi, P. (1994). Bayesian Analysis of Stochastic Volatility Models. *Journal of Business & Economic Statistics*, 12(4):371–389.

[Jacquier et al., 2002] Jacquier, E., Polson, N., and Rossi, P. (2002). Bayesian Analysis of Stochastic Volatility Models. *Journal of Business & Economic Statistics*, 20(1):69–87.

[Jia et al., 2012] Jia, B., Xin, K., and Cheng, Y. (2012). Sparse-grid quadrature nonlinear filtering. *Automatica*, 48(2):327–341.

[Jilkov and Li, 2004] Jilkov, V. and Li, R. (2004). Online Bayesian estimation of transition probabilities for Markovian jump systems. *IEEE Transactions on Signal Processing*, 52(6):1620–1630.

[Kim, 1994] Kim, C. J. (1994). Dynamic linear models with Markov-switching. *Journal of Econometrics*, 60(1):1 – 22.

[Kim and Nelson, 1999] Kim, C. J. and Nelson, C. (1999). *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*, volume 1. The MIT Press, 1 edition.

[Koski, 2001] Koski, T. (2001). *Hidden Markov Models for Bioinformatics*, volume 2. Springer, Netherlands.

[Kotecha and Djuric, 2003] Kotecha, J. and Djuric, P. (2003). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601.

[Lanchantin et al., 2011] Lanchantin, P., Lapuyade-Lahorgue, J., and Pieczynski, W. (2011). Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise. *Signal Processing*, 91(2):163 – 175.

[Lapuyade-Lahorgue and Pieczynski, 2012] Lapuyade-Lahorgue, J. and Pieczynski, W. (2012). Unsupervised segmentation of hidden semi-Markov non-stationary chains. *Signal Processing*, 92(1):29 – 42.

[Lerner, 2002] Lerner, U. (2002). *Hybrid Bayesian Networks for Reasoning About Complex Systems.* PhD thesis, stanford university.

[Lethanh and Adey, 2013] Lethanh, N. and Adey, B. (2013). Use of exponential hidden Markov models for modelling pavement deterioration. *International Journal of Pavement Engineering*, 14(7):645–654.

[Li and Jilkov, 2005] Li, R. and Jilkov, V. (2005). Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321.

[Li et al., 2015] Li, T., Bolic, M., and Djuric, P. M. (2015). Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86.

[Liao and Chen, 2006] Liao, J. and Chen, B. (2006). Robust Mobile Location Rstimator With NLOS Mitigation Using Interacting Multiple Model Algorithm. *IEEE Transactions on Wireless Communications*, 5(11):3002–3006.

[Lindsten et al., 2017] Lindsten, F., Johansen, A., Naesseth, C., Kirkpatrick, B., Schon, T., Aston, J., and Bouchard-Cote, A. (2017). Divide-and-Conquer with Sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, 26(2):445–458.

[Lo et al., 2016] Lo, C. C., Skindilias, K., and Karathanasopoulos, A. (2016). Forecasting Latent Volatility Through a Markov Chain Approximation Filter. *Journal of Forecasting*, 35(1):54–69.

[Logothetis and Krishnamurthy, 1999] Logothetis, A. and Krishnamurthy, V. (1999). Expectation maximization algorithms for MAP estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 47(8):2139–2156.

[Lopes and Tsay, 2011] Lopes, H. and Tsay, R. (2011). Particle filters and Bayesian inference in financial econometrics. *Journal of Forecasting*, 30(1):168–209.

[Lu and Darmofal, 2004] Lu, J. and Darmofal, D. (2004). Higher-Dimensional Integration with Gaussian Weight for Applications in Probabilistic Design. *SIAM Journal on Scientific Computing*, 26(2):613–624.

[Luceno, 1999] Luceno, A. (1999). Discrete approximations to continuous univariate distributions – an alternative to simulation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):345–352.

[Lugosi and Zeger, 1995] Lugosi, G. and Zeger, K. (1995). Nonparametric estimation via empirical risk minimization. *IEEE Transactions on information theory*, 41(3):677–687.

[Mamon and Elliott, 2007] Mamon, R. and Elliott, R. (2007). *Hidden Markov Models in Finance*, volume 4. Springer.

[Mamon and Elliott, 2014] Mamon, R. and Elliott, R. (2014). *Hidden Markov Models in Finance: Further Developments and Applications*. Springer.

[Mesot and Barber, 2007] Mesot, B. and Barber, D. (2007). Switching Linear Dynamical Systems for Noise Robust Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(6):1850–1858.

[Miller III and Rice, 1983] Miller III, A. and Rice, T. (1983). Discrete Approximations of Probability Distributions. *Management science*, 29(3):352–362.

[Monahan and Genz, 1997] Monahan, J. and Genz, A. (1997). Spherical-Radial Integration Rules for Bayesian Computation. *Journal of the American Statistical Association*, 92(438):664–674.

[Montgomery et al., 1990] Montgomery, D., Johnson, L., and Gardiner, J. (1990). *Forecasting and Time Series Analysis*. McGraw-Hill Companies.

