BAYESIAN MODELLING

OF

ULTRA HIGH-FREQUENCY FINANCIAL DATA

by

Golnaz Shahtahmassebi

A thesis submitted to the University of Plymouth in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

School of Computing and Mathematics

Faculty of Science and Technology

University of Plymouth

October 2011

Golnaz Shahtahmassebi

Bayesian modelling of ultra high-frequency financial data

Abstract

The availability of *ultra high-frequency* (UHF) data on transactions has revolutionised data processing and statistical modelling techniques in finance. The unique characteristics of such data, e.g. discrete structure of price change, unequally spaced time intervals and multiple transactions have introduced new theoretical and computational challenges. In this study, we develop a Bayesian framework for modelling integer-valued variables to capture the fundamental properties of price change. We propose the application of the *zero inflated Poisson difference* (ZPD) distribution for modelling UHF data and assess the effect of covariates on the behaviour of price change. For this purpose, we present two modelling schemes; the first one is based on the analysis of the data after the market closes for the day and is referred to as *off-line* data processing. In this case, the Bayesian interpretation and analysis are undertaken using *Markov chain Monte Carlo* methods. The second modelling scheme introduces the dynamic ZPD model which is implemented through *Sequential Monte Carlo* methods (also known as particle filters). This procedure enables us to update our inference from data as new transactions take place and is known as *online* data processing. We apply our models to a set of FTSE100 index changes. Based on the probability integral transform, modified for the case of integer-valued random variables, we show that our models are capable of explaining well the observed distribution of price change. We then apply the deviance information criterion and introduce its sequential version for the purpose of model comparison for off-line and online modelling, respectively. Moreover, in order to add more flexibility to the tails of the ZPD distribution, we introduce the zero inflated generalised Poisson difference distribution and outline its possible application for modelling UHF data.

Contents

Abstract **v**

List of figures **xiii**

List of tables **xix**

Acknowledgements **xxi**

Author's declaration **xxiii**

Abbreviations **xxv**

1 Introduction **1**

1.1 Overview 1

1.2 Empirical characteristics of UHF data 3

1.3 FTSE100 index futures 4

1.4 The state of the problem 7

1.4.1 Price change 7

1.4.2 Modelling price change 9

2 A Bayesian perspective of the decomposition model **11**

2.1 Introduction 11

2.2 Decomposition of index change 12

2.2.1 Modelling components of index change 13

2.3 Bayesian inference 14

2.3.1 Metropolis Hastings algorithm . . . 17
2.3.1.1 Gelman convergence diagnostics . . . 18
2.3.2 Predicting index movements . . . 18
2.3.3 Model checking . . . 20
2.3.3.1 Probability integral transform . . . 20
2.4 Results . . . 21
2.4.1 The activity of indices . . . 21
2.4.2 The direction of index change . . . 22
2.4.3 The size of index change . . . 23
2.5 Summary . . . 24

3 Zero inflated Poisson difference model **27**
3.1 Introduction . . . 27
3.2 Poisson difference distribution . . . 29
3.2.1 Properties . . . 29
3.2.2 Zero inflated Poisson difference distribution . . . 31
3.3 A model for index change . . . 34
3.3.1 Bayesian inference . . . 35
3.3.2 Metropolis Hastings algorithm . . . 36
3.3.3 Predictive distribution . . . 36
3.3.4 Model checking . . . 37
3.4 Results . . . 38
3.4.1 Simulation study . . . 39
3.4.2 FTSE100 futures . . . 40
3.5 Summary . . . 41

4 Modelling via particle filters **45**
4.1 Introduction . . . 45
4.2 Inference in hidden Markov models . . . 47
4.2.1 Hidden Markov models . . . 47

4.2.2 Sequential Monte Carlo methods 50
4.2.2.1 Degeneracy and resampling 53
4.2.2.2 Designs and issues 54
4.2.3 Improvement and extension 56
4.2.3.1 Auxiliary particle filters 57
4.2.3.2 Auxiliary particles filters with unknown static parameters 58
4.3 A dynamic model 63
4.3.1 Dynamic Poisson difference model 65
4.3.2 Dynamic zero inflated Poisson difference model 65
4.3.3 Predictive distribution 66
4.3.4 Diagnostics 67
4.3.4.1 Sequential deviance information criterion 68
4.4 Results 69
4.4.1 Simulation study 69
4.4.1.1 Dynamic Poisson difference model 69
4.4.1.2 Dynamic zero inflated Poisson difference model 72
4.4.2 A dynamic model for the FTSE100 index change 75
4.4.2.1 A DZPD model with covariates 76
4.4.2.2 Nonlinear state equation 77
4.4.2.3 Degeneracy 80
4.5 Summary 83

5 Generalised Poisson difference distribution **85**

5.1 Introduction 85
5.2 The generalised Poisson distribution 86
5.2.1 Generating functions 86
5.2.2 Moments of generalised Poisson distribution 88
5.2.3 Graphical presentation of GP distribution models 90
5.3 The generalised Poisson difference distribution 91
5.3.1 Differences of two GP variates 92

5.3.1.1 Cumulant generating function 94
5.3.2 Graphical presentation of GPD distribution models 98
5.3.3 Application: a model for index change 101
5.4 Summary 103

6 Conclusion **105**
6.1 Summary 105
6.2 Further developments and outlook 107
6.2.1 Bivariate models for price change and time duration 108
6.2.1.1 Off-line modelling 109
6.2.1.2 Online modelling 110
6.2.2 Modelling the proportion of excess of zeros 110
6.2.3 Extending the analysis to more than one trading day 111
6.2.3.1 Hierarchical models 111
6.2.3.2 Dynamic models 112
6.2.4 Alternative state space processes 112
6.2.5 Improvements for computational algorithms 112
6.2.6 General remarks 113

A ADS model: Gelman and Rubin's convergence diagnostics, and traces of the MCMC chains **115**

B ZPD model: Gelman and Rubin's convergence diagnostics, and traces of the MCMC chains **119**

C Dynamic zero inflated Poisson difference model: the graphical presentation **127**
C.1 DPD model: Graphical presentation of the results for the six simulated sets 128
C.2 DZPD model: Graphical presentation of the results for the three simulated data sets 135
C.3 DZPD model: Graphical presentation of the results for the the FTSE100 data set 139

D R codes **141**

D.1 ADS model 142

D.2 Zero inflated Poisson difference model 145

D.3 Dynamic zero inflated poisson models 149

D.4 Generalised Poisson difference distribution 158

Glossary of terms **161**

List of references **163**

List of Figures

1.1 FTSE100 index futures, the discrete structure of price process can be observed for (a) March 25, and (b) March 26, 2008. 5

1.2 One minute of FTSE100 indices presents the main features of UHF data, e.g. price discreteness, unequally spaced time intervals and multiple transactions. 6

2.1 A standard uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution. The first and the last 10% of tails are shown on a larger scale. 25

3.1 The comparison of the empirical distribution of the observed index change and the density of the PD distribution with the methods of moment estimates of λ_1 and λ_2. 31

3.2 The ZPD distribution with parameters $\lambda_1 = \lambda_2 = 2$ and four values of p. . 32

3.3 The behaviour of the variance for different values of λ_1 and λ_2 over the range of $p \in [0,1]$ 33

3.4 A standard uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution, (a) simulated data, (b) FTSE100 index change. The first and the last 10% of tails are shown on a larger scale. 42

4.1 Conditional structure for an HMM. SMC methods enable us to make inference about the unobserved process given the observed process. 48

4.2 Six sets of simulated data of size $n = 500$ from the DPD model. Observed values $\{z_t\}$, $\{\lambda_{1,t}\}$ and $\{-\lambda_{2,t}\}$, are shown by black points, solid red and blue lines, respectively. 70

4.3 (a) The median (red line), the 95% credible intervals (grey area) of the posterior predictive distribution and the true observed values (black points) for Dataset 4. (b) A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution. 72

4.4 Three sets of simulated data from the DZPD model, along with values of $\{\lambda_{1,t}\}$ (red line) and $\{-\lambda_{2,t}\}$ (blue line). 74

4.5 (a) The 95% credible intervals (grey area) of the posterior predictive distribution and the true observed values (black points) for Dataset 8. (b) A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution. 74

4.6 Difference of the DIC_t of the DPD model, $\mathrm{DIC}_{M_{\mathrm{DPD}}}$, from the DZPD model, $\mathrm{DIC}_{M_{\mathrm{DZPD}}}$. 75

4.7 One day index change from 25th of March 2008. 75

4.8 FTSE100 transactions data from 25/03/2008, (a) the number of transactions and (b) the time duration between two consecutive transactions, in 5-minute time intervals. 78

4.9 The posterior mean and 95% credible intervals (grey area) of the unobserved processes (the first row) and the parameters of model M_1 for the FTSE100 index change. 81

4.10 The posterior mean and 95% credible intervals (grey area) of the unobserved processes (the first row) and the parameters of model M_2 for the FTSE100 index change. 82

4.11 A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution (a) M_1, (b) M_2. The first and the last 10% of tails are shown on a larger scale. 83

4.12 The difference of the DIC_t of the M_1 from M_2. Negative values indicate a better fit of the model M_2 compare to the model M_1. 84

5.1 The GP distribution for $\theta = 0.3$ 91

5.2 The GP distribution for $\lambda = 4$ 92

5.3 The GP distribution for $\lambda_1 = \lambda_2 = 2$ 99

5.4 The GP distribution for $\theta = 0.4$ 100

5.5 The GPD distribution for different values of $\lambda_1 = \lambda_2 = \lambda$ and θ 101

6.1 A ZPD duration model jointly models the duration and index change. . . 109

A.1 Gelman and Rubing's R statistics over 10000 iterations for the parameter vector β of model M_{A} in (2.3). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistics. 116

A.2 Traces of the sampled values of the vector of parameters β (left) and the estimates of their posterior distributions (right) for model M_{A} in (2.3) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in. 116

A.3 Gelman and Rubing's R statistics over 10000 iterations for the parameter vector γ of model M_{D} in (2.4). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistic. 117

A.4 Traces of the sampled values of the vector of parameters γ (left) and the estimates of their posterior distributions (right) for model M_{D} in (2.4) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in. 117

A.5 Gelman and Rubing's R statistics over 10000 iterations for the parameter vector θ of model M_{S} in (2.5). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistic. 118

A.6 Traces of the sampled values of the vector of parameters θ (left) and the estimates of their posterior distributions (right) for model M_S in (2.5) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in. 118

B.1 Gelman and Rubing's R statistics over 10000 iterations for the model parameter vector ψ, for simulated data set. Solid black line is the median and dashed red line represents the 95% credible intervals for R statistics. . 121

B.2 Traces of the sampled values of the vector of parameters ψ of the ZPD model fitted to a set of simulated data. We set $m = 10000$ and discarded the first 5000 samples as burn-in. 122

B.3 The the estimates of the posterior distributions of the vector of parameters ψ of the ZPD model fitted to a set of simulated data after discarding the first 5000 samples as burn-in. 123

B.4 Gelman and Rubing's R statistics over 10000 iterations for the model parameter vector ψ, for FTSE100 data set. Solid black line is the median and dashed red line represents 95% credible intervals for R statistics. . . . 124

B.5 Traces of the sampled values of the vector of parameters ψ of the ZPD model fitted to the FTSE100 index change. We set $m = 10000$ and discard the first 5000 iterations samples as burn-in. 125

B.6 The estimates of the posterior distributions of the vector of parameters ψ of the ZPD model fitted to the FTSE100 index change after discarding the first 5000 samples as burn-in. 126

C.1 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 1 128

C.2 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 2 129

C.3 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 3 130
C.4 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 4 131
C.5 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 5 132
C.6 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 6 133
C.7 The DPD model, a total of 20 replications of the LWF algorithm, each one based on $N = 5000$ 134
C.8 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 7. 135
C.9 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 8. 136
C.10 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 9. 137
C.11 True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model fitted to the Dataset 8. 138
C.12 The posterior mean, the 95% credible intervals (grey area) of the unobserved processes (the first row) and the DZPD model parameters for the first 1000 index changes of the FTSE100 data set. 139

List of Tables

1.1 Relative frequencies of index change in multiple of tick size. 7

2.1 The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the index activity. 22

2.2 The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the direction of index change. 22

2.3 The plug-in estimates for the probability of the direction of index change. 23

2.4 The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the size of index change. 24

3.1 The true values, the posterior means and the 95% credible intervals of the posterior distribution of the ZPD model parameters based on the simulated data. 39

3.2 The DIC, the point-estimate, the average deviances and the estimated number of parameters for each of five models fitted to the simulated index change. 39

3.3 The posterior means, the standard deviation and the 95% credible intervals of the posterior distribution of parameters of the ZPD model based on FTSE100 data. 40

3.4 The DIC, the point-estimate, the average deviances and the estimated number of parameters for each of five models fitted to the FTSE100 index change. 40

4.1 Parameter values for the six simulated examples from the DPD model. . . 69

4.2 Parameter values for the three simulated examples from the DZPD model. 73

A.1 Gelman and Rubin's R statistics for the parameter vector β of model M_{A} in (2.3). Values less than 1.1 suggest that we could assume the convergence of the MCMC chains. 115

A.2 Gelman and Rubin's R statistics for the parameter vector γ of model M_{D} in (2.4). Values less than 1.1 suggest that we could assume the convergence of the MCMC chains. 115

A.3 Gelman and Rubin's R statistics for the parameter vector θ of model M_{S}. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains. 115

B.1 Acceptance rate of the MCMC chains for the ZPD model in (3.10) fitted to the simulated data and the FTSE100 index change. 120

B.2 Gelman and Rubin's R statistics for the parameter vectors, α and β, of the ZPD model in (3.10) for the simulated data set. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains. . . . 120

B.3 Gelman and Rubin's R statistics for the parameter vectors, α and β, of the ZPD model in (3.10) for FTSE100 data set. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains. 120

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another provided their valuable assistance in the preparation and completion of this study.

First and foremost, my utmost gratitude is to Dr. Rana Moyeed, my Director of Studies whose sincerity and encouragement I will never forget. Dr. Moyeed has been my inspiration as I hurdled over all the obstacles in the completion of this research work. I am also indebted to Dr Paul Hewson and Dr Julian Stander for their unselfish and unfailing support as my second and third supervisors, respectively. I would like to thank Dr Evan Kirk at Cantab Capital Partners LLP for providing the FTSE100 data set. I would also like to extend my thanks to my friends and fellow colleagues in the PhD program; particularly to Christopher Harvey, Yanwei Lu and Poonam Sharma for their help and encouragement.

Lastly, I would like to thank my family for all their love and inspiration: my parents who raised me with a love for science and supported me in all my pursuits, and my brother and my sister for their unabated support.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of University of Plymouth Research Scholarship, Faculty of Science and Technology, University of Plymouth.

Relevant scientific seminars and conferences were regularly attended at which work was often presented. One paper was prepared for publication.

Scientific papars :

The Application of the Zero Inflated Poisson Difference Distribution in the High-Frequency Financial Data Modelling.
Golnaz Shahtahmassebi, Rana Moyeed, Paul Hewson and Julian Stander.
(To be submitted.)

Presentations given :

2011: *High-Frequency Financial Data Analysis via Particle Filters*.
Research Students' Conference, University of Cambridge, Cambridge, UK.

2010: *Statistical and Mathematical Models for Predicting Financial Behaviour*.
Research Students' Conference, University of Warwick, Warwick , UK.

2009: *Mathematical and Statistical Models for Predicting Financial Behaviour*.
University of Plymouth, Plymouth, UK.

An Introduction to Bayesian Modelling of High-Frequency Financial Data.
Ferdowsi University of Mashhad, Mashhad, Iran.

Bayesian Modelling of Trade-By-Trade Price.
Research Students' Conference, University of Lancaster , Lancaster, UK.

Conference and courses attended :

2009: *Ordinary Meeting-Young Statistician*.
Royal Statistical Society, London, UK.

Royal Statistical Society Conference.

University of Edinburgh, Edinburgh, UK.

Market Liquidity Conference.
Oxford-Man Institute of Quantitative Finance, Oxford, UK.

Academy for PhD Training in Statistics:

- Statistical Asymptotics (April, University of Southampton, UK).
- Statistical Modelling (April, University of Southampton, UK).
- Computational Statistics (July, University of Bath, UK).
- Stochastic Processes (July, University of Bath, UK).

Word count for the main body of this thesis: 22,000

Signed: _____

Date: _____

Abbreviations

ACD	Autoregressive Conditional Duration
ACM	Autoregressive Conditional Multivariate
ADS	Active Direction Size
APF	Auxiliary Particle Filter
AR	Autoregressive
DIC	Deviance Information Criterion
DPD	Dynamic Poisson Difference
DZPD	Dynamic Zero Inflated Poisson Difference
FTSE	Financial Times Stock Exchange
GP	Generalised Poisson
GPD	Generalised Poisson Difference
HF	High-Frequency
LIFFE	London International Financial Futures and option Exchange
HMM	Hidden Markov Model
LWF	Liu and West's Filter
MCMC	Markov Chain Monte Carlo
M-H	Metropolis-Hastings
SMC	Sequential Monte Carlo
PD	Poisson Difference

PF	Particle Filters
PIT	Probability Integral Transform
SSM	State Space Model
UHF	Ultra High-Frequency
ZGPD	Zero inflated Generalised Poisson Difference
ZPD	Zero inflated Poisson Difference
cgf	Cumulant Generating Funtion
pgf	Probability Generating Funtion
pmf	Probability Mass Function
w.r.t	with respect to

Chapter 1

Introduction

1.1 Overview

Until three decades ago, most empirical studies in finance were based on low frequency (daily at finest frequency) data. For example, only the first or the last observation of the day for the variable of interest (e.g. price) was recorded, thus all intra-day events were neglected. However, due to advances in computer power and the increased automatisation of financial markets, high-frequency (intra-day) databases have been formed. Transactions, therefore, are recorded at finer time intervals (e.g. minute-by-minute or second-by-second) with their associated measurements such as price, volume, bid[1] and ask[2] quotes (Pacurar, 2008). The ultimate form of high-frequency (HF) data is represented by transaction-by-transaction or trade-by-trade data in which events are recorded one by one as they occur (McCulloch and Tsay, 2001; Tsay, 2005; Liesenfeld et al., 2006; Pacurar, 2008). These databases are now readily available and commonly known as ultra high-frequency (UHF) data (Engle, 2000). For example, all equity transactions of the New York Stock Exchange (NYSE) are recorded in the trades and quotes (TAQ) database from 1992 to the present which includes transactions on the NYSE, AMEX, NASDAQ and the regional exchanges. A similar database for options transactions from August 1976 to December 1996 is provided by the Berkley Options Data Base. Olsen Associates in Switzerland has maintained a database of foreign exchange spot quotes for many major

[1]The bid price is defined as the price that a buyer is willing to pay for a security.

[2]The ask price is defined as the price that an investor accepts to sell a security.

currency pairs since the mid 1980's, published over the Reuters' network (Goodhart and O'Hara, 1997; Wood, 2000; Yan and Zivot, 2005; Tsay, 2005). Data from many other securities and markets are continuously collected and processed.

Embracing finance, econometrics and statistics, the analysis of HF/UHF data has attracted much interest by providing a deeper understanding of market activity. Research in a variety of issues related to trading process and market microstructure studies[3] has become possible by the advent of HF/UHF data sets (Campbell et al., 1996; Wood, 2000; Tsay, 2005; Cont, 2011). These developments are not limited to academia, they have also affected the current trading environment. By increasing the power of computers and automating transactions, the speed of trading has increased enormously and day-trading is now available to all investors. This has formed a new and successful category of hedge funds known as *high-frequency finance hedge funds* (Goodhart and O'Hara, 1997; Calamia, 1999; McCulloch and Tsay, 2001; Tsay, 2005; Luca and Gallo, 2009; Pacurar, 2008).

However, analysing UHF data introduces a new challenge in the area of statistics and finance, because such data have some unique characteristics that are not found in low frequency data (Engle and Russell, 2005; Tsay, 2005; Liesenfeld et al., 2006). The purpose of this study is to learn about these special characteristics, investigate possible methods for analysing UHF data and discuss the application of the obtained results. We are mainly interested in price movement and its characteristics.

Our analysis in this thesis will be carried out within a Bayesian framework. Almost until two decades ago, Bayesian statistics was only an interesting alternative to the *classical* theory. The intractabilities involved in the calculation of the posterior distribution prevented researchers from building and developing models using such methods. In recent years, advances in computer techniques and computational methods have dramatically increased our ability to use Bayesian methods (Ntzoufras, 2009; Gamerman and Lopes, 2006).

[3] O'Hara (2006) defines: "market microstructure is the study of the process and outcomes of exchanging assets under explicit trading rules. While much of economics abstracts from the mechanics of trading, microstructure literature analyses on how specific trading mechanisms affect the price formation process."

The remainder of this chapter is organised as follows. Section 1.2 reviews some special characteristics of UHF data. A description of the available data set for this study is presented in Section 1.3. Finally, a literature review of price change modelling and a brief description of the rest of the chapters are given in Section 1.4.

1.2 Empirical characteristics of UHF data

UHF data are characterised by unique properties that make them different from low frequency data. We highlight below some of their important features.

(i) *Unequally spaced time intervals.* The time duration between transactions in UHF data is a random variable because transactions do not occur at equally spaced time intervals. Therefore, consecutive transaction prices do not form an equally spaced time series. This means that the time duration may carry important information, e.g. how fast the transaction price may move in a given market, or it may reflect the market intensity.

(ii) *Discrete valued prices.* The price change of an asset in consecutive transactions occurs in a multiple of a tick[4] size, which leads to a discrete structure of price process.

(iii) *Multiple transactions within a single second.* Considering the fact that the time is measured in seconds, in a period of heavy trading of the market, one second may be too long as a time scale. As a result, two or more transactions with equal or unequal prices that occurred within a period of one second are recorded at the same time as multiple transactions.

See Engle and Russell (2005) and Tsay (2005, Chapter 5) for a further description of UHF data.

[4] A tick is defined as the minimum amount by which the price of the market can change.

1.3 FTSE100 index futures

The data comprise the *Financial Times Stock Exchange* (FTSE)100 index futures[5] for March 25 and 26, 2008, with the maturity date[6] June 2008, traded on the *London International Financial Futures and Option Exchange* (LIFFE) which was provided by *Cantab Capital Partners* through Dr Ewan Kirk. The FTSE100 index is constructed from 100 of the largest UK companies listed on the London Stock Exchange. There are $81,706$ transactions over 2 working days. In the first day (March 25) 46,180 transactions and in the second day (March 26) 35,526 transactions have been recorded. Variables associated with each transaction are the FTSE100 index futures, the transaction volume and the time of the transaction in unix time stamp[7].

The normal trading hours of the FTSE100 are from 08:00 until 16:30. Therefore, for simplicity, any transaction beyond these hours are discarded from the analysis. Thus, the number of transactions are reduced to 45,267 and 34,495 transactions, respectively, for each of the trading days. Furthermore, one tick size is considered to be 0.5 FTSE100 index. Each contract is valued as £10 per index point, therefore the value of one tick is £5 [8].

The FTSE100 index futures is interesting to study because the market microstructure of stock index futures markets, in general, has not been explored so far to the same extent as the other markets, e.g. option and stock markets. The reason might be that their transaction data have not been accessible as frequently as the other markets (Tse, 1999). In addition, the structure and trading system of futures markets are different from other securities markets. For example:

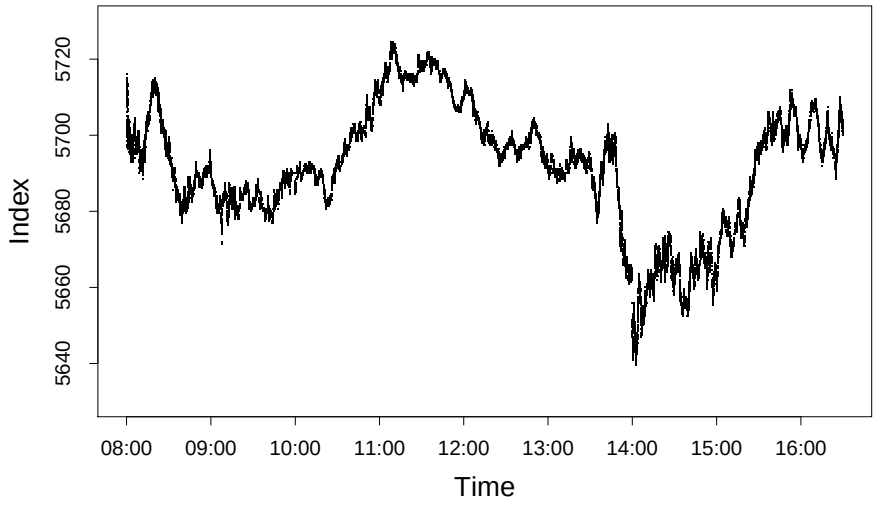
(i) There is no market maker on the futures market. All bid and ask offers are revealed to other traders in the trading pit by the process of open outcry, or through

[5] A contract that obliges the holder to buy or sell an asset at a predetermined delivery price during a specified future time period. The contract is settled daily.

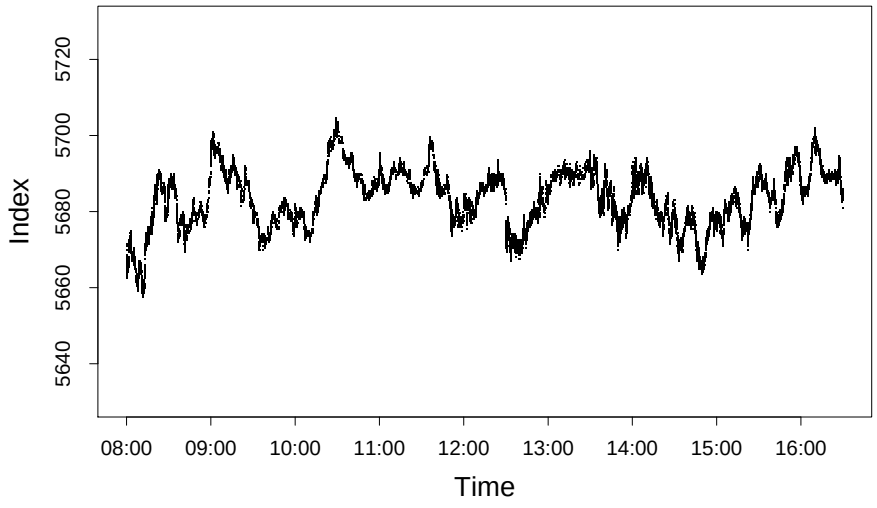
[6] The end of the life of a contract.

[7] "Unix time, or POSIX time, is a system for describing points in time, defined as the number of seconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970, not counting leap seconds"; For further information see: http://en.wikipedia.org/wiki/Unix_time.

[8] For further information see:
http://www.euronext.com/trader/contractspecifications/derivative/wide/contractspecifications-2830-EN.html?euronextCode=Z-LON-FUT.



(a)



(b)

Figure 1.1: FTSE100 index futures, the discrete structure of price process can be observed for (a) March 25, and (b) March 26, 2008.

continuous auction system of trading;

(ii) Index futures are not likely to be affected by private information;

(iii) Futures indices are influenced by market-wide information such as macroeconomic news.

As a result of these differences, the intra-day patterns of trading activity and transaction data of futures markets are different from those of stock and option markets (Tse, 1999).

