Université de Montréal

Estimation of State Space Models and Stochastic Volatility

par
Shirley Miller Lira

Département de sciences économiques
Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en sciences économiques

Septembre, 2012

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée :
Estimation of State Space Models and Stochastic Volatility

présentée par :
Shirley Miller Lira

a été évaluée par un jury composé des personnes suivantes :

Silvia Gonçalves,	président-rapporteur
William McCausland,	directeur de recherche
Éric Jacquier,	codirecteur de recherche
Francisco Ruge-Murcia,	membre du jury
Sylvia Fruehwirth-Schnatter,	examinatrice externe
Jean-Michel Cousineau,	représentant du doyen de la faculté

Thèse acceptée le 7 septembre 2012

Résumé

Ma thèse est composée de trois chapitres reliés à l'estimation des modèles espace-état et volatilité stochastique.

Dans le première article *"Simulation Smoothing for State-Space Models : A Computational Efficiency Analysis"*, nous développons une procédure de lissage de l'état, avec efficacité computationnelle, dans un modèle espace-état linéaire et gaussien. Nous montrons comment exploiter la structure particulière des modèles espace-état pour tirer les états latents efficacement. Nous analysons l'efficacité computationnelle des méthodes basées sur le filtre de Kalman, l'algorithme facteur de Cholesky et notre nouvelle méthode utilisant le compte d'opérations et d'expériences de calcul. Nous montrons que pour de nombreux cas importants, notre méthode est plus efficace. Les gains sont particulièrement grands pour les cas où la dimension des variables observées est grande ou dans les cas où il faut faire des tirages répétés des états pour les mêmes valeurs de paramètres. Comme application, on considère un modèle multivarié de Poisson avec le temps des intensités variables, lequel est utilisé pour analyser le compte de données des transactions sur les marchés financières.

Dans le deuxième chapitre *"Multivariate Stochastic Volatility"*, nous proposons une nouvelle technique pour analyser des modèles multivariés à volatilité stochastique. La méthode proposée est basée sur le tirage efficace de la volatilité de son densité conditionnelle sachant les paramètres et les données. Notre méthodologie s'applique aux modèles avec plusieurs types de dépendance dans la coupe transversale. Nous pouvons modeler des matrices de corrélation conditionnelles variant dans le temps en incorporant des facteurs dans l'équation de rendements, où les

facteurs sont des processus de volatilité stochastique indépendants. Nous pouvons incorporer des copules pour permettre la dépendance conditionnelle des rendements sachant la volatilité, permettant avoir différent lois marginaux de Student avec des degrés de liberté spécifiques pour capturer l'hétérogénéité des rendements. On tire la volatilité comme un bloc dans la dimension du temps et un à la fois dans la dimension de la coupe transversale. Nous appliquons la méthode introduite par McCausland (2012) pour obtenir une bonne approximation de la distribution conditionnelle à posteriori de la volatilité d'un rendement sachant les volatilités d'autres rendements, les paramètres et les corrélations dynamiques. Le modèle est évalué en utilisant des données réelles pour dix taux de change. Nous rapportons des résultats pour des modèles univariés de volatilité stochastique et deux modèles multivariés.

Dans le troisième chapitre *"The information content of Realized Volatility"*, nous évaluons l'information contribuée par des variations de volatilite réalisée à l'évaluation et prévision de la volatilité quand des prix sont mesurés avec et sans erreur. Nous utilisons de modèles de volatilité stochastique. Nous considérons le point de vue d'un investisseur pour qui la volatilité est une variable latent inconnu et la volatilité réalisée est une quantité d'échantillon qui contient des informations sur lui. Nous employons des méthodes bayésiennes de Monte Carlo par chaîne de Markov pour estimer les modèles, qui permettent la formulation, non seulement des densités a posteriori de la volatilité, mais aussi les densités prédictives de la volatilité future. Nous comparons les prévisions de volatilité et les taux de succès des prévisions qui emploient et n'emploient pas l'information contenue dans la volatilité réalisée. Cette approche se distingue de celles existantes dans la littérature empirique en ce sens que ces dernières se limitent le plus souvent à documenter la capacité de la volatilité réalisée à se prévoir à elle-même. Nous présentons des applications empiriques en utilisant les rendements journaliers des indices et de taux de change. Les différents modèles concurrents sont appliqués à la seconde moitié de 2008, une période marquante dans la récente crise financière.

Mots-clés : Modèles espace-état, Méthodes de Monte Carlo par chaîne de Markov, Volatilité stochastique, Volatilité réalisée, Compte de données, Données haute

fréquence.

Abstract

My thesis consists of three chapters related to the estimation of state space models and stochastic volatility models.

In the first chapter "Simulation Smoothing for State-Space Models : A Computational Efficiency Analysis", we develop a computationally efficient procedure for state smoothing in Gaussian linear state space models. We show how to exploit the special structure of state-space models to draw latent states efficiently. We analyze the computational efficiency of Kalman-filter-based methods, the Cholesky Factor Algorithm, and our new method using counts of operations and computational experiments. We show that for many important cases, our method is most efficient. Gains are particularly large for cases where the dimension of observed variables is large or where one makes repeated draws of states for the same parameter values. We apply our method to a multivariate Poisson model with time-varying intensities, which we use to analyze financial market transaction count data.

In the second chapter "Multivariate Stochastic Volatility", we propose a new technique for the analysis of multivariate stochastic volatility models, based on efficient draws of volatility from its conditional posterior distribution. It applies to models with several kinds of cross-sectional dependence. Full VAR coefficient and covariance matrices give cross-sectional volatility dependence. Mean factor structure allows conditional correlations, given states, to vary in time. The conditional return distribution features Student's t marginals, with asset-specific degrees of freedom, and copulas describing cross-sectional dependence. We draw volatility as a block in the time dimension and one-at-a-time in the cross-section. Following McCausland (2012), we use close approximations of the conditional posterior dis-

tributions of volatility blocks as Metropolis-Hastings proposal distributions. We illustrate using daily return data for ten currencies. We report results for univariate stochastic volatility models and two multivariate models.

In the third chapter "The information content of Realized Volatility", we evaluate the information contributed by (variations of) realized volatility to the estimation and forecasting of volatility when prices are measured with and without error using a stochastic volatility model. We consider the viewpoint of an investor for whom volatility is an unknown latent variable and realized volatility is a sample quantity which contains information about it. We use Bayesian Markov Chain Monte Carlo (MCMC) methods to estimate the models, which allow the formulation of the posterior densities of in-sample volatilities, and the predictive densities of future volatilities. We then compare the volatility forecasts and hit rates from predictions that use and do not use the information contained in realized volatility. This approach is in contrast with most of the empirical realized volatility literature which most often documents the ability of realized volatility to forecast itself. Our empirical applications use daily index returns and foreign exchange during the 2008-2009 financial crisis.

Keywords : State-space models, Markov chain Monte Carlo, Importance sampling, Stochastic volatility, Realized Volatility, Count data, High frequency financial data.

Table des matières

Résumé **iii**

Abstract **vi**

Table des matières **x**

Liste des figures **xii**

Liste des tableaux **xiv**

Sigles et abréviations **xv**

Dedication **xvi**

Acknowledgments **xvii**

General Introduction **1**

Chapter 1 : Simulation Smoothing for State-Space Models : A Computational Efficiency Analysis **7**

1.1 Introduction 7
1.2 Precision-Based Methods for Simulation Smoothing 11
1.3 Efficiency Analysis I : Counts of Operations 14
1.3.1 Fixed Costs 15
1.3.2 Marginal Costs 17
1.4 Efficiency Analysis II : Computational Experiments 19
1.5 An Empirical Application to Count Models 23
1.5.1 Modifications to the Algorithm for Approximating $L(\theta)$ . . . 24
1.5.2 A Multivariate Poisson Model with Time-Varying Intensities 26

1.6 Conclusions . . . 30
1.7 Appendix to Chapter 1 . . . 35
1.7.1 Derivation of Ω and c . . . 35
1.7.2 Proof of Result 1.2.1 . . . 37

Chapter 2 : Multivariate Stochastic Volatility **40**
2.1 Introduction . . . 40
2.2 The Model . . . 45
2.2.1 Alternative MSV models . . . 49
2.2.2 Prior Distributions . . . 51
2.3 Posterior inference using MCMC . . . 53
2.3.1 Draw of θ_i, α_i . . . 54
2.3.2 Draw of (B, f) . . . 57
2.3.3 Draw of f . . . 58
2.3.4 Draw of R . . . 58
2.4 Getting it Right . . . 59
2.5 Empirical Results . . . 61
2.5.1 Data . . . 61
2.5.2 Estimation Results . . . 65
2.6 Conclusions . . . 72
2.7 Appendix to Chapter 2 . . . 74
2.7.1 Computing $\bar{\Omega}^{(i)}$ and $\bar{c}^{(i)}$ . . . 74
2.7.2 Computing $\log \pi(y_t|\alpha_t, \nu, B, R)$ and its derivatives with respect to α_{it} . . . 76
2.7.3 Sampling $r|\alpha, \theta, f, B, R$ . . . 83
2.7.4 Tables of results . . . 83

Chapter 3 : The information content of Realized Volatility **94**
3.1 Introduction . . . 94
3.2 MCMC algorithms . . . 98
3.2.1 Model 1 . . . 99
3.2.2 Model 2 . . . 102
3.2.3 Priors . . . 104
3.3 Performance evaluation . . . 105
3.3.1 Performance with Model 1 . . . 108
3.3.2 Performance with Model 2 . . . 116
3.3.3 Performance with bivariate models . . . 120
3.4 Empirical Application . . . 124

3.4.1 Data 124
3.4.2 Odds ratio 126
3.4.3 Parameter Inference 130
3.4.4 Volatility estimation 131
3.4.5 S&P 500 : Realized or Implied Volatility ? 142
3.5 Conclusions 155
3.6 Appendix to Chapter 3 157
3.6.1 Model 2 - Bivariate 157
3.6.2 Comparison of JPR and KSC estimations 160
3.6.3 Particle Learning for SV-RV 164
3.6.4 Real data : posterior parameter distribution 168

General Conclusion **173**

Bibliography **174**

Table des figures

1.1 Transactions data 29
2.1 Time plots of daily returns series (in percentage) 64
2.2 Comparison of posterior parameter distributions (Part 1) 66
2.3 Comparison of posterior parameter distributions (Part 2) 67
2.4 Time series plot of the posterior mean of the factor volatility of MSV-q1 model 69
2.5 Time varying decomposition of variance into factor and idiosyncratic components 70
2.6 Comparison of annualized posterior mean of volatility 71
3.1 True vs estimated annualized volatility 116
3.2 Sequential odds ratio : British Pound - PL algorithm on M1 model 128
3.3 Logarithm of sequential odds ratio calculated with MCMC algorithm and PL filter on currencies 129
3.4 Comparison of in-sample volatility fit across models : currencies . . 132
3.5 Comparison of in-sample volatility fit across models : stocks 133
3.6 Comparison of one day ahead volatility forecast across models : currencies 135
3.7 Comparison of one day ahead volatility forecast across models : stocks 136
3.8 Comparison of one week ahead volatility forecast across models : currencies 137
3.9 Comparison of one week ahead volatility forecast across models : stocks 138
3.10 Comparison of one day ahead volatility forecast across models : stocks - new prior 139
3.11 Comparison of one week ahead volatility forecast across models : stocks - new prior 140
3.12 Plot of S&P 500, Realized Volatility and Implied Volatility Index . 143
3.13 Logarithm of sequential odds ratio : $S\&P500$ 145

3.14 S&P 500 : δ. 146
3.15 S&P 500 : β_1 and σ_η 147
3.16 S&P 500 : δ for bivariate Model 2 149
3.17 S&P 500 : annualized volatility estimation with M0, M2-RV and M2-VIX models 151
3.18 S&P 500 : posterior annualized volatility estimation comparison . . 152
3.19 S&P 500 : In sample volatility estimation 153
3.20 S&P 500 : Out of sample volatility estimation 154
3.21 Single vs multi move - $p(h_t|R_1, \ldots, R_T)$ - UK £. 163
3.22 Smoothed volatility, $\sqrt{h_t}|y^T$, filtered volatility $\sqrt{h_t}|y^t$ and learning $\delta|y^t$ 167

Liste des tableaux

1.1 Scalar multiplications needed for pre-computation. 17
1.2 Costs in ms, Time Varying Parameter model 22
1.3 Costs in ms, Dynamic Factor model 22
1.4 Computational costs per observation per additional draw of α_t . . . 28
1.5 Time cost of drawing $\alpha^{(i)}$ as a function of the number of draws N_s. Figures are in 100ths of seconds. 30
1.6 Estimation results for the model for different values of m. The standard errors are between parantheses. 31
2.1 Table of symbols 47
2.2 Parameter means and variances of prior distributions. 52
2.3 "Getting it right" sample quantiles 62
2.4 Descriptive statistics of data 63
2.5 Sample daily correlation 63
2.6 Posterior statistics of parameters of univariate SV models with student-t errors (Part 1) 84
2.7 Posterior statistics of parameters of univariate SV models with student-t errors (Part 2) 85
2.8 Posterior statistics of parameters of log volatility equation in the MSV-q0 model (Part 1) 86
2.9 Posterior statistics of parameters of log volatility equation in the MSV-q0 model (Part 2) 87
2.10 Posterior statistics of parameters of log volatility equation in the MSV-q1 model (Part 1) 88
2.11 Posterior statistics of parameters of log volatility equation in the MSV-q1 (Part 2) 89
2.12 Posterior statistics of parameters of the factor volatility for MSV-q1 model 90
2.13 Posterior statistics of loading factor matrix B for MSV-q1 model . . 90

2.14 Posterior mean of correlation matrix R_{11} for MSV-q0 model 91
2.15 Posterior mean of correlation matrix R_{11} for MSV-q1 model 92
2.16 Average of posterior mean of conditional correlation matrix $Corr(r_t|\alpha_t)$ for MSV-q1 model 93
3.1 Properties of simulated RV measures 107
3.2 Posterior parameter distribution - Model 1 on noiseless returns . . . 109
3.3 In sample properties of $\sqrt{h_t}$ - Model 1 on noiseless returns 110
3.4 Out of sample evaluation of $\sqrt{h_t}$ - Model 1 on noiseless returns . . . 111
3.5 Posterior parameter distribution - Model 1 on noisy returns 113
3.6 In sample properties of $\sqrt{h_t}$ - Model 1 on noisy returns 114
3.7 Out of sample evaluation of $\sqrt{h_t}$ - Model 1 on noisy returns 115
3.8 Posterior parameter distribution - Model 2 on noisy returns 118
3.9 In sample properties of $\sqrt{h_t}$ - Model 2 on noisy returns 119
3.10 Out of sample evaluation of $\sqrt{h_t}$ - Model 2 on noisy returns 120
3.11 Average OLS estimation for proxies 121
3.12 Posterior parameter distribution - M1 bivariate 121
3.13 Posterior parameter distribution - M2 bivariate 123
3.14 In sample fit $\sqrt{h_t}$: Model 1 and 2 bivariate 123
3.15 Out of sample fit $\sqrt{h_t}$: Model 1 and 2 bivariate 124
3.16 Description of the data 125
3.17 Logarithm of posterior odds ratio 127
3.18 Average logarithm of posterior odds Ratio by period 130
3.19 Currencies and stocks : hit rate 142
3.20 $S\&P500$: hit rate 155
3.21 Single-Move versus Multi-Move MCMC algorithm : Parameter estimation 161
3.22 Single-Move versus Multi-Move MCMC algorithm : In sample fit of $\sqrt{h_t}$ 162
3.23 Single vs multi-move - posterior analysis - UK £ 162
3.24 Currencies and Indices : posterior parameter distribution for standard Model. 169
3.25 Currencies and Indices : posterior parameter distribution for Model 1 170
3.26 Currencies and Indices : posterior parameter distribution for Model 2 171
3.27 $S\&P500$: Posterior parameter distribution - M2 172

Sigles et abréviations

ARMA	: Autoregressive Moving Average
CFA	: Cholesky Factor Algorithm
GARCH	: Generalized Autoregressive Conditional Heteroscedasticity
GMRFs	: Gaussian Markov Random Fields
IV	: Integrated Volatility
MCMC	: Monte Carlo Markov chain
MAE	: Mean Absolute Error
MSV	: Multivariate Stochastic Volatility
RMSE	: Root Mean Squared Error
RK	: Realized Kernel
RV	: Realized Volatility
SV	: Stochastic Volatility
TSRV	: Two Scaled Realized Volatility
VAR	: Vector Autorregressive
VIX	: Chicago Board Options Exchange Market Volatility Index

To my wonderful parents, Brenda and Walter Miller
To my beloved husband, Alfredo Seminario

Acknowledgments

I am very happy to have the opportunity to express my appreciation to all the people who have supported me through the years dedicated to my Ph.D. studies.

First and foremost, I would like to express my sincere gratitude to an excellent advisor, William McCausland, for his guidance, constant support and valuable feedback. I would also like to thank my co-advisor, Éric Jacquier. I feel fortunate for having the opportunity to work with him. I am very thankful to both for the learning experience.

I also thank Greg Bauer from Bank of Canada with whom I worked during my summer internship at the Financial Market Department. I greatly benefited from our discussions during my time there.

I appreciate the financial support I received throughout my studies from the economics department at *Université de Montréal*, *Institut de Finance Mathématique de Montréal*, *Faculté des Études Supérieures de Université de Montréal* and *Fonds Québécois de la Recherche sur la Société et la Culture*.

I expand my thanks to all the professors of the Economics Department at *Université de Montréal* for the devotion and inspiration conveyed through their work. I owe a great deal to the administrative staff of the Economics Department at *Université de Montréal* ; in particular Lyne Racine, Johanne Menard Simard and Josée Lafontaine. Thank you also to my student colleagues for their friendship and shared experiences.

The road to my Ph.D. has been an enriching experience with inevitable ups and downs which would not have been easy to overcome without the unconditional help and support from some very special people. Very special thanks to my wonderful

parents, Walter and Brenda, and my siblings, Vanessa and Walter, for their constant encouragement, love and for always being there for me. I am very blessed to have you in my life.

I do not have the words to express my deep gratitude towards my friends Sandra Vasquez and Andrew Freestone for dedicating time to read this thesis and to my aunt Malu Lira and her family for welcoming me into her home and unconditional availability. Thanks to my many friends in Toronto and Lima for your continuing friendship, chats and support along the way.

Last, but not least, I would like to thank my best friend, lifetime partner and beloved husband, Alfredo. Thank you for your understanding throughout the many challenging times and for the sacrifices that we have had to make, for your support and, above all, thank you for believing in me, *Te amo BB !*.

General Introduction

State space models are time series models that relate observable and latent variables. There are different specifications for a state space model depending on the specific problem of study. The basic specification considers linear relationships and Gaussian process. Extensions include non-linear systems and/or non-Gaussian process. Besides, states can be univariate or multivariate.

State space models are frequently used in finance and macroeconomics. Examples of popular applications of linear state space models include the estimation of time-varying parameters, vector autoregressive models and dynamic factor models. Non linear state space models have been very useful for the estimation of volatility, in which volatility is modeled as a latent variable that follows a stochastic process. Models of this kind are called stochastic volatility (SV) models.

Modeling and forecasting volatility is one of the most active areas of research in finance. Accurate measures and good forecasts of future volatility are critical for the implementation and evaluation of assets and derivative pricing theories as well as trading and hedging strategies.

There is an extensive literature concerned with the development of parametric models to estimate volatility. The main two parametric approaches are GARCH-type models and stochastic volatility type models. SV models differs from the GARCH type models in that the conditional volatility is treated as a latent variable and not a deterministic function of lagged returns. This feature make SV models more flexible than GARCH-type models. Jacquier, Polson, and Rossi (1994), Kim, Shephard, and Chib (1998) show that a lot can be gained from the added flexibility of the SV models over the GARCH models, especially in times of stress (Geweke

(1994)).

My first and second chapters propose efficient methods for the estimation of state space models. The first chapter is concerned with linear Gaussian models while the second chapter studies non-linear and non-Gaussian multivariate state space models through an application to multivariate stochastic volatility models. The third chapter evaluates the use of non parametric volatility estimators to improve the inference and forecasting of the latent volatility using a SV model.

The first chapter is published in *Computational Statistics and Data Analysis*. In this chapter we develop a computationally efficient procedure for state smoothing in Gaussian linear state space models. Simulation smoothing involves drawing state variables (or innovations) in discrete time state-space models from their conditional distribution given parameters and observations. Gaussian simulation smoothing is of particular interest, not only for the direct analysis of Gaussian linear models, but also for the indirect analysis of more general models.

Several methods for Gaussian simulation smoothing exist, most of which are based on the Kalman filter. Since states in Gaussian linear state-space models are Gaussian Markov random fields, it is also possible to apply the Cholesky Factor Algorithm (CFA) to draw states. This algorithm takes advantage of the band diagonal structure of the Hessian matrix of the log density to make efficient draws. We show how to exploit the special structure of state-space models to draw latent states even more efficiently. We analyze the computational efficiency of Kalman-filter-based methods, the CFA, and our new method using counts of operations and computational experiments. We show that for many important cases, our method is most efficient. Gains are particularly large for cases where the dimension of observed variables is large or where one makes repeated draws of states for the same parameter values. We apply our method to a multivariate Poisson model with time-varying intensities, which we use to analyze financial market transaction count data.

In the second chapter, we propose new Markov Chain Monte Carlo methods for Bayesian analysis of multivariate stochastic volatility (MSV) models. This approach

uses a numerically efficient method to draw volatilities as a block in the time dimension and one-at-a-time in the cross sectional dimension.

By featuring different kinds of dynamic cross-sectional dependence among multiple asset returns, multivariate volatility models can capture many different stylized facts. We show that our estimation approach is quite flexible, allowing different specifications and types of dependence. We can specify full first order VAR coefficient and covariance matrices for the evolution of volatilities. We can include a mean factor structure, which allows conditional return correlations, given asset and factor volatilities, to vary over time, and for these correlations to covary with variances. We can also model cross-sectional conditional return dependence given latent asset and factor volatilities using copulas. Copulas allow one to represent a multivariate distribution in a very flexible way by decoupling the choice of marginal distributions — which can be different from each other — from the choice of the dependence structure. We introduce a new prior for correlation matrices, which we use in the context of Gaussian copulas. It is based on a geometric interpretation of correlation coefficients. The prior is a first step towards a model for time varying correlations where assets are exchangeable, avoiding a problem with models based on the Cholesky decomposition – their predictions are not invariant to the arbitrary choice of how to order assets.

We allow heavy-tailed returns. We also depart from the usual assumption of Gaussian factors and allow Student's t factors. We applied our methodology to estimate the volatility of ten currencies. We estimate three models : a model with independent currencies, each governed by a univariate SV model with Student's t innovations, a MSV model with no factors and a MSV model with one factor. We use comparable priors in the three models and compare the posterior distribution of parameters, volatilities and correlations across models.

Finally, in the third chapter, we evaluate the use of non parametric estimations of volatility to improve the quality of inference and forecasting of volatility, which is considered a latent variable.

Recently the availability of high frequency data has made attractive the use

of totally non parametric measurements such as the Realized Variance (RV). RV is the sum of squared high frequency returns over a fixed interval. Under general conditions RV is a consistent estimator of the integrated variance of the price process. However, in a realistic setup, RV suffers from measurement errors due to the presence of jumps and microstructure noise in observed prices. To address this problem, different methods have been proposed to extract noise and determine the optimal frequency to use. Estimators are usually evaluated and compared on the basis of their forecast performance. A popular approach is to use the R-squared of a Mincer-Zarnowitz style regression, where future integrated variances are regressed on a constant and the RV estimator. As the integrated volatility is not observed it is usually replaced by some RV measure. However as RV is an estimated quantity, there can be error-in-variable problems in the estimation.

We show in a practical way the amount of information that various implementations of realized volatility can bring to the forecasting of volatility. We consider the viewpoint of an investor for whom volatility is an unknown latent variable and realized volatility is a sample quantity which contains information about it. The investor estimates the standard AR(1) stochastic volatility model by Bayesian Markov Chain Monte Carlo (MCMC) methods, which allow the formulation of the posterior densities of in-sample volatilities, and the predictive densities of future volatilities. We propose and implement two algorithms that the investor can use to naturally extend her model to incorporate intra-day information in the form of realized volatility estimates. This allows us to compare directly the volatility forecasts from predictive and posterior densities that use and do not use the information contained in realized volatility. A by-product of our algorithms is the odds ratio in favor of realized volatility. This approach is in contrast with most of the empirical realized volatility literature which generally documents the ability of realized volatility to forecast itself.

We present the results of sampling experiments conducted to asses the performance of the Bayesian inference on volatilities and parameters and the results of an empirical application using daily index returns and foreign exchange returns.

We applied the different competing models to the second half of 2008, period of stress when getting volatility right or wrong becomes very important. We extend the framework to incorporate implied volatility, using the VIX index.

Co-author's contribution

I am the first author of all of the three articles in this thesis. However, my co-authors Éric Jacquier, William McCausland, Denis Pelletier and myself have made an equal contribution to these articles.

The first two articles have been written with William McCausland and Denis Pelletier. The third article has been written with Éric Jacquier.

Chapter 1 : Simulation Smoothing for State-Space Models : A Computational Efficiency Analysis [1]

1.1 Introduction

State-space models are time series models featuring both latent and observed variables. The latent variables have different interpretations according to the application. They may be the unobserved states of a system in biology, economics or engineering. They may be time-varying parameters of a model. They may be factors in dynamic factor models, capturing covariances among a large set of observed variables in a parsimonious way.

Gaussian linear state-space models are interesting in their own right, but they are also useful devices for the analysis of more general state-space models. In some cases, the model becomes a Gaussian linear state-space model, or a close approximation, once we condition on certain variables. These variables may be a natural

1. This article was published in Computational Statistics & Data Analysis, Vol. 55, pages No. 199-212, 2011.

part of the model, as in Carter and Kohn (1996), or they may be convenient but artificial devices, as in Kim, Shephard, and Chib (1998), Stroud, Müller, and Polson (2003) and Frühwirth-Schnatter and Wagner (2006).

In other cases, one can approximate the conditional distribution of states in a non-Gaussian or non-linear model by its counterpart in a Gaussian linear model. If the approximation is close enough, one can use the latter for importance sampling, as Durbin and Koopman (1997) do to compute likelihood functions, or as a proposal distribution in a Metropolis-Hastings update, as (Shephard and Pitt 1997) do for posterior Markov chain Monte Carlo simulation.

To fix notation, consider the following Gaussian linear state-space model, expressed using notation from de Jong and Shephard (1995) :

$$y_t = X_t\beta + Z_t\alpha_t + G_t u_t, \quad t = 1, \ldots, n, \tag{1.1}$$

$$\alpha_{t+1} = W_t\beta + T_t\alpha_t + H_t u_t, \quad t = 1, \ldots, n-1, \tag{1.2}$$

$$\alpha_1 \sim \mathrm{N}(a_1, P_1), \quad u_t \sim \text{i.i.d. } \mathrm{N}(0, I_q), \tag{1.3}$$

where y_t is a $p \times 1$ vector of dependent variables, α_t is a $m \times 1$ vector of state variables, and β is a $k \times 1$ vector of coefficients. The matrices X_t, Z_t, G_t, W_t, T_t and H_t are known. Equation (1.1) is the measurement equation and equation (2.2) is the state equation. Let $y \equiv (y_1^\top, \ldots, y_n^\top)^\top$ and $\alpha \equiv (\alpha_1^\top, \ldots, \alpha_n^\top)^\top$.

We will consider the familiar and important question of simulation smoothing, which is drawing α as a block from its conditional distribution given y. This is an important component of various sampling methods for learning about the posterior distribution of states, parameters and other functions of interest.

Several authors have proposed ways of drawing states in Gaussian linear state-space models using the Kalman filter, including Carter and Kohn (1994), Frühwirth-Schnatter (1994), de Jong and Shephard (1995), and Durbin and Koopman (2002).

Rue (2001) introduces the Cholesky Factor Algorithm (CFA), an efficient way

to draw Gaussian Markov Random Fields (GMRFs) based on the Cholesky decomposition of the precision (inverse of variance) of the random field. He also recognizes that the conditional distribution of α given y in Gaussian linear state-space models is a special case of a GMRF. Knorr-Held and Rue (2002) comment on the relationship between the CFA and methods based on the Kalman filter.

Chan and Jeliazkov (2009) describe two empirical applications of the CFA algorithm for Bayesian inference in state-space macroeconomic models. One is a time-varying parameter vector autoregression model for output growth, unemployment, income and inflation. The other is a dynamic factor model for U.S. post-war macroeconomic data.

The Kalman filter is used not only for simulation smoothing, but also to evaluate the likelihood function for Gaussian linear state-space models. We can do the same using the CFA and our method. Both give evaluations of $f(\alpha|y)$ for arbitrary α with little additional computation. We can then evaluate the likelihood as

$$f(y) = \frac{f(\alpha)f(y|\alpha)}{f(\alpha|y)}$$

for any value of α. A convenient choice is the conditional mean of α given y, since it is easy to obtain and simplifies the computation of $f(\alpha|y)$.

The Kalman filter also delivers intermediate quantities that are useful for computing filtering distributions, the conditional distributions of $\alpha_1, \ldots, \alpha_t$ given $y_1, \ldots, y_t$, for various values of t. While it is difficult to use the CFA algorithm to compute these distributions efficiently, it is fairly straightforward to do so using our method.

We make four main contributions in this paper. The first is a new method, outlined in Section 1.2, for drawing states in state-space models. Like the CFA, it uses the precision and covector (precision times mean) of the conditional distribution of α given y and does not use the Kalman filter. Unlike the CFA, it generates the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\text{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ as a byproduct. These conditional moments turn out to be useful in an extension of the method, described in McCausland (2012), to non-Gaussian and non-linear

state-space models with univariate states. This is because it facilitates Gaussian and other approximations to the conditional distribution of α_t given α_{t+1} and y. With little additional computation, one can also compute the conditional means $E[\alpha_t|y_1, \ldots, y_t]$ and variances $\mathrm{Var}[\alpha_t|y_1, \ldots, y_t]$, which together specify the filtering distributions, useful for sequential learning.

The second main contribution, described in Sections 1.3 and 1.4, is a careful analysis of the computational efficiency of various methods for drawing states, showing that the CFA and our new method are much more computationally efficient than methods based on the Kalman filter when p is large or when repeated draws of α are required. For the important special case of state-space models, our new method is up to twice as fast as the CFA for large m. We find examples of applications with large p in recent work in macroeconomics and forecasting using "data-rich" environments, where a large number of observed time series is linked to a much smaller number of latent factors. See for example Boivin and Giannoni (2006), who estimate Dynamic Stochastic General Equilibrium (DSGE) models, or Stock and Watson (1999)Stock and Watson (2002) and Forni, Hallin, Lippi, and Reichlin (2000), who show that factor models with large numbers of variables give better forecasts than small-scale vector autoregressive (VAR) models do. Examples with large numbers of repeated draws of α include the evaluation of the likelihood function in non-linear or non-Gaussian state-space models using importance sampling, as in Durbin and Koopman (1997).

Our third contribution is to illustrate these simulation smoothing methods using an empirical application. In Section 1.5, we use them to approximate the likelihood function for a multivariate Poisson state-space model, using importance sampling. Latent states govern time-varying intensities. Observed data are transaction counts in financial markets.

The final contribution is the explicit derivation, in Appendix 1.7.1, of the precision and covector of the conditional distribution of α given y in Gaussian linear state-space models. These two objects are the inputs to the CFA and our new method.

We conclude in Section 1.6.

1.2 Precision-Based Methods for Simulation Smoothing

In this section we discuss two methods for state smoothing using the precision Ω and covector c of the conditional distribution of α given y. The first method is due to Rue (2001), who considers the more general problem of drawing Gaussian Markov random fields. The second method, introduced here, offers new insights and more efficient draws for the special case of Gaussian linear state-space models. Both methods involve pre-computation, which one performs once for a given Ω and c, and computation that is repeated for each draw.

We will take Ω and c as given here. In Appendix 1.7.1, we show how to compute Ω and c in terms of X_t, Z_t, G_t, W_t, T_t, H_t, a_1 and P_1, assuming that the stacked innovation $v_t \equiv ((G_t u_t)^\top, (H_t u_t)^\top)^\top$ has full rank.

The full rank condition is frequently, but not always, satisfied and we note that de Jong and Shephard (1995) and Durbin and Koopman (2002) do not require this assumption. The full rank conditional is not as restrictive as it may appear, however, for two reasons pointed out by Rue (2001).

First, we can draw α conditional on the linear equality restriction $A\alpha + b$ by drawing $\tilde{\alpha}$ unconditionally and then "conditioning by Kriging" to obtain α. This gives $\alpha = \tilde{\alpha} - \Omega^{-1}A^\top(A\Omega^{-1}A^\top)^{-1}(A\tilde{\alpha} + b)$. One can precompute the columns of $\Omega^{-1}A^\top$ in the same way as we compute $\mu = \Omega^{-1}c$ in Appendix 1.7.2, then precompute $A\Omega^{-1}A^\top$ and $-\Omega^{-1}A^\top(A\Omega^{-1}A^\top)^{-1}$.

Second, state-space models where the innovation has less than full rank are often more naturally expressed in another form, one that allows application of the CFA method. Take, for example, a model where a univariate latent variable α_t is an autoregressive process of order p and the measurement equation is given by (1.1). Such a model can be coerced into state-space form, with a p-dimensional state vector and an innovation variance of less than full rank. However, the conditional

distribution of α given y is a GMRF and one can apply the CFA method directly.

Having repeated these points, we acknowledge that the full rank condition is still quite restrictive. Conditioning by Kriging is costly when A has $O(n)$ rows, and it seems to us that simulation smoothing in autoregressive moving average (ARMA) models is impractical using precision based methods.

Rue (2001) introduces a simple procedure for drawing Gaussian random fields. We suppose α is multivariate normal, with a band-diagonal precision matrix Ω and covector c. We let N be the length of α and b be the number of non-zero subdiagonals in Ω. Ω is symmetric, so its bandwidth is $2b + 1$.

Pre-computation consists of computing the Cholesky decomposition $\Omega = LL^\top$ using an algorithm that exploits the band diagonal structure of Ω and then computing $L^{-1}c$ using band back-substitution. To draw $\alpha \sim N(\Omega^{-1}c, \Omega^{-1})$, one draws $\epsilon \sim N(0, I_N)$ and then computes $\alpha = (L^\top)^{-1}([L^{-1}c] + \epsilon)$ using band back-substitution. Here and elsewhere, we use square brackets to denote previously computed quantities. The decomposition and back-substitution operations are standard in commonly used numerical computation libraries : the LAPACK routine DPBTRF computes the Cholesky decomposition of band diagonal matrices, and the BLAS routine DTBSV solves banded triangular systems of equations using band back-substitution.

Rue (2001) recognizes that the vector of states α in Gaussian linear state-space models is an example of a Gaussian Markov random fields. In Appendix 1.7.1, we explicitly derive Ω and c. We note that for the state-space model defined in the introduction, $N = nm$ and $b = 2m - 1$.

We now introduce another method (MMP method hereafter) for drawing α based on the precision Ω and covector c of its conditional distribution given y. It is based on the following result, proved in Appendix 1.7.2.

Result 1.2.1. *If $\alpha|y \sim N(\Omega^{-1}c, \Omega^{-1})$, where Ω has the block band structure of equation (1.13), then*

$$\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y \sim \mathrm{N}(m_t - \Sigma_t\Omega_{t,t+1}\alpha_{t+1}, \Sigma_t) \quad \textit{and} \quad E[\alpha|y] = (\mu_1^\top, \ldots, \mu_n^\top)^\top,$$

where

$$\Sigma_1 = (\Omega_{11})^{-1}, \qquad m_1 = \Sigma_1 c_1,$$

$$\Sigma_t = (\Omega_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t})^{-1}, \qquad m_t = \Sigma_t (c_t - \Omega_{t-1,t}^\top m_{t-1}),$$

$$\mu_n = m_n, \qquad \mu_t = m_t - \Sigma_t \Omega_{t,t+1} \mu_{t+1}.$$

The result is related to a Levinson-like algorithm introduced by Vandebril, Mastronardi, and Van Barel (2007). Their algorithm solves the equation $Bx = y$, where B is an $n \times n$ symmetric band diagonal matrix and y is a $n \times 1$ vector. Result 1.2.1 extends the results in Vandebril, Mastronardi, and Van Barel (2007) in two ways. First, we modify the algorithm to work with $m \times m$ submatrices of a block band diagonal matrix rather than individual elements of a band diagonal matrix. Second, we use intermediate quantities computed while solving the equation $\Omega\mu = c$ for $\mu = E[\alpha|y]$ in order to compute $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$.

Pre-computation involves iterating the following computations for $t = 1, \ldots, n$:

1. Compute the Cholesky decomposition $\Sigma_t^{-1} = \Lambda_t \Lambda_t^\top$, where $\Sigma_1^{-1} = \Omega_{11}$ and $\Sigma_t^{-1} = \Omega_{tt} - [\Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t}]$ for $t > 1$.
2. Compute $\Lambda_t^{-1} \Omega_{t,t+1}$ using triangular back substitution.
3. Compute the symmetric matrix $\Omega_{t,t+1}^\top \Sigma_t \Omega_{t,t+1} = [\Lambda_t^{-1} \Omega_{t,t+1}]^\top [\Lambda_t^{-1} \Omega_{t,t+1}]$.
4. Compute m_t using triangular back substitution twice, where $m_1 = (\Lambda_1^\top)^{-1}(\Lambda_1^{-1} c_1)$ and $m_t = (\Lambda_t^\top)^{-1}(\Lambda_t^{-1}(c_t - \Omega_{t-1,t}^\top m_{t-1}))$ for $t > 1$.

To draw $\alpha \sim N(\Omega^{-1}c, \Omega^{-1})$, we proceed backwards. For $t = n, \ldots, 1$,

1. Draw $\epsilon_t \sim \mathrm{N}(0, I_m)$.
2. Compute α_t using matrix-vector multiplication and back substitution, where $\alpha_n = m_n + (\Lambda_n^\top)^{-1} \epsilon_n$ and $\alpha_t = m_t + (\Lambda_t^\top)^{-1}(\epsilon_t - [\Lambda_t^{-1} \Omega_{t,t+1}]\alpha_{t+1})$ for $t < n$.

We consider now the problem of computing the filtering distribution at time t, the conditional distribution of α_t given $y_1, \ldots, y_t$. Since α and y are jointly multivariate Gaussian, this distribution is also Gaussian and it is enough to compute the mean $E[\alpha_t|y_1, \ldots, y_t]$ and variance $\mathrm{Var}[\alpha_t|y_1, \ldots, y_t]$. It turns out we can do this with very little additional computation.

Fix t and consider the two cases $n = t$ and $n > t$. It is easy to see (in Appendix 1.7.1) that for $\tau = 1, \ldots, t-1$, the values of c_τ, $\Omega_{\tau\tau}$ and $\Omega_{\tau,\tau+1}$ do not differ betweem cases. Therefore the values of m_τ and Σ_τ do not vary from case to case either. We can use equation (1.14) (for Ω_{nn}, taking $n = t$) to compute

$$\tilde{\Omega}_{tt} \equiv Z_t^\top (G_t G_t^\top)^{-1} Z_t + A_{22,t-1},$$

and equation (1.15) (for c_n, taking $n = t$) to compute

$$\tilde{c}_t \equiv Z_t^\top (G_t G_t^\top)^{-1} (y_t - X_t\beta) - A_{21,t-1}(y_{t-1} - X_{t-1}\beta) + A_{22,t-1}(W_{t-1}\beta).$$

Then

$$\mathrm{Var}[\alpha_t | y_1, \ldots, y_t] = \tilde{\Sigma}_t \equiv (\tilde{\Omega}_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t})^{-1}$$

and

$$E[\alpha_t | y_1, \ldots, y_t] = \tilde{m}_t \equiv \tilde{\Sigma}_t (\tilde{c}_t - \Omega_{t-1,t}^\top m_{t-1}).$$

1.3 Efficiency Analysis I : Counts of Operations

We compare the computational efficiency of various methods for drawing $\alpha|y$. We consider separately the fixed computational cost of pre-computation, which is incurred only once, no matter how many draws are needed, and the marginal computational cost required for an additional draw. We do this because there are some applications, such as Bayesian analysis of state-space models using Gibbs sampling, in which only one draw is needed and other applications, such as importance sampling in non-Gaussian models, where many draws are needed.

We compute the cost of various matrix operations in terms of the number of floating point multiplications required per observation. All the methods listed in the introduction have fixed costs that are third order polynomials in p and m. The methods of Rue (2001), Durbin and Koopman (2002) and the present paper all have marginal costs that are second order polynomials in p and m. We will ignore

fixed cost terms of lower order than three and marginal cost terms of lower order than two. The marginal costs are important only when multiple draws are required.

We take the computational cost of multiplying an $N_1 \times N_2$ matrix by an $N_2 \times N_3$ matrix as $N_1 N_2 N_3$ scalar floating-point multiplications. If the result is symmetric or if one of the matrices is triangular, we divide by two. It is possible to multiply matrices more efficiently, but the dimensions required before realizing savings are higher than those usually encountered in state-space models. We take the cost of the Cholesky decomposition of a full $N \times N$ matrix as $N^3/6$ scalar multiplications, which is the cost using the algorithm in Press, Teukolsky, Vetterling, and Flannery (1992, p. 97). When the matrix has bandwidth $2b+1$, the cost is $Nb^2/2$. Solving a triangular system of N equations using back-substitution requires $N^2/2$ scalar multiplications. When the triangular system has bandwidth $b+1$, only Nb multiplications are required.

1.3.1 Fixed Costs

We first consider the cost of computing the precision Ω and covector c, which is required for the methods of Rue (2001) and the current paper.

The cost depends on how we specify the variance of v_t, the stacked innovation. The matrices G_t and H_t are more convenient for methods using the Kalman filter, while the precisions A_t are most useful for the precision-based methods. We recognize that it is often easier to specify the innovation distribution in terms of G_t and H_t rather than A_t. In most cases, however, the A_t are diagonal, constant, or take on one of a small number of values, and so the additional computation required to obtain the A_t is negligible.

There is an important case where it is in fact more natural to provide the A_t. When linear Gaussian state-space models are used as approximations of non-linear or non-Gaussian state-space models, the A_t are typically based on the Hessian matrix of the log observation density of the latter. See Durbin and Koopman (1997) and Section 1.5 of the present paper for examples.

In general, calculation of the Ω_{tt} and $\Omega_{t,t+1}$ is computationally demanding. Ho-

wever, in many cases of interest, A_t, Z_t and T_t are constant, or take on one of a small number of values. In these cases, the computational burden is a constant, not depending on n. We do need to compute each c_t, but provided that the matrix expressions in parentheses in the equations following (1.13) can be pre-computed, this involves matrix-vector multiplications, whose costs are only second order polynomials in p and m.

We now consider the cost of the Kalman filter, which is used in most methods for simulation smoothing. The computations are as follows :

$$e_t = y_t - [X_t\beta] - Z_t a_t, \quad D_t = Z_t P_t Z_t^\top + [G_t G_t^\top],$$

$$K_t = (T_t P_t Z_t^\top + [H_t G_t^\top]) D_t^{-1}, \quad L_t = T_t - K_t Z_t,$$

$$a_{t+1} = [W_t\beta] + T_t a_t + K_t e_t, \quad P_{t+1} = [T_t P_t] L_t^\top + [H_t H_t^\top] + [H_t G_t^\top] K_t$$

As before, we use square brackets for quantities, such as $[T_t P_t]$ above, that are computed in previous steps. Here and elsewhere, we also use them for quantities such as $[H_t H_t^\top]$ that are usually either constant or taking values in a small pre-computable set.

Table 1.1 lists the matrix-matrix multiplications, Cholesky decompositions, and solutions of triangular systems required for three high level operations : an iteration of the Kalman filter, the computation of $\Omega = LL^\top$ using standard methods for band diagonal Ω, and the computation of the Σ_t and m_t of Result 1.2.1. All simulation smoothing methods we are aware of use one of these high-level operations. We represent the solution of triangular systems using notation for the inverse of a triangular matrix, but no actual matrix inversions are performed, as this is inefficient and less numerically reliable. Table 1.1 also gives the number of scalar multiplications for each operation as a function of p and m. Terms of less than third order are omitted, as we ignore matrix-vector multiplications, whose costs are mere second order monomials in m and p.

There are special cases where the Kalman filter computations are less costly. In some of these, the elements of T_t and Z_t are zero or one, and certain matrix

TABLE 1.1 – Scalar multiplications needed for pre-computation.