[Morokoff and Caflisch, 1995] Morokoff, W. and Caflisch, R. (1995). Quasi-Monte Carlo Integration. *Journal of computational physics*, 122(2):218–230.

[Murphy and Russell, 2001] Murphy, K. and Russell, S. (2001). Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *Sequential Monte Carlo methods in practice*, pages 499–515. Springer.

[Niederreiter, 2010] Niederreiter, H. (2010). *Quasi-Monte Carlo methods*. Wiley Online Library.

[Nikolaev et al., 2014] Nikolaev, N., Menezes, L., and Smirnov, E. (2014). Nonlinear filtering of asymmetric stochastic volatility models and Value-at-Risk estimation. In *Proceedings of the IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, pages 310–317, London, UK.

[Novak and Ritter, 1997] Novak, E. and Ritter, K. (1997). The Curse of Dimension and a Universal Method for Numerical Integration. In *Multivariate approximation and splines*, pages 177–187. Springer.

[Olteanu and Rynkiewicz, 2007] Olteanu, M. and Rynkiewicz, J. (2007). Estimating the number of regimes in a switching autoregressive model.

[Olteanu et al., 2004] Olteanu, M., Rynkiewicz, J., and Maillet, B. (2004). Non-linear Analysis of Shocks when Financial Markets are Subject to Changes in Regime. In *European Symposium on Artificial Neural Networks*, pages 87–92, Bruges, Belgium.

[Omori and Watanabe, 2008] Omori, Y. and Watanabe, T. (2008). Block sampler and posterior mode estimation for asymmetric stochastic volatility models. *Computational Statistics and Data Analysis*, 52(6):2892–2910.

[Panopoulou and Pantelidis, 2015] Panopoulou, E. and Pantelidis, T. (2015). Regime-switching models for exchange rates. *The European Journal of Finance*, 21(12):1023–1069.

[Paul, 1991] Paul, D. (1991). The Lincoln tied-mixture HMM continuous speech recognizer. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 329–332, Toronto, Canada. IEEE.

[Pavlovic et al., 2001] Pavlovic, V., R., J., and MacCormick, J. (2001). Learning Switching Linear Models of Human Motion. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 981–987. MIT Press.

[Pedrick, 1994] Pedrick, G. (1994). *A First Course in Analysis.* Springer.

[Petetin and Desbouvries, 2014] Petetin, Y. and Desbouvries, F. (2014). A Class of Fast Exact Bayesian Filters in Dynamical Models With Jumps. *IEEE Transactions on Signal Processing*, 62(14):3643–3653.

[Pieczynski, 2003] Pieczynski, W. (2003). Pairwise Markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):634–639.

[Pieczynski, 2011a] Pieczynski, W. (2011a). Exact filtering in conditionally Markov switching hidden linear models. *Comptes Rendus Mathematique*, 349(9):587–590.

[Pieczynski, 2011b] Pieczynski, W. (2011b). Exact Smoothing in Hidden Conditionally Markov Switching Linear Models. *Communications in Statistics - Theory and Methods*, 40(16):2823–2829.

[Pieczynski et al., 2003] Pieczynski, W., Hulard, C., and Veit, T. (2003). Triplet Markov chains in hidden signal restoration. In *Procedings of the SPIEs International Symposium on Remote Sensing*, volume 4885, pages 58–68, Greece, Crete.

[Poli et al., 2007] Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle Swarm Optimization. *Swarm intelligence*, 1(1):33–57.

[Potin et al., 2006a] Potin, D., Duflos, E., and Vanheeghe, P. (2006a). Landmines Ground-Penetrating Radar Signal Enhancement by Digital Filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2393–2406.

[Potin et al., 2006b] Potin, D., Vanheeghe, P., Duflos, E., and Davy, M. (2006b). An abrupt change detection algorithm for buried landmines localization. *IEEE Transactions on Geoscience and Remote Sensing*, 44(2):260–272.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Rebeschini et al., 2015] Rebeschini, P., Van Handel, R., et al. (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866.

[Ristic et al., 2004] Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman filter: particle filters for tracking applications.* Artech House.

[Rosti and Gales, 2003] Rosti, A. and Gales, M. (2003). *Switching Linear Dynamical Systems for Speech Recognition.* University of Cambridge, Department of Engineering.

[Roth et al., 2013] Roth, M., Ozkan, E., and Gustafsson, F. (2013). A Student's t filter for heavy tailed process and measurement noise. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5770–5774, Vancouver, Canada.

[Särkkä et al., 2012] Särkkä, S., Bunch, P., and Godsill, S. (2012). A Backward-Simulation Based Rao-Blackwellized Particle Smoother for Conditionally Linear Gaussian Models. *IFAC Proceedings Volumes*, 45(16):506–511.

[Simandl and Kralovec, 2000] Simandl, M. and Kralovec, J. (2000). Filtering, Prediction and Smoothing with Gaussian Sum Representation. In *Proceedings of the 12th IFAC Symposium on System Identification (SYSID)*, pages 1157 – 1162, Santa Barbara, CA, USA.

[Singer, 2015] Singer, H. (2015). Conditional Gauss-Hermite Filtering with Application to Volatility Estimation. *IEEE Transactions on Automatic Control*, 60(9):2476–2481.