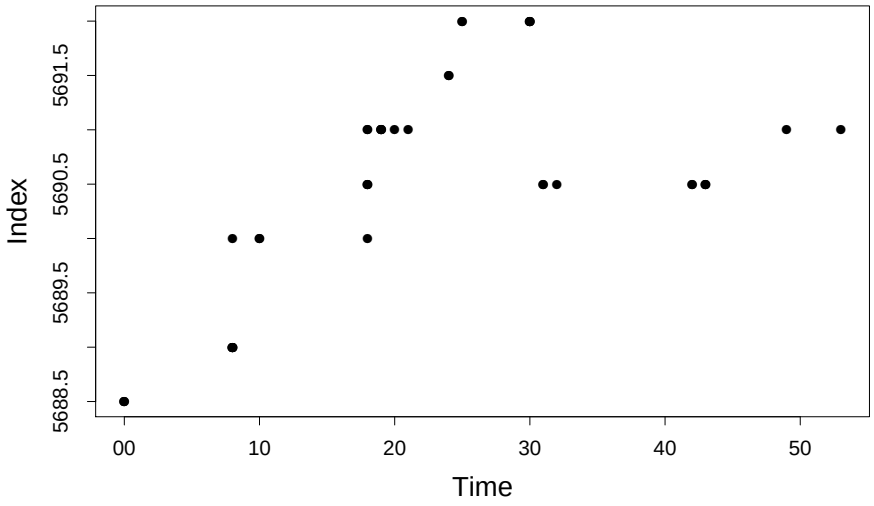


Figure 1.2: One minute of FTSE100 indices presents the main features of UHF data, e.g. price discreteness, unequally spaced time intervals and multiple transactions.

It may be noteworthy that March 25, 2008, is the Tuesday after Easter bank holiday which may explain the high intensity of the market on this day; about 10,000 more transactions than the next trading day. Figure 1.1 illustrates FTSE100 indices for the each of the trading days in the data set. The discrete structure of the FTSE100 indices is observable from the graphs in Figure 1.1. A one minute transaction period (10:00:00-10:01:00) from March 25 is illustrated in Figure 1.2 which shows unequally spaced time intervals, price discreteness and multiple transactions.

Now, let us assume that the FTSE100 index change from one transaction at time t_{i-1} to the next transaction at time t_i is given by

$$Z_i = \frac{P'_{t_i}}{ST} - \frac{P'_{t_{i-1}}}{ST} = P_{t_i} - P_{t_{i-1}}, \tag{1.1}$$

where P'_{t_i} is the FTSE100 index at time t_i, P_{t_i} is the number of ticks for the FTSE100 index at time t_i and $ST = 0.5$ is the tick size. Here t_i is a continuous time, but prices are only updated when a transaction actually occurs. Table 1.1 illustrates the relative frequencies of index change for March 25 and 26, 2008. It shows that the positive and the negative index changes are almost symmetrically distributed around zero. It can be seen that about 69% of the indices have not changed at all and 21% had a change of 1 tick size. In

Number(tick)	$\leq$ -4	-3	-2	-1	0	1	2	3	$\geq$ 4
Percentage	0.6	1.0	3.6	10.4	68.8	10.2	3.7	1.1	0.6

Table 1.1: Relative frequencies of index change in multiple of tick size.

addition, changes of 4 or more ticks are rare (about only 1%).

1.4 The state of the problem

1.4.1 Price change

UHF data are usually identified by their two main features: the irregularity of time intervals and the discreteness of price changes as discussed in Section 1.2, from the low frequency (daily or monthly) data sets. Conventional methods are mainly suitable for the purpose of analysing low frequency data that are defined over equal time intervals with continuous variables. Therefore, such methods may not be able to handle UHF data characteristics. Developing methods that consider the features of UHF data may improve our inference greatly.

A suitable technique needs to address the two characteristics of UHF data and therefore should consist of two parts. The first part is modelling time duration between two consecutive transactions (time duration). Based on the seminal work by Engle and Russell (1998), a large body of research (among the others, (Bauwens and Giot, 2000; Zhang et al., 2001; Tsay, 2005; Pacurar, 2008)) has been concerned around the further development of autoregressive conditional duration (ACD) models in order to characterise the time duration (Liesenfeld et al., 2006). The second part is the modelling of price process. Instead of modelling price process itself, Rydberg and Shephard (1999) suggested modelling price change as a non-stationary and a nonlinear process. More specifically, P'_{t_i}, the price of the most recent transaction at time t_i can be considered as follows

$$P'_{t_i} = P'_{t_0} + \sum_{i=1}^{\mathrm{N}_i} (ST \cdot Z_i), \tag{1.2}$$

where N_i is the number of transactions recorded in the interval from time 0 up to time t_i and Z_i is the index change associated with the ith transaction, given by (1.1). In practice

many of the Z_i's are zero. In contrast with the time process, modelling price change is not easy. The price change is an integer-valued variable whereas available methods are dealing with continuous variables or non-negative integer-valued variables.

Ordered probit models were commonly used for modelling price change in early studies of UHF data sets (Hausman et al., 1992; Bollerslev and Melvin, 1994). In later attempts, Russell and Engle (1998) proposed an autoregressive conditional multivariate (ACM) model and Darolles et al. (2000) considered price change process as a Markov process having the values -1, 0 and 1. However, the main drawback of these models is that they consider a finite range for the discrete distribution of the observed price change. In addition, for both ACM and the ordered probit models, it is necessary that all potential outcomes have to occur in the sample period to guarantee the identification and estimation of the true dimension of the multinomial process. They also assume that the number of parameters increases with the outcome space. Therefore, these techniques are only suitable for the empirical analysis of financial markets which are characterised by a limited number of discrete price changes and when we are not interested in predicting the market behaviour (Rydberg and Shephard, 2003; Liesenfeld et al., 2006).

Later on, Rydberg and Shephard (2000, 2003) proposed a model, commonly known as the ADS model, which considers the price change as a combination of three components: an indicator for price change (A), the direction (D) of price change if there is a change and the size (S) of price change if a change occurs. The advantage of such a method is that the final output is an integer, unlike with the previous models which were defined over a finite range. Using the idea of decomposition, Liesenfeld et al. (2006) suggested an integer count Hurdle (ICH) model in which price is decomposed into two components: a trinomial process which indicates no price change, price increases or decreases, and a truncated-at-zero count process for the size of price change.

Since parameter estimation in the more complex models is not feasible with standard techniques such as maximum-likelihood estimation, Bayesian methods may be an alternative for analysing UHF data. These methods can be implemented via a wide range of Monte Carlo techniques such as Markov chain Monte Carlo (MCMC) methods. Us-

ing Bayesian methods, McCulloch and Tsay (2001) proposed a bivariate model for price change and time duration (a PCD model). The PCD model, similar to the ICH model, decomposes the price change into the direction and the size of price change in combination with a model for time duration. However, such a model only rests over non-zero price changes and time durations, which may lead to information loss. Another study which considered the Bayesian ADS model, is Gendron and McCausland (2005). The original ADS model was fitted to a set of HF data by Gendron and McCausland (2005), in which the size of price change was also modelled separately for negative and positive changes, similar to the ICH and PCD models. Modelling the size of price change as two separate processes may ignore the effect of the components of the size on one another. Czado and Kolbe (2004) also used the idea of decomposition in a Bayesian framework, but only modelled the size of price change which left the interpretation of the results incomplete.

However, decomposing the price change may be at the cost of introducing additional complication to the analysis of UHF data. Also our interpretation from the model is only valid when results from all the components are available. Our attempt in this study is to answer the following question. Can we model price change as a single process without breaking it into two or three components or limiting it over a finite range? Modelling price change as a single process gives us better a understanding of the behaviour of price change. Also, it may be computationally more feasible than the previous methods because we only deal with a single variable.

1.4.2 Modelling price change

We initiate the discussion about modelling price change for UHF data in Chapter 2. In this chapter we describe the Bayesian ADS model and use MCMC methods to fit the model to the FTSE100 index change. This gives us the opportunity to understand better the ADS model and the behaviour of the data set using a well-known model. We can also use the results from the Bayesian ADS model to compare with the results obtained from other models that we propose later in this study.

We introduce the Bayesian application of the zero inflated Poisson difference (ZPD) distribution for modelling price change in Chapter 3. The Poisson difference distribution

provides an integer range for index change and, by considering its zero inflated version, we are able to take into account the excess of zeros in the model. In this chapter, like in Chapter 2, we assume that the data from one trading day is available, therefore in practice the analysis can be performed after the closing time of the market. In general, we refer to this type of analysis as the *off-line* modelling of data.

However, as transactions occur sequentially, we may be interested in performing inference and prediction *online*. That is, we want to update our analysis as a new transaction takes place. This leads us to using *sequential Monte Carlo* methods (also known as particle filters) that enable us to perform an online analysis of sequential data sets. For this purpose, we propose a dynamic zero inflated Poisson difference (DZPD) model in Chapter 4 which is fitted to our data set via SMC methods. Moreover, in order to assess the model sequentially, in this chapter, we introduce the sequential deviance information criterion.

The ZPD distribution may lead to an underestimation of the tails in certain situations. Therefore, in order to add more flexibility to the tails of the ZPD distribution, we introduce the generalised Poisson difference (GPD) distribution with four parameters in Chapter 5. We investigate some important characteristics of the GPD distribution and show that the GPD distribution suggested by Consul and Famoye (2006, Chapter 9) is a special case of the GPD distribution introduced in this study. In addition, we outline a possible application of the zero inflated version of the GPD (ZGPD) distribution for modelling UHF data.

We would like to point out that in Chapters 2 and 3, we fitted models to the data for March 25 and used the next day's data, March 26, for the purpose of validating our models. In Chapter 4 we only used the data relating to March 25 for the purpose of dynamic modelling of one day's transactions. We carried out all the analysis in R 2.11.1, a software environment for statistical computing and graphics.

Finally, Chapter 6 provides a summary of our results, the strengths and the weaknesses of our proposed models and general review of possible extensions of our methods.

Chapter 2

A Bayesian perspective of the decomposition model

2.1 Introduction

This chapter introduces the ADS model in a Bayesian framework. We model the FTSE100 index change from the trading day March 25 using MCMC techniques. We suggest an algorithm for predicting index change and obtain forecasts for the next trading day (March 26). We assess the performance of the MCMC chains by Gelman and Rubin's convergence diagnostics (Gelman and Rubin, 1992), and the posterior predictive distribution with a probability integral transform (PIT) modified for the case of the discrete data (Liesenfeld et al., 2006).

This chapter is organised as follows. Section 2.2 provides a more detailed introduction to the ADS model. Section 2.3 describes the Bayesian analysis of the ADS model, including: the posterior distribution, the Metropolis-Hastings algorithm, prediction and model checking. Results are presented in Section 2.4, followed by a discussion of the strengths and weaknesses of the ADS model in Section 2.5.

2.2 Decomposition of index change

Consider an index change[1], Z_i given by (1.1), an integer-valued variable observed at time t_i, for $i = 1,\ldots,n$, where n is the number of observed index changes in a given trading day. According to the ADS model, Z_i can be decomposed into three components. These components are: the indicator for index change (A_i), the direction (D_i), and the size (S_i) of index change if a change occurs. More specifically, Z_i can be expressed as follows

$$Z_i = A_i D_i S_i, \tag{2.1}$$

where A_i is a binary variable with the following values:

- $A_i = 0$ if the index remains the same, i.e. the index is *in-active*,
- $A_i = 1$ if a change occurs, i.e. the index is *active*.

In the rest of the chapter we may refer to the indicator of index change, A_i, as index activity or simply activity, interchangeably. The direction of index change, D_i, is also a binary variable given by

- $D_i = -1$ if index change is negative,
- $D_i = \ 1$ if index change is positive.

Finally, S_i is an integer-valued variable such that $S_i > 0$. Consequently, the probability of an index change can also be decomposed as follows

$$f(Z_i|\mathbf{x}) = f_{\mathrm{A}}(A_i|\mathbf{x}_{i,\mathrm{A}}) f_{\mathrm{D}}(D_i|A_i = 1, \mathbf{x}_{i,\mathrm{D}}) f_{\mathrm{S}}(S_i|D_i, A_i = 1, \mathbf{x}_{i,\mathrm{S}}), \tag{2.2}$$

where $\mathbf{x}$ is a set of covariates which affect index change through its individual components, and $\mathbf{x}_{i,\mathrm{A}}$, $\mathbf{x}_{i,\mathrm{D}}$ and $\mathbf{x}_{i,\mathrm{S}}$ are the ith row of the matrices of covariates corresponding to each of the components of index change. In the following sections we detail the modelling of the components of index change.

[1]The ADS model was originally suggested for the *price change*, whereas here we fit the ADS model to FTSE100 index changes. Thus, we may use both *price change* and *index change* interchangeably.

2.2.1 Modelling components of index change

We may assign the following model for the probability of index activity

$$M_{\mathrm{A}}: \quad p(A_i = 1 |\, \mathbf{x}_{i,\mathrm{A}}) = p_{\mathrm{A}_i}, \quad \log\left(\frac{p_{\mathrm{A}_i}}{1-p_{\mathrm{A}_i}}\right) = \mathbf{x}_{i,\mathrm{A}}\boldsymbol{\beta}, \tag{2.3}$$

where $\boldsymbol{\beta}$ is a vector of parameters of an appropriate size. Similarly, for the direction of index change given that $A_i = 1$, we can set the following model for the probability of the direction of index change

$$M_{\mathrm{D}}: \quad p(D_i = 1 |\, A_i = 1, \mathbf{x}_{i,\mathrm{D}}) = p_{\mathrm{D}_i}, \quad \log\left(\frac{p_{\mathrm{D}i}}{1-p_{\mathrm{A}_i}}\right) = \mathbf{x}_{i,\mathrm{D}}\boldsymbol{\gamma}, \tag{2.4}$$

where $\boldsymbol{\gamma}$ is a vector of parameters of an appropriate size. In order to model the size of index change, S_i, possible choice of distributions are geometric distribution (Tsay, 2005, Chapter 5), Poisson distribution (McCulloch and Tsay, 2001) and Negative Binomial distribution (Liesenfeld et al., 2006; Rydberg and Shephard, 2003) distributions. Here, we use a geometric distribution. In order to consider zero values in the variable size we model $S_i' = S_i - 1$. It is noteworthy that Rydberg and Shephard (2003) referred to a value greater than 0 as a *large move*. We fit the following model to the size of index change

$$M_{\mathrm{S}}: \quad S_i' |\, \left(D_i,\, A_i = 1,\, \mathbf{x}_{i,\mathrm{S}'}\right) \sim g(p_{\mathrm{S}_i'}), \quad \log\left(\frac{p_{\mathrm{S}_i'}}{1-p_{\mathrm{S}_i'}}\right) = \mathbf{x}_{i,\mathrm{S}'}\,\boldsymbol{\theta}, \tag{2.5}$$

where $\boldsymbol{\theta}$ is a vector of parameters of an appropriate size and $g(p_{\mathrm{S}_i'})$ denotes a geometric distribution with parameter $p_{\mathrm{S}_i'}$ such that

$$p_{\mathrm{S}_i'} = P(S_i' = s_i' | D_i,\, A_i = 1, \mathbf{x}_{i,\mathrm{S}'}).$$

Therefore, the next transaction may fall into one of the following three categories:

(i) No index change, i.e. $A_i = 0$ with probability $1 - p_{\mathrm{A}_i}$;

(ii) The index increases, i.e. the next transaction increases with associated probability $p_{\mathrm{A}_i} p_{\mathrm{D}_i}$ and a size of $S_i = S_i' + 1$ such that $S_i' \sim g(p_{\mathrm{S}_i'})$;

(iii) The index drops, i.e. the next transaction decreases with associated probability $p_{\mathrm{A}_i}(1-p_{\mathrm{D}_i})$ and a size of $S_i = S_i' + 1$ such that $S_i' \sim g(p_{\mathrm{S}_i'})$.

2.3 Bayesian inference

In the Bayesian framework of the ADS model our inference of Z is based on the posterior distributions of the parameters of the model, β, γ and θ. Let us assume β, γ and θ are independently distributed. Therefore, the posterior distribution of the model parameters using (2.2), can be defined as follows

$$\begin{aligned} f^{\text{post}}(\beta,\, \gamma,\, \theta|z) &\propto f^{\text{like}}(z|\beta,\, \gamma,\, \theta)\, f_{\beta}^{\text{prior}}(\beta)\, f_{\gamma}^{\text{prior}}(\gamma)\, f_{\gamma}^{\text{prior}}(\theta) \\ &= f_{\mathrm{A}}^{\text{like}}(a|\, \beta)\, f_{\beta}^{\text{prior}}(\beta)\, f_{\mathrm{D}}^{\text{like}}(d|\, \gamma,\, a=1)\, f_{\gamma}^{\text{prior}}(\gamma)\, f_{\mathrm{S}}^{\text{like}}(s|\theta,\, d,\, a=1)\, f_{\theta}^{\text{prior}}(\theta) \\ &\propto f_{\beta}^{\text{post}}(\beta|\, a=1)\, f_{\gamma}^{\text{post}}(\gamma|\, d,\, a=1)\, f_{\theta}^{\text{post}}(\theta|\, s,\, d,\, a=1), \end{aligned} \tag{2.6}$$

where $f_{\mathrm{A}}^{\text{like}}(a|\beta)$, $f_{\mathrm{D}}^{\text{like}}(d|\gamma,\, a=1)$ and $f_{\mathrm{S}}^{\text{like}}(s|\theta,\, d,\, a=1)$ are the likelihood functions and $f_{\beta}^{\text{prior}}(\beta), f_{\gamma}^{\text{prior}}(\gamma)$ and $f_{\theta}^{\text{prior}}(\theta)$ are the prior distributions. The decomposition in (2.6) enables us to obtain the posterior distribution for each component separately and combine the posterior distributions in order to make inference about index change.

We start by defining the likelihood functions. Let us denote n as the number of index changes in a given day and n' as the number of index changes with active indices. Thus, there are $n-n'$ transactions with in-active indices. We assign a Bernoulli distribution to A_i, with parameter p_{A_i} given by (2.3). Thus, the likelihood is

$$\begin{aligned} f_{\mathrm{A}}^{\text{like}}(a|\, \beta) &= p(\mathrm{A}_1 = a_1) \ldots\, p(\mathrm{A}_n = a_n) \\ &= \prod_{i=1}^{n} p_{A_i}^{a_i} (1-p_{A_i})^{1-a_i} \\ &= \prod_{i=1}^{n} \left(\frac{p_{A_i}}{1-p_{A_i}} \right)^{a_i} (1-p_{A_i}). \end{aligned} \tag{2.7}$$

Similarly, if we re-parameterise D_i as follows

$$D'_i = D_i + 1 \quad \text{if} \quad D_i = -1,$$
$$D'_i = D_i + 0 \quad \text{if} \quad D_i = 1,$$

and assign a Bernoulli distribution to D_i with parameter $p_{\mathrm{D}'_i} = p_{\mathrm{D}_i}$ in (2.4), the likelihood function is given by

$$\begin{aligned} f_{\mathrm{D}}^{\text{like}}(d|\,\gamma,\, a=1) &= Pr(D'_1 = d'_1)\ldots\, Pr(D'_{n'} = d'_{n'}) \\ &= \prod_{i=1}^{n'} p_{\mathrm{D}'_i}^{d'_i}(1-p_{\mathrm{D}'_i})^{1-d'_i} \\ &= \prod_{i=1}^{n'} \left(\frac{p_{\mathrm{D}'_i}}{1-p_{\mathrm{D}'_i}}\right)^{d'_i} (1-p_{\mathrm{D}'_i}). \end{aligned} \tag{2.8}$$

The probability mass function of the size of index change, $S'_i = S_i - 1$, is a geometric distribution with parameter $p_{\mathrm{S}'_i}$ defined in (2.5)

$$g(S'_i = s'_i|p_{\mathrm{S}'_i}) = p_{\mathrm{S}'_i}(1-p_{\mathrm{S}'_i})^{s'_i}, \quad S'_i = 0,1,2,\ldots,$$

so the likelihood function is

$$\begin{aligned} f_{\mathrm{A}}^{\text{like}}(s'|\theta,\, d,\, a=1) &= Pr(S'_1 = s'_1)\ldots\, Pr(S'_1 = s'_1) \\ &= \prod_{i=1}^{n'} p_{\mathrm{S}'_i}(1-p_{\mathrm{S}'_i})^{s'_i} \\ &= \prod_{i=1}^{n'} \left(\frac{p_{\mathrm{S}'_i}}{1-p_{\mathrm{S}'_i}}\right)(1-p_{\mathrm{S}'_i})^{s'_i+1}. \end{aligned} \tag{2.9}$$

Next, we choose the prior distribution. In the absence of any specific information we set a normal prior distribution with mean zero and large variance (e.g. 10^4) for each of the parameters of the model. Therefore, for each of the variables A, D and S, the prior

distributions are

$$\beta \sim \mathrm{N}(0, \Sigma_\beta), \tag{2.10}$$

$$\gamma \sim \mathrm{N}(0, \Sigma_\gamma), \tag{2.11}$$

$$\theta \sim \mathrm{N}(0, \Sigma_\theta), \tag{2.12}$$

where $\Sigma_\beta = \sigma_1 I$, $\Sigma_\gamma = \sigma_2 I$ and $\Sigma_\theta = \sigma_3 I$, where I is an identity matrix of an appropriate size and $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 10^4$ as there is no prior information available. Thus, the posterior distribution of the parameters of the index activity using (2.3), (2.7) and (2.10) is proportional to

$$\log(f_\beta^{\text{post}}(\beta |\, a = 1)) \propto \sum_{i=1}^{n} [a_i \mathbf{x}_{i,\mathrm{A}} \beta - \log(1 + \exp(\mathbf{x}_{i,\mathrm{A}} \beta))] - \frac{1}{2} \left(\beta^{\mathrm{T}} \Sigma_\beta^{-1} \beta \right). \tag{2.13}$$

Similarly, using (2.4), (2.8) and (2.11), the posterior distribution of the parameters of the direction of index change can be obtained as follows

$$\log(f_\gamma^{\text{post}}(\gamma |\, d',\, a = 1)) \propto \sum_{i=1}^{n'} \left[d_i' \mathbf{x}_{i,\mathrm{D}'} \gamma - \log(1 + \exp(\mathbf{x}_{i,\mathrm{D}'} \gamma)) \right] - \frac{1}{2} \left(\gamma^{\mathrm{T}} \Sigma_\gamma^{-1} \gamma \right). \tag{2.14}$$

For the size of index change, considering equations in (2.5), (2.9) and (2.12) the posterior distribution is

$$\log(f_\theta^{\text{post}}(\theta |\, s',\, d',\, a = 1)) \propto \sum_{i=1}^{n'} \left[\mathbf{x}_{i,\mathrm{S}'} \theta - (s_i' + 1) \log(1 + \exp(\mathbf{x}_{i,\mathrm{S}'} \theta)) \right] - \frac{1}{2} \left(\theta^{\mathrm{T}} \Sigma_\theta^{-1} \theta \right). \tag{2.15}$$

It can be seen that the posterior distributions are not known explicitly. Therefore, in order to generate samples from the posterior distribution we use MCMC methods, more specifically, the random walk Metropolis-Hastings (M-H) algorithm. We briefly describe the M-H algorithm in the following section.

2.3.1 Metropolis Hastings algorithm

A popular way of simulating from a general posterior distribution is by using MCMC techniques. The MCMC sampling strategy sets up an irreducible, aperiodic Markov chain for which the stationary distribution equals the posterior distribution of interest (Albert, 2007; Ntzoufras, 2009; Gelman et al., 2003). The M-H algorithm is a class of MCMC methods which is applicable when the posterior distribution is known up to a normalising constant (Tsay, 2005).

Suppose that we want to draw samples from the posterior distribution, $f_{\text{post}}(\psi|z)$, where ψ is a vector of parameters of a given model and z denotes the vector of observations. The posterior distribution, $f_{\text{post}}(\psi|z)$, contains a complicated normalising constant so that a direct draw is either too time-consuming or infeasible. However, there exists a distribution, $q(.|.)$, such that the

(i) State space of ψ is the same as in the target distribution;

(ii) Samples from $q(.|.)$ converge to the stationary distribution (target distribution);

(iii) Random draws from such distribution are easily available.

This distribution is referred to as *proposal distribution* (Gelman et al., 2003; Tsay, 2005). In the kth iteration, for $k = 1, \ldots, m$, the random-walk M-H algorithm proceeds as follows

(i) Draw a candidate sample ψ^{cand} from the proposal distribution, $q(.|\psi^{(k-1)})$;

(ii) Obtain the acceptance ratio, r, which is given by

$$r = \frac{f_{\text{post}}(\psi^{\text{cand}}|z)\; q(\psi^{(k-1)}|\psi^{\text{cand}})}{f_{\text{post}}(\psi^{(k-1)}|z)\; q(\psi^{\text{cand}}|\psi^{(k-1)})}; \tag{2.16}$$

(iii) Set

$$\psi^{(k)} = \begin{cases} \psi^{\text{cand}} & \text{with probability } \min(r,1), \\ \psi^{(k-1)} & \text{otherwise.} \end{cases} \tag{2.17}$$

We choose the normal distribution to be our proposal distribution, $q(.|\psi^{(k-1)})$, such that

(i) The normal distribution is centred at the values from the previous iteration, i.e. $\psi^{\text{cand}} \sim \text{N}(\psi^{(k-1)}, \sigma^2)$, for some value of σ^2;

(ii) An important point to consider in the case of random walk chains is the choice of the value of the dispersion parameter of the proposal distribution. A large value for the variance allow a greater variation from the previous value, but will lead to a very small acceptance rate. On the other hand, a small value of the variance results in draws which are close to the previous value with a high acceptance rate (Gamerman and Lopes, 2006). The optimal choice for the variance of the normal proposal is $\sigma^2 = c^2\Sigma$, where $c \approx 2.4/\sqrt{d}$ (d is the dimension of the parameter vector) and Σ is the variance-covariance matrix based on the curvature of the posterior at the mode (Tanner, 1998).

2.3.1.1 Gelman convergence diagnostics

In order to monitor the convergence of the MCMC chains, we undertake Gelman and Rubin's convergence diagnostics (Gelman and Rubin, 1992; Brooks and Gelman, 1998). The statistic obtained from this method is known as the R statistic which for values greater than 1.1 may indicate a non-stationary chain. We may need to run the MCMC chains out till we obtain an R statistic value of less than 1.1.

The M-H algorithm described above is summarised in Algorithm 2.1

2.3.2 Predicting index movements

Let us assume we are interested in estimating the next day's index change, $Z^* = (Z_1^*, \ldots, Z_{\text{L}}^*)$, where L is the number of index changes in the next trading day. The posterior predictive distribution, using the decomposition of index change in (2.2) and (2.6), is defined as

$$\begin{aligned} f^{\text{pred}}(Z^*|z) &= \int\int\int f^{\text{like}}(Z^*|\beta,\gamma,\theta)\, f^{\text{post}}(\beta,\gamma,\theta|z) d\beta\, d\gamma\, d\theta \\ &= \int\int\int f_{\text{A}}^{\text{like}}(A^*|\beta)\, f_{\text{D}}^{\text{like}}(D^*|\gamma)\, f_{\text{S}}^{\text{like}}(S^*|\theta) f_{\beta}^{\text{post}}(\beta|a)\, f_{\gamma}^{\text{post}}(\gamma|d)\, f_{\theta}^{\text{post}}(\theta|s) d\beta\, d\gamma\, d\theta \\ &= f_{\text{A}}^{\text{pred}}(A^*|a) f_{\text{D}}^{\text{pred}}(D^*|d) f_{\text{S}}^{\text{pred}}(S^*|s). \end{aligned} \tag{2.20}$$

Algorithm 2.1 Random walk Metropolis-Hastings algorithm.