Method	Operation	Scalar multiplications
Kalman	$P_t Z_t^\top$	$m^2 p$
	$Z_t[P_t Z_t^\top]$	$mp^2/2$
	$T_t[P_t Z_t^\top]$	$m^2 p$
	$D_t = \Upsilon_t \Upsilon_t^\top$ (Cholesky)	$p^3/6$
	$[T_t P_t Z_t^\top + H_t G_t^\top](\Upsilon_t^\top)^{-1}\Upsilon_t^{-1}$	mp^2
	$K_t Z_t$	$m^2 p$
	$T_t P_t$	m^3
	$[T_t P_t] L_t^\top$	m^3
	$[H_t G_t^\top] K_t$	$m^2 p$
CFA	$\Omega = LL^\top$	$2m^3$
MMP	$(\Omega_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1} \Omega_{t-1,t}) = \Lambda_t \Lambda_t^\top$ (Cholesky)	$m^3/6$
	$\Lambda_t^{-1} \Omega_{t,t+1}$	$m^3/2$
	$\Omega_{t,t+1}^\top \Sigma_t \Omega_{t,t+1} = [\Lambda_t^{-1}\Omega_{t,t+1}]^\top [\Lambda_t^{-1}\Omega_{t,t+1}]$	$m^3/2$

multiplications do not require any scalar multiplications. In others, certain matrices are diagonal, reducing the number of multiplications by an order.

The relative efficiency of precision-based methods compared with Kalman filter based methods depends on various features of the application. We see that the precision-based methods have no third order monomials involving p. For the MMP method, the coefficient of the m^3 term is 7/6, compared with 2 for the CFA and 2 for the Kalman filter if $T_t P_t$ is a general matrix multiplication. If T_t is diagonal or composed of zeros and ones, the coefficient of m^3 drops to 1 for the Kalman filter.

1.3.2 Marginal Costs

Compared with the fixed cost of pre-processing, the marginal computational cost of an additional draw from $\alpha|y$ is negligible for all four methods we consider. In particular, no matrix-matrix multiplications, matrix inversions, or Cholesky decompositions are required. However, when large numbers of these additional draws are required, this marginal cost becomes important. It is here that the precision-

based methods are clearly more efficient than those based on the Kalman filter. We use the methods of Durbin and Koopman (2002) and de Jong and Shephard (1995) as benchmarks.

Using the modified simulation smoothing algorithm in Section 2.3 of Durbin and Koopman (2002) (DK hereafter), an additional draw from $\alpha|y$ requires the following computations. We define $\epsilon_t \equiv G_t u_t$ and $\eta_t \equiv H_t u_t$, and assume $G_t^\top H_t = 0$ and $X_t\beta = 0$, recognizing that these assumptions can be easily relaxed. The first step is forward simulation using equations (6) and (7) in that article.

$$x_1 \sim N(0, P_1), \quad v_t^+ = Z_t x_t + \epsilon_t^+ \quad x_{t+1} = T_t x_t - K_t v_t^+ + \eta_t^+,$$

where $\epsilon_t^+ \sim N(0, \Xi_t)$ and $\eta_t^+ \sim N(0, Q_t)$. The next step is the backwards recursion of equation (5) :

$$r_n = 0, \quad r_{t-1} = [Z_t D_t^{-1}] v_t^+ + L_t^\top r_t,$$

and the computation of residuals in equation (4) :

$$\hat{\eta}_t^+ = Q_t r_t.$$

A draw $\tilde{\eta}$ from the conditional distribution of η given y is given by

$$\tilde{\eta} = \hat{\eta} - \hat{\eta}^+ + \eta^+,$$

where $\hat{\eta}$ is a pre-computed vector. To construct a draw $\tilde{\alpha}$ from the conditional distribution of α given y, we use

$$\tilde{\alpha}_1 = \hat{\alpha}_1 - P_1 r_0 + x_1, \quad \tilde{\alpha}_{t+1} = T_t \tilde{\alpha}_t + \tilde{\eta}_t,$$

where $\hat{\alpha}_1$ is pre-computed.

de Jong and Shephard (1995) (DeJS hereafter) draw $\alpha|y$ using the following steps, given in equation (4) of their paper. First ϵ_t is drawn from $N(0, \sigma^2 C_t)$, where the Cholesky factor of $\sigma^2 C_t$ can be pre-computed. Then r_t is computed using the

backwards recursion

$$r_{t-1} = [Z_t^\top D_t^{-1} e_t] + L_t^\top r_t - [V_t^\top C_t^{-1}]\epsilon_t.$$

Next, α_{t+1} is computed as

$$\alpha_{t+1} = [W_t\beta] + T_t\alpha_t + \Omega_t r_t + \epsilon_t.$$

In the MMP approach, we draw, for each observation, a vector $\epsilon_t \sim N(0, I_m)$ and compute

$$\alpha_t = m_t + (\Lambda_t^\top)^{-1}(\epsilon_t - [\Lambda_t^{-1}\Omega_{t,t+1}]\alpha_{t+1}).$$

The matrix-vector multiplication requires m^2 multiplications and the triangular back-substitution requires $m(m-1)/2$ multiplications and m floating point divisions. We can convert the divisions into less costly multiplications if we store the reciprocals of the diagonal elements of Λ_t, obtained during the pre-computation of $\Lambda_t^{-1}\Omega_{t,t+1}$.

The band back-substitution used by Rue (2001) is quite similar to this. However, it is a little less efficient if one is using standard band back-substitution algorithms. These do not take advantage of the special structure of state-space models, for which Ω has elements equal to zero in its first $2m-1$ subdiagonals.

1.4 Efficiency Analysis II : Computational Experiments

The performance of a simulation smoothing method does not only depend on the number of floating point multiplications. In this section, we perform computational experiments with artificial data to illustrate some of the other issues involved. The experiments reveal that these other issues are important.

One issue is whether the method is coded in a high level interpreted language such as Matlab or a lower level programming language such as C. Depending on the

dimension of the problem, the number and depth of loops, and the availability and efficiency of relevant functions in the interpreted language, the cost of interpreting commands may dominate or be dominated by the cost of executing commands for numerical analysis.

Processing resources are also important, particularly the availability of multiple processor cores and an optimized math library that exploits them.

We use two different state-space models and generate two different artificial data sets for each one. The first model is a regression model with time-varying regression parameters. The measurement equation is

$$y_t = x_t\beta_t + \epsilon_t, \quad \epsilon_t \sim \text{i.i.d.N}(0, \sigma_\epsilon^2),$$

where y_t is a univariate observed dependent variable, x_t is an observed m-vector of explanatory variables, and β_t is an unobserved time-varying m-vector of regression coefficients. The dynamics of β_t are given by the state equation

$$\beta_1 \sim \text{N}(a, Q_1) \qquad (\beta_{t+1} - \beta_t) \sim \text{i.i.d. N}(0, Q),$$

and the ϵ_t and β_t are mutually independent. We generate two artificial data sets from the model, one with $m = 4$ and the other with $m = 8$. In both cases, $n = 1000$, $a = 0_m$, $Q_1 = I_m$, $Q = (0.001)^2(\frac{1}{2}I_m + \frac{1}{2}\iota_m\iota_m^\top)$, and $\sigma_\epsilon^2 = 0.05$. I_m is the m-dimensional identity matrix and ι_m is the m-vector with unit elements. We generate the vector of explanatory variables according to $x_{t1} = 1$ and $x_{ti} \sim N(0, 1)$ for $i = 2, \ldots, m$.

The second state-space model is a dynamic factor model of the kind used in "data-rich" environments. The observation and state equations are

$$y_t = Z\alpha_t + u_t, \quad u_t \sim N(0, D),$$

$$\alpha_1 = a + v_1, \quad v_1 \sim N(0, Q_1),$$

$$\alpha_{t+1} = T\alpha_t + v_t, \quad v_t \sim N(0, Q),$$

where y_t is a p-vector of observable dependant variables, α_t is a m-vector of latent factors, a is a fixed vector, Z and T are fixed coefficient matrices and D, Q_1 and Q are fixed covariance matrices, D being diagonal. The u_t and v_t are mutually independent.

For the simulations, we set the following parameter values. We draw the elements of the factor loading matrix Z independently, with $Z_{ij} \sim N(0, (0.001)^2)$ for $i = 1, \ldots, m$ and $j = 1, \ldots, p$. We set $T = 0.9I_m$. We assign the following values to the covariance matrices : $D = I_p$, $a = 0_m$, $Q_1 = I_m$, $Q = (0.2)^2(\frac{1}{2}I_m + \frac{1}{2}\iota_m\iota_m^\top)$.

We generate two artificial data sets from the dynamic factor model. For the first, we use $m = 4$ and $p = 10$, which are relatively small. For the second, we use $m = 10$ and $p = 100$, more typical of data rich environments. For each artificial data set we perform simulation smoothing for the following methods :

DeJS-M Method of de Jong and Shephard (1995), implemented in Matlab

CFA-M Cholesky Factor Algorithm of Rue (2001), implemented in Matlab. The matrix Ω is stored as a sparse matrix and the Cholesky decomposition exploits the sparse structure.

MMP-M Method introduced in Section 1.2, implemented in Matlab.

CFA-C Cholesky Factor Algorithm, implemented in C. The matrix Ω is stored as a band triangular matrix according to the convention of LAPACK. We use the LAPACK routine DPBTRF to compute the Cholesky decomposition of band diagonal matrices, and the BLAS routine DTBSV for band back-substitution.

MMP-C Method introduced in Section 1.2, implemented in C.

We use Matlab R2009a running on a MacBook Pro with a 2.2 GHz Intel Core Duo processor running OS X 10.6.1. We measure running times for Matlab code using the Matlab profiler, and those for C code using the XCode profiler. Results for the time varying parameter model are shown in Table 1.2. For each method, we measure the time required for pre-computation and the time required for each draw. Table 1.3 shows results for the dynamic factor model. Here costs are the total cost of pre-computation and a single draw. We do not report the marginal cost of a draw since importance sampling is impractical for higher dimensional models.

TABLE 1.2 – Costs in ms, Time Varying Parameter model

Algorithm	Pre-computation $m = 4$	Draw $m = 8$	Pre-computation $m = 8$	Draw $m = 8$
MMP-M	126.0	28.2	132.7	29.7
MMP-C	1.178	0.812	5.21	1.74
CFA-M	66.6	0.853	87.7	1.65
CFA-C	2.08	0.737	8.36	1.64
DeJS-M	277.6	64.7	299.1	66.6

TABLE 1.3 – Costs in ms, Dynamic Factor model

Algorithm	$m = 4$, $p = 10$	$m = 10$, $p = 100$
MMP-M	123.3	140.5
MMP-C	1.95	11.07
CFA-M	39.6	79.9
CFA-C	2.81	16.4
DeJS-M	416	1165

Although we report results only for $n = 1000$, experiments not reported here suggest that timing is very close to linear in the number of observations n. This is hardly surprising, given that the numbers of operations required for interpretation and numerical computation both grow linearly in n.

We see clearly that the cost of interpretation dominates the cost of numerical computation for low dimensional problems. This gives a clear advantage to the CFA method, as it does not require loops over t except for the construction of the Ω_{tt} and $\Omega_{t,t+1}$ and then the sparse matrix Ω.

When MMP and CFA are coded in C, there is no longer an interpretation cost. Here, we see that the MMP method is faster than the CFA.

Even for compiled code, we see that the relative costs of CFA and MMP do not exactly correspond to the relative numbers of floating point operations. Experiments not reported here suggest that this is because it is easier to exploit multiple cores for operations on larger matrices and vectors.

1.5 An Empirical Application to Count Models

Durbin and Koopman (1997) show how to compute an arbitrarily accurate evaluation of the likelihood function for a semi-Gaussian state-space model in which the state evolves according to equation (2.2), but the conditional distribution of observations given states is given by a general distribution with density (or mass) function $p(y|\alpha)$. To simplify, we suppress notation for dependence on θ, the vector of parameters.

The approach is as follows. The likelihood function $L(\theta)$ we wish to evaluate is

$$L(\theta) = p(y) = \int p(y, \alpha) d\alpha = \int p(y|\alpha) p(\alpha) d\alpha. \tag{1.4}$$

Durbin and Koopman (1997) employ importance sampling to approximate this integral. The approximating Gaussian model has the same state density $p(\alpha)$, a Gaussian measurement density $g(y|\alpha)$ and likelihood

$$L_g(\theta) = g(y) = \frac{g(y|\alpha)p(\alpha)}{g(\alpha|y)}. \tag{1.5}$$

Substituting $p(\alpha)$ from (1.5) into (1.4) gives

$$L(\theta) = L_g(\theta) \int \frac{p(y|\alpha)}{g(y|\alpha)} g(\alpha|y) d\alpha = L_g(\theta) E_g[w(\alpha)], \tag{1.6}$$

where

$$w(\alpha) \equiv \frac{p(y|\alpha)}{g(y|\alpha)}.$$

One can generate a random sample $\alpha^{(1)}, \ldots, \alpha^{(N_s)}$ from the density $g(\alpha|y)$ using any of the methods for drawing states in fully Gaussian models, then compute a Monte Carlo approximation of $L(\theta)$.

The approximating state-space model has the form

$$y_t = \mu_t + Z\alpha_t + \epsilon_t, \tag{1.7}$$

where the ϵ_t are independent $N(0, \Xi_t)$ and independent of the state equation innovations. The Gaussian measurement density $g(y|\alpha)$ is chosen such that the Hessian (with respect to α) of $\log g(y|\alpha)$ matches the Hessian of $\log p(y|\alpha)$ at $\hat{\alpha}$, the conditional mode of α given y. Durbin and Koopman (1997) use routine Kalman filtering and smoothing to find $\hat{\alpha}$.

1.5.1 Modifications to the Algorithm for Approximating $L(\theta)$

We propose here three modifications of the Durbin and Koopman (1997) method for approximating $L(\theta)$. The modified method does not involve Kalman filtering.

First, we use the MMP algorithm to draw α from its conditional distribution given y.

Second, we compute $L_g(\theta)$ as the extreme right hand side of equation (1.5). The equation holds for any value of α; convenient choices which simplify computations include the prior mean and the posterior mean. We use the posterior mean.

Finally, we calculate $\hat{\alpha}$ using Result 1.2.1, as described in the rest of this section. As in Durbin and Koopman (1997), the method is essentially the Newton method. The difference lies in the implementation.

We iterate the following steps until convergence.

1. Using the current value of $\hat{\alpha}$, find the precision $\bar{\bar{H}}$ and co-vector $\bar{\bar{c}}$ of a Gaussian approximation to $p(\alpha|y)$ based on a second-order Taylor expansion of $\log p(\alpha) + \log p(y|\alpha)$ around the point $\hat{\alpha}$.
2. Using the current values of $\bar{\bar{H}}$ and $\bar{\bar{c}}$, compute $\hat{\alpha} = \bar{\bar{H}}^{-1}\bar{\bar{c}}$, the mean of the Gaussian approximation, using Result 1.2.1.

We compute the precision $\bar{\bar{H}}$ as $\bar{H} + \tilde{H}$, and the co-vector $\bar{\bar{c}}$ as $\bar{c} + \tilde{c}$, where $\bar{H}$ and $\bar{c}$ are the precision and co-vector of the marginal distribution of α (detailed formulations are provided for our example in the next section), and $\tilde{H}$ and $\tilde{c}$ are the precision and co-vector of the Gaussian density with mean $\hat{\alpha}$ and variance equal

to the negative inverse of the Hessian of $\log p(y|\alpha)$ at $\hat{\alpha}$. Since $\tilde{H}$ is block-diagonal and $\bar{H}$ is block-band-diagonal, $\bar{\bar{H}}$ is also block-band-diagonal.

We compute $\tilde{H}$ and $\tilde{c}$ as follows. Let $a(\alpha_t) \equiv -2\log[p(y_t|\alpha_t)]$. We approximate $a(\alpha_t)$ by $\tilde{a}(\alpha_t)$, consisting of the first three terms of the Taylor expansion of $a(\alpha_t)$ around $\hat{\alpha}_t$:

$$a(\alpha_t) \approx \tilde{a}(\alpha_t) = a(\hat{\alpha}_t) + \frac{\partial a(\hat{\alpha}_t)}{\partial \alpha_t}(\alpha_t - \hat{\alpha}_t) + \frac{1}{2}(\alpha_t - \hat{\alpha}_t)^\top \frac{\partial^2 a(\hat{\alpha}_t)}{\partial \alpha_t \partial \alpha_t^\top}(\alpha_t - \hat{\alpha}_t).$$

If we complete the square, we obtain

$$\tilde{a}(\alpha_t) = (\alpha_t - h_t^{-1}c_t)^\top h_t(\alpha_t - h_t^{-1}c_t) + k,$$

where

$$h_t = \frac{1}{2}\frac{\partial^2 a(\hat{\alpha}_t)}{\partial \alpha_t \partial \alpha_t^\top}, \quad c_t = h_t\hat{\alpha}_t - \frac{1}{2}\frac{\partial a(\hat{\alpha}_t)}{\partial \alpha_t},$$

and k is an unimportant term not depending on α_t. Note that h_t and c_t are the precision and co-vector of a multivariate normal distribution with density proportional to $\exp[-\frac{1}{2}\tilde{a}(\alpha_t)]$.

Since $\log p(y|\alpha)$ is additively separable in the elements of α, it means that it is reasonably well approximated, as a function of α, by $\prod_{t=1}^n \exp[-\frac{1}{2}\tilde{a}(\alpha_t)]$, which is proportional to a multivariate normal distribution with precision $\tilde{H}$ and co-vector $\tilde{c}$, given by

$$\tilde{H} \equiv \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_n \end{bmatrix} \quad \text{and} \quad \tilde{c} \equiv \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

1.5.2 A Multivariate Poisson Model with Time-Varying Intensities

As an example of a semi-Gaussian state-space model, let us consider a case where $y_t \equiv (y_{t1}, \ldots, y_{tp})$ is a vector of observed counts. We assume that the y_{ti} are conditionally independent Poisson with intensities λ_{ti}, so that the conditional density of y_t given $\lambda_{t1}, \ldots, \lambda_{tp}$ is

$$p(y_{t1}, \ldots, y_{tp} | \lambda_{t1}, \ldots, \lambda_{tp}) = \prod_{i=1}^{p} \frac{\exp(-\lambda_{ti}) \lambda_{ti}^{y_{ti}}}{y_{ti}!}. \tag{1.8}$$

The latent count intensities $\lambda_{t1}, \ldots, \lambda_{tp}$ are assumed to follow a factor model :

$$\lambda_{ti} = \exp\left(\sum_{j=1}^{m} z_{ij} \alpha_{tj} \right), \quad i = 1, \ldots, n, \tag{1.9}$$

$$\alpha_{t+1,j} = (1 - \phi_j)\bar{\alpha}_j + \phi_j \alpha_{tj} + \eta_{tj}, \quad j = 1, \ldots, m, \tag{1.10}$$

where the η_{tj} are independent $N(0, Q_j)$ and the distribution of α_1 is the stationary distribution, so that the $\alpha_{1,j}$ are independent $N(\bar{\alpha}_j, Q_j/(1 - \phi_j^2))$.

Denote by Q the diagonal matrix $\mathrm{diag}(Q_1, \ldots, Q_m)$. The vector of model parameters is $\theta \equiv (\bar{\alpha}_j, \phi_j, Q_j, z_{ij})_{i \in \{1,\ldots,p\}, j \in \{1,\ldots,m\}}$. To ensure identification [2], we impose $z_{ii} = 1$ and $z_{ij} = 0$ for $j > i$.

We now turn to the problem of estimating the likelihood $L(\theta)$ of this particular semi-Gaussian model using the approach of Durbin and Koopman (1997). For this

2. See for example Heaton and Solo (2004).

example, the precision $\bar{H}$ and co-vector $\bar{c}$, are given by

$$\bar{H} = \begin{bmatrix} \bar{H}_{11} & \bar{H}_{12} & 0 & \cdots & 0 & 0 \\ \bar{H}_{21} & \bar{H}_{22} & \bar{H}_{23} & \cdots & 0 & 0 \\ 0 & \bar{H}_{32} & \bar{H}_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{H}_{n-1,n-1} & \bar{H}_{n-1,n} \\ 0 & 0 & 0 & \cdots & \bar{H}_{n,n-1} & \bar{H}_{nn} \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \\ \vdots \\ \bar{c}_{n-1} \\ \bar{c}_n \end{bmatrix},$$

where

$$\begin{aligned} \bar{H}_{11} &= \bar{H}_{nn} = Q^{-1}, \\ \bar{H}_{tt} &= \begin{bmatrix} (1+\phi_1^2)/Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (1+\phi_m^2)/Q_m \end{bmatrix}, \quad t = 2, \ldots, n-1, \\ \bar{H}_{t,t+1} &= \bar{H}_{t+1,t} = \begin{bmatrix} -\phi_1/Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\phi_m/Q_m \end{bmatrix}, \quad t = 1, \ldots, n-1, \\ \bar{c}_1 &= \bar{c}_n = \begin{bmatrix} \bar{\alpha}_1(1-\phi_1)/Q_1 \\ \vdots \\ \bar{\alpha}_m(1-\phi_m)/Q_m \end{bmatrix}, \\ \bar{c}_t &= \begin{bmatrix} \bar{\alpha}_1(1-\phi_1)^2/Q_1 \\ \vdots \\ \bar{\alpha}_m(1-\phi_m)^2/Q_m \end{bmatrix}, \quad t = 2, \ldots, n-1. \end{aligned}$$

We compare the computational efficiency of all three methods for estimating the likelihood for this semi-Gaussian state-space model. We do so by counting operations and profiling code. Since a large number of draws from $g(\alpha|y)$ is required for a good approximation of $L(\theta)$, we focus on the marginal computational cost of an additional draw, the overhead associated with the first draw being small. For

TABLE 1.4 – Computational costs per observation per additional draw of α_t

Algorithm	$\times$	$N_{0,1}$
DeJS	$(3p^2+p)/2+2mp+m^2$	p
DK	$(5p^2+p)/2+4mp+2m+m^2$	$p+m$
CFA	$2m^2+pm$	m
MMP	$(3m^2+m)/2+pm$	m

all four methods, we compute $\hat{\alpha}$ using the fast method presented in Section 1.5.1.

We have already seen how to make an incremental draw using the various methods. For both MMP and CFA, we add $p \times m$ multiplications for each of the $Z\alpha_t$, which are required to evaluate $p(y|\alpha)$. The computational costs per observation for an additional draw of α_t are summarized in Table 1.4.

We profile code for all four methods to see how they perform in practice. We use data from the New York Stock Exchange Trade and Quote database on the stocks of four gold mining companies : Agnico-Eagle Mines Limited, Barrick Gold Corporation, Gold Fields Limited and Goldcorp Inc. For each stock, we observe transaction counts for 195 consecutive two minute intervals covering trading hours on November 6, 2003. The data are plotted in Figure 1.1.

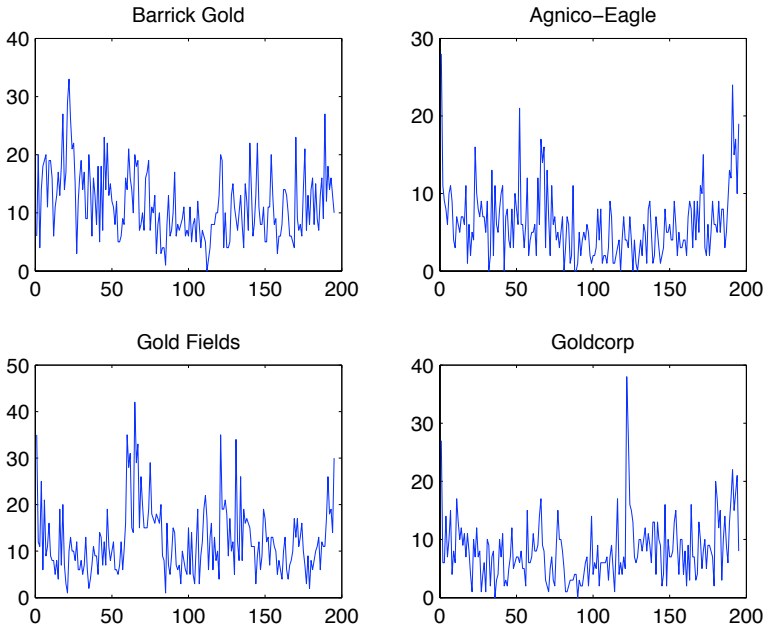
For the case where the number of factors is equal to the number of series, that is $m = p = 4$, and for various values of N_s, Table 1.5 gives the time cost in 100ths of seconds of generating N_s draws of α. All times are averaged over 10,000 replications [3]. We report results for two implementations of the MMP and CFA algorithms, one Matlab only (MMP-M, CFA-M) and a second where precomputation (Cholesky decomposition of Ω for CFA, steps 1 and 2 of MMP) and draws (band back-substitution for CFA, steps 3 and 4 of MMP) are coded in C (MMP-C, CFA-C). The implementation of MMP in C gives a better comparison with the Matlab implementation of CFA that is able to use specialized libraries to compute the banded Cholesky decomposition and perform the band back-substitution [4].

3. The simulations were performed on a MacBook Pro 2.4 GHz with Matlab R2008b.

4. By declaring Ω to be a sparse matrix, Matlab can use cholmod, a sparse Cholesky factorization package.

First, we see that for a single draw, DK is slightly faster than DeJS and MMP (Matlab). For larger numbers of draws, MMP is fastest. Second, these first three methods are dominated for every value of N_s by the CFA-M algorithm. This is the result of requiring less operations (compared to DeJS and DK) and being very efficiently implemented in Matlab. There are no loops over t, which reduces interpretation costs. Third, implementing CFA in C so it uses LAPACK and BLAS routines for banded triangular matrices and systems is computationally more efficient than Matlab's built-in functions. Fourth, we see that MMP-C is faster than CFA-C. As a point of reference, Durbin and Koopman (1997) consider $N_s = 200$ (combined with antithetic and control variables) as an acceptable value in an empirical example they consider.

FIGURE 1.1 – Transactions data



We next discuss the results of the estimation of this multivariate count data

TABLE 1.5 – Time cost of drawing $\alpha^{(i)}$ as a function of the number of draws N_s. Figures are in 100ths of seconds.

Method	$N_s = 1$	$N_s = 10$	$N_s = 50$	$N_s = 150$	$N_s = 250$
DeJS	8.14	8.43	8.89	10.61	11.78
DK	5.56	6.25	7.32	11.01	13.49
MMP-M	5.88	6.12	6.28	6.88	7.47
CFA-M	1.88	1.98	2.16	2.65	3.10
MMP-C	1.11	1.18	1.34	1.90	2.30
CFA-C	1.13	1.21	1.44	2.19	2.73

model. The estimates, standard errors [5] and log-likelihood values are presented in Table 1.6 for different values of m, the number of latent factors. These results are obtained with $N_s = 500$ and antithetic variables. To select a value for m we cannot use a test statistic such as the likelihood ratio test with the usual χ^2 limit distribution. For example, a likelihood ratio test for $m = 1$ versus $m = 2$ where we test $z_{32} = z_{42} = 0$ leaves the parameters $\bar{\alpha}_2$, ϕ_2 and Q_2 unidentified under the null. An alternative is to use an information criterion such as $AIC = -2 \log L(\theta) + 2 \dim(\theta)$ and $SIC = -2 \log L(\theta) + \log(pn) \dim(\theta)$. See Song and Belin (2008) for an example. These two criteria both suggest that m should equal four. For the model with $m = 4$, we can see that the first factor is the more persistent with $\hat{\phi}_1 = 0.7710$. It is also the factor with the highest innovation variance and the highest factor loadings, the three largest z's being z_{21}, z_{31} and z_{41}.

1.6 Conclusions

In this paper we introduce a new method for drawing state variables in Gaussian state-space models from their conditional distribution given parameters and observations. The method is quite different from standard methods, such as those of de Jong and Shephard (1995) and Durbin and Koopman (2002), that use Kalman filtering. It is much more in the spirit of Rue (2001), who describes an efficient me-

5. See Durbin and Koopman (2001, Chapter 12).

TABLE 1.6 – Estimation results for the model for different values of m. The standard errors are between parantheses.

	$m=1$		$m=2$		$m=3$		$m=4$	
$\bar{\alpha}_1$	2.0569	(0.0101)	2.0207	(0.0050)	1.9999	(0.0140)	2.0013	(0.0049)
$\bar{\alpha}_2$			0.4022	(0.0718)	0.7311	(0.2105)	0.7004	(0.0821)
$\bar{\alpha}_3$					0.5360	(0.0218)	0.2939	(0.0184)
$\bar{\alpha}_4$							0.3858	(0.0489)
ϕ_1	0.7873	(0.0007)	0.7402	(0.0255)	0.7595	(0.0079)	0.7710	(0.0059)
ϕ_2			0.1780	(0.0549)	0.1233	(0.0353)	0.2829	(0.0144)
ϕ_3					0.0886	(0.0162)	0.0412	(0.0020)
ϕ_4							0.1182	(0.0025)
Q_1	0.1582	(0.0019)	0.2225	(0.0207)	0.2250	(0.0041)	0.2142	(0.0077)
Q_2			0.0321	(0.0030)	0.1316	(0.0628)	0.1378	(0.0133)
Q_3					0.1451	(0.0262)	0.1678	(0.0074)
Q_4							0.1405	(0.0120)
z_{21}	0.9928	(0.0064)	0.8170	(0.0358)	0.6626	(0.1000)	0.6714	(0.0388)
z_{31}	0.9965	(0.0065)	0.5830	(0.1243)	0.6449	(0.0380)	0.6738	(0.0242)
z_{41}	0.9882	(0.0065)	0.6073	(0.1192)	0.5785	(0.0148)	0.6098	(0.0359)
z_{32}			2.2228	(0.2558)	0.3012	(0.0665)	0.5203	(0.0078)
z_{42}			2.0586	(0.2555)	0.6674	(0.4069)	0.4394	(0.0247)
z_{43}					0.7747	(0.3121)	0.3856	(0.0105)
$\log L(\hat{\theta})$	-2432.71		-2378.44		-2350.67		-2324.22	

thod for drawing Gaussian random vectors with band diagonal precision matrices. As Rue (2001) recognizes, the distribution $\alpha|y$ in linear Gaussian state-space models is an example.

Our first contribution is computing Ω and c for a widely used and fairly flexible state-space model. These are required inputs for both the CFA of Rue (2001) and the method we described here.

Our second contribution is a new precision-based state smoothing algorithm. It is more computationally efficient for the special case of state-space models, and delivers the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ as a byproduct. These conditional moments turn out to be very useful in an extension of the method, described in McCausland (2012), to non-linear and non-Gaussian state-space models with univariate states.

The algorithm is an extention of a Levinson-like algorithm introduced by Vandebril, Mastronardi, and Van Barel (2007), for solving the equation $Bx = y$, where B is an $n \times n$ symmetric band diagonal matrix and y is a $n \times 1$ vector. The algorithm extends theirs in two ways. First, we modify the algorithm to work with $m \times m$ submatrices of a block band diagonal matrix rather than individual elements of a band diagonal matrix. Second, we use intermediate quantities computed while solving the equation $\Omega\mu = c$ for the mean μ given the precision Ω and co-vector c in order to compute the conditional means $E[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$ and conditional variances $\mathrm{Var}[\alpha_t|\alpha_{t+1}, \ldots, \alpha_n, y]$.

Our third contribution is a computational analysis of several state smoothing methods. One can often precompute the Ω_{tt} and $\Omega_{t,t+1}$, in which case the precision-based methods are more efficient than those based on the Kalman filter. The advantage is particularly strong when p is large or when several draws of α are required for each value of the parameters. Kalman filtering, which involves solving systems of p equations in p unknowns, requires $O(p^3)$ scalar multiplications. If the A_t can be pre-computed, or take on only a constant number of values, the precision-based methods require no operations of higher order than p^2, in p. If the Z_t and T_t can also be pre-computed, or take on only a constant number of values, the order drops

to p. For large m, our method involves half as many scalar multiplications as CFA.

Illustrations with artificial data reveal that performance does not depend only on the number of floating point multiplications. Whether numerical computations are implemented in high level interpreted code or low level compiled code is important when m and p are small and, consequently, the relative burden of interpreting code in loops is high. Even when computations are performed in compiled code, operations on higher dimension vectors and matrices may be relatively more efficient if they can exploit multiple cores.

We consider an applications of our methods to the evaluation of the log-likelihood function for a multivariate Poisson model with latent count intensities.

We have learned several things relevant to the choice of a simulation smoothing method for a given state-space model. It is clear that no method dominates the others in all cases, and that much depends on the details of the state-space model, its dimensions, whether the user is using a high level language such as Matlab or a low level language such as C, the number of draws required for each value of the parameters, and whether or not sequential learning is important.

The two precision-based methods are naturally suited for models with large values of p, such as those used in data rich environments, or when one needs large numbers of repeated draws, as when one applies importance sampling for non-linear or non-Gaussian models. On the other hand, they are not well suited for state-space models such as ARMA models that cannot be expressed in a form where the variance of the stacked innovation term has full rank. They may also be less efficient than methods of de Jong and Shephard (1995) or Durbin and Koopman (2002), based on Kalman filtering, when the computation of the Ω_{tt} or $\Omega_{t,t+1}$ requires a number of operations that is third order in m and p. This is the case when Z_t or T_t or the innovation precision A_t are full matrices taking on different values at every value of t.

Of the two precision-based methods, the CFA method is best suited for low-dimensional models implemented in interpreted languages such as Matlab, provided that the language has routines for efficient Cholesky decomposition and back sub-

stitution, either for sparse matrices or for banded matrices. The MMP method is better suited for larger dimensional models. It is for these models that the benefits of coding in a compiled language are greatest. Once the decision to use a compiled language is made, the MMP method offers further computational efficiency by avoiding multiplications by zero. The MMP method is also valuable when sequential learning is required.

1.7 Appendix to Chapter 1

1.7.1 Derivation of Ω and c

Here we derive expressions for the precision Ω and covector c of the conditional distribution of α given y, for the Gaussian linear state-space model described in equations (1.1), (2.2) and (1.3). The matrix Ω and vector c are required inputs for the CFA method and our new method.

Let v_t be the stacked period-t innovation :

$$v_t = \begin{bmatrix} G_t u_t \\ H_t u_t \end{bmatrix}.$$

We will assume that the variance of v_t has full rank.

We define the matrix A_t as the precision of v_t and then partition it as :

$$A_t \equiv \begin{bmatrix} G_t G_t^\top & G_t H_t^\top \\ H_t G_t^\top & H_t H_t^\top \end{bmatrix}^{-1} = \begin{bmatrix} A_{11,t} & A_{12,t} \\ A_{21,t} & A_{22,t} \end{bmatrix},$$

where $A_{11,t}$ is the leading $p \times p$ submatrix.

Clearly α and y are jointly Gaussian and therefore the conditional distribution of α given y is also Gaussian. We can write the log conditional density of α given y as

$$\log f(\alpha|y) = -\frac{1}{2}\left[\alpha^\top \Omega \alpha - 2c^\top \alpha\right] + k, \tag{1.11}$$

where k is an unimportant term not depending on α. Using the definition of the model in equations (1.1), (2.2) and (1.3) we can also write

$$\log f(\alpha|y) = \log f(\alpha, y) - \log f(y) = -\frac{1}{2} g(\alpha, y) + k', \tag{1.12}$$

where

$$\begin{aligned} g(\alpha, y) &= (\alpha_1 - a_1)^\top P_1^{-1}(\alpha_1 - a_1) \\ &+ \sum_{t=1}^{n-1} \begin{bmatrix} y_t - X_t\beta - Z_t\alpha_t \\ \alpha_{t+1} - W_t\beta - T_t\alpha_t \end{bmatrix}^\top A_t \begin{bmatrix} y_t - X_t\beta - Z_t\alpha_t \\ \alpha_{t+1} - W_t\beta - T_t\alpha_t \end{bmatrix} \\ &+ (y_n - X_n\beta - Z_n\alpha_n)^\top (G_n G_n^\top)^{-1}(y_n - X_n\beta - Z_n\alpha_n), \end{aligned}$$

and k' is a term not depending on α.

Matching linear and quadratic terms in the α_t between equations (1.11) and (1.12), we obtain

$$\Omega \equiv \begin{bmatrix} \Omega_{11} & \Omega_{12} & 0 & \dots & 0 \\ \Omega_{12}^\top & \Omega_{22} & \Omega_{23} & \ddots & \vdots \\ 0 & \Omega_{23}^\top & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \Omega_{n-1,n-1} & \Omega_{n-1,n} \\ 0 & \dots & 0 & \Omega_{n-1,n}^\top & \Omega_{nn} \end{bmatrix} \quad c \equiv \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \tag{1.13}$$

where

$$\Omega_{11} \equiv Z_1^\top A_{11,1} Z_1 + Z_1^\top A_{12,1} T_1 + T_1^\top A_{21,1} Z_1 + T_1^\top A_{22,1} T_1 + P_1^{-1},$$

$$\Omega_{tt} \equiv Z_t^\top A_{11,t} Z_t + Z_t^\top A_{12,t} T_t + T_t^\top A_{21,t} Z_t + T_t^\top A_{22,t} T_t + A_{22,t-1}, \quad t = 2, \dots, n-1,$$

$$\Omega_{nn} \equiv Z_n^\top (G_n G_n^\top)^{-1} Z_n + A_{22,n-1}, \tag{1.14}$$

$$\Omega_{t,t+1} \equiv -Z_t^\top A_{12,t} - T_t^\top A_{22,t}, \quad t = 1, \dots, n-1,$$

$$c_1 \equiv (Z_1^\top A_{11,1} + T_1^\top A_{21,1})(y_1 - X_1\beta) - (Z_1^\top A_{12,1} + T_1^\top A_{22,1})(W_1\beta) + P_1^{-1} a_1,$$

$$\begin{aligned} c_t &\equiv (Z_t^\top A_{11,t} + T_t^\top A_{21,t})(y_t - X_t\beta) - (Z_t^\top A_{12,t} + T_t^\top A_{22,t})(W_t\beta) \\ &\quad - A_{21,t-1}(y_{t-1} - X_{t-1}\beta) + A_{22,t-1}(W_{t-1}\beta), \quad t = 2, \dots, n-1, \end{aligned}$$

$$c_n \equiv Z_n^\top (G_n G_n^\top)^{-1}(y_n - X_n\beta) - A_{21,n-1}(y_{n-1} - X_{n-1}\beta) + A_{22,n-1}(W_{n-1}\beta). \quad (1.15)$$

1.7.2 Proof of Result 1.2.1

Suppose $\alpha|y \sim N(\Omega^{-1}c, \Omega^{-1})$ and define

$$\Sigma_1 = \Omega_{11}^{-1}, \qquad m_1 = \Sigma_1 c_1,$$

$$\Sigma_t = (\Omega_{tt} - \Omega_{t-1,t}^\top \Sigma_{t-1}\Omega_{t-1,t})^{-1}, \qquad m_t = \Sigma_t(c_t - \Omega_{t-1,t}^\top m_{t-1}).$$

Now let $\mu_n \equiv m_n$ and for $t = n-1, \ldots, 1$, let $\mu_t = m_t - \Sigma_t\Omega_{t,t+1}\mu_{t+1}$. Let $\mu = (\mu_1^\top, \ldots, \mu_n^\top)^\top$.

We first show that $\Omega\mu = c$, which means that $\mu = E[\alpha|y]$:

$$\begin{aligned} \Omega_{11}\mu_1 + \Omega_{12}\mu_2 &= \Omega_{11}(m_1 - \Sigma_1\Omega_{12}\mu_2) + \Omega_{12}\mu_2 \\ &= \Omega_{11}(\Omega_{11}^{-1}c_1 - \Omega_{11}^{-1}\Omega_{12}\mu_2) + \Omega_{12}\mu_2 = c_1. \end{aligned}$$

For $t = 2, \ldots, n-1$,

$$\begin{aligned} &\Omega_{t-1,t}^\top\mu_{t-1} + \Omega_{tt}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &\quad = \Omega_{t-1,t}^\top(m_{t-1} - \Sigma_{t-1}\Omega_{t-1,t}\mu_t) + \Omega_{tt}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &\quad = \Omega_{t-1,t}^\top m_{t-1} + (\Omega_{tt} - \Omega_{t-1,t}^\top\Sigma_{t-1}\Omega_{t-1,t})\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &\quad = \Omega_{t-1,t}^\top m_{t-1} + \Sigma_t^{-1}\mu_t + \Omega_{t,t+1}\mu_{t+1} \\ &\quad = \Omega_{t-1,t}^\top m_{t-1} + \Sigma_t^{-1}(m_t - \Sigma_t\Omega_{t,t+1}\mu_{t+1}) + \Omega_{t,t+1}\mu_{t+1} \\ &\quad = \Omega_{t-1,t}^\top m_{t-1} + (c_t - \Omega_{t-1,t}^\top m_{t-1}) = c_t. \end{aligned}$$

$$\begin{aligned}\Omega_{n,n-1}\mu_{n-1} + \Omega_{nn}\mu_n &= \Omega_{n,n-1}(m_{n-1} - \Sigma_{n-1}\Omega_{n-1,n}\mu_n) + \Omega_{nn}\mu_n \\ &= \Omega_{n,n-1}m_{n-1} + \Sigma_n^{-1}\mu_n \\ &= \Omega_{n,n-1}m_{n-1} + \Sigma_n^{-1}m_n \\ &= \Omega_{n,n-1}m_{n-1} + (c_n - \Omega_{n,n-1})m_{n-1} = c_n.\end{aligned}$$

We will now prove that $E[\alpha_t|\alpha_{t+1},\ldots,\alpha_n,y] = m_t - \Sigma_t\Omega_{t,t+1}\alpha_{t+1}$ and that $\mathrm{Var}[\alpha_t|\alpha_{t+1},\ldots,\alpha_n,y] = \Sigma_t$. We begin with the standard result

$$\alpha_{1:t}|\alpha_{t+1:n},y \sim \mathrm{N}\left(\mu_{1:t} - \Omega_{1:t,1:t}^{-1}\Omega_{1:t,t+1:n}(\alpha_{t+1:n} - \mu_{t+1:n}), \Omega_{1:t,1:t}^{-1}\right),$$

where μ, α and Ω are partitioned as

$$\mu = \begin{bmatrix}\mu_{1:t}\\ \mu_{t+1:n}\end{bmatrix}, \quad \alpha = \begin{bmatrix}\alpha_{1:t}\\ \alpha_{t+1:n}\end{bmatrix}, \quad \Omega = \begin{bmatrix}\Omega_{1:t,1:t} & \Omega_{1:t,t+1:n}\\ \Omega_{t+1:n,1:t} & \Omega_{t+1:n,t+1:n}\end{bmatrix},$$

with $\mu_{1:t}$, $\alpha_{1:t}$ and $\Omega_{(11)}$ having dimensions $tm\times 1$, $tm\times 1$, and $tm\times tm$ respectively. Note that the only non-zero elements of $\Omega_{(12)}$ come from $\Omega_{t,t+1}$. We can therefore write the univariate conditional distribution $\alpha_t|\alpha_{t+1:n}$ as

$$\alpha_t|\alpha_{t+1:n} \sim \mathrm{N}(\mu_t - (\Omega_{1:t,1:t}^{-1})_{tt}\Omega_{t,t+1}(\alpha_{t+1} - \mu_{t+1}), (\Omega_{1:t,1:t}^{-1})_{tt}).$$

The following inductive proof establishes the result $\mathrm{Var}[\alpha_t|\alpha_{t+1},\ldots,\alpha_n,y] = \Sigma_t$:

$$(\Omega_{11})^{-1} = \Sigma_1$$

$$\begin{aligned}(\Omega_{1:t,1:t}^{-1})_{tt} &= (\Omega_{tt} - \Omega_{t,1:t-1}\Omega_{1:t-1,1:t-1}^{-1}\Omega_{1:t-1,t})^{-1} \\ &= (\Omega_{tt} - \Omega_{t-1,t}^{\top}\Sigma_{t-1}\Omega_{t-1,t})^{-1} = \Sigma_t.\end{aligned}$$

As for the conditional mean,

$$E[\alpha_t|\alpha_{t+1},\ldots,\alpha_n,y] = \begin{cases} \mu_t - \Sigma_t\Omega_{t,t+1}(\alpha_{t+1}-\mu_{t+1}) & t=1,\ldots,n-1 \\ \mu_n & t=n. \end{cases}$$

By the definition of μ_t, $m_t = \mu_t + \Sigma_t\Omega_{t,t+1}\mu_{t+1}$, so we obtain

$$E[\alpha_t|\alpha_{t+1},\ldots,\alpha_n,y] = \begin{cases} m_t - \Sigma_t\Omega_{t,t+1}\alpha_{t+1} & t=1,\ldots,n-1 \\ m_n & t=n. \end{cases}$$

Chapter 2 :
Multivariate Stochastic Volatility

2.1 Introduction

Multivariate volatility models are a powerful inferential tool. By featuring different kinds of dynamic cross-sectional dependence among multiple asset returns, they can capture many different stylized facts.

It is well known that asset return volatility varies over time, changing in response to news and revised expectations of future performance. It tends to cluster, so that large price changes tend to be followed by other large changes. Volatility is not independent across markets and assets, and this cross-sectional dependence is time-varying. Cross-sectional correlations increase substantially in periods of high market volatility, especially in bear markets. The distribution of returns is heavy tailed compared with the normal distribution, even when one conditions on current market conditions. There is an asymmetric relation between price and volatility changes known as the "leverage effect" : increases in volatility are associated more with large decreases in price than with large increases.