[Snyder et al., 2008] Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to High-Dimensional Particle Filtering. *Monthly Weather Review*, 136(12):4629–4640.

[So et al., 1998] So, M., Lam, K., and Li, W. (1998). A stochastic volatility model with Markov switching. *Journal of Business & Economic Statistics*, 16(2):244–253.

[Straka et al., 2011] Straka, O., Dunik, J., and Simandl, M. (2011). Gaussian sum unscented Kalman filter with adaptive scaling parameters. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, Chicago, IL, USA.

[Sun et al., 2016] Sun, Y., Mark, B., and Ephraim, Y. (2016). Collaborative Spectrum Sensing via Online Estimation of Hidden Bivariate Markov Models. *IEEE Transactions on Wireless Communications*, 15(8):5430–5439.

[Tauchen, 1986] Tauchen, G. (1986). Finite State Markov-Chain Approximations to Univariate and Vector Autoregressions. *Economics letters*, 20(2):177–181.

[Terry and Knotek, 2011] Terry, S. and Knotek, E. (2011). Markov-chain approximations of vector autoregressions: Application of general multivariate-normal integration techniques. *Economics Letters*, 110(1):4–6.

[Togneri et al., 2001] Togneri, R., Ma, J., and D., L. (2001). Parameter estimation of a target-directed dynamic system model with switching states. *Signal Processing*, 81(5):975 – 987.

[Toledo-Moreo et al., 2007] Toledo-Moreo, R., Úbeda, B., Skarmeta, A., and Zamora I., M. A. (2007). High Integrity IMM-EKF Based Road Vehicle Navigation with Low Cost GPS/INS. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):491–511.

[Tsay, 2005] Tsay, R. (2005). *Analysis of Financial Time Series*, volume 543. John Wiley & Sons.

[Verge et al., 2015] Verge, C., Dubarry, C., Del Moral, P., and Moulines, E. (2015). On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, 25(2):243–260.

[Vidyasagar, 2014] Vidyasagar, M. (2014). *Hidden Markov Processes: Theory and Applications to Biology.* Princeton University Press.

[Wasilkowski and Wozniakowski, 1995] Wasilkowski, G. and Wozniakowski, H. (1995). Explicit Cost Bounds of Algorithms for Multivariate Tensor Product Problems. *Journal of Complexity*, 11(1):1–56.

[Wasserman, 2004] Wasserman, L. (2004). *All of statistics: a concise course in statistical inference.* Springer-Verlag New York.

[Weiss et al., 2004] Weiss, K., Kaempchen, N., and Kirchner, A. (2004). Multiple-model tracking for the detection of lane change maneuvers. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 937–942, Parma, Italy.

[West et al., 1985] West, M., Harrison, P., and Migon, H. (1985). Dynamic Generalized Linear Models and Bayesian Forecasting. *Journal of the American Statistical Association*, 80(389):73–83.

[Wu et al., 2004] Wu, W., Black, M., Mumford, D., Gao, Y., Bienenstock, E., and Donoghue, J. (2004). Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Transactions on Biomedical Engineering*, 51(6):933–942.

[Yu, 2016] Yu, S. (2016). *Hidden Semi-Markov Models.* Elsevier, Boston.

[Zhao and Liu, 2012] Zhao, S. and Liu, F. (2012). State estimation in non-linear Markov jump systems with uncertain switching probabilities. *IET control theory & applications*, 6(5):641–650.

[Zheng et al., 2016] Zheng, F., Derrode, S., and Pieczynski, W. (2016). Parameter estimation in conditionally Gaussian pairwise Markov switching models and unsupervised smoothing. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Vietri sul Mare, Salerno, Italie.

[Zhong et al., 2008] Zhong, Z., Meng, H., and Wang, X. (2008). Extended target tracking using an IMM based Rao-Blackwellised unscented Kalman filter. In *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pages 2409–2412. IEEE.

[Zhu and Rahman, 2015] Zhu, X. and Rahman, S. (2015). A regime-switching NelsonâĂŞSiegel term structure model of the macroeconomy. *Journal of Macroeconomics*, 44:1–17.

[Zoeter, 2007] Zoeter, O. (2007). Bayesian Generalized Linear Models in a Terabyte World. In *Proceedings of the Image and Signal Processing and Analysis (ISPA)*, pages 435–440, Istanbul, Turkey.

[Zoeter and Heskes, 2006] Zoeter, O. and Heskes, T. (2006). Deterministic approximate inference techniques for conditionally Gaussian state space models. *Statistics and Computing*, 16(3):279–292.

[Zoeter et al., 2004] Zoeter, O., Ypma, A., and Heskes, T. (2004). Improved unscented Kalman smoothing for stock volatility estimation. In *Proceedings of the Machine Learning for Signal Processing (MLSP)*, pages 143–152, Sao Luis, Brazil.

[Zoeter et al., 2006] Zoeter, O., Ypma, A., and Heskes, T. (2006). Deterministic and Stochastic Gaussian Particle Smoothing. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*, pages 228–231, Cambridge, UK.

List of Figures