1. Set an arbitrary initial value $\psi^{(0)}$ at $k = 0$;

2. For $k = 1, \ldots, m$ repeat the following steps:

   2.1 Draw ψ^{cand} from the proposal distribution, $q(.|\psi^{(k-1)})$;

   2.2 Obtain the acceptance ratio, r, which is given by

$$r = \frac{f_{\text{post}}(\psi^{\text{cand}}|z)\; q(\psi^{(k-1)}|\psi^{\text{cand}})}{f_{\text{post}}(\psi^{(k-1)}|z)\; q(\psi^{\text{cand}}|\psi^{(k-1)})}; \tag{2.18}$$

$$\tag{2.19}$$

   2.3 Sample $u \sim \text{Uniform}(0,1)$, if $u < \min(1, r)$, set $\psi^{(i)} = \psi^{\text{cand}}$, otherwise let $\psi^{(k)} = \psi^{(k-1)}$.

Therefore, by obtaining the predictive distribution for each component of index change we may be able to evaluate the predictive distribution of index change. We generate samples from the predictive distribution for A, D and S (and therefore Z) as follows. In the kth iteration, for $k = 1, \ldots, m$, repeat the following steps for $l = 1, \ldots, L$:

(1) Obtain $p^{(k)}_{\text{A}_{n+l}}$ using (2.3) as follows

$$p^{(k)}_{\text{A}_{n+l}} = \frac{\exp(\mathbf{x}_{n+l,\text{A}}\boldsymbol{\beta}^{(k)})}{1 + \exp(\mathbf{x}_{n+l,\text{A}}\boldsymbol{\beta}^{(k)})},$$

and draw ${A^*_{n+l}}^{(k)} \sim \text{Bernoulli}(p^{(k)}_{\text{A}_{n+l}})$;

(2) If $A_{n+l} = 0$, repeat (1), otherwise

(2.1) Obtain $p^{(k)}_{\text{D}'_{n+l}}$ using (2.4) as follows

$$p^{(k)}_{\text{D}'_{n+l}} = \frac{\exp(\mathbf{x}_{n+l,\text{D}'}\boldsymbol{\gamma}^{(k)})}{1 + \exp(\mathbf{x}_{n+l,\text{D}'}\boldsymbol{\gamma}^{(k)})},$$

and sample ${D'^*_{n+l}}^{(k)} \sim \text{Bernoulli}(p^{(k)}_{\text{D}'_{n+l}})$;

- For $D'_{n+l} = 1$, set

$$D_{n+l} = D'_{n+l} - 0 = 1,$$

otherwise, let

$$D_{n+l} = D'_{n+l} - 1 = -1;$$

(2.2) Obtain $p^{(k)}_{S'_{n+l}}$ using (2.5) as follows

$$p^{(k)}_{S'_{n+l}} = \frac{\exp(\mathbf{x}_{n+l,S'}\boldsymbol{\theta}^{(k)})}{1+\exp(\mathbf{x}_{n+l,S'}\boldsymbol{\theta}^{(k)})},$$

and draw ${S'^{*}_{n+l}}^{(k)} \sim g(p^{(k)}_{S'_{n+l}})$. Let ${S^{*}_{n+l}}^{(k)} = {S'^{*}_{n+l}}^{(k)} + 1$.

(3) Obtain ${Z^{*}_{n+l}}^{(k)} = {A^{*}_{n+l}}^{(k)} {D^{*}_{n+l}}^{(k)} {S^{*}_{n+l}}^{(k)}$.

In practice we set $z^*_{n+l-1} = a^*_{n+l-1}d^*_{n+l-1}s^*_{n+l-1}$, for $l = 1$, as the last observed index change, $z_n = a_n d_n s_n$.

2.3.3 Model checking

After fitting the ADS model to a given data set, we need to assess how well the model is able to capture the behaviour of index change. In our case, we are dealing with an integer-valued random variable (price/index change). In addition, in our data set most of the values (68%) of index change were zero (no index change). This makes it very difficult to find suitable diagnostics measures. The only appropriate method that can be used here is the probability integral transform (PIT) modified for the case of discrete values which we describe below.

2.3.3.1 Probability integral transform

A randomised version of the PIT is implemented to measure how well the predictive distribution is able to explain the density of index change for the next trading day (Liesenfeld et al., 2006). In order to use the PIT, we have to construct intervals based on the cumulative predictive distribution of the *i*th observed index change and the *i*th observed index

change minus 1, as follows

$$u_i^u = \hat{p}(Z_i \leq z_i) = \sum_{j=-\infty}^{z_i} \hat{p}(Z = j), \tag{2.21}$$

$$u_i^l = \hat{p}(Z_i \leq z_i - 1) = \sum_{j=-\infty}^{z_i-1} \hat{p}(Z = j), \tag{2.22}$$

where z_i is the ith index change in the data set from the next trading day and $\hat{p}(.)$ represents the estimated counterpart of the conditional probability given in (2.2). It can be seen that (2.21) and (2.22) form intervals with upper and lower limits of u_i^u and u_i^l, respectively. If the model is correctly specified, random samples u_i's from such intervals will have a uniform distribution on $(0,1)$. To test the idea of the uniformity of the constructed values, one can use the Kolmogrov-Smirnov (K-S) test or plot a quantile-quantile (QQ)-plot of the standard uniform distribution against the u_i's. Here, we used both the K-S test and the QQ-plot to judge the efficacy of the predicted values.

2.4 Results

This section presents the results of the analysis of the FTSE100 index change using the ADS model. On March 25 (March 26), out of $n = 45,266$ ($n = 34,495$) index changes, there were $n' = 13,893$ ($n' = 11,031$) active and $n - n' = 31,373$ ($n - n' = 23,464$) inactive indices. In the M-H algorithm we set the number of iterations to $m = 10000$ and discard the first 5000 samples as burn-in.

Traces and the estimates of the posterior distributions of the models ($M_{\mathrm{A}}, M_{\mathrm{D}}$ and M_{S}) parameters were produced and are presented in Figures A.2, A.4 and A.6. Gelman and Rubins's diagnostics results are provided in Figures A.1, A.3 and A.5, and Tables A.1, A.2 and A.3. The R statistics are less than 1.1 for all of the models parameters which lets us assume the convergence of the MCMC chains.

2.4.1 The activity of indices

Consider model M_{A}, as given by (2.3). We may consider the effect of the previous index activity (a_{i-1}), the volume of the previous transaction (v_{i-1}) and the time duration

Coefficients	Mean	SD	95% CI	
β_0	-0.858	0.014	-0.883	-0.831
β_1	-0.309	0.022	-0.325	-0.265
β_2	0.381	0.019	0.342	0.417
Acceptance rate				0.412

Table 2.1: The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the index activity.

Coefficients	Mean	SD	95% CI	
γ_0	0.327	0.024	0.278	0.374
γ_1	-0.688	0.034	-0.756	-0.6211
Acceptance rate				0.513

Table 2.2: The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the direction of index change.

between two consecutive transactions ($\Delta t_i = t_i - t_{i-1}$) as possible covariates in M_{A}, for $i = 1, \ldots, n$. Our preliminary analysis showed that the volume of the previous transaction does not have a significant effect on the index activity A_i. Therefore, we fit a model without volume which is given by

$$\log\left(\frac{p_{\mathrm{A}_i}}{1-p_{\mathrm{A}_i}}\right) = \beta_0 + \beta_1 a_{i-1} + \beta_2 \log(\Delta t_i + 1). \tag{2.23}$$

Table 2.1 provides a summary of the posterior distribution of the parameters of the model M_{A}. It can be seen that the previous activity of index had a large effect on the activity of the current index. The negative sign of β_1 may imply that, the next index is less likely to be active if the previous index was in-active. However, the positive significant effect of Δt_i may indicate that, as the time duration between two consecutive transactions increases, the chance of having an index change at the next transaction increases.

2.4.2 The direction of index change

Consider the model M_{D} in (2.4) in terms of the previous direction of index change (d_{i-1}), the volume of the previous active transaction (v_{i-1}) and the time duration between two consecutive active transactions (Δt_{i-1}), for $i = 1, \ldots, n'$. The time duration between two consecutive active transactions and the volume of the previous active transaction did not

		D_{i-1}	
		-1	1
D_i	-1	0.419	0.589
	1	0.581	0.411

Table 2.3: The plug-in estimates for the probability of the direction of index change.

significantly affect the direction of index change. Hence, we set our model as

$$\log\left(\frac{p_{\mathrm{D}_i}}{1-p_{\mathrm{D}_i}}\right) = \gamma_0 + \gamma_1 d_{i-1}. \tag{2.24}$$

Results, summarised in Table 2.2, may suggest that the direction of index change at time t_i can be significantly affected by its previous direction. To have a better understanding of the behaviour of the direction of index change, we obtained the plug-in estimates of p_{D_i} by substituting the posterior mean of the vector of model parameters in (2.24). Table 2.3 suggests that at the next transaction both negative and positive changes are almost equally likely. Also, one can see that at the next transaction it is more likely to switch the direction of change.

2.4.3 The size of index change

This section provides results for modelling the size of index change for all active indices. To consider the dynamics of the negative and positive changes, in addition to the previous size of index change (s_{i-1}), volume of the previous active index (v_{i-1}) and time duration between two consecutive active indices (Δt_i), for $i = 1,\ldots,n'$, we consider the direction of the previous active index change (d_{i-1}) as a covariate in the model in (2.5). However, we noticed that the direction of index change was not significant, therefore our model for the size of index change can be set as follows

$$\log\left(\frac{p_{\mathrm{S}'_i}}{1-p_{\mathrm{S}'_i}}\right) = \theta_0 + \theta_1 s'_{i-1} + \theta_2 \log(\Delta t_i + 1) + \theta_3 \log(v_{i-1}). \tag{2.25}$$

From Table 2.4, one can see the significant effect of the size of the previous index change on the size of the current movement. Moreover, for active indices, a big positive index change is more likely to be followed by a negative index change with a similar magnitude.

			Quantiles	
Coefficients	Mean	SD	2.5%	97.5%
θ_0	-0.908	0.025	-0.963	-0.862
θ_1	0.318	0.015	0.291	0.348
θ_2	-0.073	0.020	-0.112	0.034
θ_3	0.090	0.019	0.055	0.131
Acceptance rate				0.328

Table 2.4: The posterior mean, the standard deviation (SD) and the 95% credible intervals for parameters of the size of index change.

We can also observe the significant effect of the volume. This suggests that a higher volume of the previous active transaction may increase the chance of a larger change at the next active index. On the other hand, one can see that as the time duration between two consecutive transactions increases, the chance of getting a large size of index change for the next transaction decreases.

Furthermore, using the algorithm in Section 2.3.2 we predicted index changes for the next trading day. In the algorithm we assumed the time duration between two consecutive transactions, the volume of the previous transaction and the number of transactions within the next trading day are known. We assessed the predicted values using the PIT. The Q-Q plot in Figure 2.1 and K-S test with the p-value of 0.21 suggest that the ADS model is able to characterise well the distribution of index change for the next trading day. That is, the predictive distribution of index change over the whole day of March 26 was a similar distribution to the observed index changes. We observed that about 99.24% of the predicted changes are within ± 4 which is very close to the value observed in the real data set (99.68%). We also saw that negative and positive changes outside ± 4 are equally distributed with 0.33% at each tail which is almost twice as the observed value (0.15%).

2.5 Summary

In this chapter, we provided an introduction to the ADS model suggested by Rydberg and Shephard (2000, 2003). The ADS model decomposes index changes into three components: the index activity, the direction and the size of index change, if an index change occurs. This decomposition is mostly helpful for high-frequency traders who are interested in the movement of indices over very small periods of time. We fitted the ADS

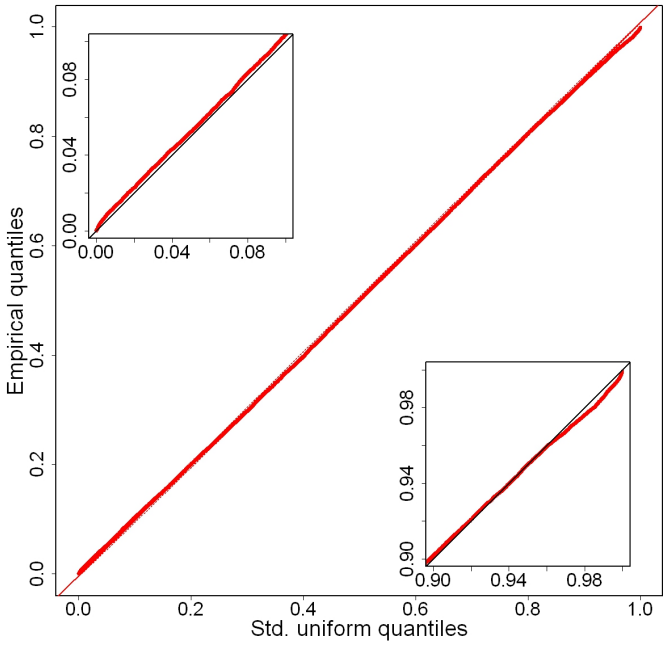


Figure 2.1: A standard uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution. The first and the last 10% of tails are shown on a larger scale.

model to a set of FTSE100 index changes using the Bayesian approach implemented via MCMC methods. We also obtained the posterior predictive distribution for the following day's transactions. In order to assess the performance of the MCMC chains and the probabilistic forecasts of index change, we used the Gelman and Rubin's convergence diagnostic and the PIT methods, respectively.

The overall results of one day's analysis confirmed that in general index change tended to be in-active, which may suggest the high level of liquidity[2] in the market. This probability increased if the previous index change was active. We also saw that both negative and positive changes had equal chances of occurrence.

The ADS model has, however, some limitations. For example, our interpretation is only valid when results from all the three models are available. In addition, modelling each component separately may lead to us ignoring the simultaneous effect of one component as well as its covariates on the other two components. For example, when we are modelling the size of index change, we do not consider the effect of the previous in-active

[2]"The ability to buy or sell significant quantities of a security quickly, anonymously, and with little price impact" (Tsay, 2005, Chapter 5).

indices. Also, we only consider the volume of the previous active index change and ignore the fact that one or more in-active changes with non-zero volumes could have happened before the current active index change.

Therefore, if we model index change as a single process, we may overcome the limitations of the ADS model that we have mentioned above. However, there is not a wide range of distributions available which are defined over the range of negative and positive integer values. So far the only choice would be the Poisson difference distribution which we introduce for modelling UHF data in the next chapter.

Chapter 3

Zero inflated Poisson difference model

3.1 Introduction

Modelling the difference of two discrete variables has attracted special interest in many research areas such as medicine (e.g. when before and after treatment measurements are considered (Karlis and Ntzoufras, 2006)), sport (e.g. the goal difference in sports involving two competing teams (Karlis and Ntzoufras, 2009)), finance (e.g. index change in the case of UHF data) and image analysis (e.g. the number of grey values in an image (Li et al., 2009)) . In the literature a wide range of techniques are available for dealing with the difference of two continuous or binary variables, while methods for modelling the difference of two non-negative integer-valued discrete random variables are rare. The available methods are mainly based on normal approximations of discrete distributions. However, in most cases normal approximations are not valid, since such data may take on a small range of integers (Karlis and Ntzoufras, 2006). Thus, techniques that are directly based on the discrete distributions can improve the inference of integer-valued random variable.

In the literature such variables are mostly decomposed into two (a trinomial variable -1, 0 and 1 indicating negative, zero and positive differences, and a non-zero positive integer-valued variable for the size of differences (Liesenfeld et al., 2006; Bien et al., 2011)) or three components (a binary variable 0 and 1 denoting zero and non-zero differences, a binary variable -1 and 1 representing negative and positive differences, and

a non-zero positive integer-valued variable corresponding to the size of differences (McCulloch and Tsay, 2001; Rydberg and Shephard, 2003)). We used the latter method in Chapter 2 and showed that when we are dealing with large data sets such method may be time consuming. In addition, it may lead to us ignoring the simultaneous effect of one component on the other components. Alternatively, we may consider a distribution that is defined over an integer domain. A possible choice for such a distribution may be the Poisson difference (PD) distribution which had been suggested, originally, for the difference of two Poisson random variables with equal parameters (Irwin, 1937) and later, for unequal values of the parameters (Skellam, 1946). This distribution can be used for the difference of any two Poisson variables, even when the two variables are not independent (Karlis and Ntzoufras, 2009)

While the PD distribution has been widely used in the literature for applications in medicine, sport (Karlis and Ntzoufras, 2009, 2006) and image analysis (Hwang et al., 2007), its application in finance, especially in modelling UHF data sets, is yet to be developed. In a recent study, Alzaid and Omair (2010) fitted a PD model to a set of high-frequency data from the Saudi stock exchange that were recorded every minute. Their study was limited to comparing the maximum likelihood and the method of moments estimators of the parameters of the PD distribution. Here, we introduce the application of the PD model using covariates for handling ultra high-frequency data in a Bayesian framework. We consider the application of the PD model to data when the difference of the two variables is only observed. To capture the (possible) excess of zeros, we use the zero inflated version of the PD model. Furthermore, we demonstrate our methodology through a simulated data set as well as FTSE100 indices.

The remainder of this chapter is organised as follows. Section 3.2 provides an overview of the PD and the zero inflated PD (ZPD) distributions. Bayesian inference, diagnostics and prediction associated with the ZPD model is described in Section 3.3. The results are presented in Section 3.4. An outline summary of the chapter is given in Section 3.5.

3.2 Poisson difference distribution

Let us assume $X \sim \mathrm{P}(\lambda_1)$ and $Y \sim \mathrm{P}(\lambda_2)$ are two independent Poisson random variables with the following joint probability function

$$p_{\mathrm{X,\,Y}}(x,y) = \frac{e^{-(\lambda_1+\lambda_2)}\lambda_1^x\lambda_2^y}{x!y!}. \tag{3.1}$$

The random variable $Z = X - Y$ follows a $\mathrm{PD}(\lambda_1,\lambda_2)$ distribution (Skellam, 1946), defined on the set of integer numbers $\mathbb{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$, with the following probability function

$$\begin{aligned} f_{\mathrm{PD}}(Z=z|\lambda_1,\lambda_2) = p_{\mathrm{Z}}(z) &= \sum_{y=0}^{\infty} p_{\mathrm{X,\,Y}}(z+y,y) \\ &= \sum_{y=0}^{\infty} \frac{e^{-(\lambda_1+\lambda_2)}\,\lambda_1^{z+y}\,\lambda_2^{y}}{(z+y)!\,y!} \\ &= e^{-(\lambda_1+\lambda_2)} \sum_{y=0}^{\infty} \frac{\lambda_1^{z+y}\,\lambda_2^{y}}{(z+y)!\,y!} \\ &= \mathrm{e}^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^{\frac{z}{2}} \mathrm{I}_{|z|}\left(2\sqrt{\lambda_1\lambda_2}\right), \end{aligned} \tag{3.2}$$

where $\lambda_1,\lambda_2 > 0$ and $\mathrm{I}_r(x)$ is the modified Bessel function of order r (Abramowitz and Stegun, 1964), given by

$$\mathrm{I}_r(x) = \left(\frac{x}{2}\right)^r \sum_{k=0}^{\infty} \frac{(x^2/4)^k}{k!\Gamma(r+k+1)}. \tag{3.3}$$

3.2.1 Properties

We list below some of the important properties of the PD distribution.

- In general, the odd and even cumulants are equal to $\lambda_1 - \lambda_2$ and $\lambda_1 + \lambda_2$, respectively. That is, the mean and the variance of the PD distribution are as follows

$$\mu = \lambda_1 - \lambda_2 \quad \text{and} \quad \sigma^2 = \lambda_1 + \lambda_2.$$

- The sign of $\lambda_1 - \lambda_2$ determines the skewness of the PD distribution. A positive (negative) skewness is a result of $\lambda_1 > \lambda_2$ ($\lambda_1 < \lambda_2$). For values of $\lambda_1 = \lambda_2$, the PD distribution is symmetric.

- The PD distribution can be approximated by a normal distribution for large value of $(\lambda_1 + \lambda_2)$.

- $\forall\ \lambda_2 \to 0$, the PD distribution becomes a Poisson distribution. On the other hand, the PD distribution tends to a Poisson distribution on the negative axis for values of λ_1 close to zero.

- The following type of symmetry holds for the PD distribution

$$f_{\text{PD}}(z|\lambda_1,\ \lambda_2) = f_{\text{PD}}(-z|\lambda_2,\ \lambda_1). \tag{3.4}$$

- For a random sample of n iid variables, $Z_i \sim \text{PD}(\lambda_1, \lambda_2)$, $i = 1, \ldots, n$, $S_n = \sum_{i=1}^{n} Z_i \sim \text{PD}(n\lambda_1, n\lambda_2)$.

Ntzoufras (2009) and Karlis and Ntzoufras (2006) have discussed the properties of the PD distribution in detail. Also, Alzaid and Omair (2010) have provided details on obtaining the maximum likelihood estimates of the parameters.

Considering the expected value and the variance of the PD distribution, the method of moments estimators of the parameters of the PD distribution, λ_1 and λ_2, can be obtained as follows

$$\tilde{\lambda}_1 = \frac{1}{2}(\mu + \sigma^2) \quad \text{and} \quad \tilde{\lambda}_2 = \frac{1}{2}(\mu - \sigma^2).$$

Moreover, for the FTSE100 data set, the method of moments estimates of the parameters λ_1 and λ_2 are 0.459 and 0.460, respectively. With these values, the PD distribution would tell us that 48% of index changes will be zero. We have already mentioned, in Chapter 2, that on March 25 out of $n = 45,266$ index changes, there were 31,373 inactive indices. That is about 69% of index changes were zero, a figure which is heavily

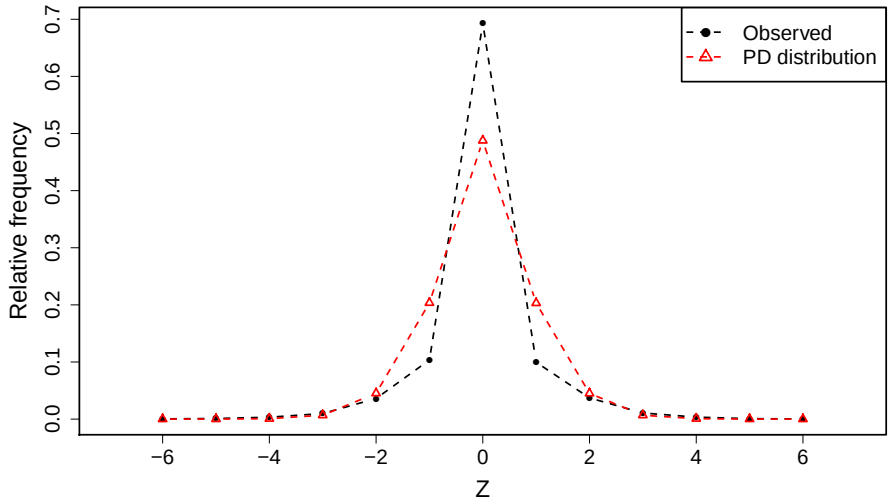


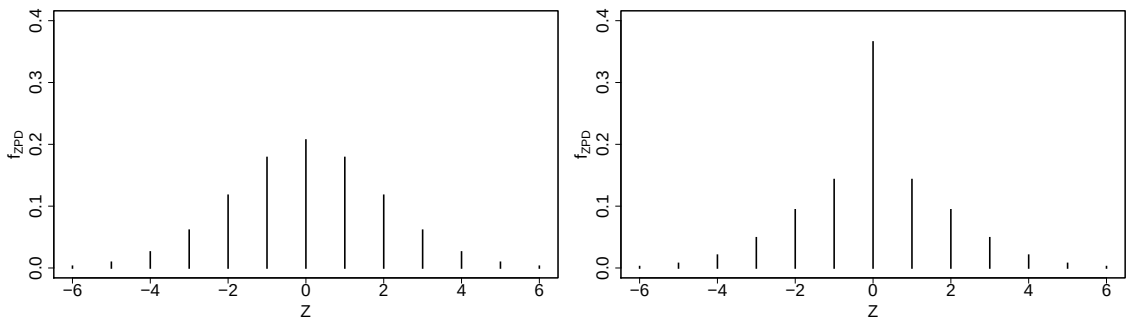
Figure 3.1: The comparison of the empirical distribution of the observed index change and the density of the PD distribution with the methods of moment estimates of λ_1 and λ_2.

underestimated by the PD distribution (Figure 3.1). Therefore, we need to use a zero inflated version of the PD distribution to capture the excess of zero values in the data (Karlis and Ntzoufras, 2006).

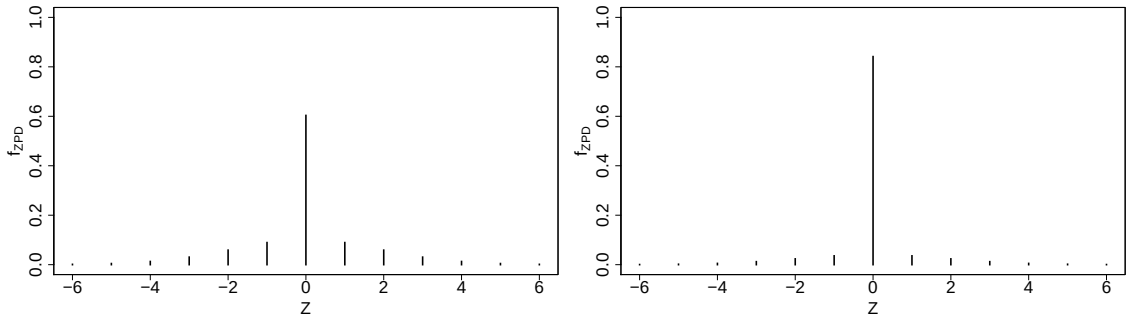
3.2.2 Zero inflated Poisson difference distribution

In real life applications count (or integer-valued) data sets, such as dental (Karlis and Ntzoufras, 2006; Mwalili, 2008), spatial (Agarwal et al., 2002) and sports (Karlis and Ntzoufras, 2009), may contain an excess of zero values, i.e. more than what the model would predict; see Ntzoufras (2009, Chapt. 8) and references therein for further examples. By introducing an extra probability parameter, zero inflated distributions enable us to capture the excess of zero values.

The zero inflated version of the PD distribution was first introduced by Karlis and Ntzoufras (2006) in order to model dental epidemiology data and also it was used by Karlis and Ntzoufras (2009) to model football data. Let $f_{\mathrm{PD}}(z;\lambda_1,\ \lambda_2)$ be the probability function of the PD distribution with the parameters λ_1 and λ_2 given by (3.2). The probability



(a) $p = 0$ *(b)* $p = 0.2$



(c) $p = 0.5$ *(d)* $p = 0.8$

Figure 3.2: The ZPD distribution with parameters $\lambda_1 = \lambda_2 = 2$ and four values of p.

function of the zero inflated Poisson difference distribution can be defined as follows

$$f_{\text{ZPD}}(z|\lambda_1,\lambda_2,p) = \begin{cases} p + (1-p) f_{\text{PD}}(z \mid \lambda_1,\lambda_2) & \text{if } z = 0, \\ (1-p) f_{\text{PD}}(z \mid \lambda_1,\lambda_2) & \text{if } z \neq 0, \end{cases} \tag{3.5}$$

for $z \in \mathbb{Z}$, where p is the proportion of extra zero values. According to (3.5), the first part of the model represents the probability of all zero values, while the probability of $z \neq 0$ is given by the second part. Figure 3.2 compares the ZPD distribution for $\lambda_1 = \lambda_2 = 2$ and different values of p. It can be seen that as the value of p increases the ZPD distribution has smaller range with a mean around zero.