Multivariate volatility models that can capture these stylized facts are in high demand in finance given their many important applications, especially in modern portfolio management. Learning about the joint distribution of asset returns is a key element for the construction, diversification, evaluation and hedging of portfolios. Accurate estimation of the covariance matrix of multiple asset returns allows

the investor to timely identify opportunities or risks associated with a particular portfolio. It is important to track changes in correlations to assess the risk of a portfolio, especially during periods of market stress. Financial crises usually have a strong impact on correlation and diversification is least effective at reducing risk at the very times when risk is highest.

As with univariate volatility models, there are two main types of multivariate volatility models : observation-driven and parameter-driven. In observation-driven models, volatility is a deterministic function of observed variables, which allows straightforward evaluation of the likelihood function. This advantage has made the observation-driven GARCH model and its extensions very popular for univariate problems.

In parameter-driven volatility models, known as stochastic volatility (SV) models, volatility is a latent stochastic process. Jacquier, Polson, and Rossi (1994) and Geweke (1994) give evidence suggesting that SV models are more realistic. They are also more natural discrete time representations of the continuous time models upon which much of modern finance theory is based. Unfortunately, computation of the likelihood function, which amounts to integrating out latent states, is difficult. However, since the introduction of Bayesian Markov chain Monte Carlo (MCMC) methods by Jacquier, Polson, and Rossi (1994) for univariate SV models, inference for these models has become much more feasible. These methods require the evaluation of the joint density of returns, states and parameters, which is straightforward. In addition, simulation methods for Bayesian inference make exact finite sample inference possible.

This paper focuses on multivariate stochastic volatility (MSV) models, which are parameter-driven. For a literature review of multivariate GARCH type models, which are observation-driven, see Bauwens, Laurent, and Rombouts (2006). We propose new MCMC methods for Bayesian analysis of MSV models, based on efficient draws of volatility from its conditional posterior distribution.

There are many different types of MSV models. In Section 2, we describe a MSV model that encompasses several special cases of interest and compare it to other

models. Two difficulties arise when we extend volatility models to the multivariate case. First, the conditional variance of returns given states must be positive definite at every point in time. Second, there is a severe trade-off between parsimony and flexibility. As the number of assets increases, the number of potential parameters increases quickly, leading to a danger of overfitting. Reining in the number of parameters forces the modeler to make choices, and much of the difference between MSV model specifications reflects a choice about how to do this. This has implications on which stylized facts can be captured by the model.

We show that our estimation approach is quite flexible and we do not rely much on any special structure for the MSV model considered. It applies to models with several kinds of cross-sectional dependence. We can specify full first order VAR coefficient and covariance matrices for the evolution of volatilities. We can include a mean factor structure, which allows conditional return correlations, given asset and factor volatilities, to vary over time, and for these correlations to covary with variances. We can also model cross-sectional conditional return dependence given latent asset and factor volatilities using copulas. Copulas allow one to represent a multivariate distribution in a very flexible way by decoupling the choice of marginal distributions — which can be different from each other — from the choice of the dependence structure. Copulas have been used in multivariate GARCH-type models, but to our knowledge, this is the first study to introduce copulas in MSV models.

We introduce a new prior for correlation matrices, which we use in the context of Gaussian copulas. It is based on a geometric interpretation of correlation coefficients. The prior is a first step towards a model for time varying correlations where assets are exchangeable, avoiding a problem with models based on the Cholesky decomposition – their predictions are not invariant to the arbitrary choice of how to order assets.

We allow heavy-tailed returns. In our applications, we use Student's t marginals, but this is not an essential choice, and we don't rely on data augmentation to obtain conditional Gaussianity, unlike with many models using Student's t dis-

tributions. In general, we allow the marginal distribution to vary by asset, which in our applications translates to asset-specific degrees of freedom parameters. We also depart from the usual assumption of Gaussian factors and allow Student's t factors.

Different MCMC methods have been proposed for inference in MSV models and sometimes they are quite model specific. The estimation technique proposed by Chib, Nardari, and Shephard (2006) (CNS) is one of the most popular, especially when analyzing a large number of asset returns. The CNS model includes factors in mean, heavy tailed errors for returns, and jumps. Factor volatilities and the volatilities of the idiosyncratic components of returns are conditionally independent given parameters. Factors are Gaussian.

An important feature of the CNS procedure is sampling the factor loading matrix and the latent factors as a single block. This is more numerically efficient than using separate blocks to draw factor loadings and factors. The procedure exploits the conditional independence of volatilities to draw all volatilities and some associated parameters as a single block, using the procedure proposed by Kim, Shephard, and Chib (1998) (KSC) for univariate SV models.

The procedure in Kim, Shephard, and Chib (1998) is an example of the auxiliary mixture approach to inference in state space models, whereby non-linear or non-Gaussian state space models are first transformed into linear models and then the distribution of the transformed error is approximated by a mixture of Gaussian distributions. The mixture can be dealt with using data augmentation — adding mixture component indicators yields a linear Gaussian model when one conditions on them. The transformation is model specific, but many other models have yielded to this approach. Some relevant articles are Chib, Nardari, and Shephard (2002) and Omori, Chib, Shephard, and Nakajima (2007) for other univariate SV models ; Stroud, Müller, and Polson (2003) for Gaussian, but non-linear, state space models with state dependant variances ; Frühwirth-Schnatter and Wagner (2006) for state space models with Poisson counts ; and Frühwirth-Schnatter and Frühwirth (2007) for logit and multinomial logit models.

The approximation of the transformed error distribution as a mixture can be corrected by reweighting, as in Kim, Shephard, and Chib (1998) or by an additional Metropolis accept-reject, as in Stroud, Müller, and Polson (2003), Frühwirth-Schnatter and Wagner (2006) and Frühwirth-Schnatter and Frühwirth (2007).

CNS draw log volatilities, component indicators and some parameters based on the approximate transformed model. These Metropolis-Hastings updates preserve an approximate posterior distribution implied by the approximate model. All other updates of unknown quantities preserve the exact posterior distribution. Thus the stationary distribution of a sweep through all the blocks is neither the exact nor the approximate posterior distribution. We cannot expect the method to be simulation consistent.

McCausland (2012) proposed an alternative procedure to draw all latent states in univariate state space models as a block, preserving their exact conditional posterior distribution. This HESSIAN method is fast and numerically efficient and does not require data augmentation. It can be used to draw joint blocks of states and parameters. It is based on a non-Gaussian proposal distribution that captures some of the departure from Gaussianity of the conditional posterior distribution of the states. The HESSIAN method uses routines to compute derivatives of the log measurement density at a point, but is not otherwise model specific.

While the HESSIAN method is only for univariate states, we can apply it to draw volatilities as a single block in the time dimension but one-at-a-time in the cross-section dimension. We will see that the conditional distribution of one state sequence, given the others, parameters and data, can be seen as the conditional posterior distribution of states in a univariate state space model. So, following McCausland (2012), we obtain very close approximations to these conditional posterior distributions, which we use as proposal distributions. We are also able to draw a single volatility sequence, together with some of its associated parameters, as a single block. Because of strong dependence between volatilities and these parameters, the result is higher numerical efficiency.

To apply the HESSIAN method in this way, we require only that the multiva-

riate state sequence be a Gaussian first-order vector autoregressive process and that the conditional distribution of the observed vector depend only on the contemporaneous state vector. This requirement is satisfied for a wide variety of state space models, including but not limited to multivariate stochastic volatility models.

In Section 2.3, we describe in detail our methods for posterior simulation. In Section 2.4, we validate the correctness of our proposed algorithm using a test of program correctness similar to that proposed by Geweke (2004). In Section 2.5, we present an empirical application using a data set of daily returns of foreign exchange rates and compare the results of different specifications of the MSV model with the results for univariate SV models. Finally, in Section 2.6, we conclude and outline some possible extensions.

2.2 The Model

This section describes the most general discrete-time MSV model considered in this paper, and identifies some special cases of interest. We compare it to other specifications in the literature. We also describe prior distributions used in our empirical applications. Table 2.1 describes all of the model's variables. The notation is similar to that in Chib, Nardari, and Shephard (2006).

There are p observed return sequences, q factors and $m = p + q$ latent log volatility states. The conditional distribution of the factor vector $f_t = (f_{1t}, \ldots, f_{qt})$ and the return vector $r_t = (r_{1t}, \ldots, r_{pt})$ given the contemporaneous state vector α_t is given by

$$r_t = Bf_t + V_t^{1/2}\epsilon_{1t}, \quad f_t = D_t^{1/2}\epsilon_{2t},$$

or alternatively

$$y_t = \begin{bmatrix} r_t \\ f_t \end{bmatrix} = \begin{bmatrix} V_t^{1/2} & BD_t^{1/2} \\ 0 & D_t^{1/2} \end{bmatrix} \epsilon_t, \tag{2.1}$$

where B is a $p \times q$ factor loading matrix, $V_t = \text{diag}(\exp(\alpha_{1t}), \ldots, \exp(\alpha_{pt}))$ and $D_t = \text{diag}(\exp(\alpha_{p+1,t}), \ldots, \exp(\alpha_{p+q,t}))$ are matrices of idiosyncratic and factor volatilities, and $\epsilon_t = (\epsilon_{1t}^\top, \epsilon_{2t}^\top)^\top$ is an vector of innovations, specified below, in terms

of parameters ν and R.

Given parameters $\bar{\alpha}$, A and Σ, the state is a Gaussian first order vector autoregression, given by

$$\alpha_1 \sim N(\bar{\alpha}, \Sigma_0), \qquad \alpha_{t+1}|\alpha_t \sim N((I - A)\bar{\alpha} + A\alpha_t, \Sigma), \tag{2.2}$$

where the derived parameter Σ_0 is chosen to make the state sequence stationary :

$$\text{vec } \Sigma_0 = (I_{m^2} - A \otimes A)^{-1}\text{vec } \Sigma.$$

See Hamilton (1994, p.265) for details on computing the marginal variance Σ_0.

We assume the conditional independence relationships implied by the following joint density decomposition :

$$\begin{aligned}\pi(\bar{\alpha}, A, \Sigma, \nu, B, R, \alpha, f, r) = \pi(\bar{\alpha}, A, \Sigma, \nu, B)\pi(R) \\ \cdot \pi(\alpha_1|\bar{\alpha}, A, \Sigma) \prod_{t=1}^{n-1} \pi(\alpha_{t+1}|\alpha_t, \bar{\alpha}, A, \Sigma) \\ \cdot \prod_{t=1}^{n} \left[\pi(f_t|\alpha_t)\pi(r_t|B, R, f_t, \alpha_t)\right].\end{aligned}$$

We specify the distribution of $\epsilon_t = (\epsilon_{1t}, \ldots, \epsilon_{mt})$ by providing marginal distributions, which may differ, and a copula function describing dependence. See Patton (2009) for an overview of the application of copulas in the modelling of financial time series and Kolev, dos Anjos, and de M. Mendez (2006) for a survey and contributions to copula theory.

The marginal distribution of ϵ_{it} is given by the cumulative distribution function (cdf) $F_\epsilon(\epsilon_{it}|\theta_i)$. Let $\pi(\epsilon_{it}|\theta_i)$ be its density function. Sklar (1959) provides a theorem on the relationship between marginal distributions, joint distributions and a copula function. It states that if $F(\epsilon_1, \ldots, \epsilon_m)$ is an m-dimensional cdf with marginals $F_1(\epsilon_1), \ldots, F_m(\epsilon_m)$, then there exists a unique copula function C such that F that

TABLE 2.1 – Table of symbols

Symbol	dimensions	description
$\bar{\alpha}$	$m \times 1$	mean of state α_t
A	$m \times m$	coefficient matrix for α_t
Σ	$m \times m$	variance of state innovation
B	$p \times q$	factor loading matrix
ν	$m \times 1$	vector of degrees of freedom parameters
R	$m \times m$	Gaussian copula parameter
ϵ_t	$m \times 1$	period t return/factor innovation
α_t	$m \times 1$	period t state
r_t	$p \times 1$	period t return vector
f_t	$q \times 1$	period t factor
y_t	$m \times 1$	$(r_t^\top, f_t^\top)^\top$

can be written as :

$$F(\epsilon_1, \ldots, \epsilon_m) = C(F_1(\epsilon_1), \ldots, F_m(\epsilon_m)).$$

A copula function is a cdf on $[0,1]^m$ with marginal distributions that are uniform on $[0,1]$. Conversely, if $\epsilon = (\epsilon_1, \ldots, \epsilon_m)$ is a random vector with cdf F and continuous marginal cdfs F_i, $i = 1, \ldots, m$, then the copula of ϵ, denoted C, is the cdf of $(u_1, \ldots, u_m)$, where u_i is the probability integral transform of $\epsilon_i : u_i = F_i(\epsilon_i)$. The distribution of the u_i is uniform on $[0,1]$. Thus

$$C(u_1, \ldots, u_m) = F(F_1^{-1}(u_1), \ldots, F_m^{-1}(u_m)).$$

In this paper, we assume Student's t marginals with asset-specific degrees of freedom. This allows for fat tails. However, the Student-t cdf could be replaced by another one and most of the derivations presented below would still hold. We choose a Gaussian copula with variance matrix

$$R = \begin{bmatrix} R_{11} & 0 \\ 0 & I_q \end{bmatrix},$$

where R_{11}, and thus R, are correlation matrices. One could replace the Gaussian copula with another, and the derivations below could be modified accordingly. However, there would be a computational cost. We take advantage of the fact that the derivatives of a log Gaussian density are non-zero only up to second order.

We denote the Gaussian copula with correlation matrix R as C_R :

$$C_R(u_1, \ldots, u_m) = \Phi_R(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_m)).$$

Here Φ denotes the standard univariate Gaussian cdf and ϕ, its density. Φ_R and ϕ_R denote the cdf and density of the m-variate Gaussian distribution with mean zero and covariance R. Then the multivariate density of ϵ_t is the product of the Gaussian copula density and the Student-t marginal density functions :

$$\pi_\epsilon(\epsilon_t|\theta) = c_R(F_\epsilon(\epsilon_{1t}|\theta_1), \ldots, F_\epsilon(\epsilon_{mt}|\theta_m)) \prod_{i=1}^{m} \pi(\epsilon_{it}|\theta_i), \tag{2.3}$$

where

$$c_R(u_1, \ldots, u_m) = \frac{\partial^{(m)} C_R(u_1, \ldots, u_m)}{\partial u_1 \cdots \partial u_m} = \frac{\phi_R(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_m))}{\prod_{i=1}^{m} \phi(\Phi^{-1}(u_i))}.$$

Letting $x_i \equiv \Phi^{-1}(u_i)$, $i = 1, \ldots, m$ and $x \equiv (x_1, \ldots, x_m)$, we can write

$$\log c_R(u_1, \ldots, u_m) = -\frac{1}{2}(\log|R| + \log(2\pi) + x^\top (R^{-1} - I))x. \tag{2.4}$$

We use the notation π_ϵ here instead of the generic π to clarify that it is the density function of ϵ_t. We can now write the conditional density of y_t given α_t, B, ν and R as

$$\pi(y_t|\alpha_t, B, \nu, R) = \pi_\epsilon \left(\begin{bmatrix} V_t^{-1/2}(r_t - Bf_t) \\ D_t^{-1/2} f_t \end{bmatrix} \middle| \nu, R \right) \prod_{i=1}^{m} \exp(-\alpha_{it}/2). \tag{2.5}$$

2.2.1 Alternative MSV models

As mentioned before, different MSV model specifications reflect, to a large extent, different restrictions chosen by the modeller to balance flexibility and parsimony. In our model, we can impose restrictions on the parameters governing the marginal distribution of volatility in (2.2), the parameters governing the conditional distribution of returns and factors given volatility, equation (2.1), or both. These choices have different implications for the stylized facts that a MSV model can capture.

First let us consider restrictions on the marginal distribution of volatilities. At one extreme, giving the most flexibility for volatility dynamics, we can specify A and Σ in equation (2) as full matrices. At another extreme, we can impose prior independence among log volatilities by specifying diagonal matrices for A and Σ. This can be much less computationally demanding, which makes it especially attractive when the number of volatilities to estimate is large. Several intermediate possibilities are possible, including the relatively parsimonious specification in Section 3.2.3, where A and Σ are not diagonal, but have $O(m)$ free elements.

We now consider cross-sectional dependence arising from the conditional distribution of returns given parameters and volatilities, marginal of latent factors. For comparison purposes, it will be helpful to write out the conditional variance of returns given returns and factor volatilities :

$$\mathrm{Var}[r_t|\alpha_t] = V_t^{1/2} R_{11} V_t^{1/2} + B D_t B^\top. \tag{2.6}$$

In the case where we have no factors, $q = 0$, then the second term disappears. The conditional variance varies in time, but the conditional correlation R_{11} is constant. Models with constant correlations have been studied by Harvey, Ruiz, and Shephard (1994), Danielsson (1998), Smith and Pitts (2006) and So, Li, and Lam. (1997). Other authors, including Yu and Meyer (2006), Philipov and Glickman (2006), Gourieroux (2006), Gourieroux, Jasiak, and Sufana (2004), Carvalho and West (2006) and Asai and McAleer (2009), consider models in which the re-

turn innovation correlation is time-varying, which is more realistic. However, as the number of assets increases, the estimation of a separate time varying correlation matrix becomes very challenging. Furthermore, when the dynamics of correlation and volatility are modelled separately, it is difficult to capture the empirical regularity that correlation and volatility covary.

Introducing latent factors in mean is another way to introduce time-varying correlations. Factors in mean models exploit the idea that co-movements of asset returns are driven by a small number of common underlying variables, called factors. The factors are typically modelled as univariate SV processes. Usually, factor MSV models give R_{11} as the identity matrix, in which case $\text{Var}(r_t|\alpha_t) = V_t + BD_tB^\top$. The main attractions of mean factor models is that they are parsimonious, they lead to time varying conditional correlations and they have a natural link with the arbitrage pricing theory (APT), an influential theory of asset pricing. APT holds that the expected return of a financial asset can be modelled as a linear function of various factors. In addition, the mean factor structure allows the conditional correlations and conditional variances to covary. This is an important feature for portfolio analysis, especially when there are turbulent periods. See Longin and Solnik (2001) and Ang and Chen (2002) for empirical studies showing the positive correlation of the conditional variances and conditional correlations. Given all these characteristics, factor MSV models have become very popular in the literature, and different versions have been proposed. The basic model assumed normal returns, a constant factor loading matrix and zero factor mean. See, for example, Jacquier, Polson, and Rossi (1995), Pitt and Shephard (1999) and Aguilar and West (2000). Other studies proposed some extensions to the basic structure such as jumps in the return equation and heavy-tailed returns (Chib, Nardari, and Shephard (2006)), time varying factor loading matrices and regime-switching factors (Lopes and Carvalho (2007)) or first-order autoregressive factors (Han (2006)). See Chib, Omori, and Asai (2009) for a brief description and comparison of the different types of MSV models mentioned. Allowing for heavy tails in the distributions of returns is desirable because empirical evidence has shown that returns present higher conditional

kurtosis than a Gaussian distribution does.

If we compare these models to the one described at the beginning of this section, we notice that the MSV model specification that we work with is fairly general and incorporates some other specifications as special cases. In its most general version, without parameter restrictions, the model allows for cross-sectional volatility dependence. It allows time-varying conditional correlations through the specification of a mean factor structure. It also incorporates cross-sectional conditional return dependence through copulas. The conditional variance matrix of returns in equation (2.6) is time-varying. The conditional correlation matrix is also time varying, and covaries with the conditional variances.

We can impose some parameter restrictions and obtain some interesting special cases :

- Independent states in cross section : A and Σ diagonal.
- Conditionally independent returns given factors and states : R diagonal.
- No factors : $q = 0$. In this case, the conditional variance-covariance matrix of returns is given by $\text{Var}(r_t|\alpha_t) = V_t^{1/2} R_{11} V_t^{1/2}$ which is still time-varying but the conditional correlation matrix will be R_{11} which is constant.

2.2.2 Prior Distributions

Prior for $\bar{\alpha}$, A, Σ, ν, and B

We now describe a prior for a low dimensional specification of $\bar{\alpha}$, A, Σ, ν, and B.

We parameterize A and Σ in the following parsimonious way :

$$\Sigma = (\text{diag}(\sigma))^2 + \begin{bmatrix} \beta\beta^T & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \text{diag}(\lambda) + \begin{bmatrix} (1/p)\delta\iota_p^\top & 0 \\ 0 & 0 \end{bmatrix}.$$

where σ and λ are $m \times 1$ vectors, β and δ are $p \times 1$ vectors and ι_p is the $p \times 1$ vector of ones.

We organize the parameters associated with each series i (a return for $i =$

$1, \ldots, p$ or a factor for $i = p+1, \ldots, m$) as

$$\theta_i = \begin{cases} (\bar{\alpha}_i, \tanh^{-1}(\lambda_i), \tanh^{-1}(\lambda_i + \delta_i), \log \sigma_i, \beta_i/\sigma_i, \log \nu_i, B_{i1}, \ldots, B_{iq})^\top, & 1 \leq i \leq p, \\ (\tanh^{-1}(\lambda_i), \log \sigma_i, \log \nu_i)^\top, & p+1 \leq i \leq m, \end{cases}$$

and organize the vector of all these parameters as $\theta = (\theta_1^\top, \ldots, \theta_m^\top)^\top$.

We suppose that the θ_i are a priori independent, multivariate normal, and that the parameters have the prior means and variances given in Table 2.2. For each

TABLE 2.2 – Parameter means and variances of prior distributions.

Parameter	mean	variance
$\bar{\alpha}_i$	-11.0	$2^2 = 4$
$\tanh^{-1}(\lambda_i)$	2.1	$(0.25)^2 = 0.0625$
$\tanh^{-1}(\lambda_i + \delta_i)$	2.3	$(0.25)^2 = 0.0625$
$\log \sigma_i$	-2.0	$(0.5)^2 = 0.25$
β_i/σ_i	0.0	$(0.5)^2 = 0.25$
$\log \nu_i$	3.0	$(0.5)^2 = 0.25$

$i = 1, \ldots, m$, the correlation coefficient between σ_i and $\tanh^{-1}(\lambda_i)$ is -0.8. All other correlations are zero. The prior probability that the A matrix is such that α is not stationary is close enough to zero that we have not seen an example in prior simulations.

Prior for R

We can interpret the correlations in the $p \times p$ correlation matrix R as the cosines of angles between vectors in $\mathbb{R}^l$, where $l \geq p$. There are p vectors, one for each asset, and the $\binom{p}{2}$ angles between distinct vectors give the various correlations.

We reparameterize the information in R. The new parameter is an $p \times l$ matrix V whose rank is p and whose rows have unit Euclidean length. The rows of V give p points on the surface of the unit l-dimensional hypersphere centred at the origin. In putting a prior on V, we induce a prior on $R = VV^\top$. It is easy to see that $VV^\top$ is a $p \times p$ symmetric positive definite matrix with unit diagonal elements. In

other words, it is a full rank correlation matrix. Conversely, for any full correlation matrix R and any $l \geq p$, there is an $p \times l$ real matrix V with rows of unit length and rank p such that $VV^\top = R$: take the Cholesky decomposition $R = LL^\top$ and let $V = [L \quad 0_{p,l-p}]$.

We choose a prior such that the rows v_i of V are independent and identically distributed. This ensures that the prior does not depend on how the assets are ordered. We could relax this to exchangeable v_i and retain this advantage. This kind of invariance is difficult to achieve if one specifies a prior on the Cholesky decomposition of the correlation matrix. A disadvantage of the V parameterization is that the number of non-zero elements of V is lp, while the number of non-zero elements of the Cholesky factor is $p(p+1)/2$. Another issue is that V is not identified. However, since $VV^\top$ is identified, this is not a problem.

We will call the vector $(1, 0, ..., 0)$ in R^l the north pole of the hypersphere. Let $\zeta_i \equiv \cos^{-1}(V_{i1})$, the angle between v_i and the north pole. We specify a marginal density $\pi(\zeta_i)$ and let the conditional distribution $v_i|\zeta_i$ be uniform on the set of points on the surface of the unit hyperphere at an angle of ζ_i from the north pole. This set is the surface of an $(l-1)$ dimensional hypersphere of radius $\sin\zeta_i$.

This gives the following density for v_i on the unit l-dimensional hypersphere :

$$\pi(v_i) = \pi(\zeta_i) 2 \frac{\pi^{(l-1)/2}}{\Gamma(\frac{l-1}{2})} \sin^{l-2}\zeta_i.$$

In our applications, we use $\zeta_i/\pi \sim \text{Be}(4,4)$.

2.3 Posterior inference using MCMC

We use MCMC methods to simulate the posterior distribution, with density $\pi(\bar{\alpha}, A, \Sigma, \nu, B, R, \alpha, f|r)$. We use a multi-block Gibbs sampler. The result is an ergodic chain whose stationary distribution is the target distribution. The sequence of steps in a single sweep through the blocks is

1. For $i = 1, \ldots, m$, update (θ_i, α_i) as described in 2.3.1, preserving the conditio-

nal posterior distribution $\theta_i, \alpha_i|\theta_{-i}, \alpha_{-i}, R, B_{-i}, f, r$, where α_{-i} is the vector of all state sequences except the i'th and θ_{-i} is the vector of all parameter values in θ except those in θ_i.

2. Update (B, f) as described in 2.3.2, preserving the conditional distribution $B, f|\theta, \alpha, R, r$.
3. Update f as described in 2.3.3, preserving the conditional distribution $f|\theta, \alpha, R, B, r$.
4. Update R as described in 2.3.4, preserving the conditional distribution $R|\theta, \alpha, B, f, r$.

In the following subsections, we describe each of these steps.

2.3.1 Draw of θ_i, α_i

We draw (θ_i, α_i) as a single Metropolis-Hastings block. Drawing a volatility sequence together with its associated parameters in one block is more efficient than drawing them separately because of their posterior dependence.

Our proposal of (θ_i, α_i) consists of a random walk proposal of θ_i^* followed by a (conditional) independence proposal of α_i^* given θ_i^*. This gives a joint proposal that we accept or reject as a unit. The acceptance probability is given by

$$\min\left(1, \frac{\pi(\theta_i^*)\pi(\alpha_i^*|\theta_i^*, \theta_{-i}, \alpha_{-i})\pi(y_t|\alpha_i^*, \alpha_{-i}, \theta_i^*, \theta_{-i}, R)}{\pi(\theta_i)\pi(\alpha_i|\theta, \alpha_{-i})\pi(y_t|\alpha, \theta, R)} \cdot \frac{g(\alpha_i^*|\theta_i^*, \theta_{-i}, \alpha_{-i}, R)}{g(\alpha_i|\theta, \alpha_{-i}, R)}\right),$$

where $g(\alpha_i^*|\theta_i^*, \theta_{-i}, R)$ is an independence (it does not depend on α_i) conditional proposal density for α_i^* given θ_i^*.

A key issue for independence proposals is the specification of the proposal density. To obtain high numerical efficiency for the draw of a vector with thousands of observations, we need an extremely close approximation. We will see that the conditional posterior distribution of α_i has the form of the target distributions approximated in McCausland (2012). These approximations are very close, and we will exploit them here.

Draw of $\theta_i^*|\theta_i, \alpha_{-i}^*, \omega$

We use a random walk Metropolis proposal for θ_i^*. The random walk $(\theta_i^* - \theta_i)$ is Gaussian with mean zero and covariance matrix Ξ. We obtain Ξ using an adaptive random walk Metropolis algorithm, described in Vihola (2011), during a burn-in period — the random walk proposal variance is adjusted after each draw to track a target acceptance probability. We use the final value of Ξ at the end of the burn-in period as the proposal covariance matrix for all future draws. Thus our posterior simulator is a true Markov chain after the burn-in period and so standard MCMC theory applies to the retained posterior sample.

Draw of $\alpha_i^*|\theta_i^*, \omega$

We now discuss the draw of the conditional proposal $\alpha_i^*|\theta_i^*, \theta_{-i}, \alpha_{-i}, R$ using the HESSIAN method in McCausland (2012).

The HESSIAN method is for simulation smoothing in state space models with univariate Gaussian states and observable vectors that are not necessarily Gaussian. It involves a direct independence Metropolis-Hastings update of the entire sequence of states as a single block. The proposal is a much closer approximation of the target distribution than is any multivariate Gaussian approximation. The result is a Metropolis-Hastings update that is not only tractable, but very numerically efficient. One can also update states jointly with parameters by constructing joint proposal distributions, as we do here.

Drawing states as a block is much more efficient than one-at-a-time draws in the usual case where the posterior autocorrelation of states is high. Adding parameters to the block leads to even higher numerical efficiency when there is strong posterior dependence between parameters and states. The HESSIAN method does not require data augmentation or model transformations, unlike auxiliary mixture sampling methods, where the model is transformed and augmented so that conditioning on auxiliary variables yields a linear Gaussian state space model. The auxiliary mixture approach has been used for univariate state space models by Omori, Chib, Shephard, and Nakajima (2007) and Kim, Shephard, and Chib (1998)

. Approximating distributions of the transformed model by mixtures of Gaussian random variables results in slightly incorrect posterior draws. In some cases, this is corrected using reweighting or an additional accept/reject step. We have seen that in Chib, Nardari, and Shephard (2006), some blocks update the true posterior and some blocks update the approximate (mixture approximation) posterior. The stationary distribution is neither the approximate distribution nor the true distribution, and it is not clear to us how one could compensate for the error. Draws from the HESSIAN approximate distribution are exact, in the sense that draws of α_i^* are consistent with the evaluation of the proposal density used to compute the Metropolis-Hastings acceptance probability.

The HESSIAN method uses an approximation $g(\alpha|y)$ of $\pi(\alpha|y)$ for univariate models in which $\alpha \sim N(\bar{\Omega}^{-1}\bar{c}, \bar{\Omega})$, with $\bar{\Omega}$ tridiagonal and $\pi(y|\alpha) = \prod_{t=1}^{n} \pi(y_t|\alpha_t)$. One needs to specify $\bar{\Omega}$, the precision, and $\bar{c}$, the co-vector, and provide routines to compute the first five derivatives of $\log \pi(y_t|\alpha_t)$. The approximation $g(\alpha|y)$ is so close to $\pi(\alpha|y)$ that we can use it as a proposal distribution to update the entire sequence $\alpha = (\alpha_1, \ldots, \alpha_n)$ as a block.

Here states are multivariate, but we can draw state sequences one at a time in the cross-sectional dimension, using approximations of the conditional distribution of each state sequence α_i given the rest of the states (α_{-i}), parameters and data. The conditional density we need to approximate is

$$\pi(\alpha_i|\alpha_{-i}, y) \propto \pi(\alpha_i|\alpha_{-i}) \prod_{t=1}^{n} \pi(y_t|\alpha_t).$$

In Appendix 2.7.1, we show that $\alpha_i|\alpha_{-i} \sim N((\bar{\Omega}^{(i)})^{-1}\bar{c}^{(i)}, \bar{\Omega}^{(i)})$, where the co-vector $\bar{c}^{(i)}$ is a $n \times 1$ vector and the precision $\bar{\Omega}^{(i)}$ is a tridiagonal $n \times n$ matrix, as required by the HESSIAN method. We also describe there how to compute the elements of $\bar{\Omega}^{(i)}$ and $\bar{c}^{(i)}$ in terms of the elements of $\bar{\Omega}$ and $\bar{c}$.

We just need to compute five derivatives of $\log \pi(y_t|\alpha_{it}, \alpha_{-i,t})$ with respect to α_{it}. We do not need to write down the complete analytical expressions of these derivatives, we just need to evaluate them at a point. To do this, we use automatic

routines to combine derivatives of primitive functions according to Faa di Bruno's rule, which is a generalization of the chain rule to higher derivatives. It allows us to take two vectors of derivative values and call a function that returns a vector of derivatives of a composite function. Appendix 2.7.2 describes the Faa di Bruno formula and how we use it to evaluate five derivatives of $\log \pi(y_t|\alpha_{it}, \alpha_{-i,t})$.

2.3.2 Draw of (B, f)

In this block, we update B and f simultaneously in a way that preserves the posterior distribution of B and f given everything else but does not change the value of the matrix-vector products Bf_t. Adding this block improves the posterior mixing of the poorly identified scale of the B matrix. At the same time, it is fairly cheap computationally, because the Bf_t do not change.

We first draw a random $q \times q$ matrix Λ. The diagonal elements are iid, with $n\Lambda_{ii} \sim \chi^2(n)$, and the non-diagonal elements are zero. With probability 1/2, we form proposals $B^* = B\Lambda$, $f_t^* = \Lambda^{-1} f_t$, $t = 1, \ldots, n$ and with complementary probability, we form $B^* = B\Lambda^{-1}$, $f_t^* = \Lambda f_t$, $t = 1, \ldots, n$. In the first case, we accept with probability

$$\min\left(1, |\Lambda|^{-(n-p)} \frac{\pi(B^*) \prod_{t=1}^n \pi(f_t^*|\alpha, \nu)}{\pi(B) \prod_{t=1}^n \pi(f_t|\alpha, \nu)}\right),$$

and in the second case, we accept with probability

$$\min\left(1, |\Lambda|^{(n-p)} \frac{\pi(B^*) \prod_{t=1}^n \pi(f_t^*|\alpha, \nu)}{\pi(B) \prod_{t=1}^n \pi(f_t|\alpha, \nu)}\right).$$

The factors $|\Lambda|^{-(n-p)}$ and $|\Lambda|^{(n-p)}$ are products of the Jacobian matrices for the multiplicative transformations of n vectors f_t and p rows of B.

2.3.3 Draw of f

We draw each f_t from its conditional posterior distribution using a random walk proposal. Because the random walk involves only two function evaluations, it is quite cheap computationally. We use a proposal variance matrix $(2.38)^2(B^\top V_t^{-1}B + D_t^{-1})^{-1}$. The matrix $(B^\top V_t^{-1}B + D_t^{-1})^{-1}$ is a crude but cheap approximation of the conditional posterior variance of f_t, obtained by setting $\nu_i = \infty$, $i = 1, \ldots, m$, and $R = I$. The scaling factor $(2.38)^2$ comes from Gelman, Roberts, and Gilks (1996), and it is optimal when the target distribution is univariate Gaussian.

2.3.4 Draw of R

We draw the rows of V one-at-a-time. We use a random walk M-H proposal to update the row vector v_i. It is a random walk on the l-dimensional unit hypersphere : the direction of the walk is uniform and the angle of the walk has some arbitrary distribution. Let d be the direction vector, normalized so that it has unit length. To draw the proposal v_i^* :

1. Draw the angle ζ_i between the proposal v_i^* and the current state. We use $\zeta_i/\pi \sim \text{Be}(1, 199)$.
2. Draw the direction d from the uniform distribution on the unit l-dimensional hypersphere [6].
3. Compute $d_\perp$, the projection of d onto the hyperplane perpendicular to v_i :

$$d_\perp = d - \frac{v_i d}{||v_i||^2} v_i$$

4. Compute :

$$v_i^* = \cos\zeta_i \cdot v_i + \sin\zeta_i \cdot \frac{d_\perp}{||d_\perp||}$$

6. We can draw from a uniform distribution on a unit hypersphere by drawing a spherically symmetric normal random vector of the same dimension, and dividing it by its length.

5. Accept with probability

$$\min\left(1, \frac{\pi(f, r|\alpha, \theta, B, R^*, f)\pi(\zeta_i^*)}{\pi(f, r|\alpha, \theta, B, R, f)\pi(\zeta_i)} \frac{\zeta_i}{\zeta_i^*}\right).$$

2.4 Getting it Right

Here we perform a computational experiment with artificial data to put the implementation of our methods to the test. We use a simulation strategy similar to that proposed by Geweke (2004) for testing the correctness of posterior simulators and detecting any analytical and coding errors there may be. This procedure replaces the common exercise of generating a single artificial data set using known values of the parameters, applying a simulation method to these data and verifying that the "true value" falls in a region of high posterior probability.

Like the approach of Geweke (2004), our approach is based on the simulation of the joint distribution of parameters, states, factors and data. We use a single simulator, a Gibbs sampler that alternates between updates of the posterior distribution, described in the previous section, and draws of returns given parameters, states and factors, described in Appendix 2.7.3. If the simulator works correctly, then the marginal distribution of the parameters must agree with the specified prior distributions. We can test a wide range of implications of this condition.

This formal approach is a more stringent way to verify the correctness of posterior simulators, as not all errors lead to obviously incorrect results. Reasonable but incorrect results are worse than obvious errors, because they can mislead. The test applied here can discriminate much more effectively between correct code and alternatives with minor coding errors. Also, simulation results often provide clues to the source of any errors.

Here in detail is how we generate a sample from the joint distribution of $\bar{\alpha}$, A, Σ, B, ν, R, α, f and r. The first draw $(\bar{\alpha}^{(1)}, A^{(1)}, \Sigma^{(1)}, B^{(1)}, \nu^{(1)}, R^{(1)}, \alpha^{(1)}, f^{(1)}, r^{(1)})$ comes directly from the model. See Appendix 2.7.3 for a description of how to draw from $\pi(r|\bar{\alpha}, A, \Sigma, B, \nu, R, \alpha, f)$. Then, we draw subsequent values by iterating the

following Gibbs blocks :

1. For $i = 1, \ldots, m$, update θ_i, α_i as described in Section 2.3.1.
2. For $t = 1, \ldots, n$, update f_t as described in Section 2.3.3.
3. Update B and $(f_1, \ldots, f_n)$ as described in Section 2.3.2.
4. Update R as described in Section 2.3.4.
5. Update r as described in Appendix 2.7.3.

We obtain a sample $\{\theta_i^{(j)}\}_{j=1}^J$ of size $J = 10^8$ for $i = 1, \ldots, m$. We construct, for $i = 1, \ldots, m$ and $j = 1, \ldots, J$ the vectors

$$z^{(i,j)} \equiv L_i^{-1}(\theta_i^{(j)} - \mu_i),$$

where μ_i is the prior mean and L_i is the lower Cholesky factor of the prior variance of θ_i. If the $\theta_i^{(j)}$ are truly multivariate Gaussian with variance $L_i L_i^\top$, the elements of $z^{(i,j)}$ are iid $N(0,1)$. The vectors $z^{(i,j)}$ have length $K_i = 6+q$ for $i = 1, \ldots, p$ and length $K_i = 3$ for $i = p+1, \ldots, m$. Since the $z^{(i,j)}$, $i = 1, \ldots, m$, are independent, we have $\sum_{i=1}^m z_i^\top z_i \sim \chi^2((6+q)p + 3q)$.

We construct the following sample frequencies for quantiles $Q = 0.1, 0.3, 0.5, 0.7, 0.9$, return and factor indices $i = 1, \ldots, m$, and parameter indices $k = 1, \ldots, K_i$

$$\hat{I}_{ik}^{(Q)} = \frac{1}{J} \sum_{j=1}^{J} 1\left(z_k^{(i,j)} \leq \Phi^{-1}(Q)\right),$$

as well as the sample frequencies

$$\hat{I}_{0k}^{(Q)} = \frac{1}{J} \sum_{j=1}^{J} 1\left(\sum_{i=1}^{m} (z^{(i,j)})^\top z^{(i,j)} \leq F^{-1}(Q)\right),$$

where F is the cdf of the χ^2 distribution with $(6+q)p + 3q$ degrees of freedom.

Standard results for laws of large numbers and central limit theorems for ergodic chains apply, so we should observe sample frequencies close to Q. Table 2.3

shows the sample frequencies $\hat{I}_{ik}^{(Q)}$ and their estimated numerical errors $s_{ik}^{(Q)}$, obtained using the method of batch means. We observe that for all cases, the sample frequencies are very similar to their respectively Q values. This fails to cast doubt on the correctness of the implementation of the proposed algorithm.

2.5 Empirical Results

In this section we apply our methods to historical exchange rate data. We describe the data and report estimation results for various models.

2.5.1 Data

We analyze daily returns of 10 currencies relative to the US dollar : the Swiss Franc (CHF), Euro (EUR), Australian Dollar (AUD), New Zealand Dollar (NZD), Mexican Peso (MXN), Brazil Real (BRL), British Pound (GBP), Canadian Dollar (CAD), Japanese Yen (JPY) and Singapore Dollar (SGD). The exchange rates are the noon spot rate obtained from the Federal Reserve Bank of New York. The sample covers the period from January 5, 1999 to December 31, 2008. We compute the log returns of the exchange rates and remove returns for those days when one or more of the markets was closed, giving 2503 observations for each return series.

Table 2.4 presents some descriptive statistics : annualized mean, annualized standard deviation, skewness and excess kurtosis. All series present excess kurtosis, but the magnitude varies from one currency to another, from around 2 for the Euro to about 27 for the Mexican Peso. Sample volatility varies a lot across currencies, with the Brazilian Real, and the Australian and New Zealand Dollars being the most volatile currencies. Although the sample statistics differ substantially across currencies, we can also observe some commonalities in Figure 2.1. This shows time plots of the 10 return series and we notice that all returns exhibit their most volatile episodes at the end of the sample, which corresponds to the financial crisis of 2008.

In Table 2.5 we show the sample correlation matrix for the entire period. Correlation coefficients vary from -0.9 to 0.8. The strongest negative correlation is for the

TABLE 2.3 – "Getting it right" sample quantiles

i	k	$\hat{I}^{(0.1)}_{i,k}$	$s^{(0.1)}_{i,k}$	$\hat{I}^{(0.3)}_{i,k}$	$s^{(0.3)}_{i,k}$	$\hat{I}^{(0.5)}_{i,k}$	$s^{(0.5)}_{i,k}$	$\hat{I}^{(0.7)}_{i,k}$	$s^{(0.7}_{i,k}$	$\hat{I}^{(0.9)}_{i,k}$	$s^{(0.9)}_{i,k}$
1	1	0.1004	0.00068	0.3003	0.00084	0.5001	0.00077	0.6997	0.00063	0.8997	0.00036
1	2	0.0997	0.00018	0.2994	0.00030	0.4994	0.00034	0.6996	0.00029	0.8998	0.00018
1	3	0.1000	0.00015	0.3000	0.00025	0.4998	0.00026	0.6997	0.00024	0.8996	0.00017
1	4	0.0997	0.00016	0.2996	0.00027	0.4998	0.00028	0.6998	0.00026	0.8998	0.00016
1	5	0.1001	0.00016	0.3002	0.00027	0.5001	0.00027	0.7000	0.00027	0.9001	0.00018
1	6	0.1002	0.00015	0.3001	0.00024	0.5001	0.00027	0.7003	0.00028	0.9005	0.00017
1	7	0.0993	0.00052	0.2993	0.00098	0.4998	0.00110	0.7000	0.00090	0.9002	0.00043
1	8	0.1002	0.00050	0.3008	0.00096	0.5010	0.00110	0.7003	0.00092	0.9004	0.00048
2	1	0.1008	0.00089	0.3005	0.00100	0.5007	0.00098	0.7004	0.00078	0.9002	0.00041
2	2	0.0999	0.00018	0.3003	0.00034	0.5001	0.00033	0.7001	0.00031	0.9003	0.00016
2	3	0.0999	0.00016	0.2996	0.00032	0.4995	0.00037	0.6999	0.00033	0.9000	0.00020
2	4	0.1000	0.00019	0.3001	0.00034	0.5005	0.00029	0.7004	0.00025	0.9003	0.00014
2	5	0.0999	0.00016	0.2996	0.00033	0.4999	0.00038	0.7002	0.00029	0.8999	0.00017
2	6	0.1001	0.00017	0.3002	0.00033	0.5001	0.00033	0.7002	0.00024	0.9002	0.00015
2	7	0.1001	0.00043	0.3001	0.00085	0.4997	0.00106	0.6997	0.00084	0.9001	0.00045
2	8	0.0990	0.00043	0.2980	0.00088	0.4974	0.00116	0.6983	0.00091	0.8994	0.00050
3	1	0.0999	0.00009	0.3000	0.00015	0.4999	0.00016	0.6999	0.00013	0.8999	0.00009
3	2	0.1000	0.00010	0.3000	0.00015	0.4999	0.00017	0.7000	0.00015	0.9001	0.00011
3	3	0.1000	0.00009	0.3000	0.00016	0.5000	0.00018	0.6999	0.00016	0.8999	0.00009
4	1	0.1000	0.00008	0.2999	0.00015	0.4998	0.00015	0.6998	0.00015	0.8999	0.00009
4	2	0.1000	0.00010	0.3001	0.00013	0.4999	0.00015	0.7000	0.00013	0.9001	0.00009
4	3	0.0998	0.00009	0.2999	0.00014	0.5000	0.00015	0.7001	0.00015	0.9000	0.00009

Table 2.4 – Descriptive statistics of data

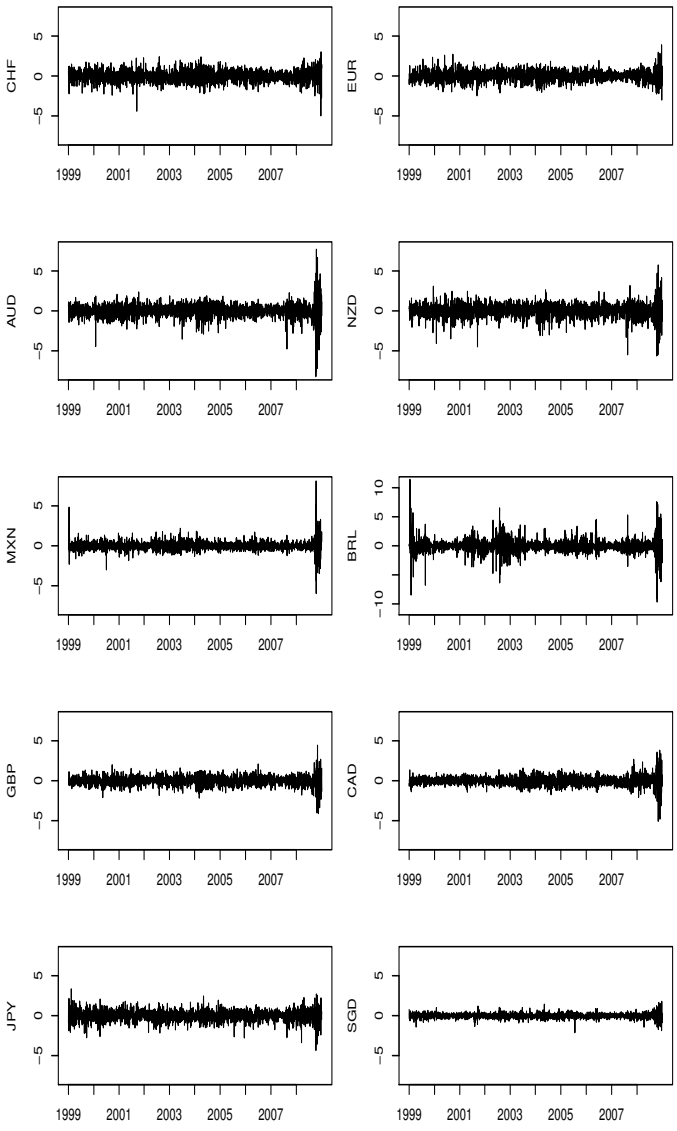
	Mean	Std. Dev.	Skewness	Excess Kurtosis
CHF	-2.56	10.81	-0.30	2.45
EUR	1.45	9.98	0.10	1.90
AUD	1.27	13.45	-0.88	16.66
NZD	0.83	13.40	-0.60	5.75
MXN	3.49	9.48	1.31	26.91
BRL	6.67	19.33	0.45	14.13
GBP	-1.19	8.97	-0.29	5.04
CAD	-2.22	8.80	-0.20	9.36
JPY	-2.05	10.35	-0.36	2.70
SGD	-1.44	4.75	-0.19	4.44

pair (EUR,CHF) and the strongest positive correlation is for the pair (AUD,NZD). The MXN and BRL are the least correlated with the rest of currencies.

Table 2.5 – Sample daily correlation

	CHF	EUR	AUD	NZD	MXN	BRL	GBP	CAD	JPY	SGD
CHF	1.00	-0.90	-0.37	-0.37	-0.05	0.01	-0.61	0.31	0.38	0.42
EUR	-0.92	1.00	0.52	0.50	-0.07	-0.12	0.70	-0.41	-0.27	-0.47
AUD	-0.37	0.52	1.00	0.82	-0.36	-0.32	0.50	-0.57	-0.02	-0.45
NZD	-0.37	0.50	0.82	1.00	-0.27	-0.25	0.49	-0.49	-0.03	-0.43
MXN	-0.05	-0.07	-0.36	-0.27	1.00	0.48	-0.15	0.30	-0.16	0.16
BRL	0.01	-0.12	-0.32	-0.25	0.48	1.00	-0.16	0.24	-0.09	0.21
GBP	-0.61	0.69	0.50	0.49	-0.15	-0.16	1.00	-0.39	-0.16	-0.40
CAD	0.31	-0.41	-0.57	-0.49	0.30	0.24	-0.39	1.00	0.01	0.35
JPY	0.38	-0.27	-0.02	-0.03	-0.16	-0.09	-0.16	0.01	1.00	0.36
SGD	0.42	-0.47	-0.45	-0.43	0.16	0.21	-0.40	0.39	0.36	1.00

FIGURE 2.1 – Time plots of daily returns series (in percentage)



2.5.2 Estimation Results

We estimate three models : a model with independent currencies, each governed by a univariate SV model with Student's t innovations (SVt), a MSV model with no factors (MSV-q0) and a MSV model with one factor (MSV-q1). We use comparable priors in the three models and compare the posterior distribution of parameters, volatilities and correlations across models.

Figures 2.2 and 2.3 show posterior densities of the parameters of the volatility equation across currencies and models. These are computed in R using the default kernel density estimation method [7]. The solid line corresponds to the univariate SVt model, the dashed line to the MSV-q0 model and the dotted line to the MSV-q1 model. Tables 2.6 to 2.13 in Appendix 2.7.4 give posterior parameter means, standard deviations, numerical standard errors (NSE) for the mean, and relative numerical efficiency (RNE) for the mean. The NSE and RNE are are computed using the R library coda, using a time series method based on an estimate of the spectral density at 0. Estimations are based on 45,000 draws after discarding the first 6,000 draws.

For the SVt model, the A and Σ matrices are diagonal, so that $\bar{\alpha}_i, A_{ii}$ and σ_{ii} are the parameters of the i'th univariate SV model, $i = 1, \ldots, 10$. For the MSV-q0 and MSV-q1 models there are non-zero off-diagonal elements. In the SVt and MSV-q0 models, the α_{ti}, governed by the $\bar{\alpha}$, A and Σ matrices, are the only source of volatility, while in the MSV-q1 model they give the idiosyncratic volatility, the part of volatility not attributable to the common factor.

We observe that the posterior density of $\bar{\alpha}_i$ and A_{ii} for the MSV models is shifted left compared with the univariate SVt models for all the currencies except MXN and BRL, for which the three posterior densities of $\bar{\alpha}_i$ are very similar [8]. At the same time, the posterior densities of σ_{ii} are shifted right, relative to the

7. The default algorithm disperses the mass of the empirical distribution function over a regular grid of at least 512 points and then uses the fast Fourier transform to convolve this approximation with a discretized version of the kernel and then uses linear approximation to evaluate the density at the specified points.

8. A_{ii} denotes the AR(1) coefficient in the volatility equation (2).

FIGURE 2.2 – Comparison of posterior parameter distributions (Part 1)

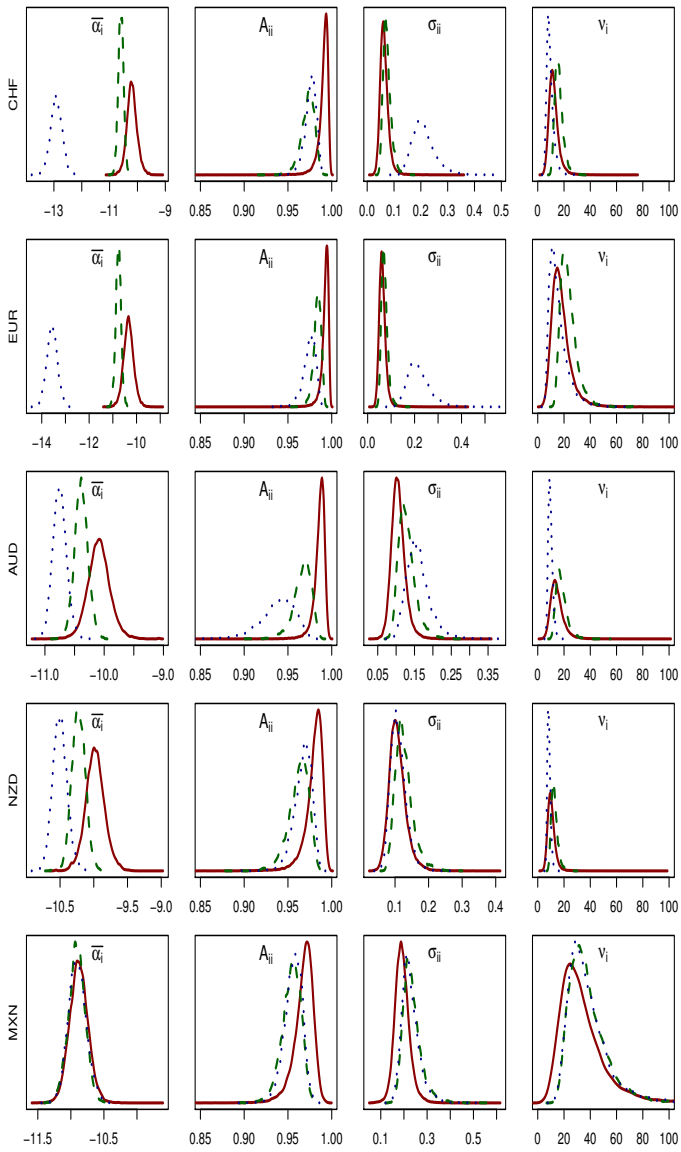
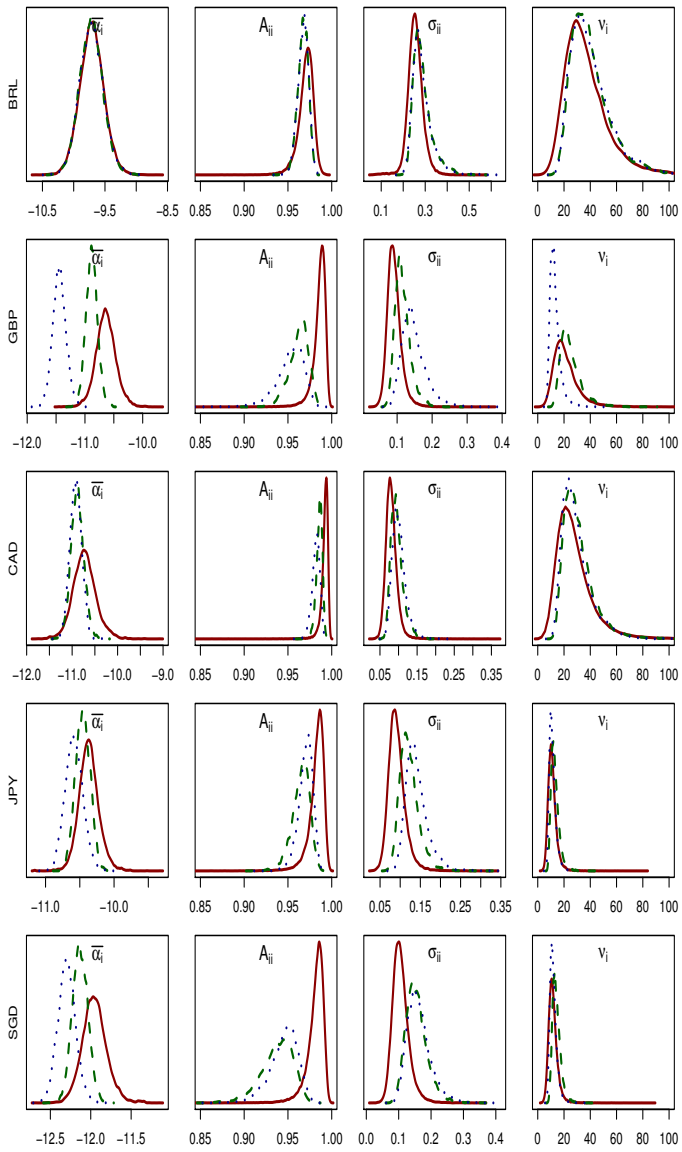


FIGURE 2.3 – Comparison of posterior parameter distributions (Part 2)



univariate models [9]. With respect to the parameter ν_i, for half of the currencies the posterior distribution looks very similar, while for the other half there are some differences, but without a clear pattern.

Passing from the univariate SVt models to the multivariate MSV-q0 model, we obtain in most cases a lower mean, lower persistence and higher volatility of idiosyncratic volatility. The MSV-q0 model allows returns to be conditionally correlated but still with currency-specific degrees of freedom. We see that the posterior mean of the degrees of freedom parameter varies from one currency to another in line with what we observed in the descriptive statistics.

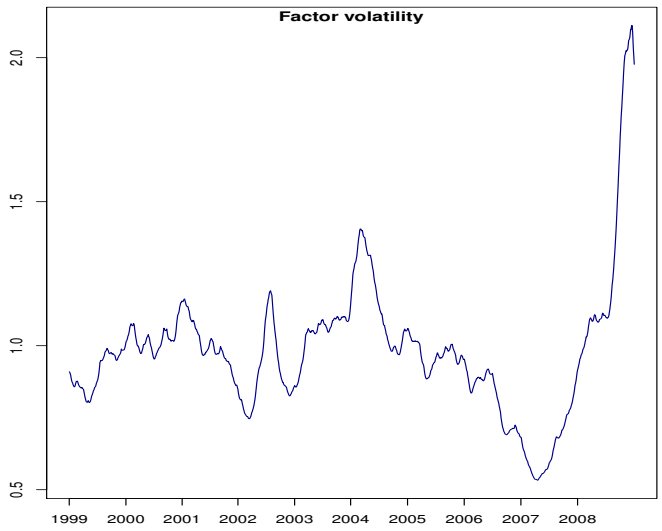
In the MSV-q1 model, there is both idiosyncratic and factor volatility. Figure 2.4 show a plot of the factor volatility and Table 11 presents the posterior parameter distribution statistics of the factor volatility equation. In our model the Bf_t are identified but not B and the f_t separately. The posterior distribution of B is thus quite sensitive to the priors for B and the parameters of the factors [10]. We set $\bar{\alpha}_{11} = 0$ to normalize the variance of the factor to one. Other normalization strategies are possible. Note that there are only two parameters to estimate for the factor volatility equation : $A_{11,11}$, the persistence parameter, and ν_{11} the factor volatility's degree of freedom. The posterior mean of $A_{11,11}$ is 0.99, indicating that the factor volatility is more persistent than the idiosyncratic volatilities. The posterior mean of ν_{11} is around 21, which suggest the conditional factor distribution is not much more fat-tailed than a Gaussian distribution.

We calculate the time varying decomposition of variance into factor and idiosyncratic components and plot them in Figure 2.5. The solid line correspond to the factor component and the dashed line to the idiosyncratic component. We see that the factor is capturing most of the co-movement among the CHF, EUR and GBP currencies. The factor volatility contribution for CHF and EUR currencies is

9. In the case of the MSV models, σ_{ii} represents the square root of the diagonal elements of the variance matrix Σ correspondent to the volatility equation. It measures the volatility of volatility.

10. Table 12 present the posterior parameter of the elements of the B matrix. The relative numerical efficiency of these parameters are low but the efficiency is improved for the Bf_t.

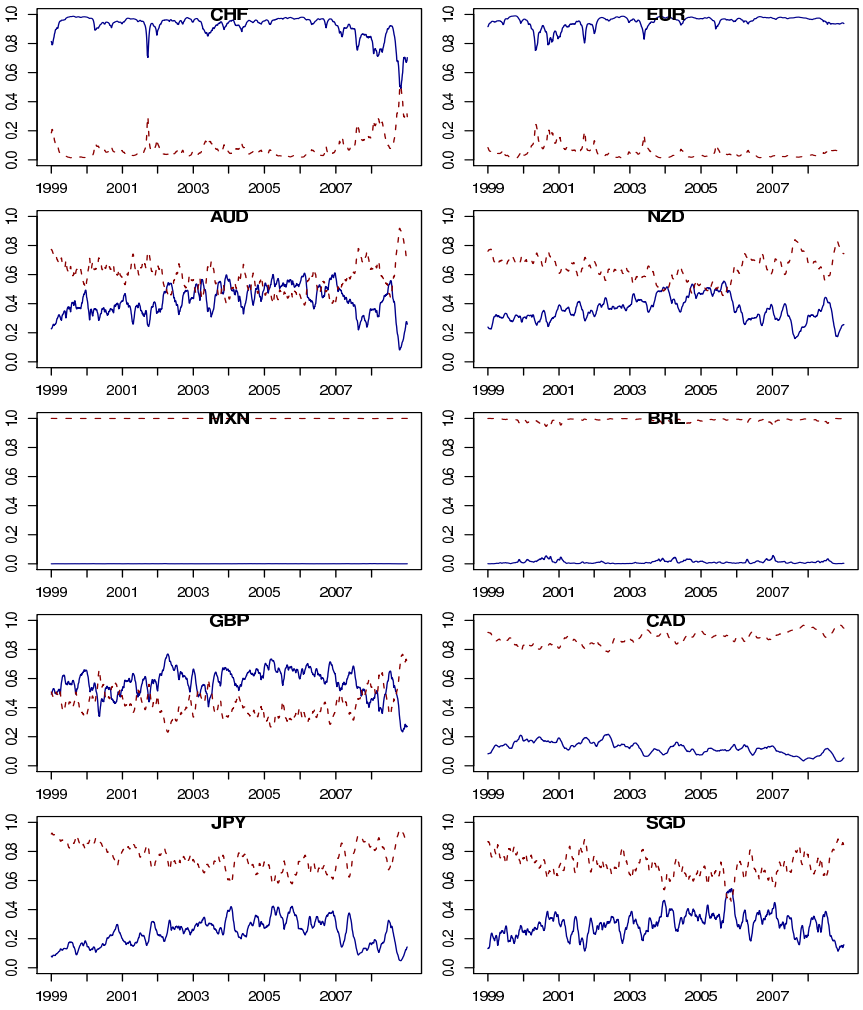
FIGURE 2.4 – Time series plot of the posterior mean of the factor volatility of MSV-q1 model



more than 80 percent for most of the period and for the GBP is slightly greater than 50 percent. For the rest of currencies, the idiosyncratic contribution is higher than the factor, specially for the case of BRL and MXN currencies where the factor contribution is close to zero. This is consistent with the low correlation between these two currencies and the rest. Thus, this suggest that the factor can be identified as an "European" factor, as the CHF, EUR and GBP currencies are the three European currencies in our sample and the factor seems to capture the shocks that affect this region.

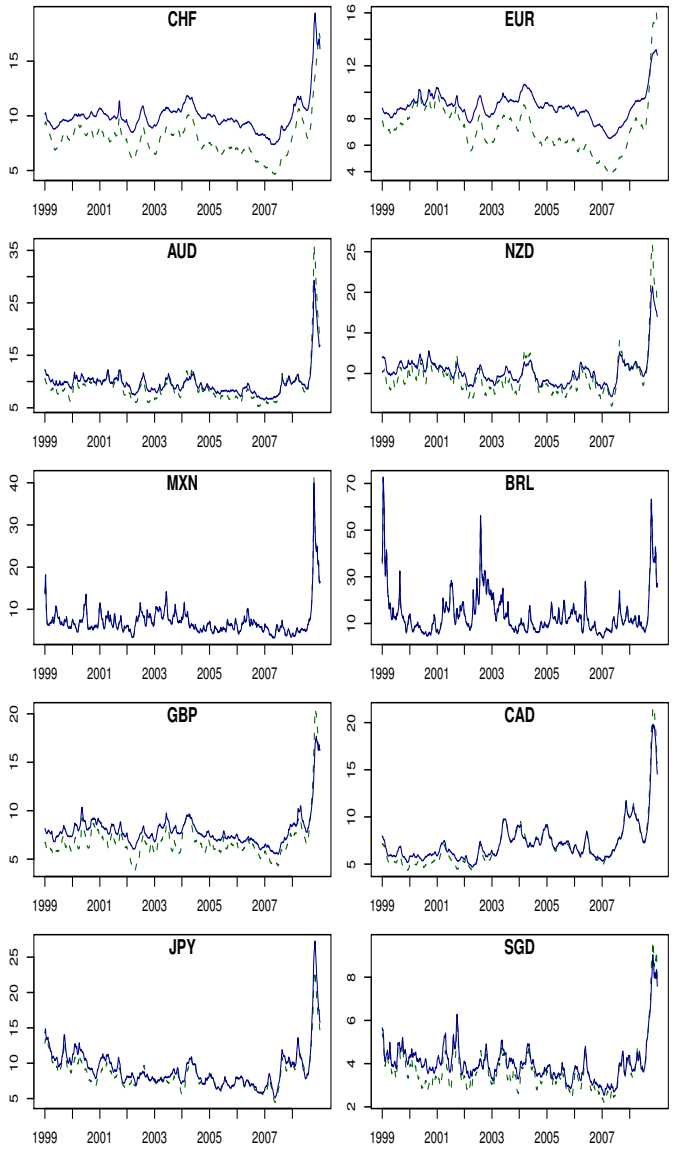
These results also explain why we see a big move to the left in the posterior distribution of the $\bar{\alpha}_i$ (mean idiosyncratic log volatility) for CHF, EUR and GBP. In Figure 2.6 we present the time series plot of the annualized total volatility for the 10 currencies analyzed obtained with MSV-q0 and MSV-q1.The dashed line corresponds to MSV-q0 and the solid line to MSV-q1. We see that estimates are similar across models except for the three currencies with the higher factor contributions, where we notice that for the MSV-q1 model the idiosyncratic volatility

FIGURE 2.5 – Time varying decomposition of variance into factor and idiosyncratic components



has higher mean and is more persistent, compared with the MSV-q0 model.

FIGURE 2.6 – Comparison of annualized posterior mean of volatility



We now analyze estimates of correlations between currencies, across models. As we have discussed in previous sections, the MSV-q0 model, with no factors, implies a time varying variance matrix, but a time invariant correlation matrix; while the MSV model with factors implies that both the variance and the correlation matrices of returns are time-varying. Tables 2.14 and 2.15 in Appendix 2.7.4 show the posterior mean of the R_{11} matrix for MSV-q0 and MSV-q1, respectively. In the case of MSV-q0, R_{11} is the conditional correlation matrix of the returns, $\text{Corr}(r_t|\alpha_t)$. For the MSV-q1 model, we show in Table 2.16 the average across the time dimension of the posterior mean of the corresponding $\text{Corr}(r_t|\alpha_t)$ matrix. If we compare these results with those of MSV-q0 and the sample correlation matrix showed in Table 2.5 we can see that the estimate of the correlation matrix for the MSV-q1 model is closer to the corresponding sample correlation matrix. The estimated conditional correlation matrix for the MSV-q0 model agrees with respect to sign but the magnitudes of the correlation estimates are much smaller.

2.6 Conclusions

We have introduced a new approach for estimating multivariate stochastic volatility models. This approach uses a numerically efficient method to draw volatilities as a block in the time dimension and one-at-a-time in the cross sectional dimension. The proposed algorithm is flexible, allowing different specifications and types of dependence. We can model time-varying conditional correlation matrices by incorporating factors in the return equation, where the factors are independent SV processes with Student's t innovations. Furthermore, we can incorporate copulas to allow conditional return dependence given volatility, allowing different Student's t marginals to capture return heterogeneity. We have tested the correctness of our implementation of the proposed method using procedures similar to those suggested by Geweke (1994).

We apply the proposed method to an exchange rate data set and compare posterior distributions of parameters and volatility with those obtained with univariate

SV models with Student's t innovations. We estimate two multivariate models, one in which we do not include factors and another in which we introduce one factor. We find that for most of the currencies, the multivariate approach with no factors gives a lower mean, lower persistence and higher volatility of volatility than the univariate model. The factor in the factor multivariate model seems to be a kind of "European" factor, as it is mainly capturing co-movement of three European currencies. The factor volatility is more persistent than the idiosyncratic volatilities. It would be interesting to introduce additional factors to see if we can capture other co-movements.

Applying the HESSIAN method one-at-a-time in the cross section only requires that the multivariate state sequence be a Gaussian first-order vector autoregressive process and that the conditional distribution of the observed vector depend only on the contemporaneous state vector. This requirement is satisfied for a wide variety of state space models, including but not limited to multivariate stochastic volatility models.

Using the HESSIAN method overcomes two disadvantages of the auxiliary mixture approach. First, it is less model specific — it does not require the researcher to find a suitable transformation for the model at hand. Second, it is exact — we do not need to correct for mixture approximation, using reweighting or additional Metropolis-Hastings steps, or settle for simulators that are not simulation consistent.

We hope to extend this work to compute marginal likelihoods and to compare the results from different specifications. Also, we hope to extend the model to incorporate leverage effects.

2.7 Appendix to Chapter 2

2.7.1 Computing $\bar{\Omega}^{(i)}$ and $\bar{c}^{(i)}$

We show here how to compute $\bar{\Omega}^{(i)}$ and $\bar{c}^{(i)}$, the conditional precision and covector of the conditionally Gaussian distribution $\alpha_i|\alpha_{-i}$. We start by defining $\bar{\Omega}$ and $\bar{c}$, the prior precision and covector of α. The precision $\bar{\Omega}$ is a $nm \times nm$ block band-diagonal matrix. We will use the notation $\bar{\Omega}_{st}$, $s, t = 1, \ldots, n$, to denote the $m \times m$ submatrix starting at row $(s-1)m+1$ and column $(t-1)m+1$. The non-zero submatrices are the diagonal blocks $\bar{\Omega}_{tt}$ and the off-diagonal blocks $\bar{\Omega}_{t,t+1}$ and $\bar{\Omega}_{t-1,t}$, given by

$$\begin{aligned}
\bar{\Omega}_{tt} &= \Sigma^{-1} + A^\top \Sigma^{-1} A, \quad t = 2, \ldots, n-1, \\
\bar{\Omega}_{11} &= \Sigma_0^{-1} + A^\top \Sigma^{-1} A, \\
\bar{\Omega}_{nn} &= \Sigma^{-1}, \\
\bar{\Omega}_{t,t+1} &= -A^\top \Sigma^{-1}, \quad t = 1, \ldots, n-1, \\
\bar{\Omega}_{t-1,t} &= -\Sigma^{-1} A, \quad t = 1, \ldots, n-1.
\end{aligned} \tag{2.7}$$

The co-vector is a $nm \times 1$ vector stacking n $m \times 1$ subvectors $\bar{c}_t$, given by :

$$\begin{aligned}
\bar{c}_t &= \Sigma^{-1}(I - A)\bar{\alpha} - A^\top \Sigma^{-1}(I - A)\bar{\alpha}, \quad t = 2, \ldots, n-1 \\
\bar{c}_1 &= \Sigma_0^{-1}\bar{\alpha} - A^\top \Sigma^{-1}(I - A)\bar{\alpha}, \\
\bar{c}_n &= \Sigma^{-1}(I - A)\bar{\alpha}.
\end{aligned} \tag{2.8}$$

We now derive the $n \times n$ precision $\bar{\Omega}^{(i)}$ and $n \times 1$ co-vector $\bar{c}^{(i)}$ of the conditional distribution $\alpha_i|\alpha_{-i}$. We know that the conditional density $\pi(\alpha_i|\alpha_{-i})$ is proportional to the joint density $\pi(\alpha)$. Matching coefficients of the first- and second-order monomial terms of $\log \pi(\alpha_i|\alpha_{-i})$ gives the non-zero elements

$$\bar{\Omega}^{(i)}_{tt} = (\bar{\Omega}_{tt})_{ii}, \quad \bar{\Omega}^{(i)}_{t,t+1} = \bar{\Omega}^{(i)}_{t+1,t} = (\bar{\Omega}_{t,t+1})_{ii}.$$

$$\bar{c}_t^{(i)} = (\bar{c}_t)_i - \sum_{j \neq i} \left[(\bar{\Omega}_{tt})_{ji} \alpha_{tj} + (\bar{\Omega}_{t,t+1})_{ji} \alpha_{t+1,j} + (\bar{\Omega}_{t-1,t})_{ji} \alpha_{t-1,j} \right].$$

2.7.2 Computing $\log \pi(y_t|\alpha_t, \nu, B, R)$ and its derivatives with respect to α_{it}

Using equations (2.3), (2.4), and (2.5), we can write $\log \pi(y_t|\alpha_t, B, \nu, R)$ in the following way :

$$\begin{aligned}\log \pi(y_t|\alpha_t, \nu, B, R) = &-\frac{1}{2}\left\{\log|R| + \log 2\pi + x_t^\top (R^{-1} - I)x_t\right\} \\ &-\frac{1}{2}\left\{\sum_{i=1}^{m}\left[\alpha_{it} + (\nu_i + 1)\log\left(1 + \frac{\epsilon_{it}^2}{\nu_i}\right)\right]\right\} \\ &+\sum_{i=1}^{m}\left[\log\Gamma\left(\frac{\nu_i+1}{2}\right) - \log\Gamma\left(\frac{\nu_i}{2}\right) - \frac{1}{2}\log(\nu_i\pi)\right],\end{aligned}$$

where $x_t = (x_{1t}, \ldots, x_{mt})$ and for $i = 1, \ldots, m$,

$$x_{it} = \Phi^{-1}(u_{it}), \quad u_{it} = F_\epsilon(\epsilon_{it}|\nu_i)),$$

$$\epsilon_{it} = \begin{cases}\exp(-\alpha_{it}/2)(r_{it} - \sum_{j=1}^{q} B_{ij} f_{jt}), & i = 1, \ldots, p, \\ \exp(-\alpha_{it}/2) f_{i-p,t}, & i = p+1, \ldots, m.\end{cases}$$

We can evaluate $\log \pi(y_t|\alpha_t, B, \nu, R)$ as a function of α_{it} bottom up, evaluating the ϵ_{it} at α_{it}, then the u_{it} at ϵ_{it}, then the x_{it} at u_{it} then $\log \pi(y_t|\alpha_t, B, \nu, R)$ at ϵ_t and x_t.

We require five derivatives of $\log \pi(y_t|\alpha_t, B, \nu, R)$ with respect to α_{it}, evaluated at α_{it}. Because it is a multi-level compound function of the α_{it}, computing these derivatives in closed form would be extremely tedious and prone to error. Fortunately, we do not need to. Instead, we compute any values we need, bottom up, using Faà di Bruno's formula (2.7.2 below) at each step to compute derivatives of a compound function by combining derivatives of its component functions.

We proceed using the following steps.

1. Compute five derivatives of $\psi(\alpha_{it}) \equiv \log \pi_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)$ with respect to α_{it} at α_{it}, as described in 2.7.2.

2. Compute five derivatives of $x^\top(R^{-1} - I)x$ with respect to x_{it} at x_{it}, as described in 2.7.2.
3. Compute five derivatives of x_{it} with respect to u_{it} at u_{it}, as described in 2.7.2.
4. Compute five derivatives of u_{it} with respect to α_{it} at α_{it}, as described in 2.7.2.
5. Use the Faà di Bruno formula, described in 2.7.2, to compute five derivatives of x_{it} with respect to α_{it} at α_{it}. Inputs are the derivatives of x_{it} with respect to u_{it} at step 3 and the derivatives of u_{it} with respect to α_{it} at step 4.
6. Use the Faà di Bruno formula to compute five derivatives of $x^\top(R^{-1} - I)x$ with respect to α_{it} at α_{it}. Inputs are the derivatives of $x^\top(R^{-1} - I)x$ with respect to x_{it} at step 2 and the derivatives of x_{it} with respect to α_{it} at step 5.
7. Compute five derivatives of $\log \pi(y_t|\alpha_t, \theta, B, R)$ with respect to α_{it} at α_{it} directly using the derivatives at steps 1 and 6.

For convenience, we define

$$\eta_t = \begin{bmatrix} \eta_{1t} \\ \vdots \\ \eta_{mt} \end{bmatrix} = \begin{bmatrix} r_t - Bf_t \\ f_t \end{bmatrix},$$

Derivatives of $\psi(\alpha_{it})$ with respect to α_{it}

For the special case of Student's t F,

$$\pi_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|v_i) = \frac{\Gamma(\frac{\nu_i+1}{2})}{\sqrt{\nu_i\pi}\Gamma(\frac{\nu_i}{2})}\left(1+\frac{e^{-\alpha_{it}}\eta_{it}^2}{\nu_i}\right)^{-\frac{\nu_i+1}{2}}$$

$$\psi(\alpha_{it}) = \log\left[\frac{\Gamma(\frac{\nu_i+1}{2})}{\sqrt{\nu_i\pi}\Gamma(\frac{\nu_i}{2})}\right] - \frac{\nu_i+1}{2}\log(1+s_{it})$$

where $s_{it} \equiv e^{-\alpha_{it}}\eta_{it}^2/\nu_i$. Noting that $\partial s_{it}/\partial\alpha_i = -s_{it}$, we compute

$$\psi'(\alpha_{it}) = \frac{\nu_i+1}{2}\frac{s_{it}}{1+s_{it}}, \quad \psi''(\alpha_{it}) = -\frac{\nu_i+1}{2}\frac{s_{it}}{(1+s_{it}^2)},$$

$$\psi'''(\alpha_{it}) = \frac{\nu_i+1}{2}\frac{s_{it}(1-s_{it})}{(1+s_{it})^3}, \quad \psi^{(4)}(\alpha_{it}) = -\frac{\nu_i+1}{2}\frac{s_{it}(1-4s_{it}+s_{it}^2)}{(1+s_{it})^4},$$

$$\psi^{(5)}(\alpha_{it}) = \frac{\nu_i+1}{2}\frac{s_{it}(1-11s_{it}+11s_{it}^2-s_{it}^3)}{(1+s_{it})^5}.$$

Derivatives of $x^\top(I - R^{-1})x$ with respect to x_{it}

In this section we show how to compute partial derivatives of $\log c(u_1, \ldots, u_m)$ with respect to the u_i. We can write

$$\begin{aligned}\log c_R(u_1, \ldots, u_m) &= \log \phi_R(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_m)) - \sum_{i=1}^{m} \log \phi(\Phi^{-1}(u_i)) \\ &= \frac{1}{2}|H| + \frac{1}{2}x^\top(I - R^{-1})x,\end{aligned}$$

where $x = (x_1, \ldots, x_m) = (\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_m))$.

The gradient and Hessian of $\log(c_R)$ with respect to u are

$$\frac{\partial \log c(u)}{\partial x} = (I - R^{-1})x,$$

$$\frac{\partial \log c(u)}{\partial x \partial x^\top} = I - R^{-1}.$$

All third order partial derivatives and higher are zero.

Derivatives of x_{it} with respect to u_{it}

We now use the relationship $\Phi(x_i) = u_i$ to compute derivatives of x_i with respect to u_i. Differentiating with respect to u_i gives $\phi(x_i)\frac{\partial x_i}{\partial u_i} = 1$, and thus

$$\frac{\partial x_i}{\partial u_i} = \frac{1}{\phi(x_i)}.$$

Taking further derivatives gives

$$\frac{\partial^2 x_i}{\partial u_i} = 2\pi e^{x_i^2} x_i,$$

$$\frac{\partial^3 x_i}{\partial u_i} = (2\pi)^{3/2} e^{3x_i^2/2}(2x_i^2 + 1),$$

$$\frac{\partial^4 x_i}{\partial u_i} = (2\pi)^2 e^{2x_i^2}(6x_i^3 + 7x_i),$$

$$\frac{\partial^5 x_i}{\partial u_i} = (2\pi)^{5/2} e^{5x_i^2/2}(24x_i^4 + 46x_i^2 + 7).$$

Derivatives of $F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)$

We describe here how to compute five derivatives of $F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)$ with respect to α_{it}. We write down the derivatives in terms of $\psi(\alpha_{it}) \equiv \log \pi_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)$:

$$\frac{\partial F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)}{\partial \alpha_{it}} = \pi_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)\left(-\frac{1}{2}e^{-\alpha_{it}/2}\eta_{it}\right) = -\frac{\eta_{it}}{2}e^{-0.5\alpha_{it}+\psi(\alpha_{it})}$$

Then

$$\frac{\partial^2 F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)}{\partial \alpha_{it}^2} = -\frac{\eta_{it}}{2}e^{-0.5\alpha_{it}+\psi(\alpha_{it})}[-0.5+\psi'(\alpha_{it})]$$

$$\frac{\partial^3 F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)}{\partial \alpha_{it}^3} = -\frac{\eta_{it}}{2}e^{-0.5\alpha_{it}+\psi(\alpha_{it})}\left[\psi''(\alpha_{it})+(-0.5+\psi'(\alpha_{it})^2\right]$$

$$\frac{\partial^4 F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)}{\partial \alpha_{it}^4} = -\frac{\eta_{it}}{2}e^{-0.5\alpha_{it}+\psi(\alpha_{it})}\left[\psi'''(\alpha_{it}) + 3(-0.5+\psi'(\alpha_{it}))\psi''(\alpha_{it}) + (-0.5+\psi'(\alpha_{it}))^3\right]$$

$$\frac{\partial^5 F_\epsilon(e^{-\alpha_{it}/2}\eta_{it}|\theta_i)}{\partial \alpha_{it}^5} = -\frac{\eta_{it}}{2}e^{\psi(\alpha_{it})}\left[\psi^{(4)}(\alpha_{it}) + 4(-0.5+\psi'(\alpha_{it}))\psi'''(\alpha_{it}) + 3(\psi''(\alpha_{it}))^2 + 6(-0.5+\psi'(\alpha_{it}))^2\psi''(\alpha_{it}) + (-0.5+\psi'(\alpha_{it}))^4\right]$$

Faà di Bruno Formula

The Faà di Bruno Formula combines the derivatives of primitive functions to obtain the derivatives of composite functions. We can use it to evaluate exact multiple derivatives of compound functions at a point without needing to write out the derivatives of the compound function in closed form.

For the composite function $h = f \circ g$, the Faà di Bruno formula gives

$$h' = f'g',$$

$$h'' = f'g'' + f''(g')^2,$$

$$h''' = f'g''' + 3f''g'g'' + f'''(g')^3,$$

$$h^{(4)} = f'g^{(4)} + 4f''g'g''' + 3f''(g'')^2 + 6f'''(g')^2g'' + f^{(4)}(g')^4,$$

$$h^{(5)} = f'g^{(5)}+5f''g'g^{(4)}+10f''g''g'''+15f'''(g'')^2g'+10f'''g'''(g')^2+10f^{(4)}g''(g')^3+f^{(5)}(g')^5.$$

If $f^{(j)} = 0$ for $j > 2$, the third and higher derivatives simplify to

$$h''' = f'g''' + 3f''g'g'',$$

$$h^{(4)} = f'g^{(4)} + 4f''g'g''' + 3f''(g'')^2,$$

$$h^{(5)} = f'g^{(5)} + 5f''g'g^{(4)} + 10f''g''g'''.$$

2.7.3 Sampling $r|\alpha, \theta, f, B, R$

We draw r from $\pi(r|\alpha, \theta, f, B, R)$ using the following steps :

1. Compute the Cholesky decomposition $R = LL^\top$ of the correlation matrix R.
2. For each $t = 1, ..., n$:
   (a) Draw $z \sim N(0, I_m)$.
   (b) Set $g = Lz$
   (c) Compute the integral probability transform $u_i = \Phi(g_i)$, $i = 1, ..., m$, where Φ is the standard univariate Gaussian cdf.
   (d) Transform each of the u_i to a Student's t with ν_i degree of freedom : $t_i = F^{-1}(u_i)$, where F^{-1} is the inverse cdf of a Student's t distribution with ν_i degrees of freedom.
   (e) Scale each of the t_i random variables to form $\epsilon_{ti} = t_i \exp(0.5\alpha_{ti})$.
   (f) Form $r_t = Bf_t + \epsilon_t$.

2.7.4 Tables of results

Table 2.6 to 2.13 present the posterior parameter distributions for univariate SV-t models, MSV-q0 and MSV-q1 models. Table 2.14 and 2.15 present the posterior mean of correlation matrices.