This can be confirmed by obtaining the mean and the variance of such a distribution

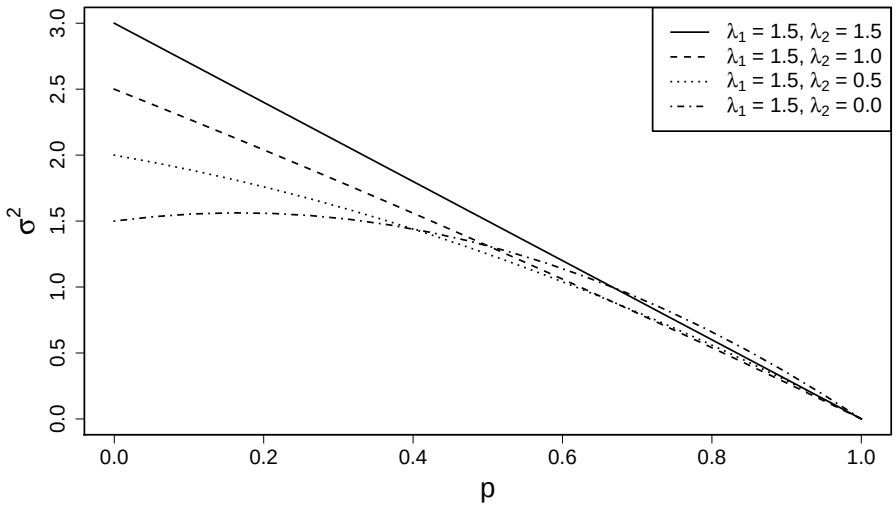


Figure 3.3: The behaviour of the variance for different values of λ_1 and λ_2 over the range of $p \in [0,1]$

as follows.

$$\begin{aligned} \mathrm{E}(Z) &= \sum_{i=1}^{n} z_i f_{\mathrm{ZPD}}(\lambda_1,\lambda_2,p) = \sum_{i=1}^{n} 0\, I_{\{z=0\}} p + z_i(1-p) f_{\mathrm{PD}}(\lambda_1,\lambda_2) \\ &= (1-p)\, \sum_{i=1}^{n} z_i\, f_{\mathrm{PD}}(\lambda_1,\lambda_2) = (1-p)\, \mathrm{E}(Z_{\mathrm{PD}}), \end{aligned} \tag{3.6}$$

where $Z_{\mathrm{PD}} \sim \mathrm{PD}(\lambda_1,\lambda_2)$, and

$$I_{\{0\}} = \begin{cases} 1 & \text{if} \quad z = 0, \\ 0 & \text{if} \quad z \neq 0, \end{cases} \tag{3.7}$$

therefore $\mathrm{E}(Z) = (1-p)(\lambda_1 - \lambda_2)$. Similarly $\mathrm{E}(Z^2)$ is given by $(1-p)\mathrm{E}(Z_{\mathrm{PD}}^2)$, therefore

$$\begin{aligned} \mathrm{Var}(Z) &= \mathrm{E}(Z^2) - [\mathrm{E}(Z)]^2 = (1-p)\left[\mathrm{Var}(Z_{\mathrm{PD}}) + [\mathrm{E}(Z_{\mathrm{PD}})]^2 - (1-p)\ [\mathrm{E}(Z_{\mathrm{PD}})]^2\right] \\ &= (1-p)\left[\mathrm{Var}(Z_{\mathrm{PD}}) + p\ [\mathrm{E}(Z_{\mathrm{PD}})]^2\right] \\ &= (1-p)\left[(\lambda_1+\lambda_2) + p(\lambda_1-\lambda_2)^2\right]. \end{aligned} \tag{3.8}$$

Figure 3.3 illustrates the behaviour of the variance for different values of λ_1 and λ_2 over $p \in [0,1]$. It can be seen that when $\lambda_1 = \lambda_2$, the variance is a linear function of p and when

$\lambda_1 \neq \lambda_2$, the variance becomes a nonlinear function of p with the nonlinearity increasing as the difference between λ_1 and λ_2 gets bigger.

3.3 A model for index change

Let us assume index change, Z_i, for $i = 1, \ldots, n$, where n is the number of index changes in a given data set, is the response variable such that

$$Z_i \sim \text{ZPD}(\lambda_{1,i}, \lambda_{2,i}, p).$$

Here, we adopt the same model structure as Karlis and Ntzoufras (2009) for modelling the goal difference, thus we can set the ZPD model parameters as follows

$$\begin{aligned} \log(\lambda_1) &= \mathbf{x}\alpha, \\ \log(\lambda_2) &= \mathbf{x}\beta, \end{aligned} \tag{3.9}$$

where $\mathbf{x}$ is a matrix of covariates, and α and β are parameter vectors of appropriate size. Similar to Chapter 2, in our model we may consider the effect of the previous index change z_{i-1}, previous volume of transactions v_{i-1} and the time duration between two consecutive transactions $\Delta t_i = t_i - t_{i-1}$. Thus, (3.9) in terms of the covariates can be written as follows

$$\begin{aligned} \log(\lambda_{1,i}) &= \alpha_0 + \alpha_1 z_{i-1} + \alpha_2 \log(v_{i-1}) + \alpha_3 \log(\Delta t_i + 1), \\ \log(\lambda_{2,i}) &= \beta_0 + \beta_1 z_{i-1} + \beta_2 \log(v_{i-1}) + \beta_3 \log(\Delta t_i + 1), \end{aligned} \tag{3.10}$$

for $i = 2, \ldots, n$. We denote the parameter vectors as $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)^{\mathrm{T}}$ and $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^{\mathrm{T}}$ and consider the matrix of covariates $\mathbf{x}$ as an $n \times 4$ matrix given by:

$$\mathbf{x} = \begin{pmatrix} 1 & z_1 & \log(v_1) & \log(\Delta t_2 + 1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_i & \log(v_i) & \log(\Delta t_i + 1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_{n-1} & \log(v_{n-1}) & \log(\Delta t_n + 1) \end{pmatrix}. \tag{3.11}$$

3.3.1 Bayesian inference

Let us, in the model (3.10), denote the parameter vector by $\psi = (\alpha^{\mathrm{T}},\ \beta^{\mathrm{T}},\ p)$. In the Bayesian framework we obtain our inference based on the posterior distribution of the parameter vector ψ. In the absence of any specific information, we choose a normal prior distribution with mean zero and a large variance (e.g. 10^4) for all of the parameters, excepting p which is uniform in the range of $(0,1)$. Here we assume all nine model parameters to be independent. Thus, the posterior distribution of ψ is given by

$$f_{\text{post}}(\psi|z) \propto f_{\text{like}}(z|\psi) f_{\text{prior}}(\psi),$$

where $f_{\text{like}}(z|\psi) = \prod_{i=1}^{n} f(z_i|\psi)$ is the likelihood and $f_{\text{prior}}(\psi)$ is the prior distribution given by

$$f_{\text{prior}}(\psi) = f_{\text{P}}(p) \prod_{j=0}^{3} f_{\alpha_j}(\alpha_j) \prod_{j=0}^{3} f_{\beta_j}(\beta_j).$$

For the case of the ZPD model, the likelihood in terms of (3.9) is given by

$$\begin{aligned} f_{\text{like}}(z|\psi) &= \prod_{i=1}^{n} f_{\text{ZPD}}(z_i|\psi) = \prod_{i=1}^{n} \left[pI_{\{0\}} + (1-p) f_{\text{PD}}(z_i|\alpha,\beta) \right] \\ &= \prod_{i=1}^{n} \left[pI_{\{0\}} + (1-p)\, \mathrm{e}^{-(\lambda_{1,i}+\lambda_{2,i})} \left(\frac{\lambda_{1,i}}{\lambda_{2,i}} \right)^{z_i/2} \mathrm{I}_{|z_i|} \left(2\sqrt{\lambda_{1,i}\lambda_{2,i}} \right) \right] \\ &= \prod_{i=1}^{n} \left[pI_{\{0\}} + (1-p)\, \exp\left[-(e^{\mathbf{x}_i\alpha} + e^{\mathbf{x}_i\beta}) + \frac{z_i}{2}(\mathbf{x}_i\alpha - \mathbf{x}_i\beta) \right] \mathrm{I}_{|z_i|} \left(2e^{\frac{\mathbf{x}_i\alpha + \mathbf{x}_i\beta}{2}} \right) \right], \end{aligned} \tag{3.12}$$

where $\mathbf{x}_i$ is the ith row of the matrix of covariates in (3.11) and $\mathrm{I}_{\{0\}}$ is given by (3.7). The prior distribution is

$$f_{\text{prior}}(\psi) = \prod_{j=0}^{3} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-\alpha_i^2}{2\sigma^2}} \right) \prod_{j=0}^{3} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-\beta_i^2}{2\sigma^2}} \right), \tag{3.13}$$

where σ^2 is set to be 10^4. Thus, the posterior distribution of the model parameters is given by

$$f_{\text{post}}(\psi|z) \propto \exp\left(-\sum_{j=0}^{3}\frac{\alpha_i}{2\sigma^2}\right)\exp\left(-\sum_{j=0}^{3}\frac{\beta_i}{2\sigma^2}\right)\times$$

$$\prod_{i=1}^{n}\left[pI_{\{0\}}+(1-p)\,\exp\left(-(e^{\mathbf{x}_i\alpha}+e^{\mathbf{x}_i\beta})+\frac{z_i}{2}(\mathbf{x}_i\alpha-\mathbf{x}_i\beta)\right)\mathrm{I}_{|z_i|}\left(2e^{\frac{\mathbf{x}_i\alpha+\mathbf{x}_i\beta}{2}}\right)\right]. \tag{3.14}$$

It can be seen that the posterior is not known explicitly, therefore, in order to generate samples from the posterior, in a similar way to Chapter 2, we use the M-H algorithm.

3.3.2 Metropolis Hastings algorithm

In this chapter, we implement the random walk M-H algorithm given by Algorithm 2.1 in Section 2.3.1. Further points that we consider are as follows

(i) We update parameters one at the time.

(ii) We use the logit transform for p. Let $y = \log(p/(1-p))$, we draw $y^{\text{cand}} \sim \mathrm{N}(y^{(k-1)}, \sigma_y^2)$. Then, in order to use the uniform distribution as a proposal distribution $q(.|.)$ in the acceptance ratio given by (2.18), we transform back the normal probability density function to the probability density function of p using the following formula

$$f_P(p) = \frac{1}{p(1-p)}\, f_Y(y),$$

where $f_P(p)$ and $f_Y(y)$ are uniform and normal probability functions, respectively.

3.3.3 Predictive distribution

Let $Z^\star = (Z_1^\star, \ldots, Z_s^\star)^{\mathrm{T}}$ be the s index changes for the whole of next trading day which we wish to predict. The posterior predictive distribution is defined as

$$f_{\text{pred}}(Z^\star|z) = \int f_{\text{like}}(Z^\star|\psi) f_{\text{post}}(\psi|z) d\psi,$$

where $f_{\text{like}}(Z^{\star}|\psi)$ may be given as

$$f_{\text{like}}(Z^{\star}|\psi) = \prod_{j=1}^{s} f(Z_j^{\star}|\psi).$$

In order to generate samples from the predictive distribution, at the kth iteration, for $k = 1,\ldots,m$, we add the following steps to the M-H algorithm.

For $i = 1,\ldots,s$ repeat the following steps

(i) Obtain

$$\lambda_{1,n+i}^{(k)} = \exp(\alpha_0^{(k)} + \alpha_1^{(k)} z^{\star(k)}_{n+i-1} + \alpha_2^{(k)} \log(v_{n+i-1}) + \alpha_3^{(k)} \log(\Delta t_{n+i}))$$

and

$$\lambda_{2,n+i}^{(k)} = \exp(\beta_0^{(k)} + \beta_1^{(k)} z^{\star(k)}_{n+i-1} + \beta_2^{(k)} \log(v_{n+i-1}) + \beta_3^{(k)} \log(\Delta t_{n+i})).$$

When $i = 1$, $z^{\star}_{n+i-1} = z_n$, $v_{n+i-1} = v_n$ are the last observed index change and volume;

(ii) Draw $b_{n+i} \sim \text{Bernoulli}(\hat{p})$:

- If $b_{n+i} = 1$ let $Z^{\star}_{n+i} = 0$, otherwise draw $Z^{\star}_{n+i} \sim \text{PD}(\lambda_{1,n+i}^{(k)}, \lambda_{2,n+i}^{(k)})$.

3.3.4 Model checking

In order to assess the performance of the ZPD model we undertake model diagnostics using methods discussed in Section 2.3.3. The Gelman and Rubin's convergence diagnostics are applied to monitor the convergence of the MCMC output (described in Section 2.3.1.1). The next step is to investigate the goodness of fit of the ZPD model to the data. This is undertaken using a standard χ^2-test. Finally to assess the next day's predictions, we use the PIT discussed in Section 2.3.3.1.

Model comparison is also undertaken in order to decide how much complexity is necessary to fit the data. That is, whether the full model fits the data properly or a model with fewer parameters has a similar performance. For this purpose, we use the *deviance*

information criterion (DIC). A numerical summary of model fit is given by the deviance which is defined as -2 times the log-likelihood:

$$D(z,\psi) = -2\log f_{\text{like}}(z|\psi). \tag{3.15}$$

We define two further Bayesian measures related to the deviance; the posterior mean deviance

$$\hat{D}_{avg}(z) = \frac{1}{L}\sum_{l=1}^{L} D(z,\psi^l), \tag{3.16}$$

where L is the number of the MCMC iterations after burn-in, and the deviance at $\hat{\psi}$,

$$D_{\hat{\psi}}(z) = D(z,\hat{\psi}(z)), \tag{3.17}$$

where $\hat{\psi}$ is the plug-in estimate, such as the posterior mean of the parameter vector ψ. The DIC of a model is defined as follows

$$\text{DIC} = p_{\text{D}} + \hat{D}_{avg}(z), \tag{3.18}$$

where p_{D} represents the effect of model fitting and has been used as a measure of the effective number of parameters of a given Bayesian model M, defined as follows

$$p_{\text{D}} = \hat{D}_{avg}(z) - D_{\hat{\psi}}(z),$$

where $\hat{D}_{avg}(z)$ and $D_{\hat{\psi}}(z)$ are given by (3.16) and (3.17); see Gelman et al. (2003, Chapter 6) for further details.

3.4 Results

In this section we illustrate the application of the ZPD model. We first conduct a simulation study to show how well our model diagnostics work, and then we fit the model to the FTSE100 index changes. In the M-H algorithm we set the number of iterations to $m = 10000$ and we discard the first 5000 samples as burn-in.

Coefficients	True	Mean	95% credible intervals		Coefficients	True	Mean	95% credible intervals	
α_0	-0.237	-0.147	-0.171	-0.122	β_0	-0.263	-0.145	-0.170	-0.122
α_1	-0.091	-0.092	-0.108	-0.076	β_1	0.143	0.146	0.131	0.163
α_2	0.014	0.017	-0.009	0.044	β_2	0.067	0.049	0.025	0.074
α_3	0.248	0.239	0.207	0.272	β_3	0.232	0.241	0.210	0.273
p	0.256	0.253	0.242	0.264					

Table 3.1: The true values, the posterior means and the 95% credible intervals of the posterior distribution of the ZPD model parameters based on the simulated data.

Model	DIC	$\hat{\mathrm{D}}_\theta$	$\hat{\mathrm{D}}_{ave}$	p_{D}
Intercept	151701.40	151695.30	151698.40	3.05
z_{i-1}	134416.70	134396.10	134406.40	10.28
$z_{i-1}+\log(v_{i-1})$	134335.20	134319.10	134327.20	8.04
$z_{i-1}+\log(\Delta t_{i-1})$	134341.80	134331.40	134336.60	5.21
$z_{i-1}+\log(v_{i-1})+\log(\Delta t_i)$	118165.40	118145.50	118155.50	9.93

Table 3.2: The DIC, the point-estimate, the average deviances and the estimated number of parameters for each of five models fitted to the simulated index change.

3.4.1 Simulation study

We simulated a set of data from a ZPD model as described in (3.10) with the following arbitrary values for the model parameters

$$\psi = (-0.237, -0.097, 0.014, 248, -0.263, -0.143, 0.067, 0.232, 0.25).$$

Let us point out that in (3.10) we assumed that the values of volume and time duration between two consecutive transactions were the same as in the real data set.

Time series plots of the sampled values of the parameter vector ψ, are presented in Figure B.2. Gelman and Rubin's convergence diagnostic results are presented in Table B.2 and Figure B.1. We can see that the obtained R statistics are less than 1.1 for all of the nine parameters of the ZPD model which satisfying the convergence condition. Finally, estimates of the posterior distributions of the model parameters are illustrated in Figure B.3.

The results are summarised in Table 3.1 which provides the true value, the posterior mean, and the 95% credible intervals of the posterior distributions of the model parameters. It can be seen that the posterior means, except for the intercepts, are close to the true values of the parameters, which suggests that the fitted ZPD model can capture the behaviour of the simulated data well.

Coefficients	Mean	SD	95% credible intervals		Coefficients	Mean	SD	95% credible intervals	
α_0	-0.133	0.017	-0.166	-0.098	β_0	-0.136	0.017	-0.169	-0.100
α_1	-0.096	0.014	-0.123	-0.069	β_1	0.147	0.013	0.119	0.172
α_2	0.037	0.019	-0.000	0.073	β_2	0.081	0.018	0.045	0.116
α_3	0.261	0.021	0.222	0.306	β_3	0.253	0.021	0.210	0.295
p	0.539	0.005	0.529	0.548					

Table 3.3: The posterior means, the standard deviation and the 95% credible intervals of the posterior distribution of parameters of the ZPD model based on FTSE100 data.

Model	DIC	$\hat{\mathrm{D}}_\theta$	$\hat{\mathrm{D}}_{ave}$	p_{D}
Intercept	144856.00	144848.40	144852.20	3.79
z_{i-1}	106132.20	106107.00	106119.60	12.61
$z_{i-1}+\log(v_{i-1})$	101360.10	101343.90	101351.90	8.19
$z_{i-1}+\log(\Delta t_{i-1})$	101387.20	101375.60	101381.40	5.79
$z_{i-1}+\log(v_{i-1})+\log(\Delta t_i)$	89719.86	89698.80	89709.33	10.52

Table 3.4: The DIC, the point-estimate, the average deviances and the estimated number of parameters for each of five models fitted to the FTSE100 index change.

Table 3.2 shows that a model with three covariates has a better fit than models with fewer covariates. The posterior predictive distribution of the predicted values was almost identical to the distribution of true values. The proportion of predicted values within the interval of $[-4,4]$ is almost 99.77% which is the same as in the case of the true values. Also, about 0.23% of the predicted values had a magnitude of 4 ticks, which is the same as the true value. The χ^2 goodness of fit test resulted in a p-value of 0.87 which may suggest an adequate fit. Furthermore, the Q-Q plot in Figure 3.4(a) and the p-value of 0.57 obtained from the K-S test for assessing predicted values using the PIT, both confirmed that the model is able to characterise well the density of the simulated index changes for the whole of the next day.

3.4.2 FTSE100 futures

This section provides the results from fitted the ZPD model to the FTSE100 index change. The R statistics with values of 1.01 suggest that our MCMC chains are mixing very well (Table B.3 and Figure B.4). Traces and the estimates of the posterior distributions of the model parameters are also provided in Figures B.5 and B.6

Table 3.4 compares the values of DIC of the ZPD models with different covariates. It suggests that the model with three covariates and nine parameters has the lowest values of DIC, with the corresponding value of the effective parameters $p_{\mathrm{D}} = 10.52$. A summary

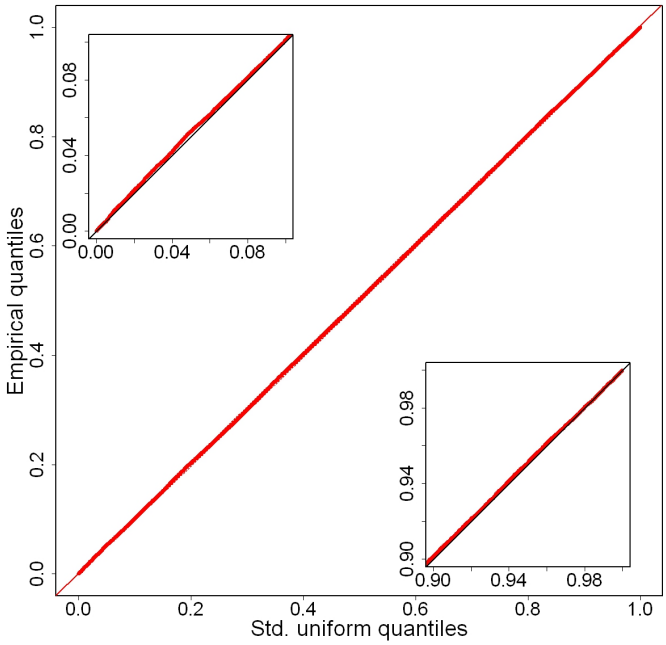
of the posterior distributions of the model parameters is provided in Table 3.3. The 95% credible intervals of the posterior distributions of the model parameters do not contain zero which may indicate the significant effect of the covariates on index change.

We may interpret the effect of the covariates in the model as follows. The previous index change has a larger effect on λ_2 compared to λ_1, which implies that in the next transaction a switch from a positive to a negative index change is more likely than a switch from a negative to a positive one. We can also see that the expected value of index change tends to be negative as the volume increases, which may suggest a reduction in the chance of getting a positive change after a transaction with a large volume. Since the posterior means of α_3 and β_3 are almost the same, the mean index change at the next transaction, irrespective of when it occurs, is expected to be zero, assuming all the other variables (previous volume and index change) are fixed.

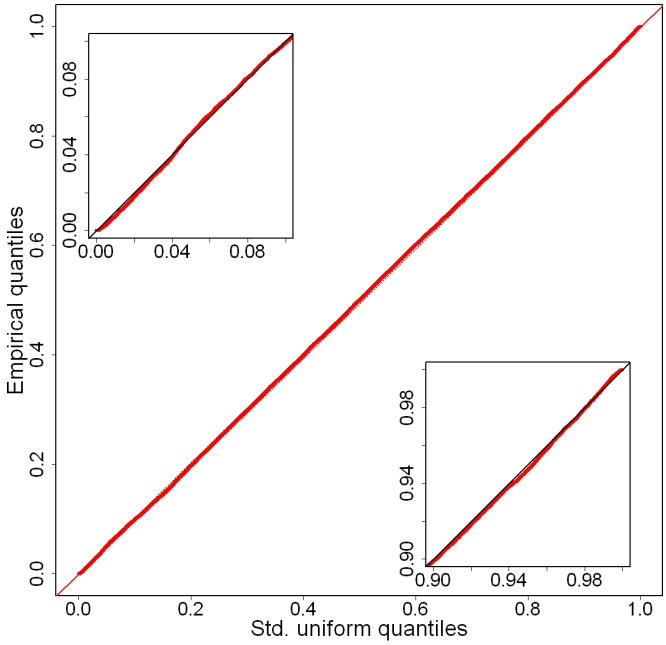
The predictive distribution of index change over the whole day of March 26 was produced and it resulted in an almost similar distribution to index change. The predictive distribution of index change indicated that about 99.88% of changes may occur within ± 4 which is very close to the real data set (99.68%). The goodness of fit was confirmed by the χ^2 test (p-value of 0.11). The standard uniform Q-Q plot (Figure 3.4(b)) and the K-S test (p-value of 0.23) resulted from the PIT analysis suggest that the predictive distribution is able to explain the behaviour of index change fairly well. However, a closer look shows that 0.32% of changes in the real data set occurred with more than 4 ticks, whereas in the case of the posterior predictive distribution this was about 0.12%. This indicates that the ZPD model may perform poorly for values in the tails. Finally, we obtain the DIC for the ADS model in order to compare the performance of the ADS and ZPD models. The DIC for the ADS model was 111630.90 which, in comparison with the DIC for the ZPD model in Table 3.4, suggests that the ZPD model performs better.

3.5 Summary

In this chapter, we provided an overview of the ZPD distribution. We proposed the application of the ZPD model to the FTSE100 index changes in a Bayesian framework via



(a)



(b)

Figure 3.4: A standard uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution, (a) simulated data, (b) FTSE100 index change. The first and the last 10% of tails are shown on a larger scale.

MCMC methods. The Bayesian estimation of the parameters of interest were presented in detail. One of our principle interests concerned the distribution of index change for the whole trading day and we have shown that our approach models the behaviour of index

change fairly well. Results from the real data revealed that the previous index change, the volume of transaction and the time duration between two consecutive transactions significantly affect the FSTE100 index change. Finally, various model diagnostics were carried out which showed that small and moderate index changes were explained well by the fitted ZPD model. However, there was some evidence of underestimation in the tail of the distribution.

The ZPD model was fitted to index change assuming that the parameters are static in time. This type of modelling is useful for the *off-line* analysis of the market when the whole data set is available and our interest lies in understanding the overall behaviour of the market. However, in practice, one would be interested in performing the analysis in real time. As transactions occur sequentially, we would like to update our knowledge of the market as new observations arrive. MCMC methods are not computationally feasible for this purpose. Alternatively, we may use *sequential Monte Carlo* (SMC) methods for performing online inference and prediction.

In the next two chapters, we explore ways for extending our current model. In Chapter 4, we introduce a dynamic Poisson difference model and its zero inflated version using latent processes. These models are fitted using particle filters. To address the problem of underestimation of the tails, there are several possible alternatives to the PD distribution, for example the distribution of the difference of two negative binomial random variables (to capture over-dispersion) or the distribution of the difference of two generalised Poisson random variables with unequal parameters (to capture over- or under-dispersion). Here, we investigate the latter as the complicated structure of the former option may make the application of such distribution infeasible. In addition, the distribution of the difference of two generalised Poisson variables may have a wider application as it introduces over- as well as under-dispersion and also can be considered as a generalisation of the PD distribution. Thus, in Chapter 5 we investigate the generalised Poisson difference distribution with four parameters and discuss its possible application for modelling UHF data.

Chapter 4

Modelling via particle filters

4.1 Introduction

In the previous chapter we modelled the FSTE100 index change as a single process using the ZPD model in a Bayesian framework. In such context, we assumed that model parameters are static in time. This type of modelling is suited for the empirical analysis of financial markets when the whole data set is available, in other words, when the data flows stops after the closing time of the market. However, in real life applications, one would be interested in performing inference and prediction as a new set of information arrives. For this purpose we need a class of dynamic models.

Dynamic models, in contemporary time series analysis, have attracted the attention of researchers and practitioners. By allowing the study and estimation of complex dynamics, such models enable us to model time series data for a wide range of response distributions. In this context, using the state space models (as we explain later) has provided a very flexible yet simple tool for analysing dynamic events. It also has led to expand the range of the application of statistical time series analysis to include non-stationary, irregular processes, systems with continuous-time, and discrete data (Petris et al., 2009).

Early efforts to develop inference for such models, by Harrison and Stevens (1976), focused on an important class of state space models known as the *Gaussian linear state space models*, or *dynamic linear models* (DLM)(Petris et al., 2009; Gamerman and Lopes, 2006). These models have been generalised for the events in which time series data

have been generated from exponential family distributions under the *dynamic generalised linear models* (DGLM) (Smith, 1979; Smith and Miller, 1986; Harvey and Fernandes, 1989; West et al., 1985); further examples can be found in Triantafyllopoulos (2009). Since then, DGLM's have been widely adopted for non-Gaussian and non-linear time series modelling (Petris et al., 2009; Triantafyllopoulos, 2009; Lopes and Tsay, 2011).