TABLE 2.6 – Posterior statistics of parameters of univariate SV models with student-t errors (Part 1)

Parameters	Mean	Std	NSE	RNE
CHF				
$\bar{\alpha}_i$	-10.195	0.189	1.900e-03	2.048e-01
A_{ii}	0.993	0.003	0.000e+00	4.056e-01
σ_{ii}	0.061	0.010	1.000e-04	5.362e-01
ν_i	12.530	3.062	2.110e-02	4.227e-01
σ_α	0.491	0.139	7.0000e-04	7.0480e-01
EUR				
$\bar{\alpha}_i$	-10.333	0.218	1.600e-03	3.843e-01
A_{ii}	0.994	0.002	0.000e+00	5.048e-01
σ_{ii}	0.061	0.010	1.000e-04	4.387e-01
ν_i	18.507	6.252	3.760e-02	5.538e-01
σ_α	0.540	0.143	6.0000e-04	1.0815e+00
AUD				
$\bar{\alpha}_i$	-10.089	0.190	1.400e-03	3.465e-01
A_{ii}	0.988	0.004	0.000e+00	3.944e-01
σ_{ii}	0.104	0.013	1.000e-04	3.839e-01
ν_i	15.225	4.279	3.350e-02	3.261e-01
σ_α	0.657	0.133	7.0000e-04	8.1990e-01
NZD				
$\bar{\alpha}_i$	-9.980	0.144	1.100e-03	3.473e-01
A_{ii}	0.983	0.006	0.000e+00	5.296e-01
σ_{ii}	0.104	0.018	1.000e-04	6.050e-01
ν_i	10.547	2.294	1.410e-02	5.258e-01
σ_α	0.557	0.131	6.0000e-04	9.1270e-01
MXN				
$\bar{\alpha}_i$	-10.880	0.142	1.000e-03	3.906e-01
A_{ii}	0.971	0.008	0.000e+00	5.010e-01
σ_{ii}	0.188	0.023	1.000e-04	5.632e-01
ν_i	33.955	13.626	7.180e-02	7.197e-01
σ_α	0.776	0.129	6.0000e-04	8.4840e-01

TABLE 2.7 – Posterior statistics of parameters of univariate SV models with student-t errors (Part 2)

Parameters	Mean	Std	NSE	RNE
BRL				
$\bar{\alpha}_i$	-9.711	0.194	1.200e-03	5.672e-01
A_{ii}	0.973	0.006	0.000e+00	6.677e-01
σ_{ii}	0.253	0.023	1.000e-04	7.442e-01
ν_i	37.990	15.446	1.012e-01	4.661e-01
σ_α	1.073	0.148	8.0000e-04	7.2330e-01
GBP				
$\bar{\alpha}_i$	-10.637	0.174	1.500e-03	2.615e-01
A_{ii}	0.989	0.004	0.000e+00	4.604e-01
σ_{ii}	0.088	0.013	1.000e-04	5.834e-01
ν_i	22.118	8.072	5.420e-02	4.429e-01
σ_α	0.566	0.135	6.0000e-04	8.6320e-01
CAD				
$\bar{\alpha}_i$	-10.729	0.255	2.100e-03	3.050e-01
A_{ii}	0.993	0.002	0.000e+00	4.161e-01
σ_{ii}	0.078	0.010	1.000e-04	4.885e-01
ν_i	28.675	11.221	6.720e-02	5.579e-01
σ_α	0.662	0.150	6.0000e-04	1.1276e+00
JPY				
$\bar{\alpha}_i$	-10.369	0.148	1.100e-03	3.370e-01
A_{ii}	0.986	0.005	0.000e+00	2.992e-01
σ_{ii}	0.087	0.014	1.000e-04	3.502e-01
ν_i	11.220	2.528	1.580e-02	5.121e-01
σ_α	0.504	0.116	6.0000e-04	7.1950e-01
SGD				
$\bar{\alpha}_i$	-11.953	0.154	1.200e-03	3.081e-01
A_{ii}	0.984	0.006	0.000e+00	3.795e-01
σ_{ii}	0.102	0.017	1.000e-04	4.366e-01
ν_i	11.814	2.681	1.560e-02	5.915e-01
σ_α	0.568	0.136	7.0000e-04	6.7830e-01

TABLE 2.8 – Posterior statistics of parameters of log volatility equation in the MSV-q0 model (Part 1)

Parameters	Mean	Std	NSE	RNE
CHF				
$\bar{\alpha_i}$	-10.594	0.095	7.1266e-03	4.4129e-03
A_{ii}	0.97	0.008	2.3179e-04	2.8595e-02
A_{ij}	0.002	0.001	2.3891e-05	2.7707e-02
σ_{ii}	0.0746	0.016	3.6917e-04	4.4277e-02
ν_i	16.19	3.216	7.7741e-02	4.2774e-02
σ_α	0.324	0.060	1.3822e-03	4.7283e-02
EUR				
$\bar{\alpha_i}$	-10.764	0.112	7.1074e-03	6.2172e-03
A_{ii}	0.98	0.004	1.0760e-04	4.2351e-02
A_{ij}	0.001	0.000	1.1393e-05	4.2358e-02
σ_{ii}	0.0722	0.014	2.9354e-04	5.4371e-02
ν_i	23.50	6.342	1.5043e-01	4.4441e-02
σ_α	0.404	0.070	1.5928e-03	4.7762e-02
AUD				
$\bar{\alpha_i}$	-10.387	0.114	8.1451e-03	4.8991e-03
A_{ii}	0.97	0.010	2.3804e-04	4.6442e-02
A_{ij}	0.003	0.001	2.6327e-05	4.8545e-02
σ_{ii}	0.1294	0.024	4.3514e-04	7.3557e-02
ν_i	18.15	4.237	8.6456e-02	6.0037e-02
σ_α	0.513	0.084	2.0306e-03	4.2290e-02
NZD				
$\bar{\alpha_i}$	-10.226	0.101	6.9267e-03	5.2884e-03
A_{ii}	0.96	0.012	2.4565e-04	5.6291e-02
A_{ij}	0.003	0.001	2.7232e-05	4.8866e-02
σ_{ii}	0.1268	0.026	6.1017e-04	4.7135e-02
ν_i	12.78	2.444	4.6843e-02	6.8060e-02
σ_α	0.472	0.082	1.8883e-03	4.6627e-02
MXN				
$\bar{\alpha_i}$	-10.911	0.127	8.3612e-03	5.7422e-03
A_{ii}	0.95	0.011	2.3910e-04	5.0846e-02
A_{ij}	0.004	0.001	2.6614e-05	5.4285e-02
σ_{ii}	0.2353	0.041	1.0775e-03	3.6274e-02
ν_i	38.12	14.615	3.3147e-01	4.8600e-02
σ_α	0.792	0.118	3.2518e-03	3.2956e-02

TABLE 2.9 – Posterior statistics of parameters of log volatility equation in the MSV-q0 model (Part 2)

Parameters	Mean	Std	NSE	RNE
BRL				
$\bar{\alpha_i}$	-9.721	0.186	1.3186e-02	4.9542e-03
A_{ii}	0.97	0.006	1.4302e-04	4.5711e-02
A_{ij}	0.002	0.001	1.6865e-05	6.1487e-02
σ_{ii}	0.2869	0.044	7.7500e-04	7.9142e-02
ν_i	40.69	16.323	3.7063e-01	4.8490e-02
σ_α	1.137	0.166	2.9253e-03	8.0389e-02
GBP				
$\bar{\alpha_i}$	-10.884	0.101	8.1928e-03	3.7836e-03
A_{ii}	0.96	0.011	3.3021e-04	2.7418e-02
A_{ij}	0.003	0.001	3.4675e-05	2.8054e-02
σ_{ii}	0.1158	0.022	4.6367e-04	5.6518e-02
ν_i	25.25	8.309	2.1649e-01	3.6826e-02
σ_α	0.433	0.072	1.5308e-03	5.5078e-02
CAD				
$\bar{\alpha_i}$	-10.884	0.142	8.3868e-03	7.1511e-03
A_{ii}	0.99	0.004	7.2136e-05	5.9020e-02
A_{ij}	0.001	0.000	8.9339e-06	5.2327e-02
σ_{ii}	0.0960	0.017	3.2542e-04	6.9751e-02
ν_i	31.07	11.796	2.4744e-01	5.6812e-02
σ_α	0.578	0.094	1.9962e-03	5.5770e-02
JPY				
$\bar{\alpha_i}$	-10.456	0.106	7.2345e-03	5.3986e-03
A_{ii}	0.96	0.010	2.4814e-04	4.2152e-02
A_{ij}	0.003	0.001	2.7392e-05	4.0232e-02
σ_{ii}	0.1227	0.024	4.1054e-04	8.7760e-02
ν_i	13.21	3.156	1.0109e-01	2.4375e-02
σ_α	0.473	0.082	1.5759e-03	6.7970e-02
SGD				
$\bar{\alpha_i}$	-12.132	0.097	7.7733e-03	3.8583e-03
A_{ii}	0.94	0.018	4.3584e-04	4.4761e-02
A_{ij}	0.006	0.002	4.8223e-05	4.1560e-02
σ_{ii}	0.1550	0.033	7.7144e-04	4.6245e-02
ν_i	14.41	3.319	7.9066e-02	4.4053e-02
σ_α	0.450	0.074	1.5790e-03	5.5276e-02

TABLE 2.10 – Posterior statistics of parameters of log volatility equation in the MSV-q1 model (Part 1)

Parameters	Mean	Std	NSE	RNE
CHF				
$\bar{\alpha_i}$	-12.913	0.210	1.1413e-02	8.5047e-03
A_{ii}	0.98	0.007	1.9454e-04	2.8507e-02
A_{ij}	0.001	0.001	2.1978e-05	2.5953e-02
σ_{ii}	0.211	0.043	1.4328e-03	2.2969e-02
ν_i	9.423	2.724	1.0473e-01	1.6920e-02
σ_α	0.974	0.165	5.8899e-03	1.9739e-02
EUR				
$\bar{\alpha_i}$	-13.565	0.222	1.4209e-02	6.0833e-03
A_{ii}	0.97	0.007	2.7650e-04	1.7289e-02
A_{ij}	0.001	0.001	3.3092e-05	1.3731e-02
σ_{ii}	0.218	0.048	1.7203e-03	1.9101e-02
ν_i	15.823	7.375	3.2775e-01	1.2660e-02
σ_α	0.977	0.163	6.6350e-03	1.5145e-02
AUD				
$\bar{\alpha_i}$	-10.751	0.116	8.6855e-03	4.4563e-03
A_{ii}	0.94	0.019	5.6233e-04	2.9813e-02
A_{ij}	0.005	0.002	5.5356e-05	2.9620e-02
σ_{ii}	0.161	0.034	8.8369e-04	3.6101e-02
ν_i	9.641	1.708	3.8678e-02	4.8760e-02
σ_α	0.474	0.079	1.9784e-03	3.9497e-02
NZD				
$\bar{\alpha_i}$	-10.503	0.107	7.1830e-03	5.5007e-03
A_{ii}	0.97	0.010	2.6698e-04	3.7050e-02
A_{ij}	0.002	0.001	2.3473e-05	3.5510e-02
σ_{ii}	0.110	0.024	5.4657e-04	4.7496e-02
ν_i	8.528	1.248	3.1835e-02	3.8428e-02
σ_α	0.439	0.076	2.1354e-03	3.1525e-02
MXN				
$\bar{\alpha_i}$	-10.902	0.136	8.8611e-03	5.8568e-03
A_{ii}	0.96	0.010	2.8022e-04	3.2519e-02
A_{ij}	0.003	0.001	2.6870e-05	3.8055e-02
σ_{ii}	0.229	0.039	8.4206e-04	5.4197e-02
ν_i	37.830	14.900	4.0716e-01	3.3482e-02
σ_α	0.788	0.117	2.5907e-03	5.1014e-02

TABLE 2.11 – Posterior statistics of parameters of log volatility equation in the MSV-q1 (Part 2)

Parameters	Mean	Std	NSE	RNE
BRL				
$\bar{\alpha_i}$	-9.715	0.190	1.2265e-02	6.0073e-03
A_{ii}	0.97	0.006	1.3283e-04	5.4991e-02
A_{ij}	0.002	0.001	2.0593e-05	3.3084e-02
σ_{ii}	0.286	0.046	1.0295e-03	4.9610e-02
ν_i	40.773	16.229	4.3679e-01	3.4516e-02
σ_α	1.136	0.170	3.9321e-03	4.6466e-02
GBP				
$\bar{\alpha_i}$	-11.442	0.116	8.6300e-03	4.4953e-03
A_{ii}	0.95	0.016	4.2810e-04	3.3885e-02
A_{ij}	0.004	0.001	4.1659e-05	3.1862e-02
σ_{ii}	0.147	0.034	6.8056e-04	6.3592e-02
ν_i	13.247	4.334	1.4878e-01	2.1219e-02
σ_α	0.491	0.085	1.8577e-03	5.2150e-02
CAD				
$\bar{\alpha_i}$	-10.919	0.129	7.0184e-03	8.3930e-03
A_{ii}	0.98	0.005	1.2659e-04	3.1920e-02
A_{ij}	0.001	0.000	1.1682e-05	3.2690e-02
σ_{ii}	0.101	0.019	4.3968e-04	4.5543e-02
ν_i	29.651	11.388	2.8737e-01	3.9266e-02
σ_α	0.540	0.087	1.8511e-03	5.5488e-02
JPY				
$\bar{\alpha_i}$	-10.584	0.131	8.0320e-03	6.7004e-03
A_{ii}	0.97	0.008	2.1608e-04	3.5409e-02
A_{ij}	0.002	0.001	2.1152e-05	3.5898e-02
σ_{ii}	0.141	0.027	6.0917e-04	4.9221e-02
ν_i	11.341	2.611	6.5062e-02	4.0255e-02
σ_α	0.588	0.097	2.3360e-03	4.2833e-02
SGD				
$\bar{\alpha_i}$	-12.297	0.109	7.8026e-03	4.9218e-03
A_{ii}	0.95	0.016	5.0045e-04	2.5373e-02
A_{ij}	0.004	0.001	4.3480e-05	2.9719e-02
σ_{ii}	0.163	0.037	1.1153e-03	2.7135e-02
ν_i	11.883	2.633	5.5718e-02	5.5822e-02
σ_α	0.504	0.085	2.4041e-03	3.1497e-02

TABLE 2.12 – Posterior statistics of parameters of the factor volatility for MSV-q1 model

	Mean	Std	NSE	RNE
$AR(1)$	0.9903	0.003	6.2776e-05	4.453e-02
ν	21.25	7.831	2.0041e-01	3.8173e-02

TABLE 2.13 – Posterior statistics of loading factor matrix B for MSV-q1 model

	Mean	Std	NSE	RNE
B_1	-6.0683e-03	4.0507e-04	3.4856e-05	3.3765e-03
B_2	5.5567e-03	3.7097e-04	3.2516e-05	3.2540e-03
B_3	3.8570e-03	2.6975e-04	2.2083e-05	3.7304e-03
B_4	3.8550e-03	2.7787e-04	2.1401e-05	4.2145e-03
B_5	-8.6580e-05	8.8846e-05	2.0822e-06	4.5518e-02
B_6	-7.0001e-04	1.4292e-04	6.0137e-06	1.4121e-02
B_7	3.7751e-03	2.6207e-04	2.1915e-05	3.5753e-03
B_8	-1.5098e-03	1.3428e-04	8.2632e-06	6.6019e-03
B_9	-2.6726e-03	2.1028e-04	1.4747e-05	5.0834e-03
B_{10}	-1.3541e-03	1.0205e-04	7.4755e-06	4.6590e-03

TABLE 2.14 – Posterior mean of correlation matrix R_{11} for MSV-q0 model

	CHF	EUR	AUD	NZD	MXN	BRL	GBP	CAD	JPY	SGD
CHF	1.000	-0.354	-0.148	-0.133	-0.035	-0.007	-0.245	0.092	0.163	0.151
EUR	-0.354	1.000	0.186	0.169	-0.003	-0.037	0.262	-0.120	-0.130	-0.166
AUD	-0.148	0.186	1.000	0.313	-0.080	-0.089	0.163	-0.186	-0.083	-0.161
NZD	-0.133	0.169	0.313	1.000	-0.060	-0.070	0.156	-0.152	-0.057	-0.147
MXN	-0.035	-0.003	-0.080	-0.060	1.000	0.204	-0.016	0.081	-0.042	0.049
BRL	-0.007	-0.037	-0.089	-0.070	0.204	1.000	-0.038	0.076	-0.008	0.077
GBP	-0.245	0.262	0.163	0.156	-0.016	-0.038	1.000	-0.095	-0.105	-0.131
CAD	0.092	-0.120	-0.186	-0.152	0.081	0.076	-0.095	1.000	0.036	0.103
JPY	0.163	-0.130	-0.083	-0.057	-0.042	-0.008	-0.105	0.036	1.000	0.197
SGD	0.151	-0.166	-0.161	-0.147	0.049	0.077	-0.131	0.103	0.197	1.000

TABLE 2.15 – Posterior mean of correlation matrix R_{11} for MSV-q1 model

	CHF	EUR	AUD	NZD	MXN	BRL	GBP	CAD	JPY	SGD
CHF	1.000	-0.055	0.134	0.126	-0.095	-0.101	0.034	-0.082	0.054	-0.056
EUR	-0.055	1.000	-0.045	-0.047	0.008	-0.004	-0.027	0.025	0.075	0.055
AUD	0.134	-0.045	1.000	0.308	-0.101	-0.093	0.051	-0.160	-0.010	-0.100
NZD	0.126	-0.047	0.308	1.000	-0.074	-0.069	0.058	-0.118	0.013	-0.090
MXN	-0.095	0.008	-0.101	-0.074	1.000	0.204	-0.032	0.091	-0.042	0.057
BRL	-0.101	-0.004	-0.093	-0.069	0.204	1.000	-0.030	0.077	-0.016	0.078
GBP	0.034	-0.027	0.051	0.058	-0.032	-0.030	1.000	-0.020	-0.011	-0.021
CAD	-0.082	0.025	-0.160	-0.118	0.091	0.077	-0.020	1.000	-0.010	0.063
JPY	0.054	0.075	-0.010	0.013	-0.042	-0.016	-0.011	-0.010	1.000	0.174
SGD	-0.056	0.055	-0.100	-0.090	0.057	0.078	-0.021	0.063	0.174	1.000

TABLE 2.16 – Average of posterior mean of conditional correlation matrix $Corr(r_t|\alpha_t)$ for MSV-q1 model

	CHF	EUR	AUD	NZD	MXN	BRL	GBP	CAD	JPY	SGD
CHF	1.000	-0.937	-0.593	-0.546	-0.005	0.071	-0.721	0.311	0.475	0.508
EUR	-0.937	1.000	0.621	0.573	-0.019	-0.099	0.734	-0.330	-0.455	-0.518
AUD	-0.593	0.621	1.000	0.575	-0.090	-0.137	0.517	-0.336	-0.322	-0.417
NZD	-0.546	0.573	0.575	1.000	-0.072	-0.116	0.484	-0.294	-0.282	-0.385
MXN	-0.005	-0.019	-0.090	-0.072	1.000	0.205	-0.037	0.092	-0.027	0.060
BRL	0.071	-0.099	-0.137	-0.116	0.205	1.000	-0.096	0.107	0.036	0.121
GBP	-0.721	0.734	0.517	0.484	-0.037	-0.096	1.000	-0.273	-0.373	-0.423
CAD	0.311	-0.330	-0.336	-0.294	0.092	0.107	-0.273	1.000	0.157	0.236
JPY	0.475	-0.455	-0.322	-0.282	-0.027	0.036	-0.373	0.157	1.000	0.392
SGD	0.508	-0.518	-0.417	-0.385	0.060	0.121	-0.423	0.236	0.392	1.000

Chapter 3 :
The information content of Realized Volatility

3.1 Introduction

Modeling and forecasting volatility is one of the most active areas of research in finance, e.g., portfolio design and risk management. However volatility is recognized to be time varying, and is not directly observable, rather it is a latent variable. The recent availability of ultra-high frequency data has made the use of non parametric measurements such as the Realized Volatility (RV) estimator proposed by (Andersen and Bollerslev 1998) more attractive. The basic RV estimator is computed as the sum of squared intra-day returns.

$$RV_t = \sum_{j=1}^{m} r_{t,j}^2$$

Andersen and Bollerslev (1998) and Andersen, Bollerslev, Diebold, and Lays (2001) show that, under general conditions, RV is a consistent estimator of the integrated variance of the price process $IV_t = \int_{t-1}^{t} \sigma_\tau^2 d\tau$ as m goes to infinity. However, measurement errors in prices, known as microstructure effects and the possibility of jumps can cause RV to be a biased estimator of integrated volatility [11]. Therefore,

11. Microstructure effects capture a variety of frictions inherent in the trading process : bid ask bounces, discreteness of price changes, differences in trade sizes or informational content of

multiple variations of this basic realized volatility now exist to address these issues. To control for microstructure noise, the first solution proposed was to use sparse RV estimators, ignoring very high frequency data and focusing on determining an optimal frequency (see for example Ait-Sahalia, Mykland, and Zhang (2005), Zhou (1996)). More recent studies propose estimators that explicitly deal with microstructure noise while taking advantage of the highest frequency available. In this line are the two scales realized volatility estimator of Zhang, Mykland, and Ait-Sahalia (2005) and the realized kernel estimator of Barndorff-Nielsen, E., Hansen, Lunde, and Shephard (2008). To distinguish between jumps and diffusion movements, estimators such as the bi-power variations are used (see Barndorff-Nielsen and Shephard (2004), Andersen, Bollerslev, and Diebold (2007)).

The realized volatility literature typically attempts to evaluate these competing measures by their ability to predict integrated volatility, IV_{t+1}. A popular approach to compare volatility forecasts is to use the R^2 of a Mincer-Zarnowitz style regression [12] :

$$IV_{t+1} = \alpha + \beta RV_t + \epsilon_t$$

However, since IV_{t+1} is never exactly known, it is typically replaced by a RV_{t+1} measure ; and as RV is an estimated quantity, there is an error-in-variable problem in the estimation. Patton (2008) proposes a data-based ranking of RV estimators. For account of practicality it is often assumed that the true volatility process follows a random walk. However, Wright (1999), Jacquier, Polson, and Rossi (1994), Jacquier, Polson, and Rossi (2004) show strong evidence against a random walk in volatility. This is a crucial difference, especially for multi-step forecasts.

The empirical realized volatility literature most often documents the ability of RV estimators to forecast themselves. In contrast, we take the standpoint of an

price changes, gradual response of prices to a block trade, strategic component of the order flow, inventory control effects, etc. (For more discussion about microstructure noise see Hasbrouck (1993))

12. See for example Andersen, Bollerslev, Diebold, and Lays (2003), Andersen, Bollerslev and Meddahi [2005, 2006], Corsi (2009).

investor who recognizes that the daily variance is a latent variable, h_t, and that RV_t is only an observable proxy with information on this latent variable, not the object to be predicted. Our objective is to measure the contribution of different RV estimators to the quality of inference and forecast of the latent variance h_t in periods of stress, when getting volatility right is most important.

We use a classic stochastic volatility (SV) model as the benchmark model. SV models and GARCH-type models are the most popular parametric models used to estimate unobserved volatility. SV models differs from the GARCH type models in that the conditional volatility is treated as a latent variable and not a deterministic function of lagged returns. This feature makes SV models more flexible than GARCH-type models. Jacquier, Polson, and Rossi (1994), Kim, Shephard, and Chib (1998) show that a lot can be gained from the added flexibility of the SV models over the GARCH models, especially in times of stress (Geweke (1994)).

The investor, absent of intra-day information, can consider reduced form SV models of the type :

$$\begin{aligned} r_t &= \sqrt{h_t}\epsilon_t \\ \log h_{t+1} &= \alpha + \delta \log h_t + \sigma_v v_{t+1} \end{aligned} \tag{3.1}$$

ϵ_t, v_t can be correlated and have fat tails, $0 < \delta < 1$.

She can initially incorporate the intra-day information into this parametric model under the form of an exogenous variable (X_t). This leads to an augmented model of the type (Model 1) :

$$\log h_{t+1} = \alpha + \beta X_t + \delta \log h_t + \sigma_v v_{t+1} \tag{3.2}$$

The addition of exogenous variables to the volatility equation is a technically simple but potentially useful extension of the SV model. Other variables of interest can be incorporated in the SV equation, such as implied volatility or the number of non-trading days between observations.

In (3.2), RV is treated as an exogenous variable. Competing measures can be introduced simply as multiple right-hand side variables. As mentioned previously, it can suffer from the error-in-variable problem. Alternatively we can characterize the link between RV_t and h_t as :

$$\begin{aligned} \log h_{t+1} &= \alpha + \delta \log h_t + \sigma_v v_{t+1} \\ \log RV_t &= \beta_0 + \beta_1 \log h_t + \sigma_\eta \eta_t, \end{aligned} \tag{3.3}$$

This specification, denoted Model 2, explicitly reflects the fact that RV_t is a noisy estimate of $\log h_t$, possibly allowing for its error η_t to be correlated with v_{t+1}. The parameters β_0, β_1 and σ_η are informative about the link between measure and volatility. Competing volatility measures can be introduced via seemingly unrelated measurement equations of possible correlated measure errors.

We provide Bayesian MCMC algorithms to implement these three models, deriving the exact optimal smoothers and filters for volatility. We perform posterior analysis, smoothing and prediction on (3.1), (3.2) and (3.3). We conduct simulations to document what reduction in volatility uncertainty can be expected by incorporating RV_t through (3.2) and (3.3). Using the root-mean-squared error of the posterior mean, we show that, for simulated data, RV measures improve out-of-sample volatility forecasts. We also find that (3.3) exploits the information content of the RV estimators better, providing improved in sample and out of sample estimation of volatility than using specification (3.2).

We apply the models on four daily country index returns and three foreign exchange currencies. Our data covers the period between 2006-2009 which allows us to evaluate the information of RV estimators during periods of stress, when getting volatility estimations right or wrong becomes very important. In addition, we also evaluate the information content of implied volatility and compare it to the results obtained using the realized volatility estimator. We use odds ratios to compare the models. The odds are calculated for the entire period, and sequentially using MCMC methods and a particle filter. We show that the odds ratios are in

favor of the model with realized volatility compared to the benchmark model. When introducing implied volatility to study $S\&P500$'s volatility, we find strong evidence that the implied volatility index contains all the information in the RV estimator and more.

Approaches similar to ours, combining SV models with intra-day variance estimators, are those of Koopman, Jungbackera, and Hol (2005), Takahashi, Omori, and Watanabe (2009), Brandt and Jones (2005). Koopman, Jungbackera, and Hol (2005) consider a model similar to (3.2) and compare it with other models to evaluate the one day ahead volatility forecasting of the $S\&P100$. Realized volatility is taken as the proxy for assessing the relative forecast accuracy of models. Takahashi, Omori, and Watanabe (2009) consider a model similar to (3.3). They propose a Bayesian MCMC approach to estimate it and apply it to study the Tokyo stock price index volatility. They focus on inference and find that the effect of non trading hours is more important than that of the microstructure noise and they show that considering asymmetry between the return equation and the latent volatility equation improves the fit of the model. Brandt and Jones (2005) proposed a MCMC algorithm to estimate SV models using a daily range estimator. They show, by simulations, that using the daily range estimator improves the fit of the model. However, their results are only in-sample and do not include RV estimators.

This chapter is organized as follows : Section 3.2 explains the MCMC algorithms used to estimate the models. Section 3.3 presents the performance evaluation of the models used. Section 3.4 describes the data and presents the empirical results. Finally, Section 3.5 summarizes our main conclusions and future extensions.

3.2 MCMC algorithms

In this section we describe the MCMC algorithms used to estimate Model 1 and Model 2 and the choice of priors.

3.2.1 Model 1

Model 1 refers to specification (3.2). We extend the MCMC algorithm proposed by Jacquier, Polson, and Rossi (2004) (JPR) to include exogenous variables. We review the algorithm for a general case with fat tails and correlated errors below. The general model specification is given by :

$$\begin{aligned} r_t &= \sqrt{h_t \lambda_t}\epsilon_t \\ \log h_{t+1} &= \alpha + \beta X_t + \delta \log h_t + \sigma_v v_{t+1}, \quad u_{t+1} = \sigma_v v_{t+1} \\ \frac{\nu}{\lambda} &\sim \chi^2_\nu \\ (\epsilon_t, u_{t+1}) &\sim N(0, \Sigma), \quad corr(v_{t+1}, \epsilon_t) = \rho \end{aligned}$$

Yu (2005) shows that the timing of the variables in the original JPR algorithm for correlated errors implied a non zero expected return, which makes the interpretation of the correlation parameter difficult. The timing problem is related to the fact that JPR specifies $corr(v_t, \epsilon_t) = \rho$ instead of using $corr(v_{t+1}, \epsilon_t) = \rho$. In this paper we adjust the algorithm to work with the correct specification. The strategy for drawing the parameters and the volatilities involves cycling through the following steps :

1. Rewrite the model as $r_t^* = r_t/\sqrt{\lambda} = \sqrt{h}\epsilon_t$, $\epsilon_t \sim N(0,1)$.

2. Draw $\omega = (\alpha, \beta, \delta)$: $p(\omega|r^*, \sigma_v, h, X) = p(\omega|\sigma_v, h, X)$ is the posterior from a linear regression. Using standard analytical results, direct draws can be made.

3. Draw (ρ, σ_v) :

   – Transform (ρ, σ_v) to (Φ, Ω) :

$$\Sigma^* = \begin{pmatrix} 1 & \Phi \\ \Phi & \Omega + \Phi^2 \end{pmatrix}$$

$$\begin{aligned}\Omega &= \sigma_v^2(1-\rho^2)\\ \Phi &= \sigma_v\rho\end{aligned}$$

– Draw Ω, Φ :

$$\begin{aligned}p(\Phi|\Omega,\omega,h) &\sim N(\bar{\Phi}, \Omega/(a_{11}+p_0))\\ p(\Omega|\omega,h,r) &\sim IG(v_0+T-1, v_0t_0^2+a_{22,1})\end{aligned}$$

where $\bar{\Phi} = (a_{11}\Phi + p_0\Phi_0)/(a_{11}+p_0)$, a_{ij} is the ij element of $A = \sum a_t {a_t}^\top$, $a_t = (\epsilon_t, u_{t+1})^\top$.

4. Draw h : $p(h_t|h_{t-1}, h_{t+1}, r^*, \Sigma^*) = p(h_t|.)$

$$p(h_t|.) \propto \frac{1}{h_t} exp\left(-\frac{1}{2} tr({\Sigma^*}^{-1} a_{t-1} {a_{t-1}}^\top)\right) \frac{1}{\sqrt{h_t}} exp\left(-\frac{1}{2} tr({\Sigma^*}^{-1} a_t {a_t}^\top)\right)$$

$$p(h_t|.) \propto \frac{1}{h_t^{(1.5)}} exp\left(\frac{-{r_t^*}^2}{2h_t}(1+\frac{\Phi^2}{\Omega}) - \frac{(u_{t+1}^2+u_t^2)}{2\Omega} + \frac{\Phi}{\Omega}\left(\frac{r_{t-1}^*}{\sqrt{h_{t-1}}}u_t + \frac{r_t^*}{\sqrt{h_t}}u_{t+1}\right)\right)$$

$$p(h_t|.) \propto \frac{1}{h_t^{\left(1.5-\frac{\Phi r_{t-1}^*}{\Omega\sqrt{h_{t-1}}}\right)}} exp\left(\frac{-{r_t^*}^2}{2h_t}(1+\frac{\Phi^2}{\Omega}) - \frac{(\log h_t - \mu_t)^2}{2\Omega/(1+\delta^2)} + \frac{\Phi}{\Omega}\left(\frac{r_t^*}{\sqrt{h_t}}u_{t+1}\right)\right)$$

where :

$$\mu_t = \frac{(\alpha(1-\delta) - \beta(\delta X_t - X_{t-1}) + \delta(\log h_{t+1} + \log h_{t-1}))}{1+\delta^2}$$

An accept/reject and Metropolis-Hastings are used to draw from the posterior

of h_t. The proposal density (q) is an inverse gamma distribution :

$$q(h_t|.) \sim IG(\phi_t, \theta_t^*) \propto \frac{1}{h_t^{(\phi_t+1)}} e^{(-\theta_t^*/h_t)}$$

where the parameters of the proposal are :

$$\begin{aligned}
\phi_t &= -\frac{\Phi y_{t-1}}{\Omega\sqrt{h_{t-1}}} + 0.5 + \phi_{LN} = -\frac{\Phi y_{t-1}}{\Omega\sqrt{h_{t-1}}} + 0.5 + \frac{1 - 2e^{\frac{\Omega}{\sqrt{1+\delta^2}}}}{1 - e^{\frac{\Omega}{\sqrt{1+\delta^2}}}} \\
\theta_t^* &= \theta_t - s\Phi r_t^*/\Omega = \frac{r_t^{*2}}{2}(1 + \frac{\Phi^2}{\Omega}) + \theta_{LN} - s\Phi r_t^*/\Omega \\
\theta_{LN} &= (\phi_{LN} - 1)e^{(\mu_t + 0.5\frac{\Omega}{1+\delta^2})}
\end{aligned}$$

s is the slope between two points around the mode of $u_{t+1}/\sqrt{h_t} = -\delta(\log h_t - \frac{(\log h_{t+1} - \alpha - \beta X_t)}{\delta})/\sqrt{h_t}$. The presence of the term $s\Phi r_t^*/\Omega$ in θ_t^* seeks to take account of the effect of the term $u_{t+1}/\sqrt{h_t}$ in the posterior distribution of h_t into the proposal distribution when ρ is different from zero. (See Jacquier, Polson, and Rossi (2004)).

5. Draw λ :

$$p(\lambda_t|r_t, h_t, \nu) \sim IG\left(\frac{\nu+1}{2}, \frac{2}{(r_t^2/h_t) + \nu}\right)$$

6. Draw ν : $(r_t|h_t, \nu) \sim t(\nu)$, so ν is a multinomial distribution (discrete) with probability mass proportional to the product of t distribution ordinates.

The introduction of an exogenous variable in the log volatility equation results in a trivial modification of the original JPR algorithm. The steps affected are (2) and (4). In step (2) we have one additional parameter, β, to estimate compared to the basic case and in step (4) the parameter μ_t of the conditional posterior of h_t takes into account the effect of the exogenous variable in the posterior mean of h_t.

This methodology describes a posterior simulator that draws volatility proposals one observation at a time. Kim, Shephard, and Chib (1998) (KSC) propose a multi-move sampler that draws volatilities and parameters as a block and as a consequence reduce autocorrelation of the draws. They use a discrete mixture of normal distributions as an approximation to the log square of the return shock. Then they employ a data augmentation technique to draw the series of log volatilities directly from their conditional distribution given parameters and discrete latent mixture component indicators. In Section 3, we compare both methodologies and show that results are basically identical.

3.2.2 Model 2

In Model 2, we incorporate a separate measurement equation to model the error of the Realized Volatility estimator (3.3). The general model is :

$$\begin{aligned} r_t &= \sqrt{h_t \lambda_t}\epsilon_t \\ \log RV_t &= \beta_0 + \beta_1 \log h_t + \sigma_\eta \eta_t, \quad z_t = \sigma_\eta \eta_t \\ \log h_{t+1} &= \alpha + \delta \log h_t + \sigma_v v_{t+1}, \quad u_{t+1} = \sigma_v v_{t+1} \\ (\epsilon_t, z_t, u_{t+1}) &\sim N(0, \Sigma) \end{aligned}$$

$$\Sigma = \begin{pmatrix} 1 & 0 & \rho_1 \sigma_v \\ 0 & \sigma_\eta^2 & 0 \\ \rho_1 \sigma_v & 0 & \sigma_v^2 \end{pmatrix}$$

To draw the parameters and the volatilities we follow the same steps as in Model 1, but modify step (2) and step (4) :

– In step (2), we now have to draw the parameters of the log volatility equation and those of the log RV equation, $\omega = (\alpha, \delta, \sigma_v, \beta_0, \beta_1, \sigma_\eta)$. With $corr(\eta_t, v_t) = 0$, we can estimate the parameters of the two regressions independently using direct draws from their respective conditional posterior distributions : $p(\alpha, \delta | h, \sigma_v)$

and $p(\beta_0, \beta_1|h, RV, \sigma_\eta)$, $p(\sigma_\eta|\beta_0, \beta_1, h, RV)$. To draw σ_v we use the same transformation given in step (3) of Model 1 algorithm.

– In step (4), we now have that Σ is a 3×3 matrix. Σ^{-1} is given by :

$$\Sigma^{-1} = \begin{pmatrix} S_{11} & S_{12} & S_{13} \\ S_{12} & S_{22} & S_{23} \\ S_{13} & S_{23} & S_{33} \end{pmatrix} = \begin{pmatrix} \frac{1}{(1-\rho_1^2)} & 0 & -\frac{\rho_1}{(1-\rho_1^2)\sigma_v} \\ 0 & \frac{1}{\sigma_\eta^2} & 0 \\ -\frac{\rho_1}{(1-\rho_1^2)\sigma_v} & 0 & \frac{1}{(1-\rho_1^2)\sigma_h^2} \end{pmatrix}$$

The posterior of volatilities is $p(h_t|h_{t+1}, h_{t-1}, r_t, \log RV_t, \Sigma, \omega_{RV}, \omega_v) = p(h_t|.)$

$$p(h_t|.) \propto h_t^{-1.5} exp\left(-\frac{1}{2}\left(tr(\Sigma^{-1}a_t a_t^\top) + tr(\Sigma^{-1}a_{t-1}a_{t-1}^\top\right)\right)$$

where $a_t = (\epsilon_t, z_t, u_{t+1})$.

Expanding the terms we obtain :

$$\propto \frac{1}{h_t^{(1.5+S_{13}\epsilon_{t-1})}} \exp\left[-\frac{1}{2}S_{11}\frac{r_t^2}{h_t} - S_{13}\epsilon_t u_{t+1} - \frac{1}{2}\frac{(\log h_t - \mu_t^*)^2}{s^{*2}}\right]$$

where the terms μ_t^* and s^{*2} are given by :

$$\mu_t^* = \frac{\mu_{1t}s_1^{-2} + \mu_{2t}s_2^{-2}}{s_1^{-2} + s_2^{-2}}, \quad s^{*2} = \frac{1}{s_1^{-2} + s_2^{-2}}$$

where :

$$\mu_{2t} = \frac{\log RV_t - \beta_0}{\beta_1}, \quad \mu_{1t} = \frac{\alpha(1-\delta) + \delta(\log h_{t+1} + \log h_{t-1})}{(1+\delta^2)}$$

$$s_2^2 = \frac{1}{S_{22}\beta_1^2}, \quad s_1^2 = \frac{1}{S_{33}(1+\delta^2)}$$

The proposal is an inverse gamma, as in Model 1 :

$$q(h_t|.) \sim IG(\phi_t, \theta_t^*) \propto \frac{1}{h_t^{(\phi_t+1)}} e^{(-\theta_t^*/h_t)}$$

where :

$$\begin{aligned}
\phi_t &= S_{13}\frac{y_{t-1}}{\sqrt{h_{t-1}}} + 0.5 + \phi_{LN} = S_{13}\frac{y_{t-1}}{\sqrt{h_{t-1}}} + 0.5 + \frac{1 - 2e^{\frac{\Omega}{\sqrt{1+\delta^2}}}}{1 - e^{\frac{\Omega}{\sqrt{1+\delta^2}}}} \\
\theta_t^* &= \theta_t - slS_{13}r_t^* = S_{11}\frac{r_t^{*2}}{2} + (\phi_{LN} - 1)e^{(\mu_t^* + 0.5s^{*2})} - sS_{13}r_t^*
\end{aligned}$$

s is the slope between two points around the mode of $u_{t+1}/\sqrt{h_t} = -\delta(\log h_t - \frac{(\log h_{t+1} - \alpha)}{\delta})/\sqrt{h_t}$.

Model 2 can be further extended to include additional proxies of volatility. In Appendix 3.6.1 we present the algorithm for a bivariate Model 2.

3.2.3 Priors

We describe the priors that we use for the parameters of Model 1 and 2. We use a standard Normal-Gamma prior for $p(\omega_h)$ and $p(\omega_{RV})$, a conjugate Inverse Gamma for $p(\lambda_t|\nu)$, a uniform discrete prior for ν, a normal prior for $p(\Phi)$ and an inverse gamma for $p(\Omega)$:

- $\alpha|\sigma_v^2 \sim N(\tilde{\alpha}, C_{11}\sigma_v^2)$
- $\beta|\sigma_v^2 \sim N(\tilde{\beta}, C_{22}\sigma_v^2)$
- $\delta|\sigma_v^2 \sim N(\tilde{\delta}, C_{33}\sigma_v^2)$
- $\beta_0|\sigma_\eta^2 \sim N(\tilde{\beta}_0, V_{11}\sigma_\eta^2)$
- $\beta_1|\sigma_\eta^2 \sim N(\tilde{\beta}_1, V_{22}\sigma_\eta^2)$
- $\sigma_\eta \sim IG(\frac{x_0}{2}, \frac{2}{x_0 q_0^2})$
- $\Phi|\Omega \sim N(\tilde{\Phi}, \Omega/p_0)$

- $\Omega \sim IG(\frac{\tilde{v}}{2}, \frac{2}{\tilde{v}t_0^2})$
- $\lambda_t|\nu \sim IG(\frac{\nu}{2}, \frac{2}{\nu}) \sim v/\chi^2_{(\nu)}$
- $\nu \sim U[\underline{\nu}, \bar{\nu}]$

The set of hyper-parameters :
$\left\{\tilde{\alpha}, \tilde{\beta}, \tilde{\delta}, \tilde{\beta}_0, \tilde{\beta}_1, C_{11}, C_{22}, C_{33}, V_{11}, V_{22}, x_0, q_0^2, \underline{\nu}, \bar{\nu}, \tilde{\Phi}, p_0, \tilde{v}, t_0^2\right\}$
is set to :
$\{0, 0, 0, 0, 0, 100, 100, 100, 100, 100, 10, 0.01, 3, 40, 0, 2, 10, 0.01\}$

3.3 Performance evaluation

In this section, we conduct sampling experiments to assess the performance of the Bayesian inference on volatilities and parameters. Specifically, we document what reduction in volatility uncertainty can be expected by incorporating RV_t to the basic SV model using Model 1 and Model 2. For example, we want to know at what rate the smoothing performance improves as the intra-day frequency used to compute the RV estimator increases, when prices are measured with and without errors.

We simulate daily volatility (h_t) from the log AR(1) SV model (3.1) with parameters : $\alpha = -0.37, \delta = 0.96, \sigma_v = 0.21$, which implies an annualized volatility of 17%. We assume constant intra-day volatility and simulate intra-day returns with and without noise. Noiseless returns are generated as :

$$\begin{aligned} r_{t,j}^{noiseless} &= p_{t,j}^* - p_{t,j-1}^* \\ p_{t,j}^* &= p_{t,j-1}^* + \sqrt{h_{t,j}}\xi_{t,j}, \quad \xi_{t,j} \sim N(0,1), \quad j = 1, \ldots, m \end{aligned}$$

and noisy returns are generated assuming an i.i.d. noise process :

$$\begin{aligned} r_{t,j}^{noisy} &= p_{t,j} - p_{t,j-1} \\ p_{t,j} &= p_{t,j}^{*} + \epsilon_{t,j}, \quad \epsilon_{t,j} \sim i.i.d.N(0, \sigma_{\epsilon}^{2}), \quad j = 1, \ldots, m \end{aligned} \tag{3.4}$$

where $h_{t,j} = h_t/m$ is the constant intra-day volatility, m is the number of intra-day observations, p^* is the logarithm of the noiseless price (true price), p is the logarithm of the noisy price (observed price) and σ_ϵ is the volatility of the i.i.d. noise ϵ. We set $\sigma_\epsilon = 0.08\%$ which is within the the range of values found in existing studies of empirical market microstructure and it implies that around 40% of the variance of the 5-minute return is attributable to microstructure noise [13] [14].

We simulate 500 samples of 1500 daily observation with 780 observations per day ($m = 780$), which corresponds to 30 second data over a 6.5-hour trading day. We construct the following estimators : RV 30sec, uses the highest frequency available (30 seconds) ; RV 5min, uses five minute returns ; and TSRV 5min, is a two-scale estimator based on sub-sampling to mitigate the effect of microstructure noise. TSRV is computed as follows. Every day, the intra-day returns are partitioned into K sub-samples of 5 minute returns and a RV estimator is calculated for each sub-sample. Then, the TSRV is calculated as :

$$TSRV_t = \frac{1}{K}\sum_{k=1}^{K} RV_t^k - \frac{\bar{m}}{m} RV_t^{all}$$

13. Ait-Sahalia, Mykland, and Zhang (2005) review different empirical studies and document that the value of σ_ϵ varies by type of security, market and time period. They found different values in the range of $(0, 1.8\%)$.

14. Using Ait-Sahalia, Mykland, and Zhang (2005) definition, the proportion of the total 5-minute return variance that is market-microstructure induced is :

$$\pi = \frac{2\sigma_\epsilon^2}{\sigma_d^2 \Delta + 2\sigma_\epsilon^2}$$

where σ_d^2 is the unconditional daily variance of the returns and $\Delta = 5/390$ considering a day of 6.5 hours.

where RV^{all} is the RV estimator calculated with all available intra-day returns, m is the the number of observations per day and $\bar{m} = m/K$ is the average size of the sub-samples.

Table 3.1 reports the magnitude of the bias inherent in each estimator when prices are measured with error. The column «Average Mean» presents the average of the estimator over the 500 samples, the column «Average Bias» shows the difference between the average true variance and the average RV estimator, and the column «Theoretical Bias» is the expected value of the bias for each estimator. The expected bias of an RV estimator is $2m\sigma\epsilon^2$, where m is the number of intra-day returns used to construct the estimator. So, the higher the frequency used to compute the RV estimator, the bigger the bias. The bias in sampling every 5 minutes is 9 times smaller than when we use 30 seconds.

TABLE 3.1 – Properties of simulated RV measures

	Average Mean	Average Bias	Theoretical Bias
True variance	0.0001278		
RV 30 sec	0.0011262	0.0009983	0.0009984
RV 5 min	0.0002278	0.0001000	0.0000998
TSRV 5 min	0.0001264	-0.0000014	0.0000000

In addition to these RV estimators we also consider the daily Range, which is another popular variance estimator proposed by Parkinson (1980) based on the difference between the highest and lowest logarithm price of the day :

$$Range_t = 0.361 \left[max(p_{t,j}) - min(p_{t,j}) \right]^2 \tag{3.5}$$

In the following sections we report the sampling distribution of the posterior means of the parameters and (in-sample) volatilities, and the predictive means of the (out-of-sample) volatilities. We consider the cases when prices are observed with and without error. For each sample, we generate 31,000 draws from the MCMC algorithm and discard the first 1,000 draws. We evaluate sampling performance through the root mean squared error (RMSE) and the percentage mean absolute

error (%MAE) of the parameters and (in-sample) volatilities. There is no reason to believe that the posterior mean is an unbiased estimator of the true parameter in the sampling sense. However, we can still use this approach to compare competing estimators of the same simulated data. The in-sample inference is based on the first 1,495 daily observations of the 500 samples (747,500 daily variances). The predictive inference is carried out as follows. We generate five day ahead volatility forecasts, for each sample estimated with 1,495 observations, that is a total of 2,500 daily forecasts, conditional on the information up to $T = 1,495$. We evaluate the performance of each of the 5 day-ahead forecasts as well as the forecast for one week volatility, $\sqrt{h_w} = \sqrt{h_{T+1} + ... + h_{T+5}}$.

3.3.1 Performance with Model 1

We start our performance evaluation using Model 1 (M1). We use the basic form which does not include correlation or fat tails :

$$\begin{aligned} r_t &= \sqrt{h_t \lambda_t} \epsilon_t \\ \log h_{t+1} &= \alpha + \beta \log RV_t + \delta \log h_t + \sigma_v v_{t+1} \end{aligned}$$

We compare the results of Model 1 to those obtained from the standard SV model (M0) in which we do not incorporate any explanatory variable in the log volatility equation. We analyze the results using the noiseless and noisy returns in order to examine the impact of microstructure noise in the efficacy of the RV estimators to provide useful information to the estimation of the true volatility.

Performance with no microstructure noise

We use the noiseless returns to estimate M1 model. If useful to the investor, the RV estimate used as an exogenous variable in the volatility equation, should result in posterior densities of volatilities closer to those simulated without the use of RV. We also expect to have an estimation of $\beta + \delta$ close to 0.96. Arguably, as RV complements the information from the the naive reduced-form AR(1), the

lagged value of h_t becomes less relevant. Table 3.2 presents the sample average of the posterior means, 5th and 95th percentile, as well as their RMSE across the 500 sample for each of the 4 cases considered : no RV, RV 30sec, RV 5 min and the Range.

TABLE 3.2 – Posterior parameter distribution - Model 1 on noiseless returns

Models	α	β	δ	σ_v	$\beta + \delta$
True value	-0.37		0.96	0.21	
M0					
Average	-0.479		0.948	0.229	
5%	-0.744		0.92	0.176	
95%	-0.268		0.971	0.291	
RMSE	0.192		0.021	0.041	
M1-RV 30sec - true prices					
Average	-0.321	0.819	0.146	0.187	0.965
5%	-1.031	0.633	0.017	0.066	0.888
95%	0.364	0.972	0.334	0.33	1.039
RMSE	0.435	0.16	0.816	0.065	0.047
M1-RV 5min - true prices					
Average	-0.524	0.683	0.259	0.195	0.942
5%	-1.09	0.509	0.08	0.077	0.881
95%	-0.097	0.852	0.446	0.327	0.988
RMSE	0.308	0.29	0.707	0.067	0.034
M1-Range - true prices					
Average	-0.461	0.282	0.66	0.155	0.942
5%	-0.864	0.212	0.561	0.07	0.898
95%	-0.138	0.358	0.747	0.255	0.977
RMSE	0.242	0.679	0.305	0.078	0.03

In the standard specification, M0, we observe that the posterior means of the parameters are very close to the simulated value. When we incorporate the different estimators as exogenous variables in the volatility equation, the results are in accordance with our expectations. The RV estimator using all intra-day prices

contains the most information about the true value of volatility, which is reflected by the lowest posterior means for the AR(1) parameter, δ, and the highest value for β, while their sum remains close to 0.96. On the other extreme, the range estimator is the least informative estimator, β is equal to 0.30 compared with the 0.70 and 0.82 for the RV 5min and RV 30sec, respectively.

Table 3.3 documents the RMSE and $\%MAE$ of the in-sample posterior means of volatility, $\sqrt{h_t}$. The presence of a variance estimator in the volatility equation improves the estimation of volatility. In all the cases the RMSE and the $\%MAE$ are below the ones obtained with M0 model. In particular, the RV 30sec estimator provides the best fit for volatility, with a 42% RMSE gain over the standard case of no RV estimators. Indeed, taking advantage of the highest frequency available when prices are measured without noise results in a massive improvement of the estimation of volatility.

TABLE 3.3 – In sample properties of $\sqrt{h_t}$ - Model 1 on noiseless returns

	RMSE	$\%MAE$
M0	2.19	16.27
M1-RV 30sec - true prices	1.27	8.98
M1-RV 5min - true prices	1.42	10.16
M1-Range - true prices	2.06	15.28

We now turn to the performance of the estimators for estimating out-of-sample volatility. For each draw of the parameters and the in-sample volatilities, the one-day ahead forecast is obtained by using the log volatility equation in Model 1. Its posterior mean is computed as :

$$\begin{aligned} E_T(\log h_{T+1}) &= \frac{1}{S}\sum_{i=1}^{S} \log h_{T+1|T}^{(i)}, \\ \log h_{T+1|T}^{(i)} &= \alpha^{(i)} + \beta^{(i)} \log RV_T + \delta^{(i)} \log h_T^{(i)} + \sigma_v^{(i)} v_T \end{aligned}$$

where S is the number of MCMC draws used to construct the posterior mean.

For multi-period forecasts we can forecast independently the RV estimators and

plug them into the log volatility equation. This procedure requires to estimate a model for each RV estimator. Alternatively, we use the same AR(1) specification as in the basic SV model, which does not include exogenous variables in the latent volatility equation and for the slope parameter we use the sum of the in-sample estimation of β and δ. This approach assumes that the proxy estimators are unbiased estimators of the latent volatility. Thus, multi-period forecast are obtained with :

$$\begin{aligned} E_T(\log h_{T+k}) &= \frac{1}{S}\sum_{i=1}^{S} \log h_{T+k|T}^{(i)}, \\ \log h_{T+k|T}^{(i)} &= \alpha^{(i)} + (\beta^{(i)} + \delta^{(i)}) \log h_{T+k-1|T}^{(i)} + \sigma_v^{(i)} v_{T+k-1}, \quad k = 2, ..., 5 \end{aligned}$$

The results are showed in Table 3.4, RMSE values are multiplied by 10^3.

TABLE 3.4 – Out of sample evaluation of $\sqrt{h_t}$ - Model 1 on noiseless returns

Models	All	$T+1$	$T+2$	$T+3$	$T+4$	$T+5$	1 week
RMSE							
M1-RV 30 sec - true prices	2.00	1.26	1.70	2.01	2.30	2.47	3.89
M1-RV 5 min - true prices	2.12	1.46	1.85	2.13	2.41	2.58	4.22
M1-Range - true prices	2.67	2.11	2.36	2.59	2.95	3.18	5.46
%**MAE**							
M1-RV 30 sec - true prices	14.87	9.41	12.59	15.46	17.90	19.00	13.19
M1-RV 5 min - true prices	16.17	10.99	13.99	16.55	19.02	20.28	14.33
M1-Range - true prices	23.59	17.09	20.12	23.64	27.28	29.82	21.69

We observe the same pattern as for in-sample volatility. Namely, the model with RV 30sec performs the best through all forecasted horizons. However, note how the accuracy of all out-of-sample forecasts decreases as the forecast horizon increases.

The performance evaluation considered so far has been based on the estimation of Model 1 using a single-move sampler algorithm. Kim, Shephard, and Chib (1998), KSC, propose an alternative algorithm based on a multi-move sampler for the estimation of the basic SV model. They model $\log \epsilon_t^2$, as a discrete mixture of

normals, augmenting the state space accordingly, which allows them to draw directly from the multivariate distribution of h. While the computational burden at each draw is higher, the resulting draws are markedly less autocorrelated, notably for σ_v, than for the single-move sampler. In Appendix 3.6.2 we compare the results of JPR and KSC using simulated and real data.

We found that SV models estimated by single-move or multi-move MCMC can deliver, period after period, posterior distributions of smoothed volatilities with a very satisfactory degree of precision. The high autocorrelation that the single-move algorithm presents does not indicate that the algorithm does not converge. It is a sign that the algorithm may accumulate information at a slower rate than it would do with a lower autocorrelation of draws. We will take the usual precautions to assess the number of draws needed to obtain a desired precision for the MC estimate of, say, the posterior mean. This is done simply by computing standard errors robust to autocorrelation (see Geweke (1992)). In fact, low autocorrelation may not even be a sign that an algorithm has converged ; it may be stuck in a region of the parameter space while exhibiting low autocorrelation in that region. With lower autocorrelation, a given desired precision for Monte Carlo estimates requires fewer draws, but this has to be weighted by the required CPU time per draw.

Performance with microstructure noise

We now evaluate the contribution of the competing RV measures to the estimation of volatility when prices are measured with error. We add the TSRV 5min estimator to the three RV measures considered above. These four estimators now use the noisy prices. We also include the RV 30sec estimator computed with the noiseless prices to contrast with the case in which the estimator can quasi perfectly extract volatility information from the intra-day prices.

Table 3.5 presents the sampling performance of the posterior means of the parameters. First, we note that price noise does not markedly diminish the sampling performance of the basic model (M0) which does not use intra-day information. The

TABLE 3.5 – Posterior parameter distribution - Model 1 on noisy returns

Models	α	β	δ	σ_v	$\beta + \delta$
True value	-0.37		0.96	0.21	
M0					
Average	-0.48		0.95	0.23	
5%	-0.76		0.92	0.17	
95%	-0.27		0.97	0.29	
RMSE	0.19		0.02	0.04	
M1-RV 30sec - true prices					
Average	-0.32	0.82	0.14	0.18	0.96
5%	-1.03	0.64	0.02	0.07	0.89
95%	0.36	0.97	0.33	0.32	1.04
RMSE	0.43	0.16	0.82	0.07	0.05
M1-RV 30sec - noisy prices					
Average	10.39	1.92	0.72	0.28	2.63
5%	7.06	1.32	0.61	0.20	2.11
95%	14.11	2.58	0.81	0.37	3.21
RMSE	11.00	1.04	0.25	0.09	1.71
M1-RV 5min - noisy prices					
Average	2.43	0.82	0.51	0.24	1.33
5%	1.56	0.60	0.36	0.12	1.22
95%	3.39	1.06	0.65	0.35	1.45
RMSE	2.86	0.20	0.46	0.07	0.38
M1-TSRV 5min - noisy prices					
Average	-0.67	0.56	0.36	0.20	0.92
5%	-1.30	0.42	0.19	0.09	0.86
95%	-0.12	0.71	0.52	0.32	0.98
RMSE	0.48	0.41	0.61	0.07	0.05
M1-Range - noisy prices					
Average	-0.22	0.31	0.66	0.17	0.98
5%	-0.52	0.23	0.56	0.08	0.94
95%	-0.03	0.40	0.75	0.27	1.00
RMSE	0.20	0.65	0.30	0.07	0.02

posterior distributions of parameters are the same as in the case of no noise (Table 3.2). However, performance deteriorates markedly when we use the RV estimators on intra-day noisy returns ; the posterior means of the parameters are far from the true ones. We also observe that sampling less frequently, sparse estimator RV 5min, or using an estimator that explicitly accounts for the noise, as with the TSRV, improves posterior performance considerably. The range estimator also appears to be robust to price noise ; the posterior parameters distributions are very similar to those in Table 3.2 for the case of no noise.

We now compare the contribution of these estimators to the estimation of volatility. Table 3.6 shows the results of the in-sample posterior means of $\sqrt{h_t}$. Again, the performance of volatility estimation with M0 model is not affected by price noise. Both RMSE and $\%MAE$ are almost the same as when prices are measured without error (Table 3.3). In contrast, in the case of RV 30sec with noisy prices, the ability to estimate volatility has deteriorated considerably, the RMSE and $\%MAE$ of the posterior means of volatility almost double with respect to the case with no price noise. The sparse estimator is also affected but not as drastically. The M1-TSRV model now provides the best in sample performance for volatility estimation, RMSE and $\%MAE$ are the lowest among all estimators that use noisy prices. The volatility performance obtained with the M1-Range model does not change much. However, it is dominated by all of the M1-RV cases.

TABLE 3.6 – In sample properties of $\sqrt{h_t}$ - Model 1 on noisy returns

	RMSE	$\%MAE$
M0	2.21	16.65
M1-RV 30sec - true prices	1.27	8.98
M1-RV 30sec - noisy prices	2.25	15.81
M1-RV 5min - noisy prices	1.82	13.58
M1-TSRV 5min - noisy prices	1.48	11.06
M1-Range - noisy prices	2.10	16.10

To see how the noise in prices affects the quality of the information contributed by the RV estimators, Figure 3.1 plots the true annualized latent volatility (hori-

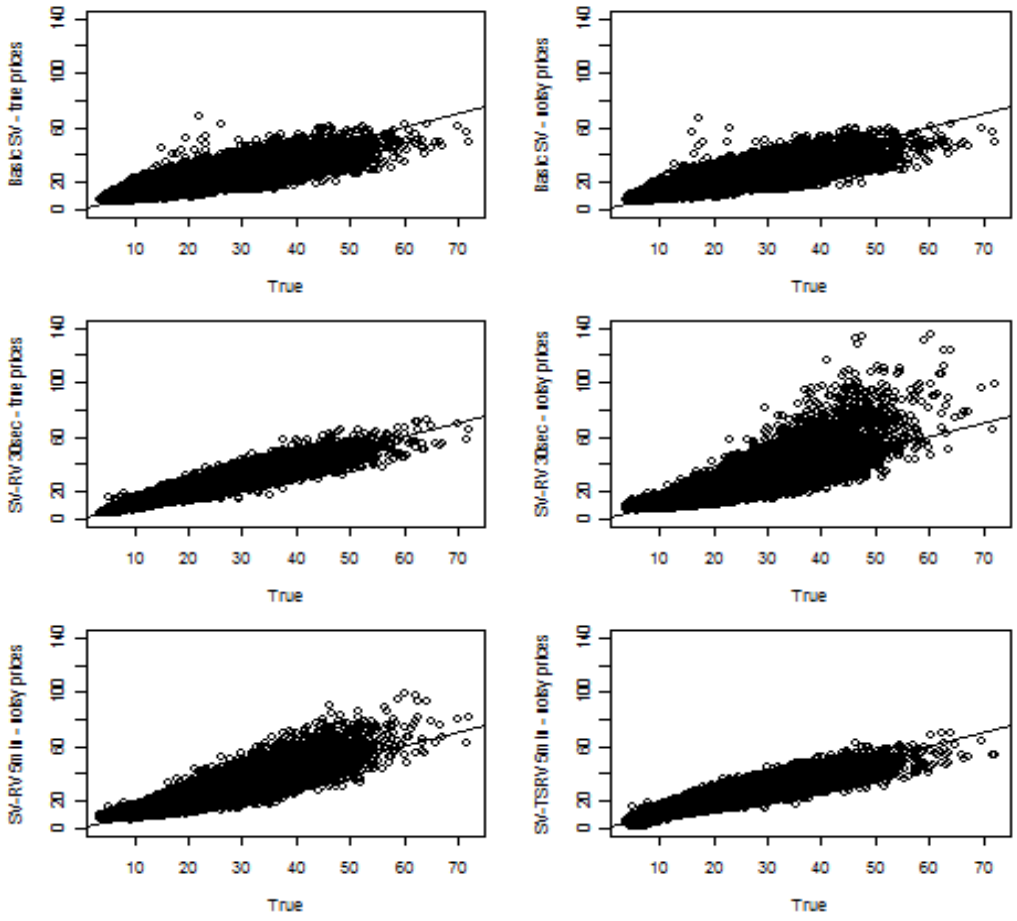
zontal axis) against the in-sample posterior means (vertical axis) obtained with the different models. We clearly observe how the posterior means are affected when one uses all available intra-day noisy prices to compute the RV. There is a non-linear effect on the estimation of volatility (top right plot). This non linearity is reduced as one samples less frequently. It disappears when one uses an estimator robust to microstructure noise such as TSRV (bottom left plot).

Table 3.7 reports on out-of-sample performance (RMSE values are multiplied by 10^3). It shows that M1-TSRV improves volatility forecasts at all the forecasting horizons considered (5 days). The largest improvement is for the one-day ahead forecast, for which RMSE and $\%MAE$ are reduced by a half from the basic case. As the forecasting horizon increases, the advantage of RV measures is reduced and the forecasting performance is comparable to that obtained with the M1-Range model.

TABLE 3.7 – Out of sample evaluation of $\sqrt{h_t}$ - Model 1 on noisy returns

Models	ALL	$T+1$	$T+2$	$T+3$	$T+4$	$T+5$	1 week
RMSE							
M0	3.00	2.89	2.93	2.97	3.08	3.14	6.40
M1-TSRV 5min - noisy prices	2.25	1.57	1.96	2.24	2.55	2.72	4.48
M1-Range - noisy prices	2.46	2.16	2.32	2.44	2.62	2.73	5.07
%**MAE**							
M0	25.39	24.44	24.75	25.68	26.11	25.95	24.50
M1-TSRV 5min - noisy prices	18.09	12.32	15.41	18.45	21.56	22.70	16.31
M1-Range - noisy prices	20.51	18.25	19.40	20.92	21.82	22.17	18.98

FIGURE 3.1 – True vs estimated annualized volatility



3.3.2 Performance with Model 2

We have shown how inference on volatility can be improved when using an unbiased RV estimator as an explanatory variable in the volatility equation. We now evaluate whether this benefit can be further enhanced by using a more flexible model, where the estimation error of the non parametric variance estimators (RV)

is explicitly modeled. We recall the basic specification for Model 2 (M2) :

$$
\begin{aligned}
r_t &= \sqrt{h_t \lambda_t}\epsilon_t \\
\log RV_t &= \beta_0 + \beta_1 \log h_t + \sigma_\eta \eta_t \\
\log h_{t+1} &= \alpha + \delta \log h_t + \sigma_v v_{t+1}
\end{aligned}
$$

We evaluate the different estimators for the case in which prices are measured with error. For each sample, we generate 30,000 draws from the MCMC algorithm and discard the first 1,000 draws. Model 2 allows us to directly model the link between the RV measure and the unobserved latent volatility, through parameters β_0 and β_1. Table 3.8, shows the parameter posteriors obtained for each case. The first row, labeled True value, recalls the parameters used for the simulations ; the second row, OLS value, reports the average OLS parameter estimate obtained for an AR(1) model on the logarithm of the true variance (the average value of the parameters are reported under column α, δ and σ_h), and a regression of the logarithm of the RV estimator computed with all intra-day noiseless returns on the true variance (the average value of the parameters are reported under columns β_0, β_1 and σ_{rv}). We then compute the root mean square error of the posterior means of the parameters. We notice that the average OLS parameter β_0 is 0 and β_1 is 1, consistent with the fact that we are using a very informative and unbiased estimator computed with noiseless returns.

The parameters are recovered with a good level of precision when we estimate Model 2 using the RV 30sec estimator computed from the noiseless returns. In the presence of microstructure noise, the AR(1) parameters are all similar and close to the true value. However, the posterior parameters obtained for the measurement equation are drastically different. The posterior means of β_0, β_1 and σ_{rv} reflect how the model adjusts to extract the relevant information from the different estimators. In the case of the RV estimator computed with all intra-day noisy returns we observed a large deviation of β_0 from zero and a low β_1 of only 0.14. For the sparse

TABLE 3.8 – Posterior parameter distribution - Model 2 on noisy returns

Models	α	δ	σ_h	β_0	β_1	σ_{rv}
True value	-0.37	0.96	0.21			
OLS value	-0.40	0.96	0.21	0.00	1.00	0.05
M2-RV 30sec - true prices						
Average	-0.36	0.96	0.20	0.02	1.00	0.07
5%	-0.47	0.95	0.18	-0.93	0.90	0.06
95%	-0.25	0.97	0.23	0.74	1.08	0.09
RMSE	0.06	0.01	0.01	0.36	0.04	0.02
M2-RV 30sec - noisy prices						
Average	-0.46	0.95	0.19	-5.49	0.14	0.06
5%	-0.63	0.93	0.16	-5.68	0.12	0.06
95%	-0.31	0.97	0.23	-5.29	0.16	0.06
RMSE	0.14	0.02	0.04	5.50	0.86	0.01
M2-RV 5min - noisy prices						
Average	-0.44	0.95	0.21	-3.48	0.54	0.17
5%	-0.58	0.94	0.18	-4.05	0.48	0.16
95%	-0.30	0.97	0.24	-2.91	0.60	0.18
RMSE	0.11	0.01	0.03	3.51	0.46	0.12
M2-TSRV 5min - noisy prices						
Average	-0.42	0.95	0.22	0.29	1.04	0.22
5%	-0.55	0.94	0.19	-0.59	0.94	0.20
95%	-0.29	0.97	0.24	1.16	1.13	0.23
RMSE	0.10	0.01	0.02	0.75	0.08	0.17

5min estimator, β_0 is still different from zero but the difference is smaller than in the previous case, and we observe a higher value of β_1, 0.54. Note that σ_{rv} increased from 0.06 to 0.17. The last RV estimator, TSRV, shows a significative improvement in the performance of the estimation of parameter β_1, 1.04. The deviation of β_0 from 0 is considerably reduced, while σ_{rv} increases to 0.22.

We now analyze the contribution of the competing RV measures to volatility estimation, via Model 2. Table 3.9 documents the performance of in-sample volatility estimation for Model 2 algorithm and the different RV estimators. If we compare these results with those for Model 1 in Table 3.6, we notice that both measures, RMSE and %MAE, are smaller for Model 2 for all the cases. This indicates that Model 2 exploits the information content on the realized volatility estimators better than Model 1.

TABLE 3.9 – In sample properties of $\sqrt{h_t}$ - Model 2 on noisy returns

	RMSE	$\%MAE$
M0	2.21	16.65
M2-RV 30sec - true prices	0.45	2.99
M2-RV 30sec - noisy prices	2.67	14.41
M2-RV 5min - noisy prices	1.57	9.82
M2-TSRV 5min - noisy prices	0.88	6.22

Table 3.10 presents the performance of the competing out-of-sample estimates of volatility under Model 2 (RMSE values are multiplied by 10^3). If we compare these results to the ones obtained with Model 1, shown in Table 3.7, we observe that the use of the RV estimators considerably improves the prediction of volatility as far as 5 days ahead for both models. The performance diminishes as we increase the number of days ahead in the forecast, especially for Model 1. For example, the %MAE obtained using TSRV estimator in Model 1 increased from 12.3 to 22.8, while in Model 2 it increased from 12.2 to 19, when evaluating one-day ahead versus five-day-ahead forecasted volatility.

TABLE 3.10 – Out of sample evaluation of $\sqrt{h_t}$ - Model 2 on noisy returns

Models	All	T+1	$T+2$	$T+3$	$T+4$	$T+5$	1 week
RMSE							
M0	3.00	2.89	2.93	2.97	3.08	3.14	6.40
M2-RV 30sec - true prices	1.97	1.41	1.77	1.99	2.23	2.33	3.87
M2-RV 30sec - noisy prices	2.68	2.44	2.59	2.64	2.80	2.90	5.62
M2-RV 5min - noisy prices	2.24	1.86	2.10	2.22	2.41	2.56	4.54
M2-TSRV 5min - noisy prices	2.10	1.58	1.90	2.11	2.34	2.45	4.19
%**MAE**							
M0	25.39	24.44	24.75	25.68	26.11	25.95	24.50
M2-RV 30sec - true prices	14.38	9.64	12.59	14.93	17.08	17.66	12.59
M2-RV 30sec - noisy prices	21.87	19.77	20.97	21.90	23.26	23.44	20.38
M2-RV 5min - noisy prices	18.25	15.26	16.83	18.66	19.89	20.58	16.59
M2-TSRV 5min - noisy prices	16.16	12.22	14.42	16.58	18.60	18.95	14.59

3.3.3 Performance with bivariate models

We are now interested in comparing Model 1 and Model 2 when we consider two volatility proxies using a bivariate version of each model. We use the simulated 500 samples of 1,500 observations of daily returns generated with noise. Our first volatility proxy is the TSRV 5min estimator built with the noisy intra-day returns and our second volatility proxy (X2) is built using the following equation :

$$X2_t = \gamma_0 + \gamma_1 \log h_t^* + \sigma_{x2}\epsilon_t \tag{3.6}$$

where h_t^* is the true variance and $\epsilon_t \sim N(0,1)$. We set the parameters value to be : $(\gamma_0, \gamma_1, \sigma_{x2}) = (-0.5, 0.81, 0.07)$. The choice of this particular set of values responds to the empirical results we find when using an alternative proxy estimator for volatility in Model 2, called the implied volatility. The parameters we use are similar to the posterior mean obtained when applying Model 2 on the $S\&P500$ returns and using an implied volatility estimator as the proxy variable (Section 3.4.5 give more detail about the data and the empirical findings when using an

implied volatility estimator).

The two proxies are very similar, the average correlation across the 500 samples is 0.94. In Table 3.11 we compare the average OLS parameter estimator across the 500 samples when regressing each proxy against the true log volatility variable. We observe that the TSRV-5min estimator seems to be a relatively better proxy for the true volatility as its constant parameter is zero, slope parameter is one and have less volatility than X2.

Table 3.11 – Average OLS estimation for proxies

Proxy	β_0	β_1	σ_{tsrv}	γ_0	γ_1	σ_{X2}
TSRV-5min	0.0	1.0	0.05			
X2				-0.46	0.81	0.07

We estimate bivariate Model 1 and Model 2. The posterior parameters distribution of Bivariate Model 1 are presented in Table 3.12, we include in the table the results of univariate Model 1 when using only TSRV -5min proxy for comparison.

Table 3.12 – Posterior parameter distribution - M1 bivariate

Models	α	β	θ	δ	σ_v	$\beta+\theta+\delta$
True value	-0.37			0.96	0.21	
M1 - TSRV 5min						
Average	-0.67	0.56		0.36	0.20	0.92
5%	-1.30	0.42		0.19	0.09	0.86
95%	-0.12	0.71		0.52	0.32	0.98
RMSE	0.48	0.41		0.61	0.067	0.05
M1 - TSRV 5min - X2						
Average	-0.17	0.21	0.67	0.19	0.19	1.07
5%	-0.87	-0.05	0.32	0.03	0.07	0.97
95%	0.51	0.47	1.05	0.37	0.32	1.18
RMSE	0.45	0.76	0.34	0.78	0.06	0.13

We observe that the slope parameters associated to the new proxy X2,(θ), is

higher compared the slope parameter of TSRV, (β), and the AR(1) parameter, (δ), for M1-TSRV-X2 model. Besides, the posterior mean of the sum of the three parameters corresponding to the proxies and to the volatility is greater than 1 which will make difficult to rely on the multi-period forecasts of this model if we use ($\beta + \theta + \delta$) as the AR(1) coefficient for multi-period forecasts, as the we do not have a stationary prediction model.

With respect to Bivariate Model 2, Table 3.13 presents the posterior distribution of the parameters and includes results from the univariate cases when estimating only with TSRV and only with X2. The posterior distribution of the parameters of the volatility equation are similar for the 3 cases presented and close to the true values. Also, the posterior parameter distribution of each volatility proxy equation is in line with their true values.

Now, we compare the models with respect to their ability to estimate volatility. Table 3.14 presents the in-sample fit of volatility. The estimation of in-sample volatility improves with the introduction of the X2 proxy in Model 1 and Model 2. The improvement for Model 2 is greater than for Model 1. However, we notice, that when we estimate univariate Model 2 with only X2 we obtain a very similar in-sample fit of volatility as when we use the bivariate version. These results indicate that there is no much more information from combining both highly correlated proxies and the proxy X2 seems to be a relative better proxy than the TSRV estimator.

In terms of out-of-sample volatility fit, Table 3.15 presents the out-of-sample performance of Bivariate Model 1 and Model 2, as well as their corresponding univariate version for comparison. Bivariate Model 2 outperforms bivariate Model 1 for all periods, this is mainly due to the inability to draw effective out-of-sample forecasts with bivariate Model 1. The sum of parameters implies explosive behavior. With only TSRV, Model 1 parameters are "reasonable" and its performance out-of-sample is adequate.

From these performance evaluations, we can conclude that multivariate versions of Model 2 extrapolate the information content on different volatility proxies in a

TABLE 3.13 – Posterior parameter distribution - M2 bivariate

Models	α	δ	σ_v	β_0	β_1	σ_{η_1}	γ_0	γ_1	σ_{η_2}
True value	-0.37	0.96	0.21				-0.5	0.81	0.07
OLS value	-0.4	0.96	0.21	0	1	0.05	-0.46	0.81	0.07
M2 - TSRV 5min									
Average	-0.42	0.95	0.22	0.29	1.04	0.22			
5%	-0.55	0.94	0.19	-0.59	0.94	0.20			
95%	-0.29	0.97	0.24	1.16	1.13	0.23			
RMSE	0.10	0.01	0.02	0.75	0.08	0.17			
M2 - X2									
Average	-0.38	0.96	0.20				-0.18	0.84	0.08
5%	-0.50	0.95	0.18				-0.76	0.78	0.07
95%	-0.27	0.97	0.22				0.43	0.91	0.09
RMSE	0.07	0.01	0.02				0.47	0.05	0.01
M2 - TSRV 5min - X2									
Average	-0.39	0.96	0.20	0.51	1.06	0.22	-0.23	0.84	0.08
5%	-0.50	0.95	0.18	-0.20	0.98	0.22	-0.78	0.78	0.07
95%	-0.27	0.97	0.22	1.26	1.14	0.23	0.36	0.90	0.08
RMSE	0.07	0.01	0.01	0.67	0.08	0.18	0.43	0.05	0.01

TABLE 3.14 – In sample fit $\sqrt{h_t}$: Model 1 and 2 bivariate

	RMSE	$\%MAE$
M1 - TSRV 5min	1.48	11.06
M1 - TSRV 5min - X2	1.31	9.58
M2 - TSRV 5min	0.88	6.22
M2 - X2	0.56	3.98
M2 - TSRV 5min - X2	0.52	3.73

TABLE 3.15 – Out of sample fit $\sqrt{h_t}$: Model 1 and 2 bivariate

Models	All	T+1	$T+2$	$T+3$	$T+4$	$T+5$	1 week
RMSE							
M1- TSRV 5min	2.25	1.57	1.96	2.24	2.55	2.72	4.48
M1- TSRV 5min - X2	6.09	1.35	3.83	6.11	7.58	8.63	9.87
M2- TSRV 5min	2.10	1.58	1.90	2.11	2.34	2.45	4.19
M2- X2	1.90	1.33	1.66	1.90	2.16	2.28	3.67
M2- TSRV 5min - X2	1.90	1.32	1.66	1.91	2.16	2.28	3.67
%MAE							
M1- TSRV 5min	18.09	12.32	15.41	18.45	21.56	22.70	16.31
M1- TSRV 5min - X2	46.93	10.22	30.38	52.19	66.26	75.61	36.66
M2- TSRV 5min	16.16	12.22	14.42	16.58	18.60	18.95	14.59
M2- X2	14.17	10.06	12.34	14.44	16.65	17.35	12.31
M2- TSRV 5min - X2	14.14	9.93	12.26	14.48	16.68	17.36	12.32

better way than multivariate versions of Model 1 which are reflected in a better in-sample and out-of-sample fit of volatility.

3.4 Empirical Application

We implement Model 1 and 2 on actual data to evaluate the contribution of the competing RV measures to the estimation of volatility. We are interested in analyzing the parameter posteriors across the fitted (smoothed) and predicted volatilities across models. Our analysis includes the financial crisis of 2008-2009 during which volatility was high and highly variable.

3.4.1 Data

The data consists of three daily exchange rates and four daily stock indices for the period between January 2, 2006 and March 1, 2009.

Table 3.16 – Description of the data

	No. obs	Std. Dev.	RK
Currencies			
British Pound	810	10.9	11.6
Euro	809	10.5	10.7
Japanese Yen	809	12.6	12.9
Indices			
S&P 500	794	26.7	20.4
TSE	778	26.1	19.0
Nikkei 250	743	31.3	18.3
FTSE 100	782	24.5	18.2

We use data from the "Oxford-Man Institute's realized volatility library" produced by Heber, Lunde, Shephard, and Sheppard (2009). It contains daily returns and RV measures constructed with intra-day prices. Specifically, we use their daily returns and the realized kernel (RK) estimator for each of the series analyzed in our study. These daily realized kernel estimators are computed in tick time from all available intra-day data, after cleaning [15], using the methodology of Barndorff-Nielsen, Hansen, Lunde, and Shephard (2006). RK estimators are similar to the TSRV estimator seen above, as they are robust to some microstructure noise. Table 3.16 reports the annualized returns and realized kernels, and the number of observations for each series in our application. We observe that the annualized squared returns are much higher than the annualized realized kernel for the stock indices due to the fact that the RK estimators use only intra-day prices, excluding the overnight return.

In addition to the RK estimator we consider another popular estimator called the implied volatility estimator, which uses option prices to imply the (risk-neutral) expectation of future volatility. We use this variable for the estimation of S&P 500 volatility. The implied volatility index is obtained from the Chicago Board Options Exchange Market Volatility Index (VIX), based on a highly liquid options market.

15. See Heber, Lunde, Shephard, and Sheppard (2009) for explanation of the cleaning process.

3.4.2 Odds ratio

The MCMC draws allow us to compute posterior odds ratios of the standard SV model to the SV-RV model. The posterior odds ratio is the product of the ratio of the marginal densities of the data, called Bayes factor (BF), times the prior odds ratio :

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(M_1)}{P(M_2)} \ \frac{Marg.Likelihood|M_1}{Marg.Likelihood|M_2}$$

In Model 1, the SV model is nested within the SV-RV model when $\beta = 0$, this allows us to use the Savage-Dickey density approach to compute the ratios $BF_{SV|SV-RV}$:

$$BF_{SV|SV-RV} = \frac{p(y^t|SV)}{p(y^t|SV-RV)} = \frac{p(\alpha_1 = 0|y^t, SV-RV)}{p(\alpha_1 = 0|SV-RV)}$$

$p(\alpha_1 = 0|SV - RV)$ is the ordinate of the prior and $p(\alpha_1 = 0|y^t, SV - RV)$ is the ordinate of the posterior. Here both ordinates correspond to a student-t distribution. See Jacquier, Polson, and Rossi (2004) for proofs. For the exchange rates, we use the basic specification of the models, that is, no correlation and not fat tail. We compare SV to SV-RV model. For the stock indices we use the correlation version of each model, that is, we compare SVC to SVC-RV model. Table 3.17 reports the logarithm of the posterior odds ratios : $BF_{SV|SV-RV}, BF_{SVC|SVC-RV}$ estimated for the entire period (January 2006 to March 2009). For all currencies and stock indices the odds ratios are in favor of the model with realized volatility.

Sequential estimation : Filtering

The above results are based upon smoothed volatilities. The MCMC algorithms produce $p(h_t|y^T)$ and $p(\theta|y^T)$, where $y^T \equiv (y_1, \ldots, y_T)$, and θ, h_t are the model parameters and the variance at time t. For a given sample of data, however, the econometrician may want, for each time t, the posterior density of both filtered vo-

Table 3.17 – Logarithm of posterior odds ratio

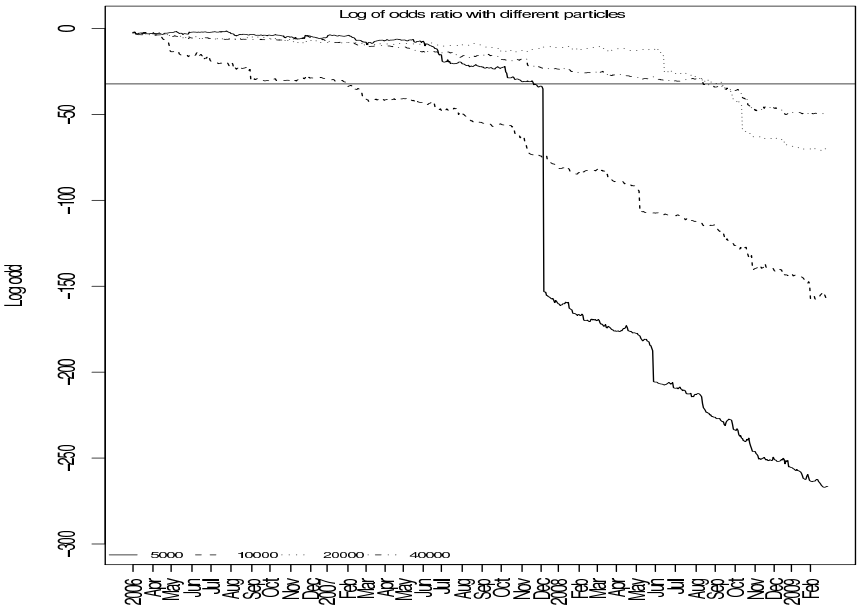
	Logarithm of odds ratio
Currencies	
British Pound	-27.57
Euro	-9.70
Japanese Yen	-3.64
Indices	
S&P 500	-38.97
TSE	-73.31
Nikkei 250	-134.01
FTSE 100	-138.65

latilities and parameters $p(h_t|y^t)$ and $p(\theta|y^t)$. Running the MCMC sampler again each time a new observation (y^{t+1}) becomes available is a feasible, but computationally unattractive, solution. Recent research has, therefore, been devoted to filtering algorithms for non-linear state space models. Early filtering algorithms solve the problem conditional on a value of θ. This is unattractive for two reasons. First, they do not incorporate the uncertainty on θ into the predictive density of h_t. Second, the most likely value of θ on which to condition, comes from the posterior distribution of a single MCMC algorithm run on the whole sample. However, conditioning on the information from the entire sample is precisely what one wants to avoid when drawing from $p(h_t|y^t)$. To solve this problem, Caravalho, Johannes, Lopes, and Polson (2010) (CJLP) introduced a particle learning algorithm.

We applied the Particle Learning (PL) algorithm to estimate the Model 1 specification with no correlations sequentially on the three currencies and to compute the sequential odds ratio. We present the algorithm in Appendix 3.6.3. It is important to understand how crucial the number of particles used is in these algorithms. We estimate the PL filter with 5, 10, 20 and 40 thousand particles and estimate the posterior odds for each t to compare the SV and SV-RV model. In Figure 3.2, we compare the logarithm of the odds ratio at the end of the sample with the logarithm of the MCMC odds ratio. The horizontal line correspond to the log of the

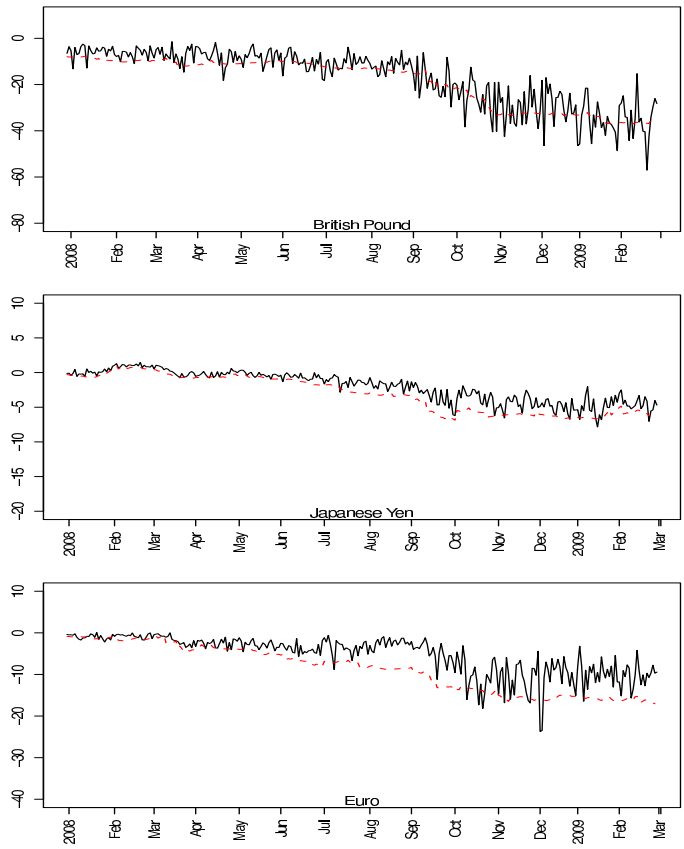
odd ratio calculated with the MCMC algorithm. Note how the PL-derived odds ratios converge very slowly to the MCMC odds ratio as the number of particles increases. This is a sign that the posterior distributions in the filtering algorithm may require a very large number of particles to be deemed reliable.

FIGURE 3.2 – Sequential odds ratio : British Pound - PL algorithm on M1 model



In Figure 3.3 we compare the sequential odds ratios obtained with the sequential estimation of MCMC algorithm and with the PL algorithm. The PL filter are based on 80,000 particles and the MCMC algorithm based on 50,000 draws. We see that for the three currencies, the log odds ratio decreased more since September 2008, which coincide with the start of the financial crisis. We also note that the PL and MCMC estimation of the odds are close, however, the odds ratios obtained with the MCMC algorithm are more volatile.

FIGURE 3.3 – Logarithm of sequential odds ratio calculated with MCMC algorithm and PL filter on currencies



In Table 3.18 we compute the average of the sequential odds ratios obtained with the MCMC algorithm for all series. The average of the sequential odds are calculated for two periods. The first period, correspond to the pre-crisis period (January 2, 2006 to August 27, 2012) and the second period includes the entire period (January 2, 2006 to March 1, 2009). Sequential estimation starts in January 2, 2008. The odds ratios in favor of RV increase for all currencies and stocks for

the financial crash period : October 2008 - March 2009.

TABLE 3.18 – Average logarithm of posterior odds Ratio by period

	Aug. 2,2006 to Aug. 28,2008	Aug. 29,2008 to Mar. 1,2009
Currencies		
British Pound	-8.38	-28.25
Euro	-0.37	-4.20
Japanese Yen	-2.32	-9.25
Indices		
S&P 500	-28.76	-49.07
TSE	-11.85	-46.47
Nikkei 250	-73.43	-120.84
FTSE 100	-60.44	-125.65

3.4.3 Parameter Inference

For all models considered, we use the basic specification of no correlation or fat tails for currencies and the case with correlated errors for stocks. The results are based on 50,000 draws after discarding the first 5,000 draws. Posterior parameter distribution form M0, M1 and M2 models are presented in Appendix 3.6.4. Table 3.24 shows the result for M0 model. The volatility persistence parameter is close to one for all series and there is a strong negative correlation value for the stock indices with posterior means that vary from -0.72 (Nikkei 250) to -0.57 (S&P 500).

Table 3.25 presents the posterior parameter distribution for Model 1 using the RV estimator as the volatility proxy. The posterior mean of $\beta + \delta$ across all series is relatively lower than their corresponding δ parameter obtained with M0 model. The posterior mean of β in the currency series is considerably lower than the values found for the stock indices. The posterior confidence intervals of α, β and δ for the stock indices show these are considerably more volatile compare to the obtained for currencies. For all stocks the confidence intervals include the value of 0 for δ and the 95% posterior percentile for the constant parameter α is considerable high for Nikkei

and FTSE. With respect to the correlation parameter, the posterior distribution of ρ shows that this coefficient is highly variable with very high posterior density near zero when we incorporate the RV estimator in the volatility equation. It seems that the interaction of the lagged RV estimator and lagged latent volatility cancel the correlation effect measured as $corr(v_{t+1}, \epsilon_t) = \rho$. These findings suggest that Model 1 with correlated errors for the stocks seems to be unstable.

Finally, Table 3.26 presents the posterior parameter distribution for Model 2 using the RV estimator as the volatility proxy. The posterior mean value of δ is close to one but slightly lower than that obtained with the basic SV model for all series. For β_1, the persistence parameter of the proxy volatility, its posterior mean varies from 0.89 (Nikkei) to 1.01 (TSE); for σ_{rv}, the posterior mean for stocks is higher than for currencies. Contrary to the results with Model 1, the posterior distribution of ρ for the indices shows evidence of a negative correlation between the returns and latent volatility, as in the case of the basic correlated model, however the value is lower. Thus, Model 2 with RV seems to be more appropriate than Model 1 with RV when analyzing return series that are known to present a leverage effect.

3.4.4 Volatility estimation

We compare the in-sample volatility estimation across models for the different series. Scatter plots of the annualized posterior mean of in-sample volatility estimated with Model 1 against the standard model, Model 2 against the standard model and Model 2 against Model 1 are presented in Figures 3.4 and 3.5. The in-sample posterior volatilities are based on the full sample for each series. In general, we see that estimations of volatility are relatively similar across models when volatility is low, but for very high volatility periods both Model 1 and Model 2 seems to have higher estimates compare to the basic model. Furthermore, Model 1 and Model 2 show discrepancies in the value of volatility when this is high.

FIGURE 3.4 – Comparison of in-sample volatility fit across models : currencies

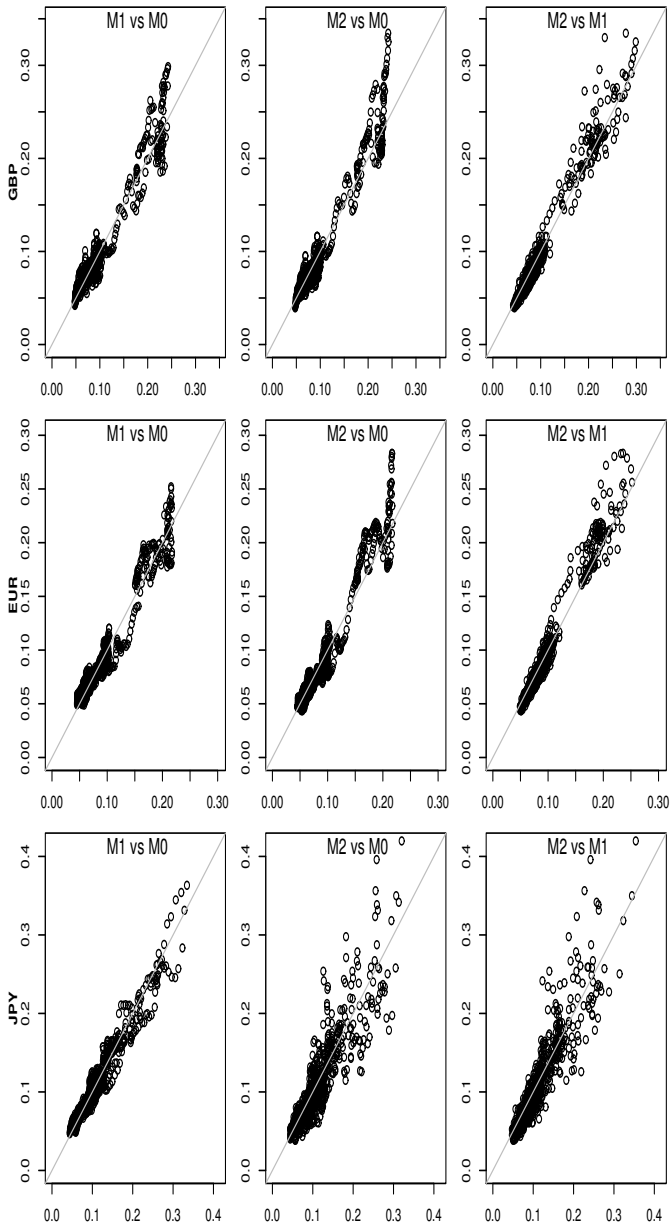
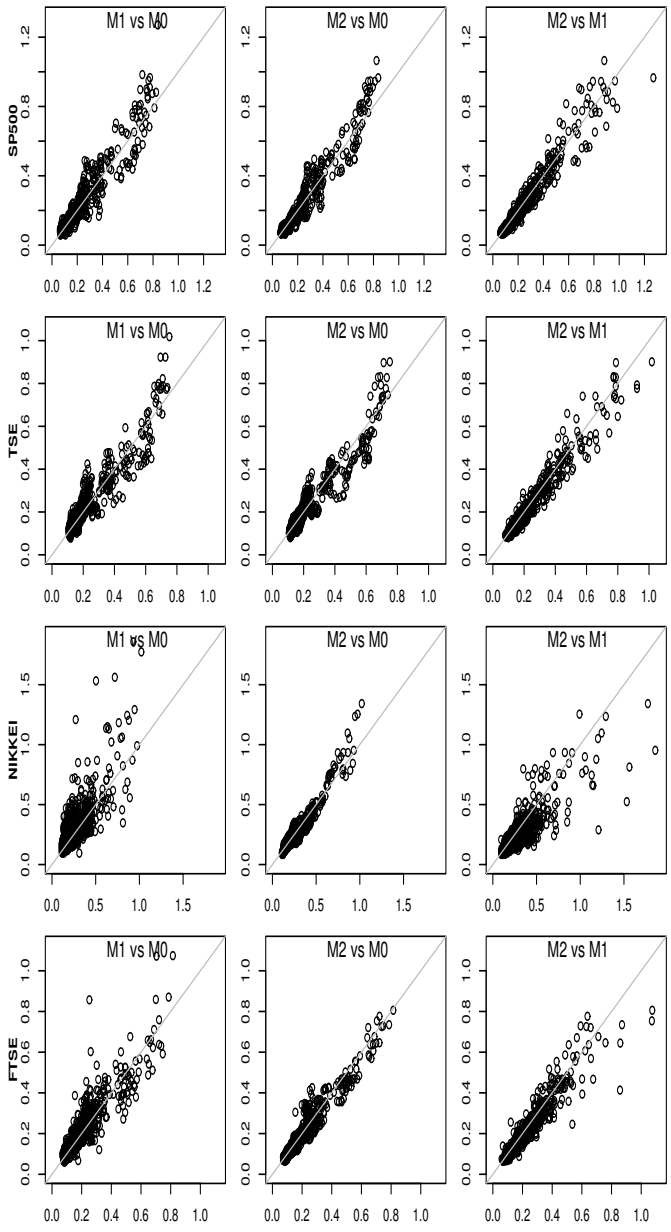


FIGURE 3.5 – Comparison of in-sample volatility fit across models : stocks



With respect to the out-of-sample volatility fit, Figures 3.6 and 3.7 presents the scatter plots of the annualized posterior mean of one-day forecasts and Figures 3.8 and 3.9 presents scatter plots of the annualized posterior mean of one-week ahead volatility forecasts. In the four figures, the panels in the left side show volatility estimated with Model 1 against the standard model, the panels in the middle show volatility estimated with Model 2 against the standard model and the panels on the right show volatility estimated with Model 2 against Model 1. Out of sample volatility forecasts are based on sequential estimation of the models for the period from January 2008 to February 2009 (around 300 observations per series).