To run estimation in real time, various approaches are available (Triantafyllopoulos, 2009; Lopes and Tsay, 2011). Most of these methods are applicable in the case of linear models with Gaussian errors, when the posterior distribution is analytically available. Complex models often lead to integrals that cannot be solved analytically. This has created an increase in the popularity of Bayesian methods that utilize Monte Carlo (MC) based approximations. A set of MC techniques are MCMC methods which are suitable for the purpose of off-line applications and we have already presented their implementation in Chapters 2 and 3. However, MCMC based algorithms are prohibitively costly for the purpose of online estimation of states and parameters (Lopes and Tsay, 2011; Gamerman and Lopes, 2006).

Sequential Monte Carlo (SMC) methods are alternative set of simulation-based algorithms for online sampling which help us to approximate analytically intractable integrals. These methods are very flexible, easy to implement, and applicable in very general settings (Doucet et al., 2001). SMC methods, also known as particle filters[1] (PF) mainly consist of three basic operations: evolution, prediction and updating (Gamerman and Lopes, 2006). In these methods, a continuous probability distribution is approximated by a discrete distribution made of weighted draws called particles. As a new observation arrives, particles are updated to approximate the distribution and their weights based only on the previous step. Therefore, these methods are feasible to use as there is no need to store the information prior to the current stage. Their application in the field of economics was suggested by Kim et al. (1998) for studying the volatility of asset price. PF have been widely used since then in various areas of financial studies such as option pricing (Jasra and Moral, 2011), volatility analysis (Carvalho and Lopes, 2007) and portfolio choice

[1] In the literature (Doucet and Johansen, 2009) particle filters are defined as a subclass of sequential Monte Carlo methods. Here we adopt the definition by Lopes and Tsay (2011).

(Johannes et al., 2008); further examples can be found in Creal (2009) and Lopes and Tsay (2011).

This chapter is organised as follows. Section 4.2 introduces the general state space model and basic particle filters and discusses the weaknesses and strengths of such methods. In Section 4.3 we first introduce the dynamic PD model and then generalise it to the DZPD model to capture the excess of zeros in the data. In Section 4.4, we illustrate the application of our models on a set of simulated data as well as the FTSE100 data. Finally, Section 4.5 concludes with some discussion and summary comments .

4.2 Inference in hidden Markov models

In this section we first introduce the notation and describe the standard inference problems that are associated with a hidden Markov model (HMM), also known as a state space model (SSM). We then describe basic PF methods and their application in HMM inference followed by a brief discussion on the strengths and limitations of PF methods. Finally, we discuss the auxiliary particle filters and Liu and West (2001) algorithms as an improvement and an extension to the basic particle filters.

4.2.1 Hidden Markov models

In the context of an HMM, a hidden Markov process $\{X_t\}_{t\geq 0}$ is characterised with its initial density $X_0 \sim \mu_\theta(x_0)$ and transition density function

$$X_t \,|\, (X_{t-1} = x_{t-1}) \sim f_\theta(x_t|x_{t-1}), \tag{4.1}$$

for some static parameter θ which may be multidimensional. The process $\{X_t\}_{t\geq 0}$ is only accessible through an observed process $\{Y_t\}_{t\geq 1}$. We assume the observations $\{Y_t\}_{t\geq 1}$ are independent given the unobserved process $\{X_t\}_{t\geq 0}$ with the following marginal distribution

$$Y_t \,|\, (X_t = x_t) \sim g_\theta(y_t|x_t). \tag{4.2}$$

The relation between an observed and an unobserved process is shown as a graph in

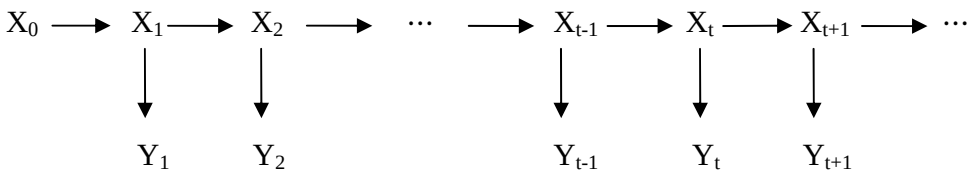


Figure 4.1: Conditional structure for an HMM. SMC methods enable us to make inference about the unobserved process given the observed process.

Figure 4.1. Let us, for any generic sequence $\{z_t\}$, denote $(z_i, \ldots, z_j)$ by $z_{i:j}$, for $i < j$. The figure shows any path connecting Y_t with one of the Y_s $(s < t)$ or with any of the states of X_s $(s < t)$, has to go through X_t; therefore, X_t separates $(X_{0:t-1}, Y_{1:t-1})$ and Y_t. Similarly, one can show that X_t and $(X_{0:t-2}, Y_{1:t-1})$ are conditionally independent given X_{t-1}.

We are interested in making inference about $\{X_t\}_{t\geq 0}$. Thus, in this context we are able to construct a Bayesian model, in which (4.1) and (4.2) define the prior and the likelihood respectively as follows

$$p_\theta(x_{0:t}) = \mu_\theta(x_0) \prod_{i=1}^{t} f_\theta(x_i|x_{i-1}), \tag{4.3}$$

$$p_\theta(y_{1:t}|x_{1:t}) = \prod_{i=1}^{t} g_\theta(y_i|x_i). \tag{4.4}$$

Let us assume $\theta \in \Theta$ is a known parameter. The inference about $X_{0:t}$, given a realisation of the observations $Y_{1:t} = y_{1:t}$, relies upon the posterior density given by

$$p_\theta(x_{0:t}|y_{1:t}) = \frac{p_\theta(x_{0:t}, y_{1:t})}{p_\theta(y_{1:t})}, \tag{4.5}$$

where

$$p_\theta(x_{0:t}, y_{1:t}) = p_\theta(x_{0:t})\; p_\theta(y_{1:t}|x_{1:t}), \tag{4.6}$$

and

$$p_\theta(y_{1:t}) = \int p_\theta(x_{0:t}, y_{1:t})\; dx_{0:t}. \tag{4.7}$$

The joint distribution of $x_{0:t}$ and $y_{1:t}$ in (4.6), also known as the *unnormalised posterior*, can be written as

$$p_\theta(x_{0:t}, y_{1:t}) = p_\theta(x_{0:t-1}, y_{1:t-1})\, f_\theta(x_t|x_{t-1})\, g_\theta(y_t|x_t). \tag{4.8}$$

Consequently, the posterior distribution in (4.5) in terms of (4.8) is given by

$$p_\theta(x_{0:t}|y_{1:t}) = p_\theta(x_{0:t-1}|y_{1:t-1})\, \frac{f_\theta(x_t|x_{t-1})\, g_\theta(y_t|x_t)}{p_\theta(y_t|y_{1:t-1})}, \tag{4.9}$$

where

$$p_\theta(y_t|y_{1:t-1}) = \int p_\theta(x_{t-1}|y_{1:t-1})\, f_\theta(x_t|x_{t-1})\, g_\theta(y_t|x_t)\, dx_{t-1:t}. \tag{4.10}$$

In the literature (Andrieu et al., 2010), approximating $p_\theta(x_{0:t}|y_{1:t})$ is referred to as optimal filtering problem. However, here we are only interested in the marginal distribution, $p_\theta(x_t|y_{1:t})$, which can be obtained by integrating (4.9) over $x_{0:t-1}$ and is given by

$$p_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t)\, p_\theta(x_t|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})}, \tag{4.11}$$

where $p_\theta(x_t|y_{1:t-1})$ is known as the prediction step and is given by

$$p_\theta(x_t|y_{1:t-1}) = \int f_\theta(x_t|x_{t-1})\, p_\theta(x_{t-1}|y_{1:t-1})\, dx_{t-1}.$$

Thus, by computing $p_\theta(x_t|y_{1:t})$ sequentially, from $p_\theta(x_{0:t}|y_{1:t})$, we then are able to evaluate the quantity of the marginal likelihood, $p_\theta(y_{1:n})$, as follows

$$p_\theta(y_{1:t}) = p(y_1) \prod_{i=2}^{t} p(y_i|y_{1:i-1}),$$

where $p(y_i|y_{1:i-1})$ is given by (4.10).

If θ is unknown, it is suggested to assign a prior density $p(\theta)$ to θ, therefore Bayesian

inference relies on the joint density of $x_{0:n}$ and θ where

$$p(\theta, x_{0:t}|y_{1:t}) \propto p_\theta(x_{0:t}, y_{1:t})\ p(\theta). \tag{4.12}$$

For nonlinear and non-Gaussian models, the posterior distributions, $p_\theta(x_{0:t}|y_{1:t})$ and $p(\theta, x_{0:t}|y_{1:t})$ are often impossible to obtain analytically. It is therefore necessary to employ numerical and approximation methods. Here, we use SMC methods for the purpose of approximating posterior distribution; see Andrieu et al. (2010) for further details.

4.2.2 Sequential Monte Carlo methods

Particle filters, which is how SMC is usually referred to in the context of HMM, is easier to understand when viewed as an extension of importance sampling. Thus, we explain these methods in the context of importance sampling. Suppose that we are interested in evaluating the expected value

$$\mathrm{E}\left[h_\theta(x_{0:t})\right] = \int h_\theta(x_{0:t}) p_\theta(x_{0:t}|y_{1:t})\ dx_{0:t}, \tag{4.13}$$

where h is any function of $x_{0:t}$. The law of large numbers confirms that sample averages tend to population moments as the number of draws increases. For complex models, it is generally impossible to draw samples directly from the target distribution because it is often unknown and non-standard.

Instead, we draw from an importance or proposal distribution $q_\theta(x_{0:}|y_{1:t})$ which is easier to sample from. The choice of the importance distribution will be discussed later in this section. By substituting the importance density in (4.13), we have

$$\begin{aligned}
\mathrm{E}\left[h_\theta(x_{0:t})\right] &= \int h_\theta(x_{0:t})\ \frac{p_\theta(x_{0:t}|y_{1:t})}{q_\theta(x_{0:t}|y_{1:t})}\ q_\theta(x_{0:t}|y_{1:t})\ dx_{0:t} \\
&= \int h_\theta(x_{0:t})\ \frac{p_\theta(y_{1:t}|x_{1:t})\ p_\theta(x_{0:t})}{p_\theta(y_{1:t})\ q_\theta(x_{0:t}|y_{1:t})} q_\theta(x_{0:t}|y_{1:t})\ dx_{0:t} \\
&= \int h_\theta(x_{0:t})\ \frac{\tilde{w}_t}{p_\theta(y_{1:t})} q_\theta(x_{0:t}|y_{1:t})\ dx_{0:t},
\end{aligned} \tag{4.14}$$

where $\tilde{w}_t$ is called the *unnormalised importance weight* and given by

$$\tilde{w}_t = \frac{p_\theta(y_{1:t}|x_{1:t})\; p_\theta(x_{0:t})}{q_\theta(x_{0:t}|y_{1:t})}.$$

Then, (4.14) is written as

$$\begin{aligned}
\mathrm{E}\left[h_\theta(x_{0:t})\right] &= \frac{1}{p_\theta(y_{1:t})} \int h_\theta(x_{0:t})\; \tilde{w}_t\; q_\theta(x_{0:t}|y_{1:t})\; dx_{0:t} \\
&= \frac{\int h_\theta(x_{0:t})\; \tilde{w}_t\; q_\theta(x_{0:t}|y_{1:t})\; dx_{0:t}}{\int p_\theta(y_{1:t}|x_{1:t})\; p_\theta(x_{0:t})\; \frac{q_\theta(x_{0:t}|y_{1:t})}{q_\theta(x_{0:t}|y_{1:t})}\; dx_{0:t}} \\
&= \frac{\int h_\theta\;(x_{0:t}) \tilde{w}_t\; q_\theta(x_{0:t}|y_{1:t})\; dx_{0:t}}{\int \tilde{w}_t\; q_\theta(x_{0:t}|y_{1:t})\; dx_{0:t}} \\
&= \frac{\mathrm{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[\tilde{w}_t\; h_\theta(x_{0:t})\right]}{\mathrm{E}_{q_\theta(x_{0:t}|y_{1:t})}\left[\tilde{w}_t\right]}.
\end{aligned} \tag{4.15}$$

Thus, by drawing N random samples, so-called *particles*, from $q_\theta(.|.)$, we can approximate the expected value of interest as follows

$$\begin{aligned}
\mathrm{E}\left[h_\theta(x_{0:t})\right] &\approx \frac{\frac{1}{N}\sum_{i=1}^{N} \tilde{w}_t^{(i)}\; h_\theta(x_{0:t}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} \tilde{w}_t^{(i)}} \\
&\approx \sum w_t^{(i)}\; h_\theta(x_{0:t}^{(i)}),
\end{aligned} \tag{4.16}$$

where the *normalised importance weights* are given by

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}},$$

such that

$$\sum_{i=1}^{N} w_t^{(i)} = 1.$$

The approximation given by (4.16) holds for every well-behaved function h. Therefore, for the sample $x^{(1)},\ldots,x^{(N)}$, with the associated weights $w^{(1)},\ldots,w^{(N)}$, the target distri-

bution $p_\theta(X)$ can be approximated by

$$\hat{p}_\theta(x) \approx \sum_{i=1}^{N} w^{(i)} \delta_{X^{(i)}}(x), \tag{4.17}$$

where $\delta_X(x)$ denotes the Dirac delta mass located at X.

However, in the filtering problem, the target distribution changes every time as a new observation arrives, i.e. we move from $p_\theta(x_{0:t-1}|y_{1:t-1})$ to $p_\theta(x_{0:t}|y_{1:t})$. Note that this does not imply that $p_\theta(x_{0:t-1}|y_{1:t-1})$ is a marginal density of $p_\theta(x_{0:t}|y_{1:t})$, although $x_{0:t-1}$ are the first components of $x_{0:t}$. To efficiently update a discrete approximation of $p_\theta(x_{0:t-1}|y_{1:t-1})$ when a new observation becomes available and therefore, to get a discrete approximation of $p_\theta(x_{0:t}|y_{1:t})$,

1. draw $x_t^{(i)}$, for each sample path, $x_{0:t-1}^{(i)}$, in the support of $\hat{p}_\theta(x_{0:t-1}|y_{1:t-1})$ to get $x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)})$ and then

2. update its weight $w_{t-1}^{(i)}$ to an appropriate weight $w_t^{(i)}$.

The weighted points, $(x_{0:t}^{(i)}, w_t^{(i)})$, provide a new discrete approximation $\hat{p}_\theta(x_{0:t}|y_{1:t})$. If we express the proposal distribution, $q_\theta(x_{0:t}|y_{1:t})$, as

$$q_\theta(x_{0:t}|y_{1:t}) = q_\theta(x_t|x_{0:t-1}, y_{1:t}) q_\theta(x_{0:t-1}|y_{1:t-1}), \tag{4.18}$$

thus we only need to sample from the *importance transition density*, $q_\theta(x_t|x_{0:t-1}, y_{1:t})$, at each time point t and update the importance density sequentially using (4.18) (Petris et al., 2009, Chapt. 5). This decomposition also allows us to sequentially update the weight as we move from $t-1$ to t. Therefore, the ith weight at time t is given by

$$\begin{aligned} w_t^{(i)} &\propto \frac{p_\theta(x_{0:t}^{(i)}|y_{1:t})}{q_\theta(x_{0:t}^{(i)}|y_{1:t})} \\ &\propto \frac{p_\theta(x_{0:t}^{(i)}, y_t|y_{1:t-1})}{q_\theta(x_{0:t}^{(i)}|y_{1:t})} \end{aligned}$$

$$\propto \frac{p_\theta(x_t^{(i)}, y_t | x_{0:t-1}^{(i)}, y_{1:t-1})\; p_\theta(x_{0:t-1}^{(i)} | y_{1:t})}{q_\theta(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})\; q_\theta(x_{0:t-1}^{(i)} | y_{1:t-1})}$$

$$\propto \frac{g_\theta(y_t | x_t^{(i)})\; f_\theta(x_t^{(i)} | x_{t-1}^{(i)})}{q_\theta(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})}\; w_{t-1}^{(i)}. \tag{4.19}$$

Therefore, for every i, after sampling from $q_\theta(x_t | x_{0:t-1}^{(i)}, y_{1:t})$, the unnormalised weights can be obtained as follows

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{g_\theta(y_t | x_t^{(i)})\; f_\theta(x_t^{(i)} | x_{t-1}^{(i)})}{q_\theta(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})}. \tag{4.20}$$

Then we scale the unnormalised weights:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}.$$

Earlier in Section 4.2.1, we mentioned that our interest in this section lies in estimating $p_\theta(x_t | y_{1:t})$ which was defined as the marginal distribution of $p_\theta(x_{0:t} | y_{1:t})$. Similarly, $p_\theta(x_t | y_{1:t})$ can be approximated by discarding the first t components of each path $x_{0:t}^{(i)}$ in $\hat{p}_\theta(x_{0:t} | y_{1:t})$, which leaves us only $x_t^{(i)}$, to obtain

$$p_\theta(x_t | y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(x).$$

4.2.2.1 Degeneracy and resampling

In practice, it is often the case that after a number of updates, a few points have relatively high weights in the support of $\hat{p}_\theta$, while the remaining have very small weights. This leads to a poor Monte Carlo approximation. This problem is known as *weight degeneracy* and having a *resampling* step may prevent such a problem by eliminating unpromising samples (Andrieu et al., 2010). Resampling might be done in several ways. The most popular algorithms in the literature are , residual, systematic and multinomial resampling. The first two algorithms are more efficient algorithms when our attempt is to keep the variance increase as small as possible. The third algorithm is the simplest algorithm but

still efficient to tackle weight degeneracy. Here we adopt the multinomial resampling scheme which can be described as follows:

(i) Draw a random sample of size N from $\hat{p}_\theta(x_{0:t}|y_{1:t})$;

(ii) Assign each sampled points equal weights and use them as the new discrete approximation of the target distribution.

While the expected value of the approximating distribution $\hat{p}_\theta(x_{0:t}|y_{1:t})$ stays the same, this technique may increase the Monte Carlo variance (Doucet and Johansen, 2009; Petris et al., 2009).

The cost of resampling at each time point t is that we introduce some additional variance (Doucet and Johansen, 2009). If particles have unnormalised weights with a small variance then the resampling might not be necessary. Therefore, in practice, it is useful to measure how different the proposal distribution is from the target distribution (Liu, 2001, pp. 35-36). This is often assessed by looking at the *effective sample size*, N_{eff}, criterion (Liu, 2001), which at time t is given by

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_t^{(i)}\right)^2}. \tag{4.21}$$

In a simple importance sampling setting N_{eff} can be interpreted as the inference based on the N weighted samples is approximately equivalent to inference based on N_{eff} perfect samples from the target distribution. Therefore, N_{eff} can change between N (when all the weights are equal) and one (when one weight is equal to one) and we only resample when N_{eff} falls bellow the pre-specified threshold N_0, typically $N_0 = N/2$; see Doucet and Johansen (2009) and references therein for further details.

4.2.2.2 Designs and issues

Importance sampling requires us to specify $q_\theta(x_t|x_{0:t-1}, y_{1:t})$. Petris et al. (2009) has highlighted guidelines on how best to select the importance density. A common choice is $q_\theta(x_t|x_{0:t-1}, y_{1:t}) = p_\theta(x_t|x_{t-1})$, that is sampling from the prior distributions of the states without considering any information from the observations. Although this leads

Algorithm 4.1 Basic particle filter algorithm

1. At time $t = 0$,
   1.1 Sample $(x_0^{(1)}, \ldots, x_0^{(N)}) \sim q_\theta(x_0)$ and set $(w_1^{(1)}, \ldots, w_1^{(N)}) = \frac{1}{N}$.
2. For $t = 1, \ldots, n$,
   2.1 For $i = 1, \ldots, N$,
      2.1.1 Sample $x_t^{(i)} \sim q_\theta(x_t | x_{0:t-1}, y_{1:t})$ and set

$$x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)});$$

      2.1.2 Obtain the unnormalised weights

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{g_\theta(y_t | x_t^{(i)})\, f_\theta(x_t^{(i)} | x_{t-1}^{(i)})}{q_\theta(x_t | x_{0:t-1}^{(i)}, y_{1:t})};$$

   2.2 Normalise the weights

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{k=1}^N \tilde{w}_t^{(k)}};$$

   2.3 Obtain

$$N_{eff} = \frac{1}{\sum_{i=1}^N \left(w_t^{(i)} \right)^2};$$

   2.4 If $N_{eff} < N_0$, resample:

      2.4.1 Draw a sample of size N from the discrete distribution

$$p(x_{0:t} = x_{0:t}^{(i)}) = w_t^{(i)},$$

      for $i = 1, \ldots, N$, and relabel this sample

$$x_{0:t}^{(1)}, \ldots, x_{0:t}^{(N)};$$

      2.4.2 Reset the weights: $w_t^{(i)} = \frac{1}{N}$, $i = 1, \ldots, N$;

   2.5 Set $\hat{p}_\theta(x_{0:t} | y_{1:t}) = \sum_{i=1}^{\mathrm{N}} w_t^{(i)} \delta_{X_{0:t}^{(i)}}(x)$.

to a straightforward calculation, it is often the case that particles fall in regions with low posterior density that result in poor approximations. To avoid such a problem, it is suggested to sample x_t from its condition distribution given x_{t-1} and $y_{1:t}$. This distribution

is referred to as the *optimal kernel density*.

Considering the strengths of SMC methods, they also suffer from well-known drawbacks. Indeed, when there is an outlier, the weights w_i will be extremely unevenly distributed and so it will require an extremely large value of N for draws to be close to samples from the empirical filtering density. This is of particular concern if the measurement density $g_\theta(y_t|x_t)$ is highly sensitive to x_t (Pitt and Shephard, 1999).

4.2.3 Improvement and extension

Over the past 15 years many techniques have been suggested to improve the performance of particle filters. These techniques are essentially categorised as

(a) techniques suggested to reduce the variance by introducing a resampling step such as the residual (Liu, 2001, Chapt. 3) or stratified (Kitagawa, 1996) resampling procedure, and

(b) techniques that aim to tackle the weight degeneracy problem (defined in Section 4.2.2.1) which include, among others, the auxiliary particle filters (APF) suggested by Pitt and Shephard (1999) and the resampling-move algorithm (Gilks and Berzuini, 2001).

Particle filters that learn about the static model parameters in a sequential manner have attracted much attenction in recent years. Algorithms for this purpose may assume the static parameters as a part of an unobserved process (Berzuini et al., 1997), or combine MCMC within the particle filter algorithms (Andrieu et al., 1999; Fearnhead, 2002; Storvik, 2002). Alternatively, they add random noise to the particles, therefore in this context, static parameters are approximated by some slowly changing dynamic parameters (West, 1993b; Liu and West, 2001), also see Lopes and Tsay (2011), Andrieu et al. (2010) and references therein for recent reviews on improvement and extension of particle filters.

We discuss the APF algorithm as an improvement to the basic particle filter algorithm. The performance of the PF depends on the specification of importance densities. However, devising an effective importance density is a very hard problem and an inappropriate

choice can lead to severe degeneracy of the weights. The APF algorithm helps us to overcome this difficulty. Then, we explain the algorithm suggested by Liu and West (2001) for estimating unknown static parameters in the model. In the literature, this algorithm is called the *LWF* algorithm and considered as an extension of the APF algorithm (Lopes and Tsay, 2011).

4.2.3.1 Auxiliary particle filters

Consider the approximation to $p_\theta(x_{0:t-1}|y_{1:t-1})$ at time $t-1$ given by

$$\hat{p}_\theta(x_{0:t-1}|y_{1:t-1}) = \sum_{i=1}^{\mathrm{N}} w_{t-1}^{(i)} \delta_{X_{0:t-1}^{(i)}}(x).$$

Our interest lies in updating $p_\theta(x_{0:t-1}|y_{1:t-1})$ to $p_\theta(x_{0:t}|y_{1:t})$ when a new data point arrives. We have

$$\begin{aligned}
p_\theta(x_{0:t}|y_{1:t}) \propto & p_\theta(x_{0:t}, y_t|y_{1:t-1}) \\
= & p_\theta(y_t|x_{0:t}, y_{1:t-1})\ p_\theta(x_t|x_{0:t-1}, y_{1:t})\ p_\theta(x_{0:t-1}|y_{1:t-1}) \\
= & g_\theta(y_t|x_t)\ f_\theta(x_t|x_{t-1})\ p_\theta(x_{0:t-1}|y_{1:t-1}) \\
\approx & g_\theta(y_t|x_t)\ f_\theta(x_t|x_{t-1})\ \hat{p}_\theta(x_{0:t-1}|y_{1:t-1}) \\
= & \sum_{i=1}^{\mathrm{N}} w_{t-1}^{(i)}\ g_\theta(y_t|x_t)\ f_\theta(x_t|x_{t-1}^{(i)})\ \delta_{X_{0:t-1}^{(i)}}(x), \qquad (4.22)
\end{aligned}$$

where (4.22) set to be our target distribution for an importance sampling step. To eliminate the summation in (4.22), $p_\theta(x_{0:t}|y_{1:t})$ can be written as a joint distribution of $x_{0:t}$ and the ith index,

$$p_\theta(x_{0:t}, I = i|y_{1:t}) \propto w_{t-1}^{(i)}\ g_\theta(y_t|x_t)\ f_\theta(x_t|x_{t-1}^{(i)})\ \delta_{x_{0:t-1}^{(i)}}(x), \qquad (4.23)$$

where $I \in \{1, 2, \ldots, N\}$ is set to be a latent variable such that $p(I = i) = w_{t-1}^{(i)}$. The importance density suggested by Pitt and Shephard (1999) for the auxiliary target distribution

in (4.23) is

$$q_\theta(x_{0:t}, I = i|y_{1:t}) \propto w_{t-1}^{(i)}\, g_\theta(y_t|\hat{x}_t^{(i)})\, f_\theta(x_t|x_{t-1}^{(i)})\, \delta_{x_{0:t-1}^{(i)}}(x), \tag{4.24}$$

where $\hat{x}_t^{(i)}$ is a central value, such as the mean or the mode, a draw of $f_\theta(x_t|X_{t-1} = x_{t-1}^{(i)})$. A sample from $q_\theta(x_{0:t}, i|y_{1:t})$ can be generated by implementing the following steps for $j = 1, \ldots, N$

(i) Draw an auxiliary variable I_j with

$$p_\theta(I_j = i) \propto w_{t-1}^{(i)}\, g_\theta(y_t|\hat{x}_t^{(i)}), \qquad i = 1, \ldots, N.$$

(ii) Given $I_j = i$, draw

$$x_t^{(j)} \sim f_\theta(x_t|x_{t-1}^{(i)})$$

and set $x_{0:t}^{(j)} = (x_{0:t-1}^{(i)}, x_t^{(j)})$.

The importance weight of the jth draw using (4.20), (4.23) and (4.24), is proportional to

$$\begin{aligned}\tilde{w}_t^{(j)} =& \frac{w_{t-1}^{(\mathrm{I}_j)}\, g_\theta(y_t|x_t^{(j)})\, f_\theta(x_t^{(j)}|x_{t-1}^{(j)})}{w_{t-1}^{(\mathrm{I}_j)}\, g_\theta(y_t|\hat{x}_t^{(\mathrm{I}_j)})\, f_\theta(x_t^{(j)}|x_{t-1}^{(j)})} \\ =& \frac{g_\theta(y_t|x_t^{(j)})}{g_\theta(y_t|\hat{x}^{(\mathrm{I}_j)})}.\end{aligned}$$

Similar to the basic particle filter algorithm in Algorithm 4.1, a resampling is commonly applied when the effective sample size falls below a threshold. The APF algorithm is summarised bellow Algorithm 4.2.

4.2.3.2 Auxiliary particles filters with unknown static parameters

In real applications when θ is unknown in the model, the target distribution is replaced by (4.12). To approximate such distribution, one might extend the state vector to include θ, defining an imaginary state $\theta_t = \theta_{t-1} = \theta$, and apply either of the PF algorithms in Sections 4.2.2 and 4.2.3.1. However, such method has a serious drawback. Since there is

Algorithm 4.2 Auxiliary particle filter algorithm

1. At time $t = 0$, sample $x_0^{(i)} \sim q_\theta(x_0)$ and set $w_0^{(i)} = \frac{1}{N}$, for $i = 1, \ldots, N$.

2. For $t = 1, \ldots, n$:

   2.1 for $j = 1, \ldots, N$:

   2.1.1 Draw I_j, with probability $p(I_j = i) \sim w_{t-1}^{(i)} g_\theta(y_t | \hat{x}_t^{(i)})$, for $i = 1, 2, \ldots, N$;

   2.1.2 Sample new particle $x_t^{(j)}$ from $p_\theta(x_t | x_{t-1} = x_{t-1}^{(\mathrm{I}_j)})$ and set

   $$x_{0:t}^{j} = (x_{0:t-1}^{(\mathrm{I}_j)}, x_t^{(j)});$$

   2.1.3 Assign each particle $x_t^{(j)}$ the corresponding importance weight

   $$\tilde{w}_t^{(j)} = \frac{g_\theta(y_t | X_t = x_t^{(j)})}{g_\theta(y_t | X_t = \hat{x}_t^{(\mathrm{I}_j)})};$$

   2.2 Normalise the weights

   $$w_t^{(j)} = \frac{\tilde{w}_t^{(j)}}{\sum_{k=1}^{N} \tilde{w}_t^{(k)}};$$

   2.3 Obtain

   $$N_{eff} = \frac{1}{\sum_{k=1}^{N} \left(w_t^{(k)} \right)^2};$$

   2.4 If $N_{eff} < N_0$, resample:

   2.4.1 Draw a sample of size N from the discrete distribution

   $$p(x_{0:t} = x_{0:t}^{(i)}) = w_t^{(i)},$$

   for $i = 1, \ldots, N$, and relabel this sample

   $$x_{0:t}^{(1)}, \ldots, x_{0:t}^{(N)};$$

   2.4.2 Reset the weights: $w_t^{(i)} = \frac{1}{N}$, $i = 1, \ldots, N$;

   2.5 Set $\hat{p}_\theta(x_{0:t} | y_{1:t}) = \sum_{i=1}^{\mathrm{N}} w_t^{(i)} \delta_{X_{0:t}^{(i)}}(x)$.

---

no evolution assigned to this imaginary state, draws at time $t > 0$, $\theta_t^{(i)}$ are same as draws at time $t = 0$, $\theta_0^{(i)}$, for $i = 1, \ldots, N$. That is, samples from the prior distribution are not representative of the posterior distribution. Therefore, though weights are updated as the new information arrives, if the values of $\theta_t^{(i)}$'s do not represent the marginal target dis-

tribution, $p(\theta|y_{1:t})$ (e.g. all happen in the tails of $p(\theta|y_{1:t})$), the approximation obtained from the algorithm is poor (Petris et al., 2009). To avoid such problem, it is suggested to refresh the sampled values of θ by discarding the current values of θ at each time the target distribution changes and generating new ones. The LWF algorithm suggested by Liu and West (2001) serves this purpose and is a combination of (i) the APF algorithms, (ii) a kernel smoothing algorithm approximation to $p(\theta|y_{1:t})$, via a mixture of multivariate normals, and (iii) a shrinkage idea to adjust the increasing variation of θ (Lopes and Tsay, 2011; West, 1993b,a). In their algorithm, the analysis only relies on the values of θ sampled from the importance density at time t and one can forget about the values of θ used at time $t-1$. Consider the approximation of $p(x_{0:t-1}, \theta|y_{1:t-1})$ at time $t-1$ given by

$$\hat{p}(x_{0:t-1}, \theta|y_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)} \, \delta_{(x_{0:t-1}^{(i)}, \theta^{(i)})}(x, \theta). \tag{4.25}$$

The distribution of θ is obtained as the marginal distribution of $p(x_{0:t-1}, \theta|y_{1:t-1})$ as follows

$$\hat{p}(\theta|y_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)} \, \delta_{\theta^{(i)}}(\theta). \tag{4.26}$$

Liu and West (2001) replaced each point mass $\delta_{\theta^{(i)}}(\theta)$ with a normal distribution, thus the discrete approximation of $p(\theta|y_{1:t-1})$ becomes a continuous distribution. We may set the normal distribution centred at $\theta^{(i)}$. However, as we discuss it below, this may increase the variance of the approximating distribution. Let $\bar{\theta}$ and Σ_θ be the mean vector and the variance matrix of θ under $\hat{p}(\theta|y_{1:t-1})$, and let

$$\tilde{p}(\theta) = \sum_{i=1}^{N} w_{t-1} \, \mathrm{N}(\theta^{(i)}, \psi_\theta). \tag{4.27}$$

Let us introduce I as a latent classification variable which is an index in the mixture in (4.27). We have

$$\mathrm{E}(\theta) = \mathrm{E}\left[\mathrm{E}(\theta|I)\right] = \mathrm{E}(\theta^{(\mathrm{I})}) = \sum_{i=1}^{N} w_{t-1}^{(i)}\, \theta^{(i)} = \bar{\theta},$$

and the variance is

$$\mathrm{Var}(\theta) = \mathrm{E}\left[\mathrm{Var}(\theta|I)\right] + \mathrm{Var}\left[\mathrm{E}(\theta|I)\right] = \mathrm{E}(\psi_\theta) + \mathrm{Var}(\theta^{(\mathrm{I})}) = \psi_\theta + \Sigma_\theta > \Sigma_\theta.$$

Therefore, having a variance larger than Σ_θ results in a kernel density function that is over-dispersed relative to the posterior samples. This results in an over-dispersed approximation of $p(\theta|y_{1:t})$, and therefore *loss of information* will accumulate as the operation is repeated over the time (Liu and West, 2001). West (1993a,b) addressed this problem by introducing the idea of shrinkage kernel locations. By changing the definition of $\tilde{p}(\theta|Y_{1:t})$ in (4.27) to

$$\tilde{p}(\theta) = \sum_{i=1}^{N} w_{t-1}^{(i)}\, \mathrm{N}\left(m^{(i)}, \left[1-a^2\right]\Sigma_\theta\right), \tag{4.28}$$

where $m^{(i)} = a\theta_{t-1}^{(i)} + (1-a)\bar{\theta}$ for some a in $(0,1)$, thus

$$\mathrm{E}(\theta) = \mathrm{E}\left[\mathrm{E}(\theta|I)\right] = \mathrm{E}\left[a\theta_{t-1}^{(\mathrm{I})} + (1-a)\bar{\theta}\right] = a\bar{\theta} + (1-a)\bar{\theta} = \bar{\theta},$$

and the variance is

$$\begin{aligned}
\mathrm{Var}(\theta) =& \mathrm{E}\left[\mathrm{Var}(\theta|I)\right] + \mathrm{Var}\left[\mathrm{E}(\theta|I)\right] \\
=& \mathrm{E}\left[\psi_\theta\right] + \mathrm{Var}\left[m^{(\mathrm{I})}\right] = \mathrm{E}\left[\left(1-a^2\right)\Sigma_\theta\right] + \mathrm{Var}\left[a\theta^{(\mathrm{I})} + (1-a)\bar{\theta}\right] \\
=& \left(1-a^2\right)\Sigma_\theta + a^2\mathrm{Var}(\theta^{(\mathrm{I})}) = \left(1-a^2\right)\Sigma_\theta + a^2\Sigma_\theta = \Sigma_\theta.
\end{aligned}$$

Therefore, θ has the same mean and variance under $\hat{p}(\theta|y_{1:t})$ and $\tilde{p}(\theta|y_{1:t})$. The choice of the *shrinkage parameter*, a, has considerable effect on the performance of the LWF algorithm, because it drives both the shrinkage and the smoothness of the normal approximation (Lopes and Tsay, 2011; Petris et al., 2009). In practice Liu and West (2001) suggested to set $a = (3\delta - 1)/(2\delta)$ for a discount factor $\delta \in (0.95, 0.99)$, which results in an $a \in (0.974, 0.995)$.

The approximation of the joint distribution in (4.25) can be replaced by $\tilde{p}(x_{0:t-1}, \theta | y_{1:t-1})$ given by

$$\tilde{p}(x_{0:t-1}, \theta | y_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)} \, \mathrm{N}\left(m^{(i)}, \left[1-a^2\right]\Sigma\right) \, \delta_{x_{0:t-1}^{(i)}}(x). \tag{4.29}$$

Hence, the target distribution $p(x_{0:t}, \theta | y_{1:t})$ can be approximated as follows

$$\begin{aligned}
p(x_{0:t}, \theta | y_{1:t}) \propto & p(x_{0:t}, \theta, y_t | y_{0:t-1}) \\
= & p(y_t | x_{0:t}, \theta, y_{1:t-1}) \; p(x_t | x_{0:t-1}, \theta, y_{1:t-1}) \; p(x_{0:t-1}, \theta | y_{1:t-1}) \\
= & g(y_t | x_t, \theta) \; f(x_t | x_{t-1}, \theta) \; p(x_{0:t-1}, \theta | y_{1:t-1}) \\
\approx & g(y_t | x_t, \theta) \; f(x_t | x_{t-1}, \theta) \; \tilde{p}(x_{0:t-1}, \theta | y_{1:t-1}) \\
= & \sum_{i=1}^{N} w_{t-1}^{(i)} \; g(y_t | x_t, \theta) \; f(x_t | x_{t-1}^{(i)}, \theta) \, \mathrm{N}\left(m^{(i)}, \left[1-a^2\right]\Sigma\right) \, \delta_{x_{0:t-1}^{(i)}}(x).
\end{aligned}$$

So that the auxiliary target distribution in (4.23) at time t can be altered as

$$p(x_{0:t}, \theta, i | y_{1:t}) \propto w_{t-1}^{(i)} \; g(y_t | x_t, \theta) \; f(x_t | x_{t-1}^{(i)}, \theta) \, \mathrm{N}\left(m^{(i)}, \left[1-a^2\right]\Sigma\right) \, \delta_{x_{0:t-1}^{(i)}}(x). \tag{4.30}$$

Liu and West (2001) suggested an appropriate choice for the importance distribution in the following form

$$\begin{aligned}
q(x_{0:t}, \theta, i | y_{1:t}) \propto & w_{t-1}^{(i)} \; g\left(y_t | x_t = \hat{x}_t^{(i)}, \theta = m^{(i)}\right) \; f(x_t | x_{t-1}^{(i)}, \theta) \\
& \mathrm{N}\left(\theta; m^{(i)}, \left[1-a^2\right]\Sigma\right) \; \delta_{x_{0:t-1}^{(i)}}(x).
\end{aligned} \tag{4.31}$$

where $\hat{x}_t^{(i)}$ is a central value, such as the mean or the mode of $p(x_t | x_t = \hat{x}_t^{(i)}, \theta = m^{(i)})$. At time t, a sample of size N from the auxiliary target distribution in (4.30) can be obtained by repeating following steps for $j = 1, \ldots, N$.

(i) Draw variable I_j with

$$p(I_j = i) \propto w_{t-1}^{(i)} \; g(y_t | x_t = \hat{x}_t^{(i)}, \theta = m^{(i)}), \qquad i = 1, \ldots, N.$$

(ii) Given $I_j = i$, sample $\theta^{(j)} \sim \mathrm{N}\left(m^{(i)}, \left[1-a^2\right]\Sigma_\theta\right)$ and set $\theta = \theta^{(j)}$.

(iii) Given $I_j = i$ and $\theta = \theta^{(j)}$ obtain

$$x_t^{(j)} \sim f(x_t | x_{t-1} = x_{t-1}^{(i)}, \theta = \theta^{(j)})$$

and set $x_{0:t}^{(j)} = \left(x_{0:t-1}^{(i)}, x_t^{(j)}\right)$.

Thus, the unnormalised importance weight using (4.20), (4.30) and (4.31)is proportional to

$$\begin{aligned}\tilde{w}_t^{(i)} =& \frac{w_{t-1}^{(\mathrm{I}_j)}\, g(y_t | x_t = x_t^{(j)}, \theta = \theta^{(j)})\, f(x_t^{(j)} | x_{t-1}^{(j)}, \theta^{(j)})\, \mathrm{N}(m^{(\mathrm{I}_j)}, (1-a^2)\Sigma_\theta)}{w_{t-1}^{(\mathrm{I}_j)}\, g(y_t | x_t = \hat{x}_t^{(\mathrm{I}_j)}, \theta = m^{(\mathrm{I}_j)})\, f(x_t^{(j)} | x_{t-1}^{(j)}, \theta^{(j)})\, \mathrm{N}(m^{(\mathrm{I}_j)}, (1-a^2)\Sigma_\theta)} \\ =& \frac{g(y_t | x_t = x_t^{(j)}, \theta = \theta^{(j)})}{g(y_t | x_t = \hat{x}_t^{(\mathrm{I}_j)}, \theta = m^{(\mathrm{I}_j)})}.\end{aligned}$$

After normalising the importance weights, the approximation $p(\theta, x_{0:t} | y_{1:t})$ can be obtained as follows

$$\hat{p}(x_{0:t}, \theta | y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{(x_{0:t}^{(i)}, \theta^{(i)})}(x, \theta).$$

Similar to the general particle filters and APF algorithms, in Sections 4.2.2 and 4.2.3.1, a resampling takes place if the number effective samples falls below the threshold. Let us point out that in order to use the kernel normal methods of Liu and West (2001), it is appropriate to work with real-valued parameters. Otherwise, if the parameters of interest do not have real line domain, we transform them so that their transformed distributions have the entire real line support. For example, in the LWF algorithm, variances are expressed in terms of their log, or probabilities in terms of their logit.

4.3 A dynamic model

In the following sections we introduce a dynamic Poisson difference (DPD) model and a dynamic zero inflated Poisson difference (DZPD) model for modelling index change. We investigate model diagnostics and the predictive distribution for one step ahead pre-

Algorithm 4.3 Liu and West's (LW) filter algorithm

1. At time $t = 0$, sample $(x_0^{(1)}, \theta^{(1)}), \ldots, (x_0^{(\mathrm{N})}, \theta^{(\mathrm{N})})$ from $q(x_0)p(\theta)$ independently and set $w_0^{(i)} = \frac{1}{N}$.

2. For $t = 1, \ldots, n$:

   2.1 For $i = 1, \ldots, N$, calculate $\bar{\theta} = \mathrm{E}_{\hat{p}_{t-1}}(\theta)$ and $\Sigma_\theta = \mathrm{Var}_{\hat{p}_{t-1}}(\theta)$ and

   $$m^{(i)} = a\theta^{(i)} + (1-a)\bar{\theta},$$

   $$\hat{x}_t^{(i)} = \mathrm{E}(x_t | x_{t-1} = x_{t-1}^{(i)}, \theta = m^{(i)});$$

   2.2 for $j = 1, \ldots, N$:

   2.5.1 Draw I_j, with probability $p(I_j = i) \sim w_{t-1}^{(i)} g(y_t | \hat{x}_t^{(i)}, \theta = m^{(i)})$;

   2.5.2 Sample $\theta^{(j)} \sim \mathrm{N}\left(m^{(\mathrm{I}_j)}, \left[1 - a^2\right]\Sigma\right)$;

   2.5.3 Sample the new particle $x_t^{(j)}$ from $p(x_t | x_{t-1} = x_{t-1}^{(\mathrm{I}_j)}, \theta = \theta^{(j)})$ and set

   $$x_{0:t}^{(j)} = (x_{0:t-1}^{(i)}, x_t^{(j)});$$

   2.5.4 Assign each particle $x_t^{(j)}$ the corresponding importance weight

   $$\tilde{w}_t^{(j)} = \frac{g(y_t | X_t = x_t^{(j)}, \theta = \theta^{(j)})}{g(y_t | X_t = \hat{x}_t^{(\mathrm{I}_j)}, \theta = m^{(\mathrm{I}_j)})};$$

   2.3 Normalise the weights

   $$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{k=1}^N \tilde{w}_t^{(k)}};$$

   2.4 Obtain

   $$N_{eff} = \frac{1}{\sum_{i=1}^N \left(w_t^{(i)}\right)^2};$$

   2.5 If $N_{eff} < N_0$, resample:

   2.2.1 Draw a sample of size N from the discrete distribution

   $$p\left[(x_{0:t}, \theta) = (x_{0:t}^{(i)}, \theta^{(i)})\right] = w_t^{(i)}, \quad \text{for} i = 1, \ldots, N$$

   for $i = 1, \ldots, N$, and relabel this sample

   $$(x_{0:t}^{(1)}, \theta^{(1)}), \ldots, (x_{0:t}^{(N)}, \theta^{(N)});$$

   2.2.2 Reset the weights: $w_t^{(i)} = \frac{1}{N}$, $i = 1, \ldots, N$;

   2.6 Set $\hat{p}(x_{0:t} | y_{1:t}, \theta) = \sum_{i=1}^{\mathrm{N}} w_t^{(i)} \delta_{(x_{0:t}^{(i)}, \theta^{(i)})}(x, \theta)$.

dictions.

4.3.1 Dynamic Poisson difference model

Let $\{Z_t\}_{t=1}^{n}$ denotes the observation process. A DPD model may be defined as follows

Observation equation

$$Z_t|(\mathbf{X}_t = \mathbf{x}_t) \sim \text{PD}(\lambda_{1,t}, \lambda_{2,t}), \tag{4.32}$$

state equation

$$\log(\boldsymbol{\lambda}_t) = \mathbf{x}_t, \tag{4.33}$$

$$\mathbf{x}_t = \boldsymbol{\theta}\mathbf{x}_{t-1} + \boldsymbol{\nu}_t, \tag{4.34}$$

where

$$\boldsymbol{\lambda}_t = \begin{pmatrix} \lambda_{1,t} \\ \lambda_{2,t} \end{pmatrix}, \quad \mathbf{x}_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix}, \quad \boldsymbol{\theta} = \begin{pmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\nu}_t = \begin{pmatrix} \nu_{1,t} \\ \nu_{2,t} \end{pmatrix} \sim \text{N}(0, \Sigma),$$

such that

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}.$$

Let us denote the vector of parameter by $\psi_{\text{DPD}} = (\theta_1, \theta_2, \sigma_1^2, \sigma_2^2)$, and set $\mathbf{x}_0 \sim \text{N}(0,\ I)$, where I is an identity matrix of appropriate size, also $f_{\psi_{\text{DPD}}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \text{N}(\boldsymbol{\theta}\mathbf{x}_{t-1},\ \Sigma)$ and $g_{\psi_{\text{DPD}}}(z_t|\mathbf{x}_t) = f_{\text{PD}}(\lambda_{1,t},\ \lambda_{2,t})$.

4.3.2 Dynamic zero inflated Poisson difference model

A DZPD model can be defined by adding the additional parameter "p" to the DPD model.

Similar to the DPD model, consider the observation process, $\{Z_t\}_{t=1}^{n}$. A DZPD model may be defined as follows

Observation equations

$$Z_t|(\mathbf{X}_t = \mathbf{x}_t) \sim \text{ZPD}(\lambda_{1,t},\ \lambda_{2,t},\ p), \tag{4.35}$$

state equations

$$\log(\boldsymbol{\lambda}_t) = \mathbf{x}_t + h(p), \tag{4.36}$$

$$\mathbf{x}_t = \boldsymbol{\theta}\mathbf{x}_{t-1} + \boldsymbol{\nu}_t, \tag{4.37}$$

where $h(p)$ is any function of p, for $0 < p < 1$. Considering that $\psi_{\text{DZPD}} = (\theta_1, \theta_2, \sigma_1^2, \sigma_2^2, p)$ stands for the vector of model parameters, we may assign $\mathbf{x}_0 \sim \text{N}(0,\ I)$, where I is an identity matrix of appropriate size, $f_{\psi_{\text{DZPD}}}(\mathbf{x}_t|\mathbf{x}_{t-1}) = \text{N}(\boldsymbol{\theta}\mathbf{x}_{t-1},\ \Sigma)$ and $g_{\psi_{\text{DZPD}}}(z_t|\mathbf{x}_t) = f_{\text{ZPD}}(\lambda_{1,t},\ \lambda_{2,t}\ p)$.

We might be able to consider p as another observed process affected by an unobserved process in the model. However, in practice p is less likely to change within one trading day. Thus, we consider p as a parameter in the model by having $h(p)$. One possible choice could be $h(p) = -\log(1-p)$ which can be explained by definition of the mean given by

$$\text{E}(z) = (1-p)(\lambda_1 - \lambda_2) = (1-p)\lambda_1 - (1-p)\lambda_2.$$

We then can model $\log[(1-p)\lambda_1]$ and $\log[(1-p)\lambda_2]$ giving $h(p) = -\log(1-p)$ in (4.36).

4.3.3 Predictive distribution

At time point t, we are interested in predicting z_{t+1}, denoted as z^*_{t+1}. According to Triantafyllopoulos (2009), the one-step ahead predictive distribution,

$$p(z_{t+1}|z_{1:t}) = \int g(z_{t+1}|x_{t+1})p(x_{t+1}|z_{1:t})dx_{t+1},$$

can be approximated by

$$\hat{p}(z^*_{t+1}|z_{1:t}) = \sum_{i=1}^{\mathrm{N}} g(z^*_{t+1}|x^{(i)}_{t+1}) w^{(i)}_t.$$

At time t, a sample from $\hat{p}(z^*_{t+1}|z_{1:t})$ can be obtained by iterating, for $i = 1,\ldots,N$, the following steps

Steps for sampling from $\hat{p}(z^*_{t+1}|z_{1:t})$ in a DPD model:

(i) Obtain $\hat{\mathbf{x}}^{(i)}_{t+1} = \boldsymbol{\theta}^{(i)}\mathbf{x}^{(i)}_t$;

(ii) Calculate $\hat{\boldsymbol{\lambda}}^{(i)}_{t+1} = \exp(\hat{\mathbf{x}}^{(i)}_{t+1})$;

(iii) Draw $z^{*(i)}_{t+1} \sim \mathrm{PD}(\hat{\lambda}^{(i)}_{1,t+1}, \hat{\lambda}^{(i)}_{2,t+1})$.

Steps for sampling from $\hat{p}(z^*_{t+1}|z_{1:t})$ in a DZPD model:

(i) Obtain $\hat{\mathbf{x}}^{(i)}_{t+1} = \boldsymbol{\theta}^{(i)}\mathbf{x}^{(i)}_t$;

(ii) Calculate $\hat{\boldsymbol{\lambda}}^{(i)}_{t+1} = \exp(\hat{\mathbf{x}}^{(i)}_{t+1} + h(\hat{p}^{(i)}))$;

(iii) Draw $z^{*(i)}_{t+1} \sim \mathrm{ZPD}(\hat{\lambda}^{(i)}_{1,t+1}, \hat{\lambda}^{(i)}_{2,t+1}, \hat{p}^{(i)})$ as follows

- Draw $u \sim Bernoulli(p)$; if $u = 1$, let $z^{*(i)}_{t+1} = 0$, otherwise draw

$$z^{*(i)}_{t+1} \sim \mathrm{PD}(\hat{\lambda}^{(i)}_{1,t+1}, \hat{\lambda}^{(i)}_{2,t+1}).$$

4.3.4 Diagnostics

In order to determine whether an PF filter works in practice the main factor is considered to be the presence or absence of degeneracy (defined in Section 4.2.2.1) which needs to be visually checked (Doucet and Johansen, 2009). To evaluate the efficiency of the suggested models, mean squared error (MSE) (Pitt and Shephard, 1999) and the PIT (defined in Chapter 2) for the predicted values are calculated. In order to compare the performance of the suggested models, we introduce a sequential version of the DIC in the following section.

4.3.4.1 Sequential deviance information criterion

Consider the likelihood function in an HMM context in (4.4), and the deviance in (3.15) in Section 3.3.4. Therefore, we propose the sequential deviance at time t as follows

$$D(z_{1:t}, x_{1:t}, \psi) = -2 \sum_{j=1}^{t} \log \left[g(z_j | x_j, \psi) \right]. \tag{4.38}$$

Now let us consider the definition of the posterior mean deviance and the deviance at $\hat{\psi}$ in (3.16) and (3.17), respectively. Using the sequential deviance in (4.38), the sequential posterior mean deviance may be obtained as

$$\hat{D}_{avg}(z_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} D(z_{1:t} | x_{1:t}^{(i)}, \psi^{(i)}), \tag{4.39}$$

$$\tag{4.40}$$

where N is the number of particles, $\left(x_{1:t}^{(i)}, \psi^{(i)}, w_t^{(i)} \right)$, $i = 1, \ldots, N$, are the particle approximation to $p(\mathbf{x}_{1:t}, \boldsymbol{\theta} | z_{1:t})$ obtained from LWF algorithm; and the sequential deviance at $\hat{\psi}$ may be given by

$$\hat{D}_{\hat{\psi}}(z_{1:t}) = D(z_{1:t}, \hat{x}_{1:t}, \hat{\psi}), \tag{4.41}$$

where $\hat{x}_{1:t}$ and $\hat{\psi}$ are central values, such as the posterior mean of the target distribution. Therefore, the sequential DIC is defined as follows

$$\mathrm{DIC}_t = p_{\mathrm{D}_t} + \hat{D}_{avg}(z_{1:t}), \tag{4.42}$$

where $p_{\mathrm{D}_t} = \hat{D}_{avg}(z_{1:t}) - \hat{D}_{\hat{\psi}}(z_{1:t})$. The lower value of the DIC_t may suggest a better fit of a given model. Here for the simplicity, we plot the difference of the DIC_t from two given models M_1 (e.g. a basic model with no covariate) and M_2 (e.g. a model with covariates) $(\mathrm{DIC}_{2,t} - \mathrm{DIC}_{1,t})$ against time. Values close to 0 indicate that the two models are broadly equivalent in performance whereas negative values suggest that the model M_2 performs better than the model M_1.

	Dataset1	Dataset2	Dataset3	Dataset4	Dataset5	Dataset6
θ	-0.9	-0.3	0.3	0.9	0.9	0.9
σ^2	0.16	0.16	0.16	0.16	0.01	0.25

Table 4.1: Parameter values for the six simulated examples from the DPD model.

4.4 Results

To illustrate the application of our model, we first conduct a simulation study. For both the simulation study and the real application, we estimate the model parameters as well as the unobserved processes. For all our examples, we use the LWF algorithm shown in Algorithm 4.3 in Section 4.2.3.2, set $N = 5000$, $N_0 = \frac{N}{2}$ and the shrinkage constant $a = 0.975$.