For the one-day-ahead volatility forecast, Model 1 and Model 2 have different estimations with respect to the standard model for all series. Between Model 1 and Model 2, it seems that estimations are more similar with the exception of Nikkei for which Model 1 estimates higher volatilities than Model 2 and the standard model. For the longer forecast period, one week, we observe that Model 1 give very different volatility forecasts for all stock series than the basic and Model 2, the estimations are in most of the cases much higher, specially for Nikkei and FTSE. As we observed in the evaluation of the posterior distribution of the parameters for Model 1, parameters are unstable with some very large estimates of the constant parameter α for the case of Nikkei and FTSE.

Given the different forecasts obtained with Model 1 for the stock's volatilities, we evaluate the sensitivity of the results to the prior specification of the constant parameter α. In general, as described in Section 3.2.3 we use very flat priors for all parameters. In particular, for the parameter α we set its prior to be $N(0, 100)$. We now adjust the variance of the prior from 100 to 1 and re-estimate Model 1 for the stocks. Figure 3.10 and 3.11 show the one-day and one-week ahead volatility estimations obtained with the tighter prior. We observe that the volatility's forecasts improve for Model 1, although we still see a bias for the estimation of one-week ahead forecast of Nikkei's volatility.

FIGURE 3.6 – Comparison of one day ahead volatility forecast across models : currencies

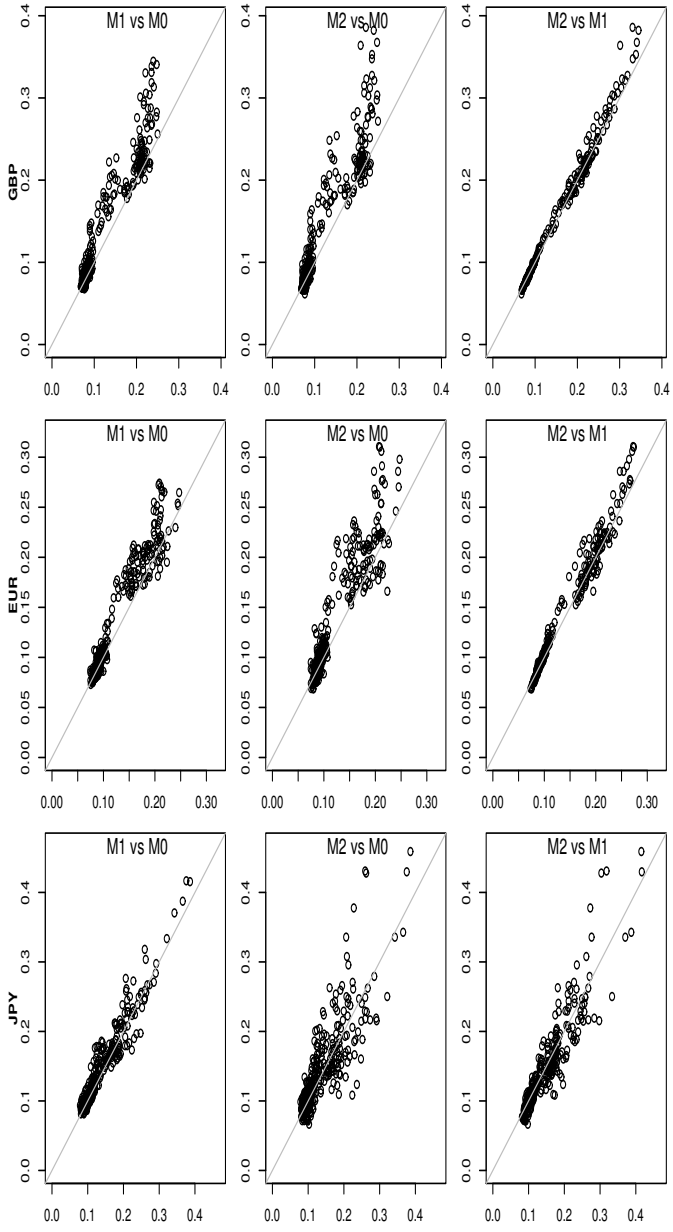


FIGURE 3.7 – Comparison of one day ahead volatility forecast across models : stocks

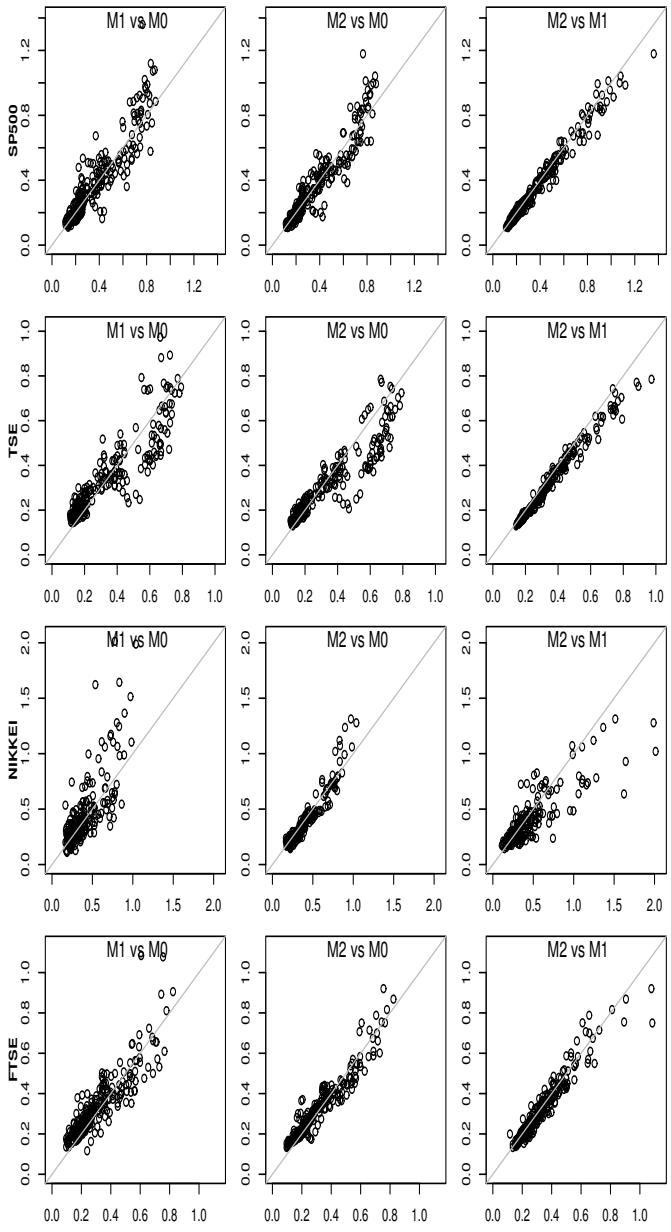


FIGURE 3.8 – Comparison of one week ahead volatility forecast across models : currencies

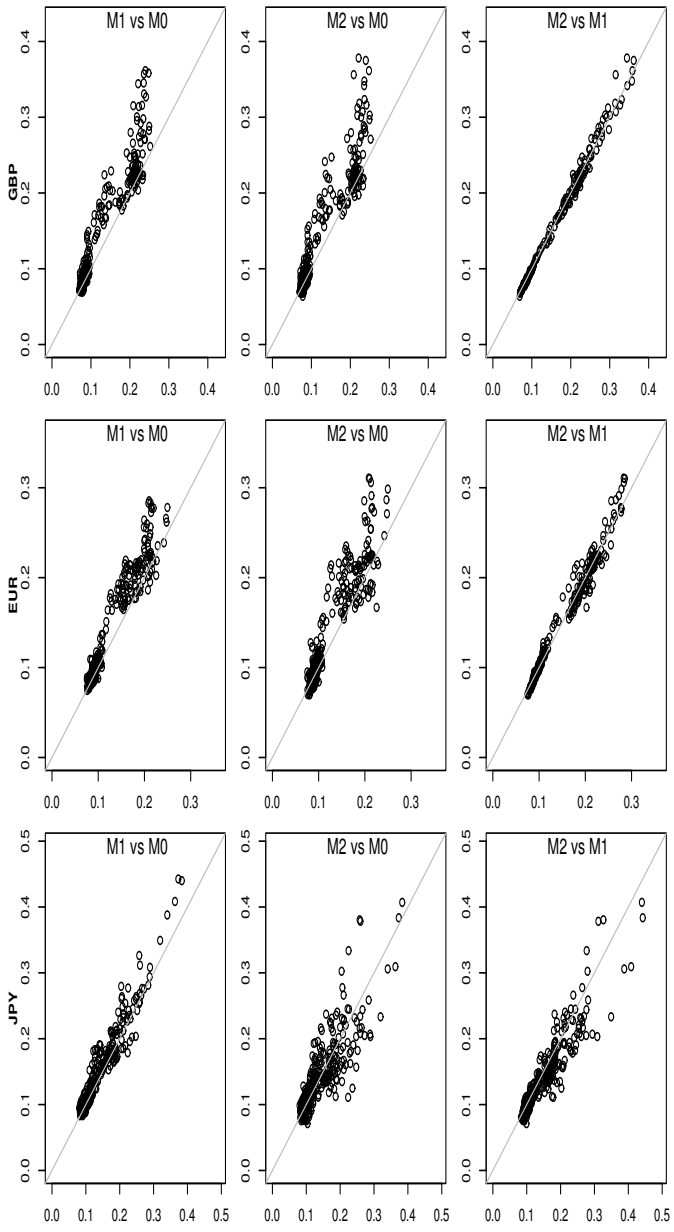


FIGURE 3.9 – Comparison of one week ahead volatility forecast across models : stocks

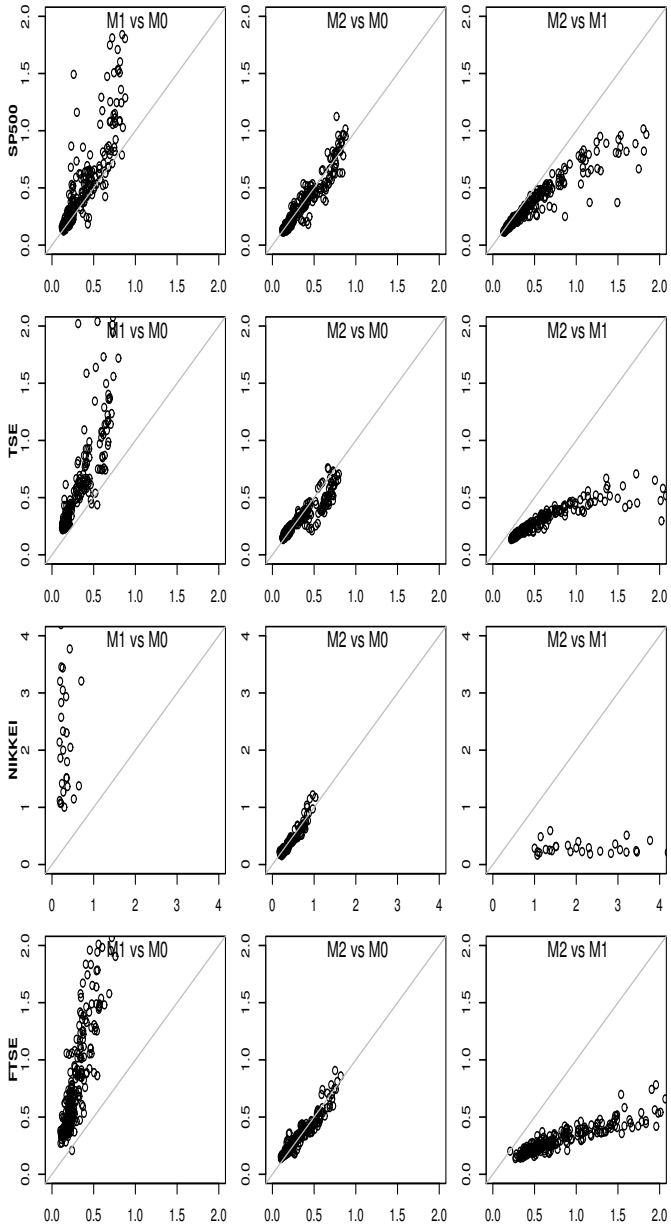


FIGURE 3.10 – Comparison of one day ahead volatility forecast across models : stocks - new prior

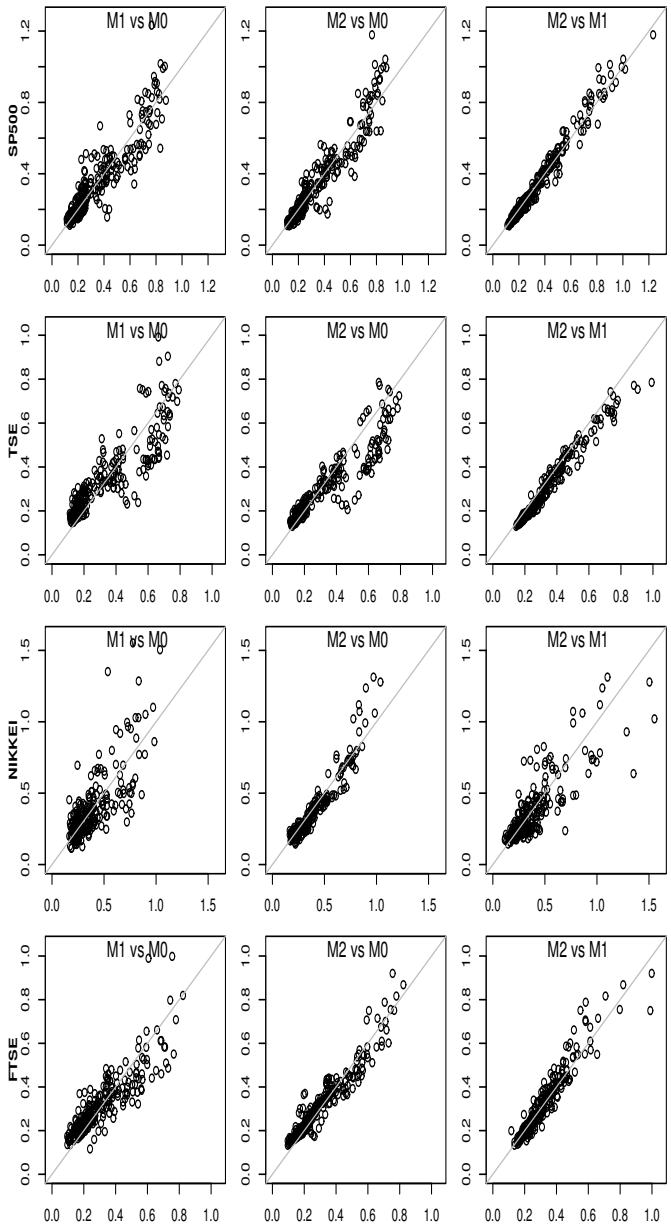
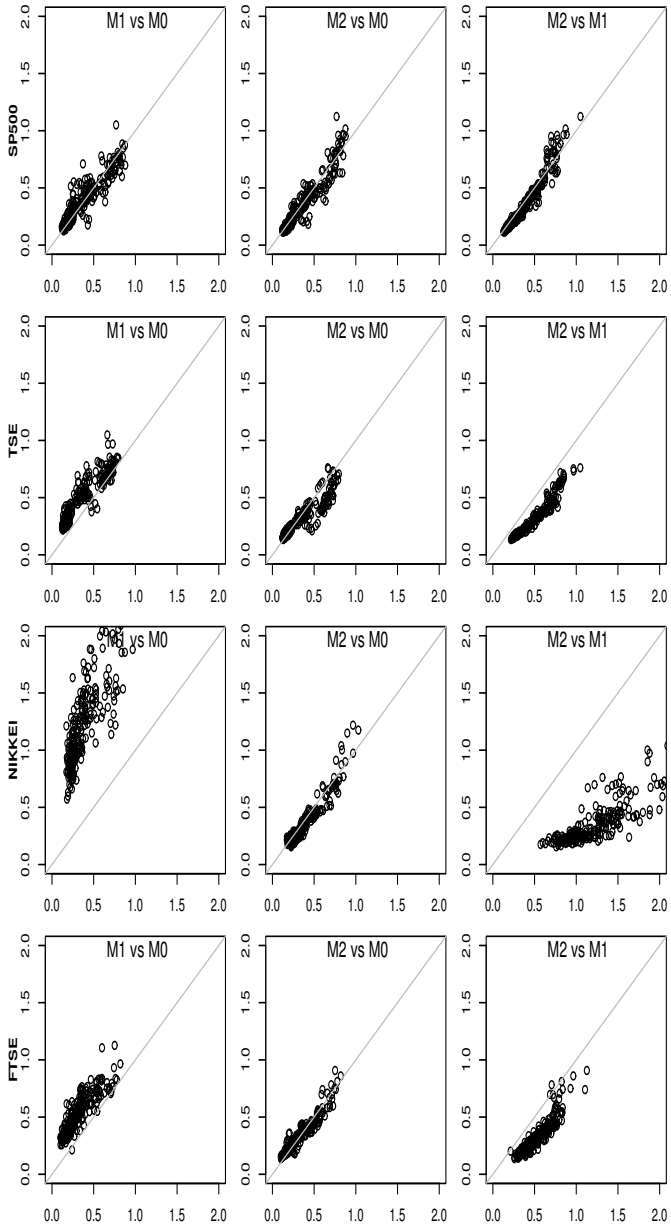


FIGURE 3.11 – Comparison of one week ahead volatility forecast across models : stocks - new prior



The comparison of the in-sample and out-of-sample volatility estimations have allowed us to identify some of the discrepancies across models in terms of volatility estimation and forecasting, but we can not say if one model is better than the other as we do not know the true volatility. We do observe the returns, though, so we use them to compute the hit rate of 95% confidence intervals for the 1-step and one week ahead future log-returns using each model. The relevant out of sample density is :

$$p(r_{t+k}|I_t) = \int p(r_{t+k}|h_{t+k})p(h_{t+k}|I_t)dh_{t+k}$$

We use the sequential estimation of each model to draw the returns from its conditional predictive distribution given volatility. The accuracy of the hit rates is evaluated using an unconditional coverage test as used in Chib, Nardari, and Shephard (2006) :

$$LR_{uc} = 2\left[\log(\hat{\alpha}^{\gamma}(1-\hat{\alpha})^{T-\gamma}) - \log(\alpha^{\gamma}(1-\alpha)^{T-\gamma})\right]$$

where α is 5%, $\hat{\alpha}$ is the estimated proportion of observations outside the 95% confidence interval, and γ is the number of hits. The null hypothesis of this test is $\hat{\alpha} = \alpha$ and the test statistic is distributed asymptotically as $\chi^2(1)$. Table 3.19 reports the average hit rate based on approximately 300 observations per series. Columns 1 to 3 report the average hit rate (in percentage) for the standard model, Model 1 and Model 2, respectively. Columns 4 to 6 report the p-value for the unconditional coverage test for the three models. None of the models performs uniformly better. For the one-day-ahead hit rate of the 95% confidence intervals of the log returns, the basic model provides a higher level of accuracy for the Japanese Yen and the Nikkei index, Model 1 is more precise for TSE and FTSE indices and Model 2 is more accurate for the British Pound, Euro and S&P 500 index. For the one-week-ahead hit rate, Model 1 performs the worst for all stock indices while Model 2 is equal or more accurate than Model 1 for all series but for the TSE index.

TABLE 3.19 – Currencies and stocks : hit rate

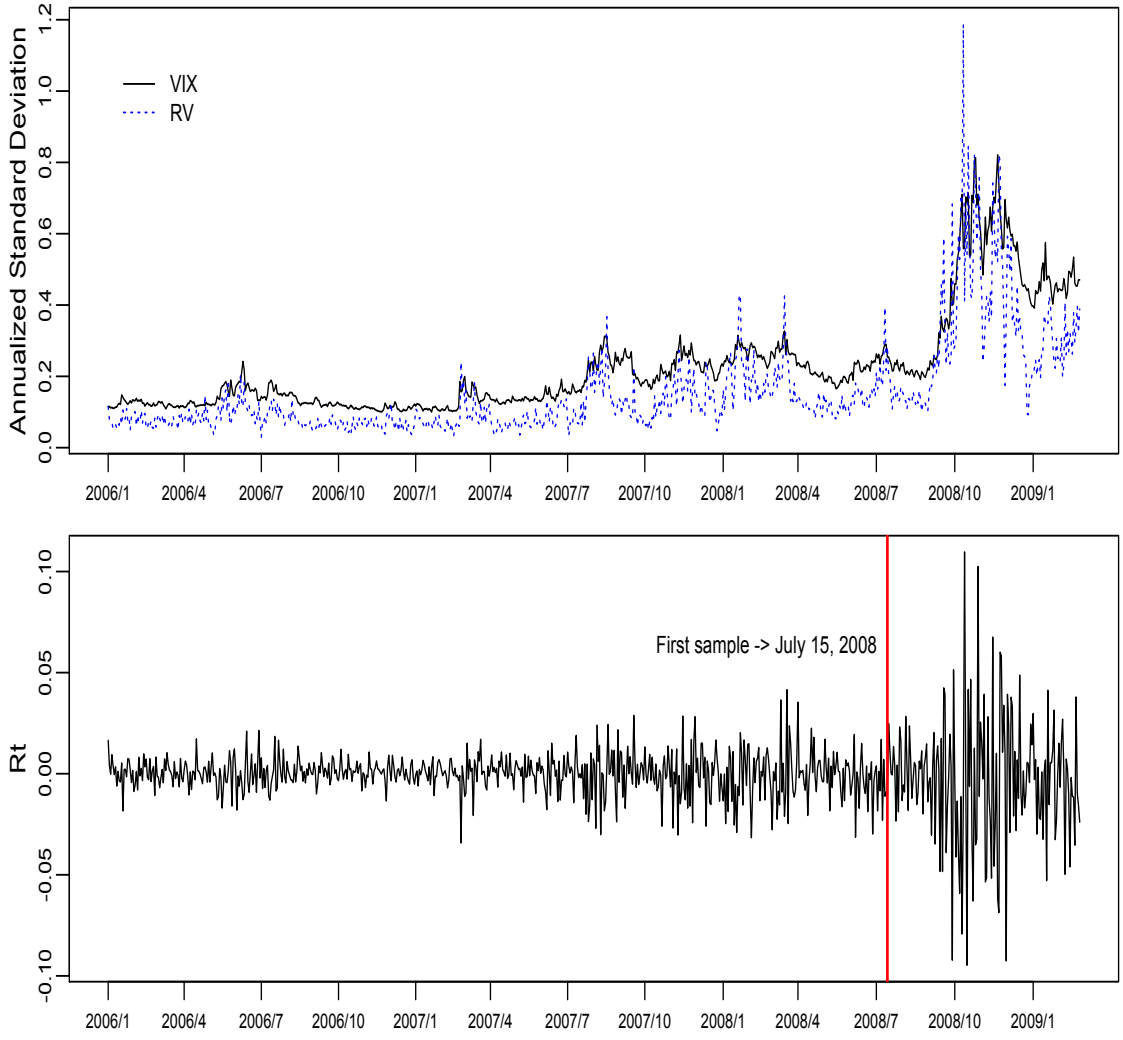
	Hit rate			p-values		
Series	M0	M1	M2	M0	M1	M2
One day ahead						
British Pound	8.36	2.01	5.02	0.01	0.01	0.99
Euro	8.72	6.71	5.70	0.01	0.20	0.58
Japanese Yen	5.03	6.04	6.38	0.98	0.42	0.30
S&P 500	7.07	6.72	6.05	0.13	0.20	0.39
TSE	9.36	5.99	10.11	0.00	0.47	0.00
Nikkei 250	4.31	6.03	6.90	0.62	0.48	0.21
FTSE 100	7.38	5.17	4.43	0.09	0.90	0.66
One week ahead						
British Pound	8.85	3.05	6.78	0.00	0.10	0.18
Euro	8.88	7.82	7.82	0.00	0.04	0.04
Japanese Yen	5.44	4.76	5.44	0.73	0.85	0.73
S&P 500	1.79	1.79	3.58	0.00	0.00	0.25
TSE	6.84	1.52	8.37	0.19	0.00	0.02
Nikkei 250	3.51	0.00	3.51	0.28	0.00	0.28
FTSE 100	3.37	0.00	4.49	0.20	0.00	0.70

3.4.5 S&P 500 : Realized or Implied Volatility ?

In this subsection, we study with more detail the volatility estimation of the S&P 500 index for which we have two alternative volatility proxies : the realized volatility estimator (RV) and the implied volatility index (VIX). We want to evaluate the difference in the estimation when using the RV and/or the VIX estimator with particular interest in learning from the information given by both estimators during the financial crisis of 2008-2009. The top panel in Figure 3.12 shows the time series plot of the two volatility proxies. The bottom panel shows the daily return. The vertical line indicates the starting point for our analysis of the sequential estimation, July 15, 2008. This starting point corresponds to a time prior to the crisis period.

Using the sequential estimation of Model 1, we compute sequential odds ratio to

FIGURE 3.12 – Plot of S&P 500, Realized Volatility and Implied Volatility Index



compare Model 1 with VIX against Model 1 with VIX and RV ($M1-VIX|M1-RV-VIX$), and Model 1 with RV against Model 1 with VIX and RV ($M1-RV|M1-RV-VIX$). Figure 3.13 presents the sequential odds ratios for each case. There is strong evidence in favor of the model with implied volatility in the presence of the RV estimator, and there is weak or even negative evidence for the RV estimator in the presence of the VIX. This suggests that the VIX may contain all the information in the RV estimator and more.

The performance evaluation of Section 3 shows that Model 2 extrapolates the information content on different volatility proxies in a better way than multivariate versions of Model 1. Therefore, we now use the sequential estimation of univariate and bivariate formulations of Model 2 to analyze the differences and consistencies with respect to parameters and volatility estimation during the crash period across alternative specifications of Model 2. First, we need to know if the crisis period affects the persistence parameter δ. Figure 3.14 plots the sequential posterior distributions of δ for the standard model(M0) and the alternative two univariate specifications (M2-RV, M2-VIX). The two top panels shows the sequential estimation of δ for the basic model (M0). The bottom left panel shows the same for the the M2-RV model and the bottom left for the M2-VIX model. It clearly shows that incorporating the RV estimator reduces the posterior persistence of volatility, while the opposite occurs when incorporating implied volatility. The persistence obtained with RV in the model is far less than that with the VIX. Finally, the plots show that the posterior persistence increases through the financial crisis period for all models.

Figure 3.15 reports the sequential posterior distribution of the measurement equation parameters for RV and the VIX, namely β_1 and σ_η. The two top panels show the sequential estimation of β_1 for the M2-RV model (left panel) and M2-VIX model (right panel). The bottom panel shows the estimation of σ_η for the the model M2-RV (left panel) and M2-VIX (right panel).

The posterior slope of the RV equation is quite stable during the entire period and higher than that obtained for the VIX equation. The slope of the VIX equation

FIGURE 3.13 – Logarithm of sequential odds ratio : $S\&P500$

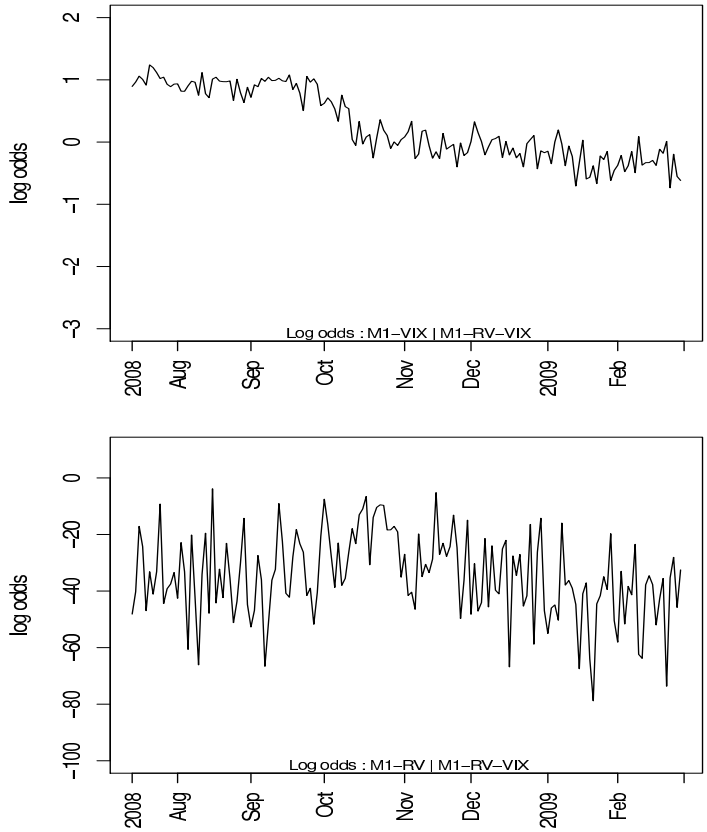


FIGURE 3.14 – S&P 500 : δ.

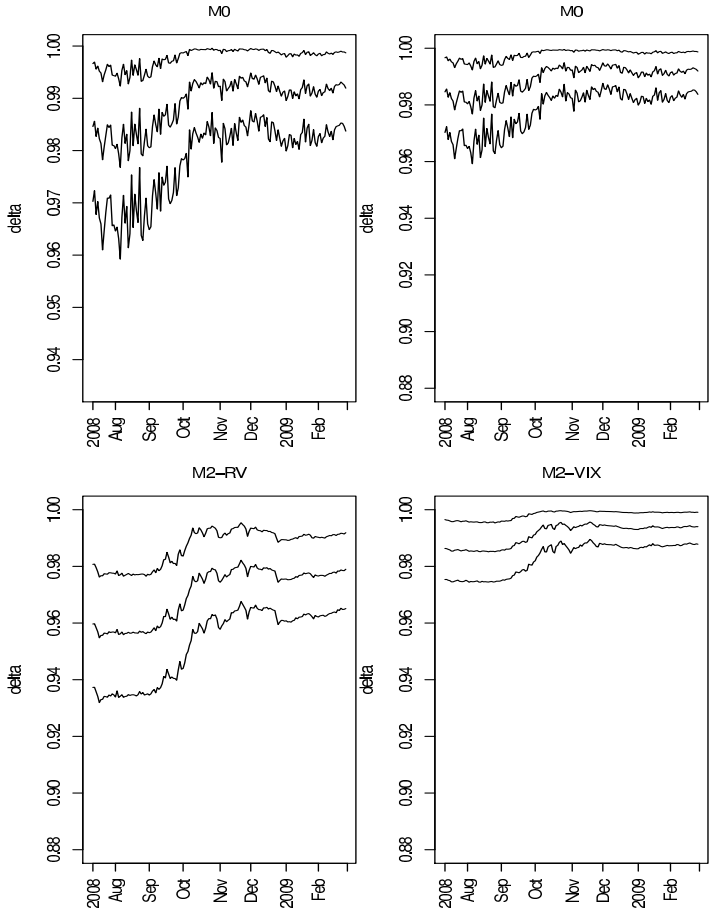
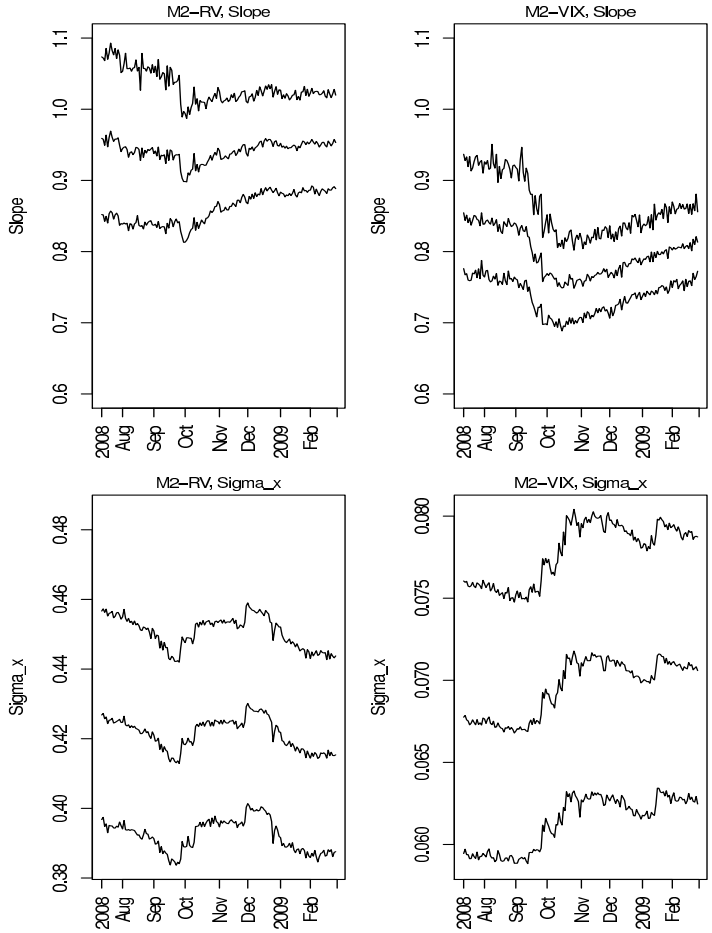


FIGURE 3.15 – S&P 500 : β_1 and σ_η



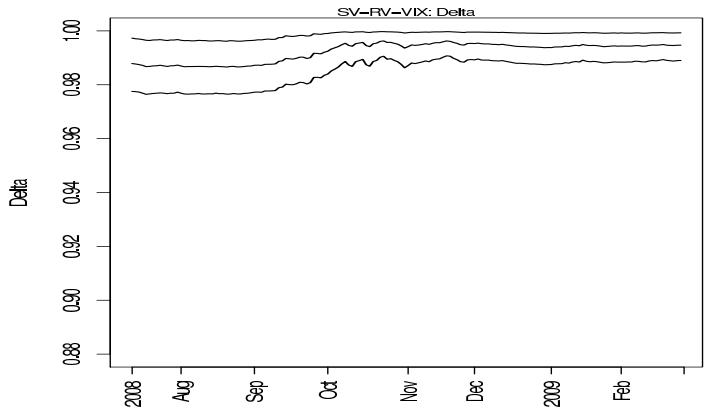
decreases after the crash period. The standard deviation of the estimation error σ_ϵ is much higher for RV than for VIX. With resect to the correlation parameter, we find that the M2-RV model estimates a stable ρ parameter along the the entire sample with posterior mean close to -0.5. However, the M2-VIX model estimates that the posterior mean of the correlation parameter is close to zero. A possible explanation for the null correlation with the M2-VIX model is that the VIX estimator is based on future expectations of volatility and may be incorporating the information capture in the correlation parameter.

Next, we evaluate how the posterior distribution of parameters changes when we estimate Model 2 incorporating the RV and the VIX estimators at the same time (M2-RV-VIX). Figure 3.16 plots the sequential estimation of the persistence parameter of the log volatility equation. The evolution of the delta parameter is very similar to that obtained when we just incorporate the VIX estimator. That is, the introduction of the RV estimator in addition to the VIX estimator does not change the persistence parameter. The posterior distribution of the slope and the volatility parameters of the two proxy equations are also very similar to their corresponding values when using only one proxy at a time. The correlation parameter is very similar to that obtained with only the VIX variable, that is it is close to null [16].

What are the implications for the S&P 500 volatility estimation ?. Is the higher error variance of the RV estimators transmitted to the posterior density of the latent volatility ?. Figure 3.17 plots the posterior mean of the latent volatility, $\sqrt{h_t}$ and its posterior 5% and 95% percentiles for the basic model (top left panel), M2-RV model (top right panel), M2-VIX (bottom left panel) and M2-RV-VIX (bottom right panel). The black solid line corresponds to the posterior mean and the green dotted lines are the posterior 5% and 95% percentiles. The period plotted is from January 3, 2006 to August 28, 2008. The higher volatility of the RV measure does translate into a more variable latent volatility posterior density. In contrast, the bottom left plot shows that the VIX imparts its persistence. The M2-RV-VIX model

16. We estimated the M2-VIX and the M2-RV-VIX models without correlation and obtained very similar results for the parameters and volatility estimations.

FIGURE 3.16 – S&P 500 : δ for bivariate Model 2



gives very similar estimates to the M2-VIX model.

Figure 3.18 compares the posterior means of annualized volatility for the M2-RV and the M2-VIX models against the standard model. The left panel shows a scatter plot of the volatility estimated with M2-RV model against the basic model. Right panel is the scatter plot of the M2-VIX volatility versus the basic model. The incorporation of RV increases posterior volatilities when they are large, and decreases them when they are small. In contrast, the incorporation of VIX does not appear to have a dramatic effect on the formulated posterior density of volatility.

Consider now the crash period itself : August 28,2008 to November 19, 2008. Figure 3.19 displays volatility posterior mean, using M0, M2-RV, M2-VIX and M2-RV-VIX models for this period. RV and VIX strongly disagree on the path of the past volatility between the beginning of the crisis period and where we are on November 19, 2008.

In terms of out-of-sample forecasts, Figure 3.20 presents scatter plots of one-day-ahead and one-week-ahead volatility forecasts for the M2-RV, M2-VIX and M2-RV-VIX models against the standard model. As in the in-sample estimation of volatility, there are discrepancies in the out-of-sample estimates between the M2-RV and M2-VIX model against the basic model and between them. The top panels show one-day ahead posterior volatility forecasts and the bottom panels show one-week-ahead volatility forecasts. The left panels show a scatter plot of the volatility estimated with M2-RV model against the standard model. Middle panel is the scatter plot of the M2-VIX volatility versus the standard model and the right panels show the scatter plot of the M2-VIX volatility versus the standard model. The forecasts obtained with M2-RV-VIX are very close to those obtained with the M2-VIX model.

FIGURE 3.17 – S&P 500 : annualized volatility estimation with M0, M2-RV and M2-VIX models

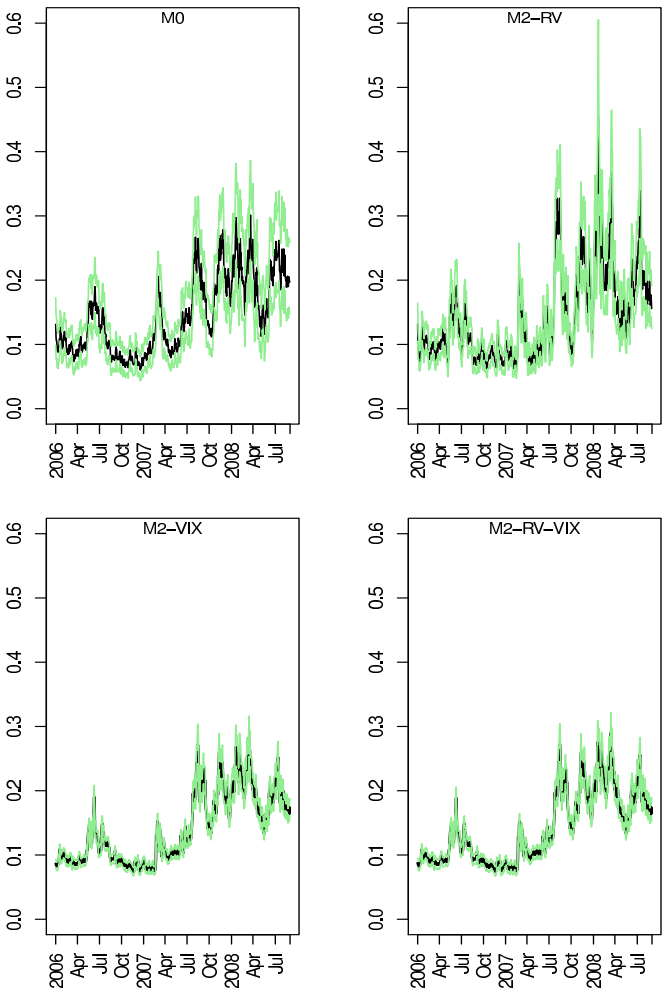


FIGURE 3.18 – S&P 500 : posterior annualized volatility estimation comparison

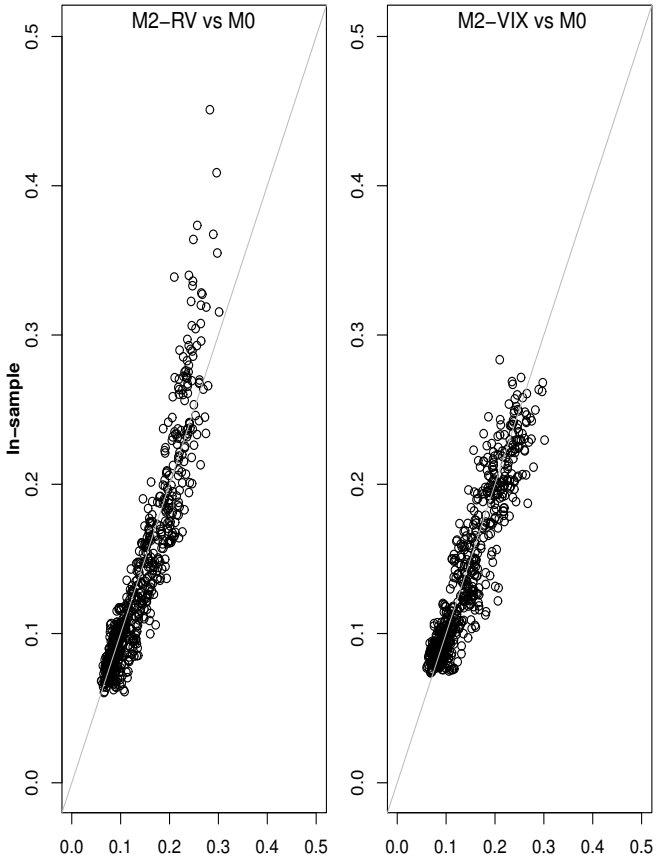


FIGURE 3.19 – S&P 500 : In sample volatility estimation

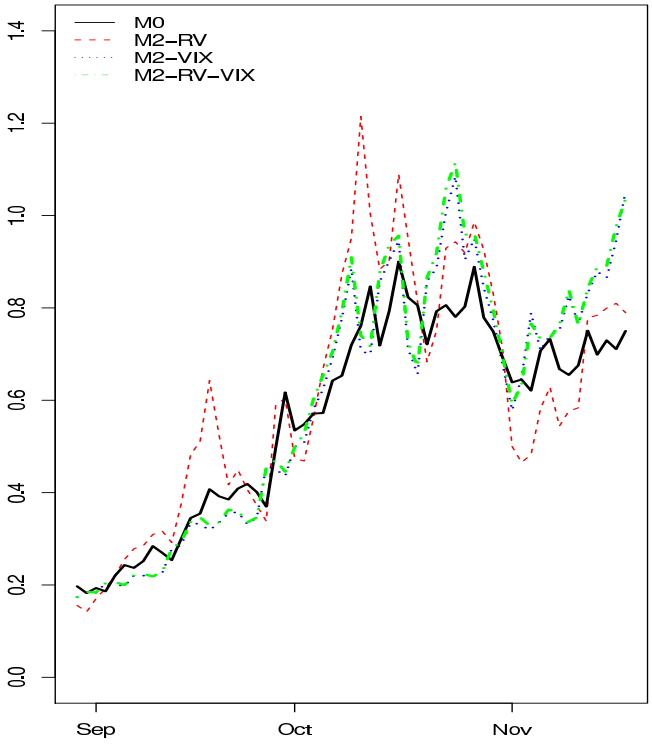


FIGURE 3.20 – S&P 500 : Out of sample volatility estimation

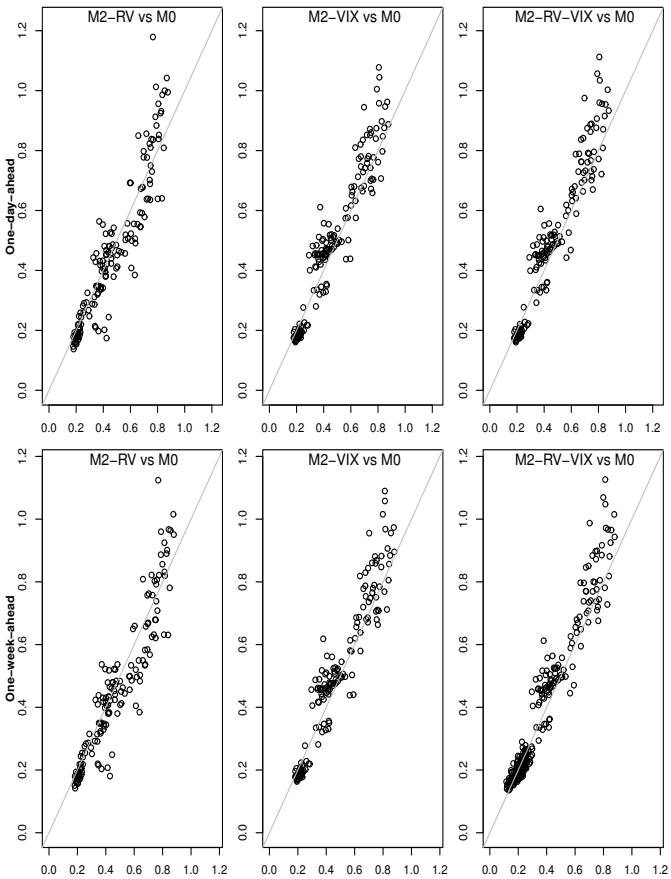


Table 3.20 compares the hit rate of 95% confidence intervals for the one-day and one-week ahead future log-returns across models. The models that incorporate information from volatility proxies (M2-RV, M2-VIX, M2-RV-VIX) never perform worse than the standard model (M0). In particular, the bivariate model M2-RV-VIX is more accurate for the one-day-ahead log return forecast while the M2-RV is better for the one-week-ahead forecast.