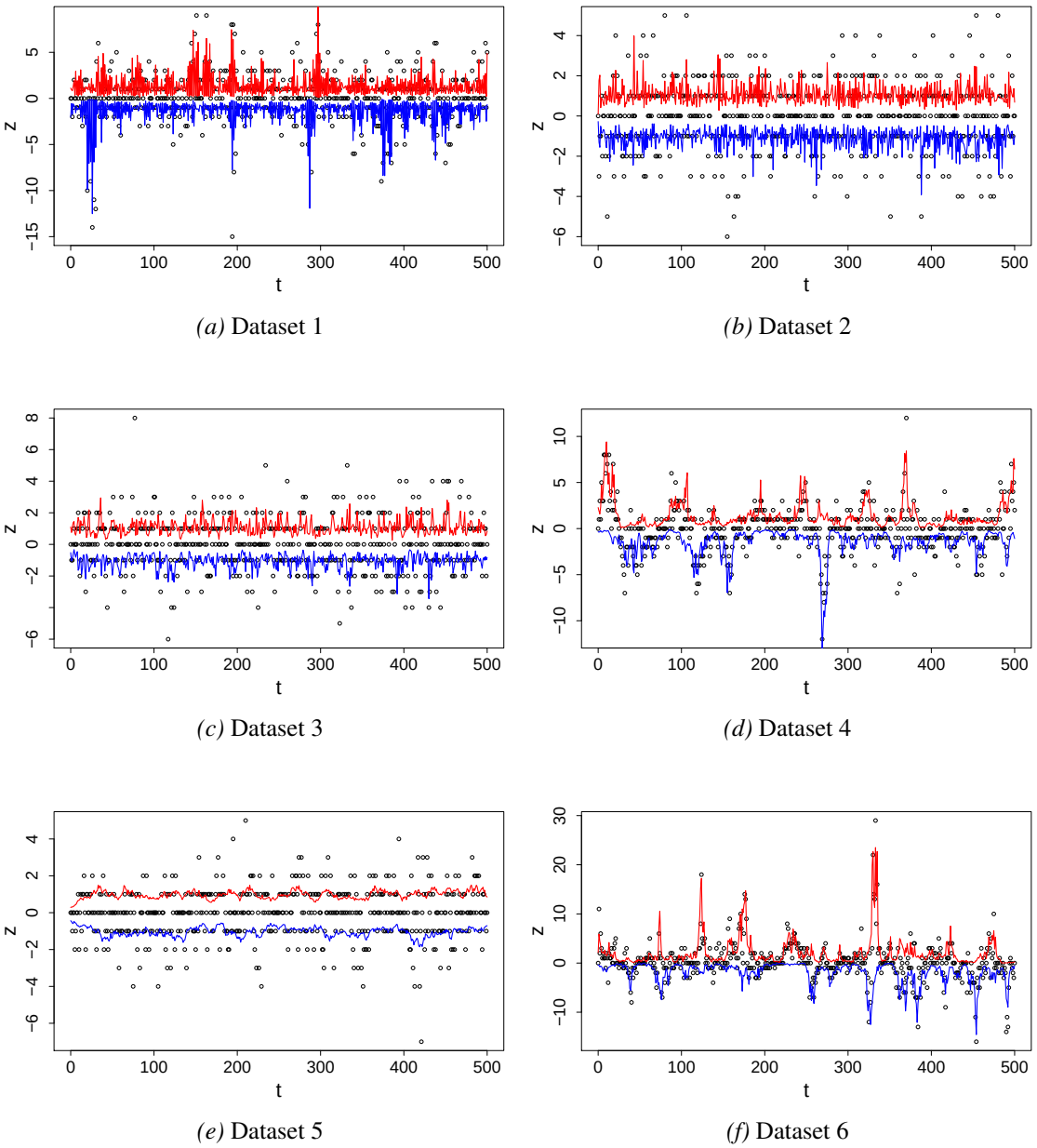
4.4.1 Simulation study

The interpretation of the model parameters may not be straightforward. Therefore, we first see how the model behaves for different values of the parameters. Then we fit the DPD and DZPD models to the corresponding simulated data sets. Finally, we show an example of model mis-specification

4.4.1.1 Dynamic Poisson difference model

Six sets of data with $n = 500$ observations each were simulated from the DPD model (4.32)-(4.34) with parameters shown in Table 4.1. We only considered the case of $\theta_1 = \theta_2 = \theta$ and $\sigma_1^2 = \sigma_2^2 = \sigma^2$, because index change is expected to be essentially symmetric over a trading day. The first four examples are considered for studying the role of θ in the DPD model with a fixed σ^2. The two other examples help us to understand the model behaviour as σ^2 varies when θ is held fixed.

Figure 4.2 shows the simulated data $\{z_t\}_{t=1}^n$ along with $\{\boldsymbol{\lambda}_t\}_{t=1}^n$ for the six models in table 4.1. For the sake of presentation, $\{\lambda_{2,t}\}_{t=1}^n$ is illustrated as $\{-\lambda_{2,t}\}_{t=1}^n$ on the graphs. This makes it easier to show the effect of the parameters λ_1 and λ_2 on the observed data. We noticed that when $|\theta|$ is close to 1 there is a high level of variation in the values of $\boldsymbol{\lambda}_t$ whereas for small values of θ the variation in $\boldsymbol{\lambda}_t$ drops significantly. The plots in Figures 4.2(e) and (f) show that as σ^2 increases, the variation in the values of $\boldsymbol{\lambda}_t$ increases



(a) Dataset 1 *(b)* Dataset 2

(c) Dataset 3 *(d)* Dataset 4

(e) Dataset 5 *(f)* Dataset 6

Figure 4.2: Six sets of simulated data of size $n = 500$ from the DPD model. Observed values $\{z_t\}$, $\{\lambda_{1,t}\}$ and $\{-\lambda_{2,t}\}$, are shown by black points, solid red and blue lines, respectively.

substantially. We may conclude that it is generally not possible to ascertain whether high (low) variation in the observed data was due to either a large (small) value of σ^2 or a large (small) value of $|\theta|$.

We will now demonstrate the performance of the LWF algorithm in recovering the latent processes and estimating the static model parameters. At time $t = 0$, to initialise the

LWF algorithm, we draw samples from

$$x_0 \sim \mathrm{N}(0, I), \quad \log(\Sigma^{(i)}) \sim \mathrm{N}(0, 4I), \quad \boldsymbol{\theta}^{(i)} \sim \mathrm{N}(0,\ 0.0625I), \quad \text{for } i = 1, \ldots, N.$$

Due to computational reasons the PD likelihood function becomes undefined for large values of σ^2 and $|\theta|$, therefore we set the parameters of our initial distributions as above.

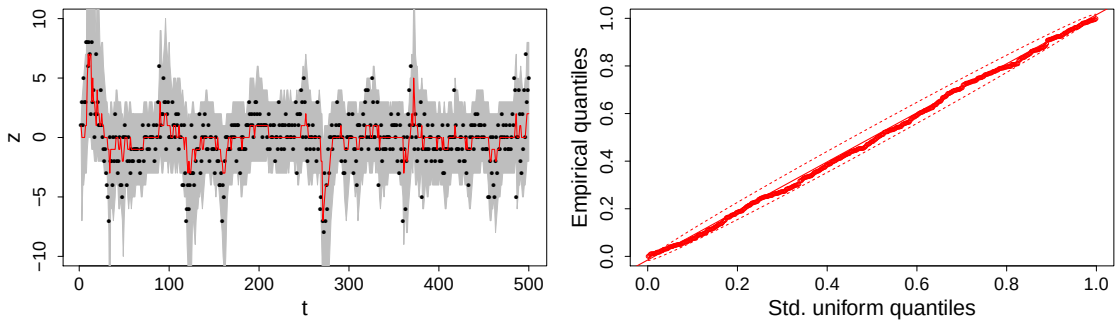
Obtained results suggest (Figures C.1-C.6) that

- for smaller values of θ, the credible intervals of the posterior distributions of the unobserved processes were wider but follow the pattern of the unobserved process $\{\boldsymbol{\lambda}_t\}$;
- for larger values of θ, the credible intervals of the posterior distributions of the unobserved processes were narrower but the true $\{\boldsymbol{\lambda}_t\}$ generally lies within the interval;
- for smaller values of σ^2, the credible intervals of the posterior distributions of the unobserved processes were wider but do not reflect changing pattern of $\{\boldsymbol{\lambda}_t\}$;
- for larger values of σ^2, the credible intervals of the posterior distributions of the unobserved processes became narrower, but the true $\{\boldsymbol{\lambda}_t\}$ generally lies within the interval.

In addition, Figures C.1-C.6 illustrate the sequential learning for the unknown static parameters of the model including: the true value, the posterior mean at time t along with the 95% credible interval of the posterior distribution for each of the parameters. We can see that how well θ was estimated depended on the true values of θ and σ^2, such that

- for large values of $|\theta|$ and σ^2, the convergence to the true value of θ was quite quick;
- for small values of $|\theta|$ and σ^2, the convergence to the true value of θ was very slow.

We noticed that the estimation of σ^2 itself did not depend on the true value of σ^2 and the LWF algorithm estimates σ^2 reasonably well.



(a) *(b)*

Figure 4.3: (a) The median (red line), the 95% credible intervals (grey area) of the posterior predictive distribution and the true observed values (black points) for Dataset 4. (b) A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution.

Among the six examples discussed in this section, the LWF algorithm performed the best for Dataset 4. Thus, we undertook a Monte Carlo study to investigate the consistency of our results. As suggested by other studies (Lopes and Tsay, 2011), we applied the LWF algorithm to Dataset 4, each time with different initial values. Results indicated that the LWF algorithm was able to recover well the unknown processes as well as static parameters for different starting values (Figures C.7).

Our main interest for using dynamic models and SMC methods lies in predicting the next index change immediately after the new observation becomes available. Thus, it is important to obtain predictions with a reasonable degree of accuracy. Figure 4.3 (a) shows that for Dataset 4, 94% of the values of $\{z_t\}$ fell within the 95% credible intervals of the predictive distribution. The QQ-plot presented in Figure 4.3 (b) indicates that the values obtained from the PIT have uniform distribution which can be confirmed by the K-S test with a p-value of 0.487. This suggests that the posterior distribution was able to explain the behaviour of the data well.

4.4.1.2 Dynamic zero inflated Poisson difference model

In this section, we fixed θ and σ^2 at 0.9 and 0.16, respectively and only investigated the effect of p on the performance of the LWF algorithm when the parameter p was added to the model. We chose three values of p: 0.2, 0.5 and 0.7. Three data sets of size 500 each were simulated from the DZPD model given in (4.35)-(4.37) with parameters as shown

	Dataset7	Dataset8	Dataset9
θ	0.9	0.9	0.9
σ^2	0.16	0.16	0.16
p	0.2	0.5	0.7

Table 4.2: Parameter values for the three simulated examples from the DZPD model.

in Table 4.2. These data sets are shown in Figure 4.4. The plots illustrate the observed process, $\{z_t\}$, along with $\{\lambda_{1,t}\}_{t=1}^{n}$ and $\{-\lambda_{2,t}\}_{t=1}^{n}$. The initial values of the parameters were exactly the same as in the previous section. For the parameter p, we used a logit transform and set

$$\log(\frac{p}{1-p}) \sim \mathrm{N}(0, 4I),$$

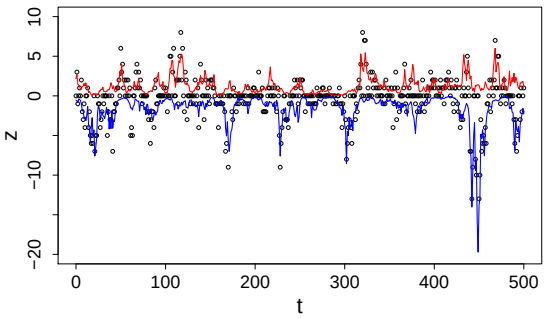
and $h(p) = -\log(1-p)$ as we discussed earlier in Section 4.3.2.

A sequential estimation of $\boldsymbol{\lambda}_t$ for Datasets 7- 9 are presented in Figures C.8-C.10. In these plots, 95% credible intervals of the posterior distribution of $\{\lambda_{1,t}\}$ and $\{\lambda_{2,t}\}$ are illustrated along with their true values. We can see that the algorithm has produced reasonable 95% credible intervals regardless of the value of p. The sequential learning of model parameters for Dataset 7- 9 were obtained and may suggest that the LWF algorithm has correctly estimated the static model parameters for all of the data sets..

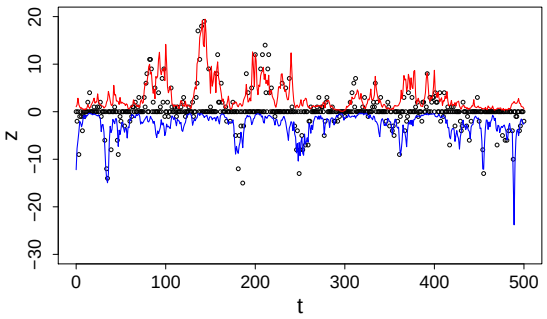
For the sake of presentation, we chose Dataset 8 and obtained one-step ahead predictions. The observed process and 95% credible interval of the predictive distribution are given in Figure 4.5 (a). Similar to the DPD model we used the PIT in order to see whether the model was correctly specified. The Q-Q plot (Figure 4.5 (b)) and the p-value of 0.85 obtained from the K-S test confirm that the DPZD model is able to model the behaviour of data quite well.

Furthermore, we fitted a DPD model to Dataset 8 to investigate model mis-specification. Results are presented in (Figure C.11). It can be seen that the values of $\{\boldsymbol{\lambda}_t\}$ were mostly outside the 95% credible intervals of the posterior distribution of $\{\boldsymbol{\lambda}_t\}$. The difference of the sequential DIC of the DPD model from the DZPD model also indicates better performance of the DZPD model (Figure 4.6).

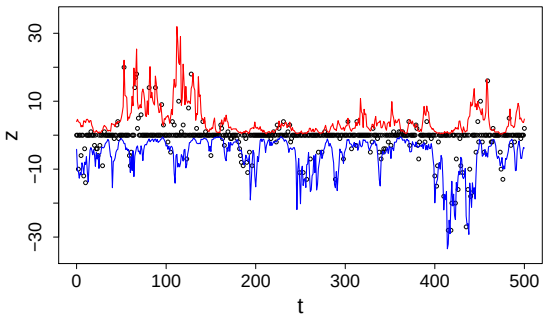
In summary, we have learnt from the simulation study that the LWF algorithm



(a) Dataset 7

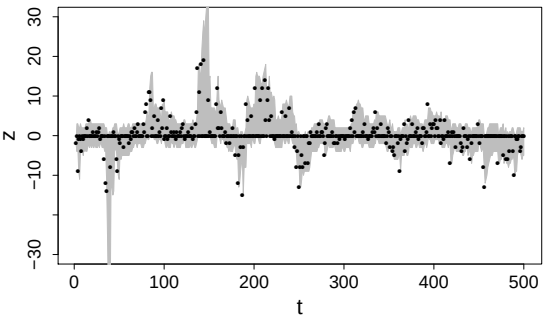


(b) Dataset 8

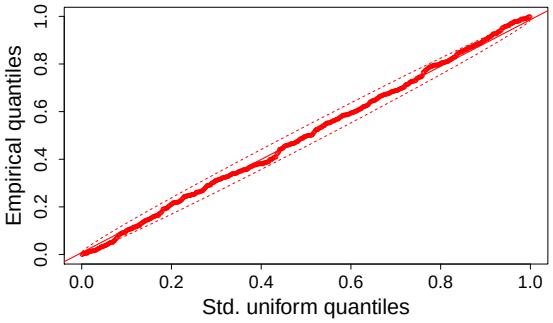


(c) Dataset 9

Figure 4.4: Three sets of simulated data from the DZPD model, along with values of $\{\lambda_{1,t}\}$ (red line) and $\{-\lambda_{2,t}\}$ (blue line).



(a)



(b)

Figure 4.5: (a) The 95% credible intervals (grey area) of the posterior predictive distribution and the true observed values (black points) for Dataset 8. (b) A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution.

(i) is able to produce reliable estimates of σ^2 and p regardless of the true values of the model parameters;

(ii) performed poorly for small values of $|\theta|$ and σ^2;

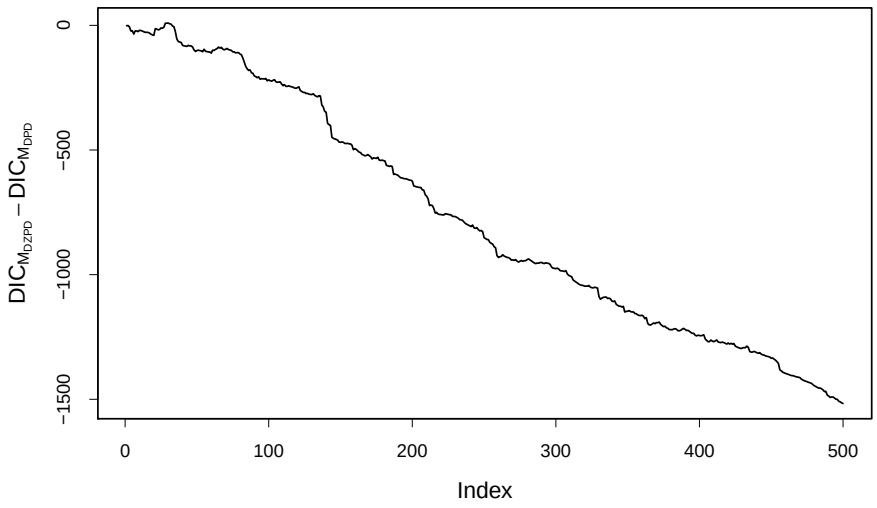


Figure 4.6: Difference of the DIC_t of the DPD model, $\mathrm{DIC}_{M_{\mathrm{DPD}}}$, from the DZPD model, $\mathrm{DIC}_{M_{\mathrm{DZPD}}}$.

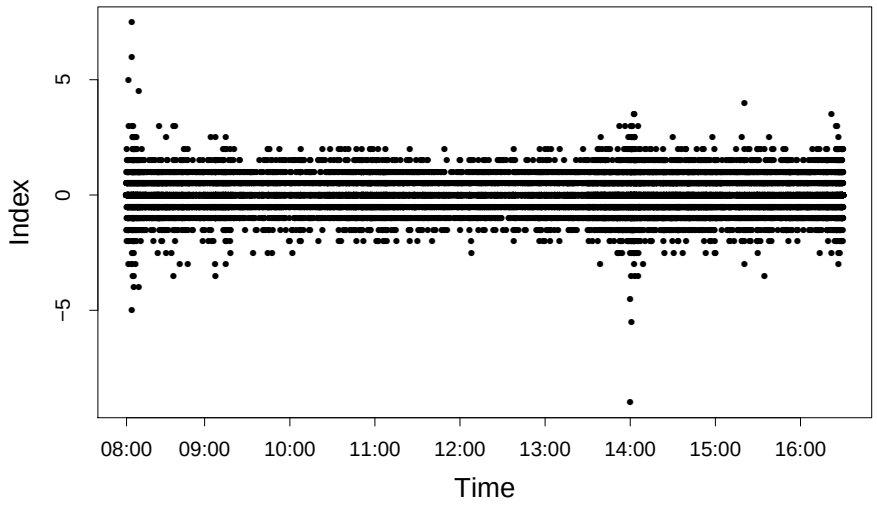


Figure 4.7: One day index change from 25th of March 2008.

(iii) is able to identify model mis-specification.

4.4.2 A dynamic model for the FTSE100 index change

Let index change be represented by $\{z_j\}_{j=1}^{n}$, where z_j is the jth index change associated with the transaction at time t_j. In this part we intend to fit the DZPD model, given by (4.35)-(4.37), to index change. Figure 4.7 shows that index change mainly fluctuates over the range of $(-4,4)$.

Our preliminary analysis of the first 1000 index changes showed that the LWF algo-

rithm performs poorly (Figure C.12). In these graphs, X-axis shows time stamp in minute, which can be read as 8:00:00, 8:01:00 and so on. We can assign 330 transactions to each minute on average. One can see that the sequential estimation of $\boldsymbol{\lambda}_j$ hardly moves after few seconds. In addition, the estimation of $\boldsymbol{\theta}$ is always close to zero, with wide credible intervals containing zero. This may imply either index change has not been affected by the unobserved processes, or the model is not correct.

In order to improve our model we need to address the following problems.

(i) Looking back at Figure 4.7, we can see that there are two time periods during the day, around 8 am and 2 pm, when the magnitude of index change is higher than in the rest of the data. This may suggest that the current form of the DZPD model may not be appropriate because the structure of data changes at these time periods.

(ii) We also noticed that as the distance between the starting point and time point t increases the weights from the LWF algorithm become degenerate. That is, the model approximation is based only on one particle instead of N.

We improve the model performance by adopting the following strategies in Sections 4.4.2.1-4.4.2.3.

4.4.2.1 A DZPD model with covariates

In Figure 4.8 we aggregated data into 5-minute time intervals (resulted in 102 time intervals) for March 25, 2008. From Figure 4.8(a) we can see that the market usually has a considerable high number of trades immediately after the opening and just before the closing hours of the market. Having a high number of transactions in a fixed time interval, suggests that there is a high intensity in the market, which means that the time duration between trades is very small and almost negligible (Figure 4.8(b)). Now, if we compare the diurnal pattern graph shown in Figure 4.8(b) and the one day index change in Figure 4.7, we can see that transactions with the bigger changes (usually more than 8 ticks) happened when the intensity in the market was high. Thus, we might be able to associate the size of index change with the time duration between two consecutive transactions. For this reason, we add the time duration between two consecutive transactions as a covariate

to the DZPD model. If we set $\Delta t_j = t_j - t_{j-1}$, thus a DZPD model with one covariate can be set as follows

Observation equation

$$Z_j|(\mathbf{X}_j = \mathbf{x}_j) \sim ZPD(\lambda_{1,j}, \lambda_{2,j}, p), \tag{4.43}$$

State equation

$$\log(\boldsymbol{\lambda}_j) = \mathbf{x}_j + h(p) + \boldsymbol{\alpha} \log(\Delta t_j + 1), \tag{4.44}$$

$$\mathbf{x}_j = \boldsymbol{\theta}\mathbf{x}_{j-1} + \boldsymbol{\nu}_j, \tag{4.45}$$

where $\mathbf{x}_i$, $\boldsymbol{\theta}$ and $\boldsymbol{\nu}_i$ are defined in Section 4.3.1 and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^{\mathrm{T}}$ is the corresponding parameter vector for time duration between trades. The parameter vector of the DZPD model can be written as $\psi_{\mathrm{DZPD}} = (\theta_1, \theta_2, \alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2, p)$.

To initialise the filter, we draw samples of $\boldsymbol{\alpha}$ from $\mathrm{N}(0, I)$, and the rest of the parameters are initialised exactly in the same way as described in Section 4.4.1.2.

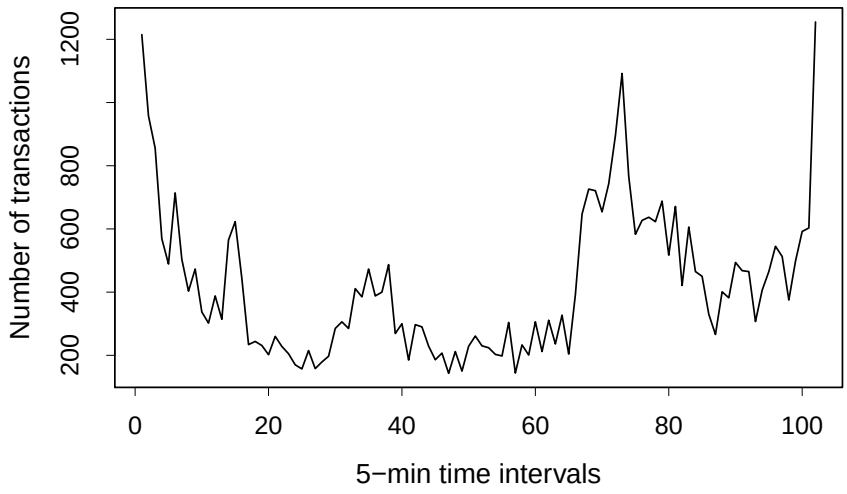
However, having time duration between two consecutive transactions as a covariate did not make any difference to our results. In order to capture the volatility in the data we may need to define the DZPD model in a way that as soon as the market condition changes, the model adapts to it. The current DZPD model is not suitable for this purpose.

4.4.2.2 Nonlinear state equation

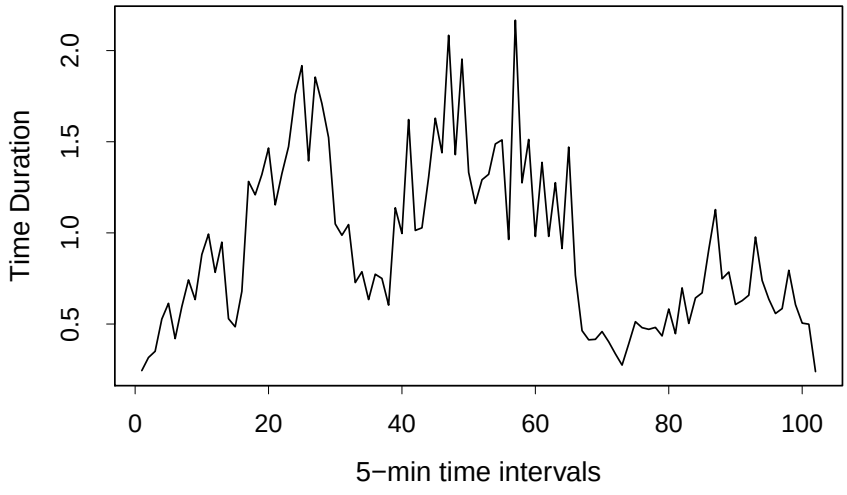
For a time series variable X_t, the linear class of autoregressive models of order p can be generalised as follows

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-p}) + \omega_t, \tag{4.46}$$

where f is some non-linear function and ω_t denotes strict white noise (Chatfield, 2001, Chapter 3). This model is called a *non-linear autoregressive model* of order p. In the case of state space models described here, the nonlinear state equation may be a nonlinear



(a)



(b)

Figure 4.8: FTSE100 transactions data from 25/03/2008, (a) the number of transactions and (b) the time duration between two consecutive transactions, in 5-minute time intervals.

autoregressive model for the case of $p = 1$. Therefore, a wide range of model variety may fall into this class of nonlinear autoregressive models. The specific feature of interest in this section is the model with time varying coefficients, given by

$$Y_t = \beta_t Y_{t-1} + \omega_t, \tag{4.47}$$

$$\beta_t = \tau_0 + \tau_1 \beta_{t-1} + \varepsilon_t, \tag{4.48}$$

where τ_0 and τ_1 are constant, $\varepsilon_t \sim \mathrm{N}(0, \Lambda_\varepsilon)$ is independent from $\omega_t \sim \mathrm{N}(0, \Lambda_\omega)$, and Λ_ε and Λ_ω are diagonal matrices of appropriate size. In the case of $\tau_1 = 0$, the above model reduces to what is called *random coefficient model* (Chatfield, 2001, Chapter 3).