TABLE 3.20 – $S\&P500$: hit rate

	Hit rate		p-values	
S&P 500	One-day-ahead	One-week-ahead	One-day-ahead	One-week-ahead
M0	7.01	1.96	0.28	0.05
M2-RV	7.01	5.88	0.28	0.63
M2-VIX	6.37	3.92	0.45	0.53
M2-RV-VIX	5.10	3.27	0.96	0.30

3.5 Conclusions

We evaluate the information contributed by (variations of) realized volatility to the forecasting of volatility when prices are measured with and without error. We use two econometric specifications to incorporate realized volatility into the inference on latent volatility. We write the MCMC algorithms for these models.

The performance results show that the ability of realized volatility to improve forecasts is seriously affected by price noise, and highlight the importance of using adjustments that are robust to the presence of noise, such as the two scaled realized volatility estimator. We also find that the second econometric specification, where the volatility proxy is explicitly linked to the latent volatility, results in better in-sample and out-of-sample estimation of volatility than the first specification. This is especially true for multi period forecasts.

The empirical analysis shows that the odds ratios favor the use of the realized volatility estimator for most indices and currencies. However, when using out-of-sample estimations of volatility to compute the 95% hit rate of confidence intervals of the log returns not one model performs uniformly better across all series. We study the modeling of the $S\&P$ 500 volatility in more detail. We compare the information contents on realized volatility and implied volatility during the financial crisis 2008-2009. We find that the odds ratios favor the model with implied volatility (VIX) against the model with realized volatility. Realized volatility has a high level of variability which is transmitted to the estimation of the latent volatility. On the

other hand, a higher persistence in volatility is transmitted when using implied volatility. We also found that both estimators generate very different estimations of volatility during the most volatile period of the financial crisis. When estimating the model with both volatility proxies, results indicate that RV does not bring any information over and above the VIX. Namely, estimation of volatility remains similar to that obtained when using only the VIX indicator. In terms of out-of-sample forecasts, the models that include information from the proxy volatilities perform better than the standard SV model.

Methodological extensions of this paper include the multivariate version of the second econometric specification to include additional proxies of volatility via seemingly unrelated measurement equations (SUR) of possibly correlated measure errors. In terms of implementation, one needs to further develop nonstatistical metrics to evaluate the out-of-sample performance of these volatility estimators, such as resulting from minimum variance or maximum expected utility strategies.

3.6 Appendix to Chapter 3

3.6.1 Model 2 - Bivariate

In this section we describe the algorithm to estimate Model 2 when using 2 competing log volatility proxies (X_1, X_2). The general model is given by :

$$\begin{aligned} r_t &= \sqrt{h_t \lambda_t}\epsilon_t \\ \log h_{t+1} &= \alpha + \delta \log h_t + \sigma_v v_{t+1}, \quad u_{t+1} = \sigma_v v_{t+1} \\ X_{1t} &= \beta_0 + \beta_1 \log h_t + \sigma_{\eta_1}\eta_{1t}, \quad z_{1t} = \sigma_{\eta_1}\eta_{1t} \\ X_{2t} &= \gamma_0 + \gamma_1 \log h_t + \sigma_{\eta_2}\eta_{2t}, \quad z_{2t} = \sigma_{\eta_2}\eta_{2t} \\ a_t &= (\epsilon_t, u_{t+1}, z_{1t}, z_{2t}) \sim N(0, \Sigma) \end{aligned}$$

$$\Sigma = \begin{pmatrix} 1 & \rho_1\sigma_v & 0 & 0 \\ \rho_1\sigma_v & \sigma_v^2 & 0 & 0 \\ 0 & 0 & \sigma_{\eta_1}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\eta_2}^2 \end{pmatrix}$$

The algorithm to estimate this model is based on the same steps described in Section 2, with modifications to steps (2) and (4).

– In step (2), we now have to estimate the parameters of the log volatility equation and those of each volatility equation : $\omega = (\alpha, \delta, \sigma_v, \beta_0, \beta_1, \sigma_{\eta_1}, \gamma_0, \gamma_1, \sigma_{\eta_2})$. In this model we assume that there is no error correlation between the proxy equations, so, we can estimate the parameters of each equation independently using direct draws :

1. Draw (α, δ) from $p(\alpha, \delta|\sigma_v, h)$ which is Normal distribution and direct draws can be performed. σ_v is draw joint with ρ_1 using the same step (3) of Model 1 and 2.
2. Draw (β_0, β_1) from $p(\beta_0, \beta_1|\sigma_{\eta_1}, h)$ which is Normal distribution and di-

rect draws can be performed. Draw σ_{η_1} from $p(\sigma_{\eta_1}|\beta_0, \beta_1, h)$ which is an Inverted Gamma distribution an direct draws are possible.

3. Draw (γ_0, γ_1) from $p(\gamma_0, \gamma_1|\sigma_{\eta_2}, h)$ which is Normal distribution and direct draws can be performed. Draw σ_{η_2} from $p(\sigma_{\eta_2}|\gamma_0, \gamma_1, h)$ which is an Inverted Gamma distribution an direct draws are possible.

– In step (4), we now write :

$$\Sigma^{-1} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{22} & 0 & 0 \\ 0 & 0 & S_{33} & S_{34} \\ 0 & 0 & S_{34} & S_{44} \end{pmatrix} = \begin{pmatrix} \frac{1}{(1-\rho_1^2)} & \frac{-\rho_1}{(1-\rho_1^2)\sigma_v} & 0 & 0 \\ \frac{-\rho_1}{(1-\rho_1^2)\sigma_v} & \frac{1}{(1-\rho_1^2)\sigma_v^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_{\eta_1}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{\eta_2}^2} \end{pmatrix}$$

Then, $p(h_t|.) \propto$:

$$h_t^{-1.5} exp\left(\frac{-1}{2}\frac{S_{11}r_t^2}{h_t} - S_{12}\epsilon_t u_{t+1} - S_{12}\epsilon_{t-1}u_t - \frac{1}{2}S_{22}(u_{t+1}^2 + u_t^2) - \frac{1}{2}S_{33}z_{1t}^2 - \frac{1}{2}S_{44}z_{2t}^2\right)$$

Developing :

1. $B = -\frac{1}{2}S_{22}(u_{t+1}^2 + u_t^2) - \frac{1}{2}S_{33}z_{1t}^2 - \frac{1}{2}S_{44}z_{2t}^2 = -\frac{1}{2}\frac{(\log h_t - \mu_t^*)^2}{s^{*2}}$
   where :

   (a) $\mu_t^* = s^{*2}\left(\mu_{1t}s_1^{-2} + \mu_{2t}s_2^{-2} + \mu_{3t}s_3^{-2}\right)$, $s^{*2} = \frac{1}{s_1^{-2}+s_2^{-2}+s_3^{-2}}$.

   (b) $\mu_{1t} = \frac{\alpha(1-\delta)+\delta(\log h_{t+1}+\log h_{t-1})}{(1+\delta^2)}$, $s_1^2 = \frac{1}{S_{22}(1+\delta^2)}$.

   (c) $\mu_{2t} = \frac{X_{1t}-\beta_0}{\beta_1}$, $s_2^2 = \beta_1^{-2}S_{33}^{-1}$.

   (d) $\mu_{3t} = \frac{X_{2t}-\theta_0}{\theta_1}$, $s_3^2 = \theta_1^{-2}S_{44}^{-1}$.

2. $C = -S_{12}\epsilon_{t-1}u_t \propto \log h_t^{\left(-S_{12}\frac{r_{t-1}}{\sqrt{h_{t-1}}}\right)}$

Replacing into the posterior of h, we have :

$p(h_t|.) \propto$

$\frac{1}{h_t^{\left(1.5+S_{12}\frac{r_{t-1}}{h_{t-1}}\right)}}$
$\exp\left[-\frac{1}{2}S_{11}\frac{r_t^2}{h_t} - \frac{1}{2}\frac{(\log h_t - \mu_t^*)^2}{s^{*2}} - S_{12}r_t\frac{u_{t+1}}{\sqrt{h_t}}\right]$
As in Model 2, the proposal density is given by :

$$q2(h_t|.) \sim IG(\phi_{2t}, \theta_t^*) \propto \frac{1}{h_t^{\phi_{2t}+1}} \exp\left(-\theta_t^*/h_t\right)$$

and the new parameters of the proposal are given by :

– $\phi_{2t} = S_{12}\frac{r_{t-1}}{h_{t-1}} + 0.5 + \phi_{LN}$

$\phi_{LN} = \dfrac{1 - 2e^{s^{*2}}}{1 - e^{s^{*2}}}$

– $\theta_t^* = \theta_t - s\delta S_{12} r_t$
$\theta_t = S_{11}\dfrac{r_t^2}{2} + \theta_{LN}$
$\theta_{LN} = (\phi_{LN} - 1)e^{\mu_t^* + 0.5s^{*2}}$
The s is to compensate for the term $\frac{u_{t+1}}{\sqrt{h_t}}$ that is in the posterior of h when ρ_1 is different from zero.

3.6.2 Comparison of JPR and KSC estimations

In order to compare the performance of each method we use the KSC algorithm to estimate the simple SV model and the SV model extended with an exogenous variable, RV 30sec, using the 500 simulated samples of 1,500 daily observations with prices measured without error. Table 3.21 reports the sampling performance of the posterior mean of the parameters based on 30,000 draws after discarding the first 1,000. Clearly, the two algorithms deliver almost the same output, parameter posteriors are almost identical for both cases studied.

We also compare the single-move and the multi-move algorithms for inference in volatility. Table 3.22 reports the RMSE and $\%MAE$ of the (in-sample) posterior means of $\sqrt{h_t}$. We observe that the single-move and multi-move algorithm produce about the same posterior means of volatility. We will show later, with actual data, that this is also true for the entire posterior distribution of volatility.

Now we run both algorithms on 809 days of the daily UK pound to US$ exchange rate. Table 3.23 shows the posterior analysis, where the two models produce nearly identical inference.

Possibly, the multi-move will result in different posterior densities for the volatilities $\sqrt{h_t}$? In Figure 3.21 we show plots of the posterior mean and the 5th and 95th quantiles of the posterior distribution of $p(\sqrt{h_t})$, for both algorithms. These are also identical. These results indicate that SV models estimated by single-move or multi-move MCMC can deliver, period after period, posterior distributions of smoothed volatilities with a very satisfactory degree of precision.

TABLE 3.21 – Single-Move versus Multi-Move MCMC algorithm : Parameter estimation

Models	α	β	δ	σ_v	$\beta + \delta$
True value	-0.37		0.96	0.21	
Single - move					
No RV					
Average	-0.479		0.948	0.229	
5%	-0.744		0.92	0.176	
95%	-0.268		0.971	0.291	
RMSE	0.192		0.021	0.041	
RV 30sec					
Average	-0.321	0.819	0.146	0.187	0.965
5%	-1.031	0.633	0.017	0.066	0.888
95%	0.364	0.972	0.334	0.33	1.039
RMSE	0.435	0.16	0.816	0.065	0.047
Multi - move KSC					
No RV					
Average	-0.443		0.952	0.221	
5%	-0.612		0.934	0.18	
95%	-0.305		0.967	0.267	
RMSE	0.16		0.017	0.037	
RV 30sec					
Average	-0.33	0.841	0.124	0.189	0.965
5%	-0.872	0.695	0.02	0.08	0.907
95%	0.187	0.962	0.271	0.311	1.021
RMSE	0.456	0.151	0.841	0.093	0.049

The sampling distributions are based on 500 samples simulated with 1500 daily observations with prices measured without error. The rows entitled Average and RMSE report the average and the mean squared errors of the 500 posterior means. The rows entitled 5% and 95% report the average of the 500 5th and 95th posterior percentiles. The posteriors are based on 30,000 draws after discarding the first 1,000 draws.

TABLE 3.22 – Single-Move versus Multi-Move MCMC algorithm : In sample fit of $\sqrt{h_t}$

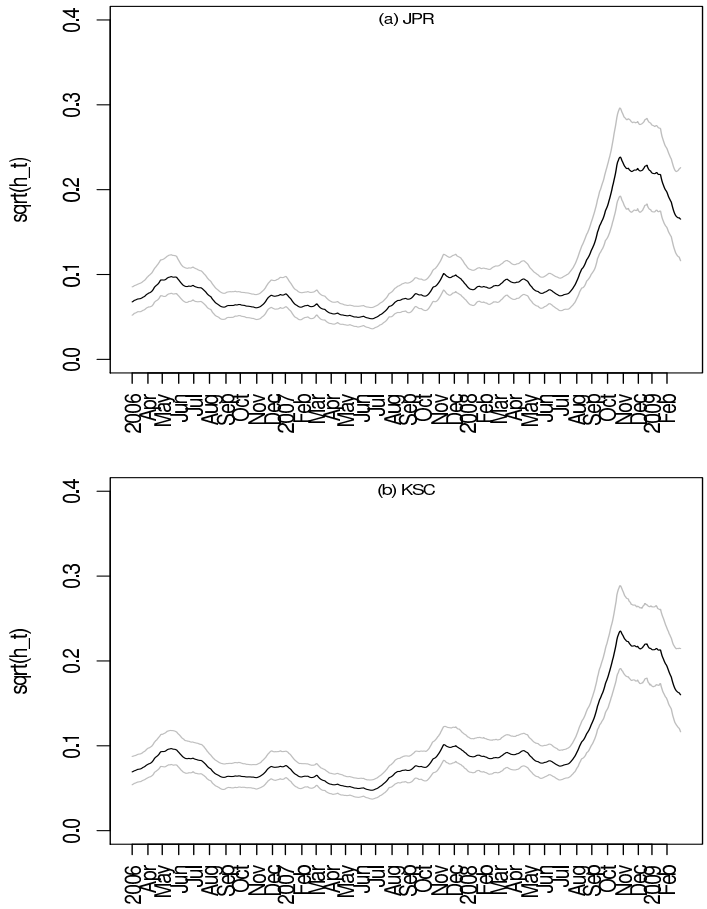
Models	RMSE	$\%MAE$
Single - move		
No RV	2.19	16.27
RV 30 sec	1.27	8.98
Multi - move KSC		
No RV	2.19	16.19
RV 30 sec	1.27	8.98

The sampling distributions are based on the 500 samples simulated with 1500 daily observations with prices measured without error. For each observation, we compute the estimation error of the posterior mean of $\sqrt{h_t}$. We report the root mean squared error, RMSE, and the average of the absolute values of % errors, $\%MAE$. The averages are computed over all the 750,000 observations.

TABLE 3.23 – Single vs multi-move - posterior analysis - UK £

	δ	σ_v
Single-move		
Mean	0.992	0.108
5% , 95%	[0.983 , 0.999]	[0.075 , 0.146]
Multi-move		
Mean	0.993	0.097
5% , 95%	[0.988 , 0.998]	[0.077 , 0.122]

FIGURE 3.21 – Single vs multi move - $p(h_t|R_1, \ldots, R_T)$ - UK £.



3.6.3 Particle Learning for SV-RV

The SV-RV model can be written as :

$$\begin{aligned} y_t &= \log h_t + z_t \\ \log h_t &= \alpha + \beta X_{t-1} + \delta \log h_{t-1} + \sigma_v v_t \end{aligned}$$

where $y_t = \log(r_t + c)^2$, c is a small quantity. Then, the error term $z_t \sim \log(\chi_1^2)$ is approximated by a discrete mixture of normals with fixed weights $\sum_{j=1}^{10} \pi_j Z_j$ with $Z_j \sim N(\mu_j, \sigma_j^2)$. An auxiliary indicator variable I_t is introduced to track the mixture component. $I_{t+1} = j$ indicates that the current volatility state is the j-th component of the mixture distribution :

$$p(y_{t+1}|\log h_{t+1}, I_{t+1} = j) = N(\log h_{t+1} + \mu_j, \sigma_j^2)$$

Assuming that, at time t, we have the particles $z_t = \left[(lh_t, s_t, \omega, I_t)^{(i)}\right]_{i=1}^N$, then the steps of the PL algorithm are :

- Compute weigths for $i = 1, \ldots, N$:

$$\begin{aligned} \omega_{t+1}^{(i)} &\propto p(y_{t+1}|lh_t^{(i)}, \theta^{(i)}, I_t^{(i)}) \\ &\propto f_n(y_{t+1}; \mu_{I_{t+1}^{(i)}} + \alpha^{(i)} + \beta^{(i)} X_t + \delta^{(i)} lh_t^{(i)}, \sigma^2_{I_{t+1}^{(i)}} + (\sigma_v^2)^{(i)}) \end{aligned}$$

- For $i = 1, \ldots, N$:
  - Re-sample from z_t with weights w_{t+1}

$$k(i) \sim Multi(\{\omega_{t+1}^{(i)}\}_{i=1}^N)$$

  - Draw indicator :

$$I_{t+1}^{(i)} \sim p(I_{t+1}|(lh_t, \omega)) = Multi_{10}(\pi_j)$$

  - Propagate states :

$$I_{t+1}^{(i)} \sim p(I_{t+1}|(lh_t, \omega) = Multi_{10}(\pi_j)$$

– Propagate states : log volatilities :

$$lh_{t+1}^{(i)} \sim p(lh_{t+1}|(lh_t, \omega, I_{t+1})^{k(i)}, y_{t+1})$$
$$lh_{t+1}^{(i)} \sim f_N(lh_{t+1}; mean_i, var_i)$$

where :

$$var_i = \left[(\sigma^{-2}_{I_{t+1}^{(k(i))}} + (\sigma_v^{-2})^{(k(i))}) \right]^{-1}$$

$$mean_i = var_i \left[\sigma^{-2}_{I_{t+1}^{(k(i))}} (y_{t+1} - \mu_{I_{t+1}^{(k(i))}}) + (\sigma_v^{-2})^{(k(i))}) (\alpha^{(i)} + \alpha_1^{(k(i))} X_t + \delta^{(k(i))} lh_t^{(k(i))}) \right]$$

– Update Recursive sufficient statistics :

- $s_{t+1}^{(i)} = (v_{t+1}, vs_t^{(i)}, m_{t+1}^{(i)}, C_{t+1}^{(i)})$
- $v_{t+1} = v_t + 1$
- $vs_{t+1} = vs_t + [m_t C_t^{-1} m_t + lh_{t+1}^2 - m_{t+1} C_{t+}^{-1} m_{t+1}]$
- $C_{t+1}^{-1} = C_t^{-1} + X_t X_t^\top$
- $m_{t+1} = C_{t+1}(C_t^{-1} m_t + X_t lh_{t+1})$

where $X_t = [1 \quad X_t \quad lh_t]$

– Sample parameters : :

- $\theta^{(i)} \sim p(\theta | s_{t+1}^{(i)})$
- $p(\alpha, \alpha_1, \delta|.) \sim N(m_{t+1}^{(i)}, C_{t+1}^{(i)})$

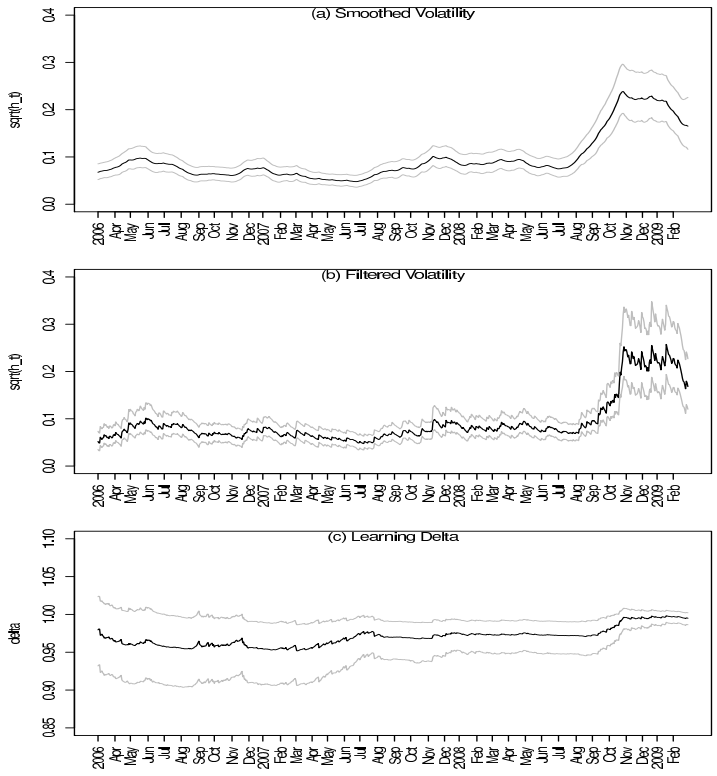
– $p(\sigma_v|.) \sim IG(v_{t+1}, vs_{t+1}^{(i)})$

Application

Compared to filtered volatilities, smoothed volatilities benefit from the information contained in future y's; therefore, one expects the posterior distributions of smoothed volatilities to have a tighter spread than those of filtered volatilities. Figure 3.22 demonstrates the magnitude of the difference for the UK Pound.

The top and middle plots show the 90% intervals for the smoothed and filtered volatility densities obtained by MCMC and the CJLP algorithm. The bottom plot demonstrates the evolution of the parameter δ as the filtering algorithm updates its posterior distribution.

FIGURE 3.22 – Smoothed volatility, $\sqrt{h_t}|y^T$, filtered volatility $\sqrt{h_t}|y^t$ and learning $\delta|y^t$



3.6.4 Real data : posterior parameter distribution

Tables 3.24 to 3.26 presents the posterior parameter distributions of the benchmark model (M0), Model 1 (M1) and Model 2 (M2) for all the series analyzed. Table 3.27 presents the posterior parameter distribution for the models applied to the $S\&P500$ index. For the currencies, we estimate the basic form of each model and for the country stock indices we estimate the correlated version.

Posterior means are based on 50,000 draws after discarding the first 5,000 draws. The rows entitled 5% and 95% report the posterior 5th and 95th percentiles.

TABLE 3.24 – Currencies and Indices : posterior parameter distribution for standard Model.

Models	α	δ	σ_v	ρ
Currencies - SV				
British Pound				
Average	-0.066	0.994	0.103	
5%	-0.148	0.986	0.079	
95%	-0.007	0.999	0.130	
Euro				
Average	-0.067	0.994	0.097	
5%	-0.145	0.986	0.076	
95%	-0.008	0.999	0.124	
Japanese Yen				
Average	-0.161	0.984	0.158	
5%	-0.309	0.970	0.119	
95%	-0.042	0.996	0.205	
Indices - SVC				
S&P 500				
Average	-0.081	0.991	0.168	-0.592
5%	-0.163	0.982	0.133	-0.731
95%	-0.013	0.998	0.210	-0.411
TSE				
Average	-0.074	0.991	0.128	-0.568
5%	-0.156	0.982	0.099	-0.745
95%	-0.011	0.999	0.162	-0.344
Nikkei 250				
Average	-0.129	0.984	0.172	-0.719
5%	-0.245	0.971	0.138	-0.824
95%	-0.032	0.996	0.212	-0.578
FTSE 100				
Average	-0.147	0.984	0.195	-0.645
5%	-0.261	0.971	0.157	-0.759
95%	-0.047	0.995	0.235	-0.494

TABLE 3.25 – Currencies and Indices : posterior parameter distribution for Model 1

Models	α	α_1	δ	σ_v	$\alpha 1 + \delta$	ρ
Currencies - SV-RV						
British Pound						
Average	-0.213	0.266	0.713	0.108	0.979	
5%	-0.479	0.153	0.552	0.076	0.953	
95%	-0.025	0.419	0.833	0.155	0.997	
Euro						
Average	-0.125	0.166	0.821	0.108	0.987	
5%	-0.318	0.082	0.691	0.075	0.968	
95%	-0.004	0.288	0.912	0.153	0.999	
Japanese Yen						
Average	-0.139	0.157	0.830	0.202	0.987	
5%	-0.358	0.061	0.701	0.129	0.965	
95%	-0.013	0.278	0.929	0.294	0.999	
Indices - SVC-RV						
S&P 500						
Average	-0.106	0.559	0.421	0.311	0.980	-0.081
5%	-0.449	0.283	-0.050	0.134	0.942	-0.297
95%	0.125	1.021	0.703	0.458	0.999	0.134
TSE						
Average	0.029	0.471	0.488	0.127	0.959	-0.032
5%	-0.366	0.186	-0.141	0.080	0.909	-0.253
95%	0.662	1.084	0.791	0.198	0.995	0.184
Nikkei 250						
Average	0.984	1.217	-0.303	0.156	0.914	-0.157
5%	-0.750	0.418	-0.665	0.087	0.751	-0.528
95%	2.531	1.644	0.513	0.277	0.998	0.262
FTSE 100						
Average	0.268	0.773	0.196	0.123	0.969	-0.042
5%	-0.260	0.328	-0.390	0.079	0.919	-0.334
95%	0.936	1.369	0.649	0.183	0.998	0.256

TABLE 3.26 – Currencies and Indices : posterior parameter distribution for Model 2

Models	α	δ	σ_h	β_0	β_1	σ_{rv}	ρ
Currencies - SV-RV							
British Pound							
Average	-0.120	0.988	0.145	0.088	1.005	0.279	
5%	-0.230	0.978	0.120	-0.830	0.917	0.261	
95%	-0.025	0.997	0.175	1.153	1.106	0.297	
Euro							
Average	-0.099	0.990	0.124	-0.421	0.960	0.326	
5%	-0.197	0.981	0.100	-1.365	0.871	0.308	
95%	-0.018	0.998	0.151	0.649	1.062	0.344	
Japanese Yen							
Average	-0.617	0.940	0.316	-0.907	0.898	0.279	
5%	-0.899	0.912	0.262	-1.753	0.815	0.247	
95%	-0.363	0.964	0.376	-0.013	0.986	0.309	
Indices - SVC-RV							
S&P 500							
Average	-0.194	0.979	0.273	-0.355	0.965	0.415	-0.543
5%	-0.322	0.965	0.236	-0.963	0.899	0.387	-0.648
95%	-0.074	0.992	0.314	0.295	1.037	0.443	-0.435
TSE							
Average	-0.178	0.980	0.202	-0.075	1.071	0.415	-0.407
5%	-0.299	0.966	0.172	-0.868	0.983	0.389	-0.534
95%	-0.064	0.992	0.234	0.783	1.167	0.442	-0.279
Nikkei 250							
Average	-0.268	0.968	0.235	-2.013	0.882	0.425	-0.584
5%	-0.428	0.949	0.196	-2.699	0.801	0.397	-0.697
95%	-0.123	0.985	0.277	-1.243	0.974	0.453	-0.463
FTSE 100							
Average	-0.136	0.985	0.190	-0.855	0.966	0.410	-0.557
5%	-0.235	0.974	0.167	-1.510	0.894	0.387	-0.680
95%	-0.043	0.995	0.217	-0.161	1.043	0.433	-0.421

TABLE 3.27 – $S\&P500$: Posterior parameter distribution - M2

Models	α	δ	σ_v	β_0	β_1	σ_{η_1}	γ_0	γ_1	σ_{η_2}	ρ
M2 - RV										
Mean	-0.194	0.979	0.273	-0.355	0.965	0.415				-0.543
5%	-0.322	0.965	0.236	-0.963	0.899	0.387				-0.648
95%	-0.074	0.992	0.314	0.295	1.037	0.443				-0.435
M2 - VIX										
Mean	-0.049	0.994	0.137				-1.374	0.811	0.073	0.090
5%	-0.104	0.988	0.123				-1.922	0.751	0.064	0.014
95%	-0.003	0.998	0.153				-0.862	0.868	0.081	0.166
M2 - RV - VIX										
Mean	-0.045	0.995	0.133	-0.488	0.958	0.566	-1.361	0.813	0.076	0.063
5%	-0.098	0.989	0.120	-1.076	0.893	0.542	-1.815	0.63	0.069	-0.020
95%	-0.002	0.999	0.146	0.155	1.030	0.590	-0.857	0.870	0.084	0.14

General Conclusion

This thesis contains three chapters that aim to contribute to the literature related to the estimation of state space models and latent volatility using stochastic volatility models.

In the first chapter, we have made several contributions to the literature of linear Gaussian state space models. We explicitly derive the precision (inverse of variance) and co-vector (precision times vector) of the conditional distribution of the state given the data. We propose a new efficient method for drawing states in state space models. We compare the computational efficiency of various methods for drawing states showing that the Cholesky Factor Algorithm and our new method are much more computationally efficient than methods based on the Kalman filter. The method we propose is best suited for high dimension problems or when repeated draws of the state are required. We consider an application of our methods to the evaluation of the log-likelihood function for a multivariate Poisson model with latent count intensities.

In the second chapter, we have contributed to the estimation of multivariate stochastic volatility models. We presented a new flexible approach which allows capturing many of the stylized facts observed in returns. We allow for different types of dependence. We can model time-varying conditional correlation matrices by incorporating factors in the return equation, where the factors are independent SV processes with Student's t innovations. Furthermore, we can incorporate copulas to allow conditional return dependence given volatility, allowing different Student's t marginals to capture return heterogeneity.

The two key advantage of our method over auxiliary mixture approaches is that our method is less model specific and it is an exact method. We draw volatilities one-at-a-time in the cross section dimension and as a block in the time dimension using the HESSIAN method introduced by McCausland (2012). This requires that the multivariate state sequence be a Gaussian first-order vector autoregressive process and that the conditional distribution of the observed vector depend only on the contemporaneous state vector. This requirement is satisfied for a wide variety of state space models, including but not limited to multivariate stochastic volatility models.

Finally, in the third chapter we have contributed to assess the information content of different realized volatility estimators to the estimation and forecasting of volatility. We considered two econometric specifications to incorporate realized volatility into the inference on latent volatility.

After our performance evaluation, we conclude that the ability of realized volatility to improve forecasts is seriously affected by price noise, and highlight the importance of using adjustments that are robust to the presence of noise. We also find that the second econometric specification, where the volatility proxy is explicitly linked to the latent volatility, results in better in-sample and out-of-sample estimation of volatility than the first specification. This is especially true for multi period forecasts.

The results from our empirical application using currencies and stock indices showed that the odds are in favor of the models that incorporate the information of realized volatility estimators. Besides, this evidence has increased for the recent financial crisis period of 2008-2009. However, the out-of-sample evaluation of volatility forecasts does not allow us to conclude whether one model dominates the the others.

Bibliographie

Aguilar, O., and West, M. (2000). 'Bayesian dynamic factor models and variance matrix discounting for portfolio allocation', Journal of Business and Economic Statistics, 18 : 338–357.

Ait-Sahalia, Mykland, and Zhang (2005). 'How often to sample a continous-time process in the presence of market microstructure noise', Review of Financial Studies, 18 : 351–416.

Andersen, and Bollerslev (1998). 'Answering the Skeptics : Yes, Standard Volatility Models Do Provide Accurate Forecasts', International Economic Review, 39 : 885–905.

Andersen, Bollerslev, and Diebold (2007). 'Roughing it up : Including jump components in the measurement, modeling and forecasting of return volatility', Review of Economics and Statistics, 89 : 701–720.

Andersen, Bollerslev, Diebold, and Lays (2001). 'The Distribution of Exchange Rate Volatility', Journal of the American Statistical Association, 96 : 42–55.

——— (2003). 'Modeling and Forecasting Realized Volatility', Econometrica, 71 : 579–625.

Ang, A., and Chen, J. (2002). 'Asymmetric correlations of equity portfolios', Journal of Financial Econometrics, 63 : 443–494.

Asai, and McAleer (2009). 'The structure of dynamic correlations in multivariate stochastic volatility models', Journal of Econometrics, 150 : 182–192.

Barndorff-Nielsen, E., O., Hansen, P. R., Lunde, A., and Shephard, N. (2008). 'Designing Realised Kernels to Measure the Ex-Post Variation of Equity Prices in the Presence of Noise', Econometrica, 76 : 1481–1536.

Barndorff-Nielsen, Hansen, Lunde, and Shephard (2006). 'Subsampling realised kernels', Unpublished manuscript.

Barndorff-Nielsen, and Shephard (2004). 'Power and bipower variation with stochastic volatility and jumps', Journal of Financial Econometrics, 2 : 1–48.

Bauwens, L., Laurent, S., and Rombouts, J. V. K. (2006). 'Multivariate GARCH : A survey', Journal of Applied Econometrics, 21 : 79–109.

Boivin, J., and Giannoni, M. (2006). 'DSGE Models in a Data-Rich Environment', Working Paper 12772, National Bureau of Economic Research.

Brandt, M. W., and Jones, C. S. (2005). 'Bayesian Range-Based Estimation of Stochastic Volatility Models', Finance Research Letters 2, pp. 201–209.

Caravalho, Johannes, Lopes, and Polson (2010). 'Particle Learning and Smoothing', To apper in Statistical Science.

Carter, C. K., and Kohn, R. (1994). 'On Gibbs Sampling for State Space Models', Biometrika, 81(3) : 541–553.

——— (1996). 'Markov chain Monte Carlo in conditionally Gaussian State Space Models', Biometrika, 83(3) : 589–601.

Carvalho, C. M., and West, M. (2006). 'Dynamic matrix-variate graphical models', Bayesian Analysis, 1 : 1–29.

Chan, J. C. C., and Jeliazkov, I. (2009). 'Efficient Simulation and Integrated Likelihood Estimation in State Space Models', Working paper.

Chib, S., Nardari, F., and Shephard, N. (2002). 'Markov Chain Monte Carlo Methods for Stochastic Volatility Models', Journal of Econometrics, 108 : 281–316.

——— (2006). 'Analysis of High Dimensional Multivariate Stochastic Volatility Models', Journal of Econometrics, 134 : 341–371.

Chib, S., Omori, Y., and Asai, M. (2009). 'Multivariate Stochastic Volatility.', in T. G. Andersen, R. A. Davis, J.-P. Kreib, and T. Mikosch (eds.), Handbook of Financial Time Series, pp. 365–400. Springer.

Corsi, F. (2009). 'A Simple Approximate Long-Memory Model of Realized Volatility', Journal of Financial Econometrics, 7 : 174–196.

Danielsson, J. (1998). 'Multivariate stochastic volatility models : Estimation and comparison with VGARCH models.', Journal of Empirical Finance, 5 : 155–173.

de Jong, P., and Shephard, N. (1995). 'The Simulation Smoother for Time Series Models', Biometrika, 82(1) : 339–350.

Durbin, J., and Koopman, S. J. (1997). 'Monte Carlo maximum likelihood estimation for non-Gaussian state space models', Biometrika, 84(3) : 669–684.

——— (2001). Time series analysis by state space methods, vol. 24 of Oxford Statistical Science Series. Oxford University Press, Oxford.

——— (2002). 'A Simple and Efficient Simulation Smoother for State Space Time Series Analysis', Biometrika, 89(3) : 603–615.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). 'The Generalized Dynamic Factor Model : Identification and Estimation', Review of Economics and Statistics, 82(4) : 540–554.

Frühwirth-Schnatter, S. (1994). 'Data augmentation and Dynamic Linear Models', Journal of Time Series Analysis, 15 : 183–202.

Frühwirth-Schnatter, S., and Frühwirth, R. (2007). 'Auxiliary mixture sampling with applications to logistic models', Computational Statistics and Data Analysis, 51 : 3509–3528.

Frühwirth-Schnatter, S., and Wagner, H. (2006). 'Auxiliary mixture sampling for parameter-driven models of time series of counts with applications to state space modelling', Biometrika, 93 : 827–841.

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996). 'Efficient Metropolis Jumping Rules', pp. 599–607.

Geweke, J. (1994). 'Bayesian Comparison of Econometric Models', Working Paper, Federal Reserve Bank of Minneapolis Research Department.

Geweke, J. (2004). 'Getting it Right : Joint Distribution Tests of Posterior Simulators', Journal of the American Statistical Association, 99 : 799–804.

Gourieroux, C. (2006). 'Continous time Wishart process for stochastic risk.', Econometric Reviews, 25 : 177–217.

Gourieroux, C., Jasiak, J., and Sufana, R. (2004). 'The Wishart autorregresive process of multivariate stochastic volatility.', Discussion Paper, University of Toronto.

Hamilton, J. D. (1994). Time Series Analysis. Princeton University Press, Princeton, NJ.

Han (2006). 'The economics value of volatility modelling : Asset allocation with a high dimensional dynamic latent factor multivariate stochastic volatility model.', Review of Financial Studies, 19 : 237–271.

Harvey, A. C., Ruiz, E., and Shephard, N. (1994). 'Multivariate stochastic variance models', Review of Economic Studies, 61 : 247–264.

Hasbrouck, J. (1993). 'Assessing the Quality of a Security Market : A New Approach to Transaction Cost Measurement', Review of Financial Studies, 6 : 191–212.

Heaton, C., and Solo, V. (2004). 'Identification of causal factor models of stationary time series', Econometrics Journal, 7(2) : 618–627.

Heber, G., Lunde, A., Shephard, N., and Sheppard, K. (2009). 'OMI realised measure library. Version 0.1', Oxford-Man Institute, University of Oxford.

Jacquier, E., Polson, N., and Rossi, P. (1994). 'Bayesian Analysis of Stochastic Volatility Models', Journal of Business and Economic Statistics, 12(4) : 371–388.

Jacquier, E., Polson, N., and Rossi, P. (1995). 'Models and prior distributions for multivariate stochastic volatility.', Working paper, Graduate School of Business, University of Chicago.

Jacquier, E., Polson, N., and Rossi, P. (2004). 'Bayesian Analysis of Stochastic Volatility Models with Leverage Effect and Fat tails', Journal of Econometrics, pp. 185–212.

Kim, S., Shephard, N., and Chib, S. (1998). 'Stochastic Volatility : Likelihood Inference and Comparison with ARCH Models', Review of Economic Studies, 65(3) : 361–393.

Knorr-Held, L., and Rue, H. (2002). 'On Block Updating in Markov Random Field Models for Disease Mapping', Scandinavian Journal of Statistics, 29 : 597–614.

Kolev, N., dos Anjos, U., and de M. Mendez, B. V. (2006). 'Copulas : a review and recent developments.', Stochastic Models, 22 : 617–660.

Koopman, S. J., Jungbackera, B., and Hol, E. (2005). 'Forecasting daily variability of the SP 100 stock index using historical, realised and implied volatility measurements', Journal of Empirical Finance, 12 : 445–475.

Longin, F., and Solnik, B. (2001). 'Extreme correlations of international equity markets.', Journal of Finance, 56 : 649–676.

Lopes, H. F., and Carvalho, C. M. (2007). 'Factor stochastic volatility with time varying loadings and Markov switching regimes.', Journal of Statistical Planning and Inference, 137 : 3082–3091.

McCausland, W. J. (2012). 'The HESSIAN Method : Highly Efficient State Smoothing, in a Nutshell', Forthcoming, Journal of Econometrics.

Omori, Y., Chib, S., Shephard, N., and Nakajima, J. (2007). 'Stochastic volatility with leverage : fast and efficient likelihood inference', Journal of Econometrics, 140 : 425–449.

Parkinson, M. (1980). 'The Extreme Value Method for Estimating the Variance of the Rate of Return', The Journal of Business, 53 : 61–65.

Patton (2008). 'Data-Based Ranking of Realised Volatility Estimators', Working Paper, University of Oxford.

Patton, A. J. (2009). 'Copula-Based Models for Financial Time Series.', in T. G. Andersen, R. A. Davis, J.-P. Kreib, and T. Mikosch (eds.), Handbook of Financial Time Series, pp. 767–784. Springer.

Philipov, A., and Glickman, M. E. (2006). 'Multivariate stochastic volatility via Wishart processes', Journal of Business and Economic Statistics, 24 : 313–328.

Pitt, M. K., and Shephard, N. (1999). 'Time varying covariances : a factor stochastic volatility approach with discussion.', in J. Bernardo, J. Berger, A. Dawid, and A. Smith (eds.), Bayesian Statistics, pp. 547–570. Oxford University Press, Oxford.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). Numerical recipes in C. Cambridge University Press, Cambridge, second edn., The art of scientific computing.

Rue, H. (2001). 'Fast Sampling of Gaussian Markov Random Fields with Applications', Journal of the Royal Statistical Society Series B, 63 : 325–338.

Shephard, N., and Pitt, M. K. (1997). 'Likelihood Analysis of Non-Gaussian Measurement Time Series', Biometrika, 84(3) : 653–667.

Sklar, A. (1959). 'Fonctions de répartition à n dimensions et leurs marges.', Publications de l'Institut Statistique de l'Université de Paris, 8 : 229–231.

Smith, M., and Pitts, A. (2006). 'Foreign exchange intervention by the Bank of Japan : Bayesian analyisis using a bivariate stochastic volatility model.', Econometric Reviews, 25 : 425–451.

So, M. K. P., Li, W. K., and Lam., K. (1997). 'Multivariate modelling of the autorregresive random variance process.', Journal of Time Series Analysis, 18 : 429–446.

Song, J., and Belin, T. R. (2008). 'Choosing an appropriate number of factors in factor analysis with incomplete data', Computational Statistics and Data Analysis, 52(7) : 3560–3569.

Stock, J. H., and Watson, M. W. (1999). 'Forecasting Inflation', Journal of Monetary Economics, 44 : 293–335.

——— (2002). 'Macroeconomic Forecasting Using Diffusion Indexes', Journal of Business and Economic Statistics, 20 : 147–162.

Stroud, J. R., Müller, P., and Polson, N. G. (2003). 'Nonlinear State-Space Models With State-Dependent Variances', Journal of the American Statistical Association, 98 : 377–386.

Takahashi, M., Omori, Y., and Watanabe, T. (2009). 'Estimating Stochastic Volatility Models Using Daily Returns and Realized Volatility Simultaneously', Computational Statistics and Data Analysis, 53 : 2404–2426.

Vandebril, R., Mastronardi, N., and Van Barel, M. (2007). 'A Levinson-like algorithm for symmetric strongly nonsingular higher order semiseparable plus band matrices', Journal of Computational and Applied Mathematics, 198(1) : 74–97.

Vihola, M. (2011). 'Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate', arXiv eprint.

Wright (1999). 'Testing for a Unit Root in the Volatility of Asset Returns', Journal of Applied Econometrics, 14 : 309–318.

Yu, J. (2005). 'On leverage in a stochastic volatility model', Journal of Econometrics, 127 : 165–178.

Yu, J., and Meyer, R. (2006). 'Multivariate stochastic volatility models : Bayesian estimation and model comparison.', Econometric Reviews, 25 : 361–384.

Zhang, Mykland, and Ait-Sahalia (2005). 'A tale of two time scales : Determining integrated volatility with noisy high frequency data', Journal of the American Statistical Association, 100 : 1394–1411.

Zhou, B. (1996). 'High-Frequency Data and Volatility in Foreign-Exchange Rates', Journal of Business and Economic Statistics, 14 : 45–52.