Thus, in the DZPD model context, a time varying coefficient for the state model is given by

Observation equation

$$Z_j|(\mathbf{X}_j = \mathbf{x}_j) \sim \mathrm{ZPD}(\lambda_{1,j}, \lambda_{2,j}, p), \tag{4.49}$$

State equation

$$\log(\boldsymbol{\lambda}_j) = \mathbf{x}_j + h(p) + \boldsymbol{\alpha} \log(\Delta t_j + 1), \tag{4.50}$$

$$\mathbf{x}_j = \boldsymbol{\theta}_j \mathbf{x}_{j-1} + \boldsymbol{\nu}_j, \tag{4.51}$$

$$\boldsymbol{\theta}_j = \boldsymbol{\gamma} + \boldsymbol{\omega}_j, \tag{4.52}$$

where $\mathbf{x}_j$, $\boldsymbol{\theta}$ and $\boldsymbol{\nu}_j$ are given in Section 4.3.1 and

$$\boldsymbol{\gamma} = \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\omega}_j = \begin{pmatrix} \omega_{1,j} \\ \omega_{2,j} \end{pmatrix} \sim \mathrm{N}(0, \boldsymbol{\varepsilon}),$$

such that

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1^2 & 0 \\ 0 & \varepsilon_2^2 \end{pmatrix}.$$

Equations in (4.51) and (4.52) can be combined as follows

$$\mathbf{x}_j = \boldsymbol{\gamma}\mathbf{x}_{j-1} + \boldsymbol{\omega}_j \mathbf{x}_{j-1} + \boldsymbol{\nu}_j. \tag{4.53}$$

If we denote the vector of the parameters by $\psi_{\mathrm{NLDZPD}} = (\gamma_1, \gamma_2, \alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2, \varepsilon_1^2, \varepsilon_2^2, p)$, then we have $\mathbf{x}_0 \sim \mathrm{N}(0, I)$, $f_{\psi_{\mathrm{NLDZPD}}}(\mathbf{x}_j|\, \mathbf{x}_{j-1}) = \mathrm{N}(\boldsymbol{\gamma}\mathbf{x}_{j-1}, \boldsymbol{\omega}\mathbf{x}_{j-1}^2 + \Sigma)$ and $g_{\psi_{\mathrm{NLDZPD}}}(Z_j|\, \mathbf{X}_t = \mathbf{x}_j) = f_{\mathrm{ZPD}}(\lambda_{1,j}, \lambda_{2,j}, p)$.

4.4.2.3 Degeneracy

It is a general case for SMC methods that as i increases, such methods become less efficient (Doucet and Johansen, 2009; Petris et al., 2009). This happens because as i increases

(i) successive resampling steps lead to particle degeneracy. That is, the approximation is based only on a single unique particle;

(ii) we are effectively targetting the joint posterior distribution $\hat{p}_\theta(x_{0:j}|y_{1:j})$, so we are trying to track a distribution in an increasingly large number of dimensions;

(iii) errors accumulate as we move from time j to $j+1$ because the jth approximation at time t_j is based on the $(j-1)^{\text{th}}$ approximation at time t_{j-1} which results in accumulated errors;

A practical solution to this problem is to run the particle filter over a fixed time interval of length k. The sequential data is considered in blocks of size k, such that the rth block will consist of all observations at time points $(t_{(r-1)k+1},\ldots,t_{rk})$ for $r = 1,2,\ldots$. The PF algorithm is applied to each block independently. However, at the beginning of each new block $t_{(r-1)k+1}$, $r = 2,3,\ldots$, we restart the algorithm by assigning the posterior means of the parameters obtained at the last time point $t_{(r-1)k}$, $r = 2,3,\ldots$, of the previous block as our initial values. A problem with this procedure is that we do not know the value of k. We might have a poor approximation for small values of k, on the other hand a large value of k may lead to the degeneracy well before reaching the end of the chosen interval. A practical way to choose k is to run the algorithm with different values of k to ascertain approximately the time point from where the degeneracy sets in.

After some experimentations, we first found that for our data a good choice of k was to be 200, which is well before the degeneracy starts. Therefore, the data is divided into 226 intervals with the first 225 containing 200 observations each and the last one containing of the remaining 266 observations.

Before applying the above strategies to the LWF algorithm, it would be useful to investigate the effect of the $h(p)$ over the number of estimated zeros in the model, to make sure the model is able to capture the excess of zeros in the data. The current choice of

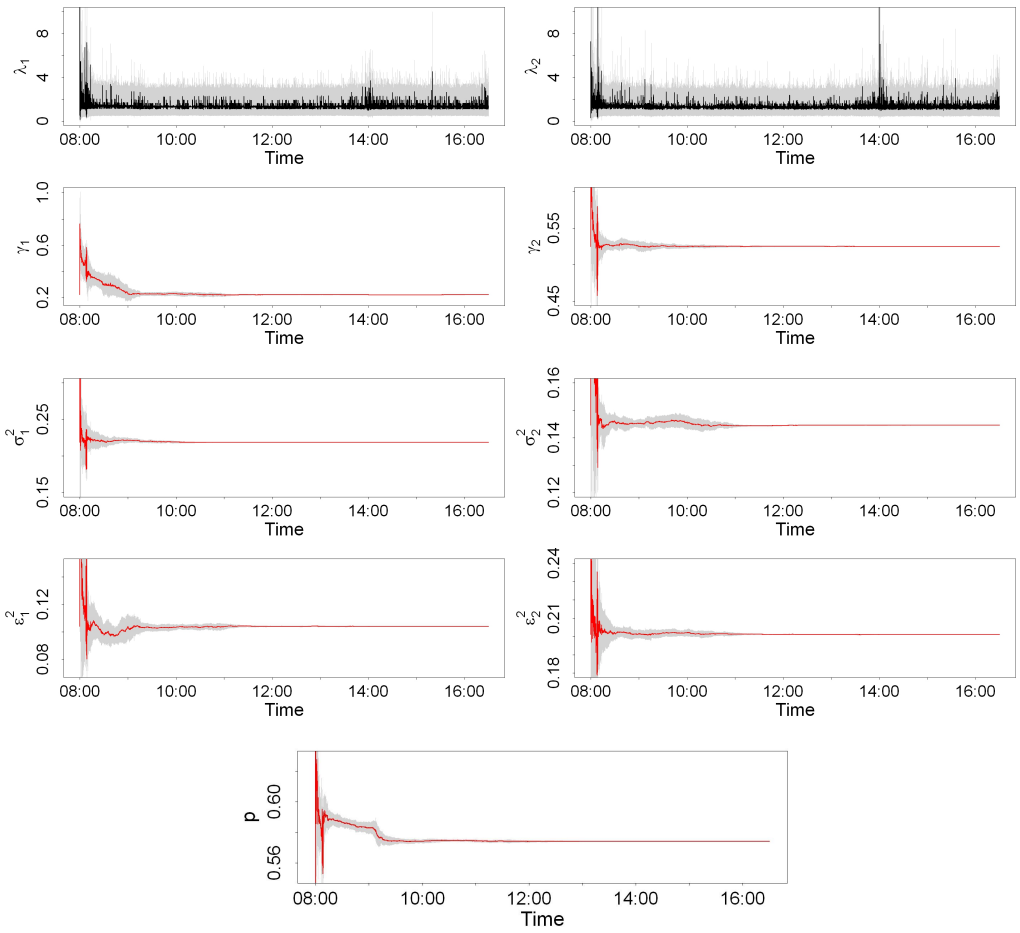


Figure 4.9: The posterior mean and 95% credible intervals (grey area) of the unobserved processes (the first row) and the parameters of model M_1 for the FTSE100 index change.

$h(p)$ results in only positive values. Also it underestimates the number of zeros by 4%. If we change our choice to a function over the real line, for example $h(p) = \log[p/(1-p)]$, we may add more flexibility to the model and overcome the problem of underestimation of the number of zero index changes.

By adopting these strategies, the LWF algorithm performance improved significantly. Results are presented in Figures 4.9 and 4.10 which suggest that the DZPD models (i) a basic model, i.e., a model without any covariates, M_1 and (ii) a model with the time duration between two consecutive transactions as a covariate, M_2, both are correctly able to capture the behaviour of the data.

Although, results from both the models look almost similar, we need to perform model diagnostics to decide which of the two models fits better. For the purpose of model comparison we first compared the MSE from one-step ahead predictions. The MSEs were

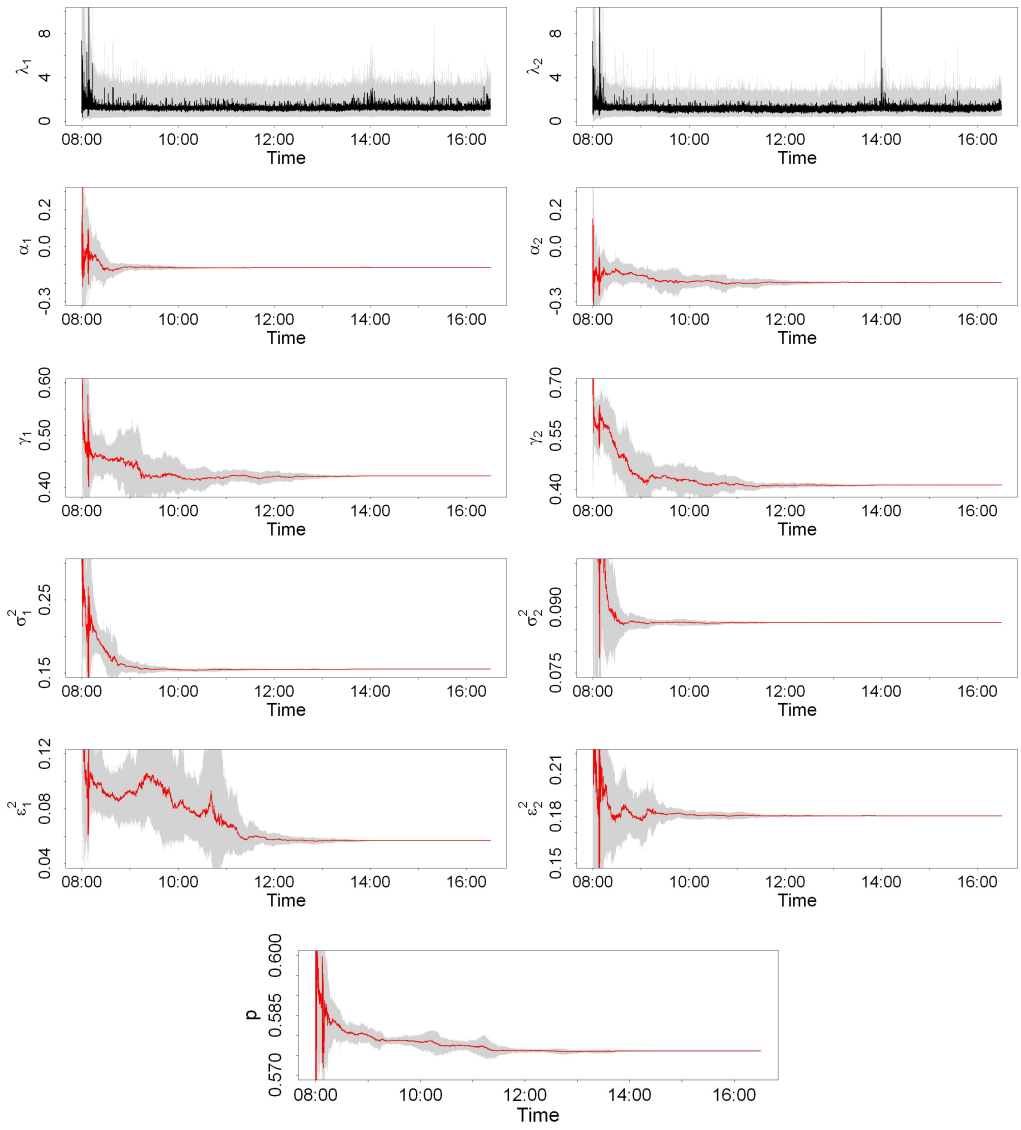
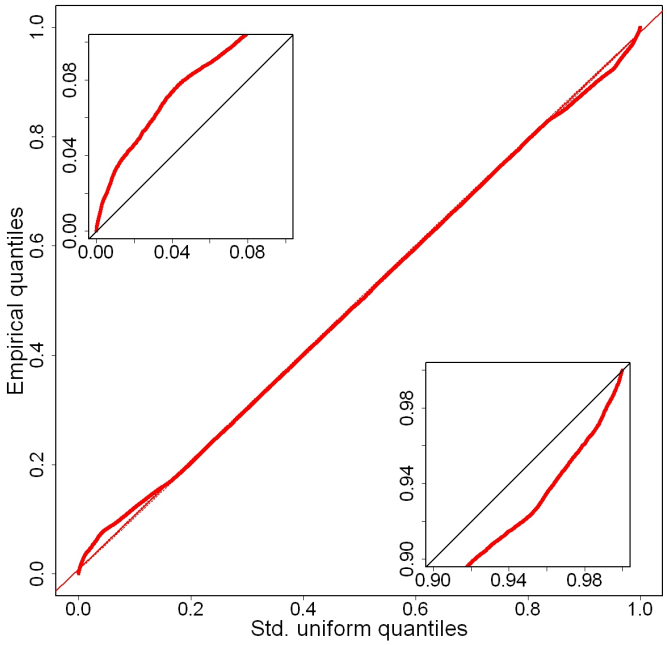
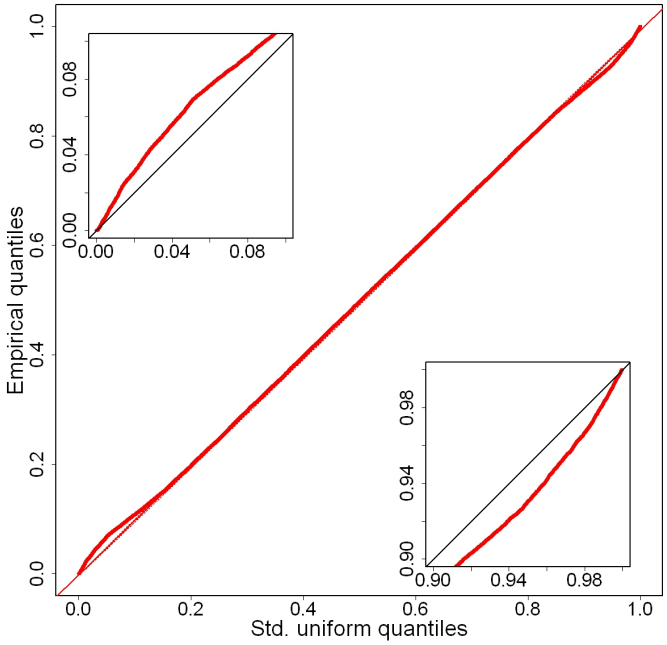


Figure 4.10: The posterior mean and 95% credible intervals (grey area) of the unobserved processes (the first row) and the parameters of model M_2 for the FTSE100 index change.

0.843, 0.853 for models M_1 and M_2, respectively. In addition, using the PIT, we saw that both models are performing equally well (Figure 4.11). However, after comparing the marginal predictive distribution of index change of both models over a whole day, we saw that model M_1 estimated about 68% of index changes as zero, whereas for model M_2 this proportion was about 68.8% which was closer to 69%, the observed proportion of zero values in the real data set. Figure 4.12 illustrates the difference of the DIC_t for M_1 from M_2. The obtained negative values suggests that M_2 performed better than M_1. This indicates that, though M_1 explained the behaviour of the data well, still the model has been improved further by adding additional information to it.



(a)



(b)

Figure 4.11: A uniform Q-Q plot of random samples drawn from intervals based on the cumulative predictive distribution (a) M_1, (b) M_2. The first and the last 10% of tails are shown on a larger scale.

4.5 Summary

In this chapter we provided a dynamic framework for modelling ultra high-frequency financial data using ZPD distribution via SMC methods. For this purpose, we introduced

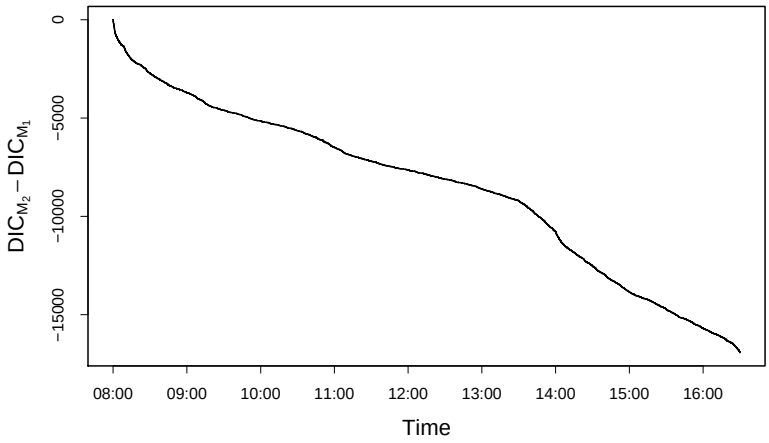


Figure 4.12: The difference of the DIC_t of the M_1 from M_2. Negative values indicate a better fit of the model M_2 compare to the model M_1.

SMC methods. We also discussed one well-known improvement (APF) and an extension (LWF algorithm) to the basic particle filter algorithms to overcome the problem of degeneracy and estimating unknown static model parameters.

The main part of this chapter focused on the DZPD model and its application. We first introduced basic the DPD and DZPD models. Using the LWF algorithm, we fitted our models to nine sets of simulated data and identified limitations of such models and discussed model mis-specification.

Furthermore, the DZPD model was fitted to the real data set. Our study showed that our original model did not explain the behaviour of the data well and the algorithm suffered from severe degeneracy. To improve the DZPD model we adopted a non-linear state equation, introduced a covariate in the model and applied the LWF algorithm over a fixed length of time. Using the sequential DIC, proposed for the sequential assessing of the dynamic models in this chapter, we showed that the non-linear DZPD model performs better if the time duration between two consecutive transactions is included as a covariate in the model.

Moreover, the improved DZPD was able to capture fluctuations of index change and perform online prediction. We would like to point out that there is a considerable scope of extending the DZPD model which we discuss in Chapter 6.

Chapter 5

Generalised Poisson difference distribution

5.1 Introduction

The Poisson distribution has been used in a wide variety of situations as the underlying distribution for the purpose of modelling count data (Cameron and Trivedi, 1996; Gendron and McCausland, 2005; Gourieroux et al., 1984; Tsay, 2005; McCulloch and Tsay, 2001). Such data frequently arise as outcomes of an underlying count process in continuous time (Winkelmann, 2008). A Poisson distribution is characterised by one parameter, suggesting the equal mean and variance. As the mean and variance of the Poisson distribution are equal, we say that Poisson distribution satisfies the equi-dispersion property (Consul and Jain, 1973; Famoye, 2010; Consul, 1989; Consul and Famoye, 2006; Ismail and Jemain, 2007). In addition, under the Poisson model, it is assumed that events satisfy the principle of independence. However, there are several examples where these principles do not hold (Consul and Jain, 1973; Famoye, 2010). For example, number of purchased units of different commodities (Consul and Famoye, 2006), or the number of major derogatory reports in the credit history (Greene, 1994), may be serially correlated or have considerably greater (smaller) sample variance than the sample mean.

These problems have been identified by the literature and led to the development of so-called *generalized* probability models (Consul, 1989; Consul and Famoye, 2006; Consul and Jain, 1973; Nikoloulopoulos and Karlis, 2008). The generalised Poisson (GP) dis-

tribution has been introduced and used when the either data is over- or under-dispersed, as an alternative to the Poisson distribution. The main advantage of using the GP distribution is that such distribution has one additional parameter. The additional parameter introduces over-, equi- or under-dispersion which allows the variance to be greater, equal to or less than the mean according to whether is positive, zero or negative. Both the mean and the variance tend to increase or decrease as the rate parameter, increases or decreases.

5.2 The generalised Poisson distribution

The following form of the GP distribution is originally suggested by Consul and Jain (1973) and discussed in detail; see (Consul and Famoye, 2006, Chapter 9) for further details. Let us assume X is a non-negative discrete random variable and $f_{\mathrm{GP}}(\lambda,\theta)$ denotes the probability distribution of $\mathrm{X}=x$ which is defined as follows

$$f_{\mathrm{GP}}(x|\lambda,\theta)=\begin{cases}\lambda(\lambda+\theta x)^{x-1}e^{-\lambda-\theta x}/x!, & x=0,1,2,\ldots,\\ 0, & \text{for} \quad x>m \quad \text{if } \theta<0,\end{cases} \tag{5.1}$$

where $\lambda>0$, $\max(-1,-\lambda/m)\leq\theta\leq 1$, and $m(\geq 4)$ is the largest positive integer for which $\lambda+m\theta>0$ when $\theta<0$. This condition is considered to ensure that there are at least five classes with non-zero probability when θ is negative. It can be seen that the GP distribution introduces over- or under-dispersion whether θ is positive or negative. Also, it reduces to the Poisson distribution when $\theta=0$.

The distribution, for all $x>m$, become a truncated distribution at zero when $\theta<0$ which results in $\sum_{x=0}^{m} f_{\mathrm{GP}}(x|\lambda,\theta)<1$. However, this truncation error is less than 0.5% when $m\geq 4$ which is negligible in practical applications. Consul and Famoye (2006, Chapter 9) have discussed some strategies in order to modify this truncation error.

5.2.1 Generating functions

Consul and Famoye (2006) discussed the properties of the GP distribution under the class of the general Lagrangian distributions. Let us start by defining the Lagrange expansion and the general Lagrangian distribution.

Consider z to be a function of u, such that $z = u\,g(z)$, for $-1 < z < 1$. Also, let $f(z)$ and $g(z)$ be two analytic functions of z, which are infinitely differentiable with respect to (w.r.t) z and $g(0) \neq 0$. The Lagrange's inversion theorem states that any function of z can be expressed as a power series in u which converges for sufficiently small u (Consul and Famoye, 2006, Chapter 2). According to this theorem, the following power series expansion can be obtained

$$z = \sum_{x=1}^{\infty} \frac{u^x}{x!} \left[\frac{d^{x-1}}{dz^{x-1}} g(z)^x \right]_{z=0}, \tag{5.2}$$

$$f(z) = f(0) + \sum_{x=1}^{\infty} \frac{u^x}{x!} \left[\frac{d^{x-1}}{dz^{x-1}} \left(g(z)^x \frac{d}{dz} f(z) \right) \right]_{z=0}. \tag{5.3}$$

The expansion in (5.3) forms the probability generating function (pgf) of the discrete general Lagrangian probability distribution (Consul and Famoye, 2006, Chapter 2). Consequently, the probability mass function of such distributions is given by

$$P(X = 0) = f(0), \tag{5.4}$$

$$P(X = x) = (1/x!) \left[\frac{d^{x-1}}{dz^{x-1}} \left(g(z)^x \frac{d}{dz} f(z) \right) \right]_{z=0}. \tag{5.5}$$

In this case, the two functions $g(z)$ and $f(z)$ are called the transformer and the transformed functions, respectively. Each set of $g(z)$ and $f(z)$, satisfying the following conditions

$$\left[\frac{d^{x-1}}{dz^{x-1}} \left(g(z)^x \frac{d}{dz} f(z) \right) \right]_{z=0} \geq 0, \quad x \in \mathbf{N},$$

$g(0) \neq 0$, $g(1) = 1$ and $f(1) = 1$, together provide a general Lagrangian distribution. Therefore, a large number of general Lagrangian distributions can be generated.

For the case of GP distribution, Consul and Famoye (2006) suggested $g(z) = e^{\theta(z-1)}$ and $f(z) = e^{\lambda(z-1)}$. It can be seen that by substituting $f(z)$ and $g(z)$ in (5.5), the GP

distribution in (5.1) can be obtained. In this case, the pgf is expressed as follows

$$G(u) = f(z) = e^{\lambda(z-1)}, \text{ where } z = u\, e^{\theta(z-1)}. \tag{5.6}$$

In order to obtain the moment generating function (mgf) for the GP distribution, in (5.6), let $z = e^s$ and $u = e^{\beta}$. Thus the mgf of the GP distribution is given by

$$M_x(\beta) = e^{\lambda(e^s-1)}, \text{ where } s = \beta + \theta(e^s - 1). \tag{5.7}$$

Thus, the cumulative generating function (cgf) of the GP distribution can be obtained by taking the logarithm of the mgf as follows

$$\psi(\beta) = \ln M_x(\beta) = \theta(e^s - 1), \text{ where } s = \beta + \theta(e^s - 1). \tag{5.8}$$

5.2.2 Moments of generalised Poisson distribution

Consider the pgf of the GP distribution in (5.6), in which z can be shown as a function of u such that $z = u\, g(z) = \psi(u)$, where $\psi(u)$ is given by (5.2). The descending factorial moments of the GP distribution can be obtained by successively differentiating (5.6) w.r.t u, for $z = u = 1$. The first two derivatives in terms of $f(z)$ and $g(z)$ are

$$\frac{\partial f(\psi(u))}{\partial u} = \frac{dz}{du}\, f'(z), \tag{5.9}$$

$$\frac{\partial^2 f(\psi(u))}{\partial u^2} = \frac{d^2z}{du^2}\, f''(z) + \left(\frac{dz}{du}\right)^2 f'(z), \tag{5.10}$$

where dz/du and d^2z/du^2 are given by

$$\frac{dz}{du} = \frac{g(z)}{1 - u\, g'(z)}, \tag{5.11}$$

$$\frac{d^2z}{du^2} = \frac{2g(z)\, g'(z)}{(1 - u\, g'(z))^2} + \frac{z\, g(z)\, g''(z)}{(1 - u\, g'(z))^3}. \tag{5.12}$$

By substituting (5.11)-(5.12) in (5.9)-(5.10) and then $z = u = 1$, the first two descending factorial moments $\mu'_{(1)}(u)$ and $\mu'_{(2)}(u)$ are obtained as follows

$$\mu'_{(1)}(u) = \mu = \frac{f'(1)}{1-g'(1)}, \tag{5.13}$$

$$\mu'_{(2)}(u) = \frac{f''(1)+2g'(1)\,f'(1)}{[1-g'(1)]^2} + \frac{f'(1)\,g''(1)}{[1-g'(1)]^3}, \tag{5.14}$$

where μ is the first central moment and therefore, the mean of the GP distribution. Using the first two factorial moments, the variance, σ^2 is obtained as

$$\begin{aligned}\sigma^2 &= \mu'_{(2)}(u) + \mu'_{(1)}(u) - \left(\mu'_{(1)}(u)\right)^2, \\ &= \frac{f''(1)+g'(1)\,f'(1)+f'(1)-(f'(1))^2}{[1-g'(1)]^2} + \frac{f'(1)\,g''(1)}{[1-g'(1)]^3}. \end{aligned} \tag{5.15}$$

Now let $f(z) = e^{\lambda(z-1)}$ and $g(z) = e^{\theta(z-1)}$, thus the the mean and the variance of the GP distribution are

$$\mu = \lambda(1-\theta)^{-1}, \quad \sigma^2 = \lambda(1-\theta)^{-3}.$$

Therefore, when $0 < \theta < 1$, $\theta = 0$ and $\max[-1, -\lambda/m] < \theta \leq 0$, the GP distribution displays over-dispersion ($\sigma^2 > \mu$), equi-dispersion ($\sigma^2 = \mu$) and under-dispersion ($\sigma^2 < \mu$), respectively. This implies that in the case of under-dispersion, the parameter space is restricted. We can obtain the maximum likelihood estimates of λ and θ by solving the following equations

$$\begin{aligned}\sum_{i=1}^{n} \frac{x_i(x_i-1)}{\bar{x}+(x_i-\bar{x})} - n\bar{x} &= 0, \\ \hat{\lambda} &= \bar{x}(1-\hat{\theta}),\end{aligned}$$

where $x_1, \ldots, x_n$ are the values of n independent random variables, $X_1, \ldots, X_n$, with probability distribution $f_{\mathrm{GP}}(\lambda, \theta)$ (Jiang et al., 2006, Chapter 10).

Using the recurrence formula provided by Consul and Famoye (2006, Chapter 9), the

coefficients of skewness (β_1) and kurtosis (β_2) are given by

$$\beta_1 = \frac{1+2\theta}{\sqrt{\lambda(1-\theta)}}, \tag{5.16}$$

$$\beta_2 = 3 + \frac{1+8\theta+6\theta^2}{\lambda(1-\theta)}. \tag{5.17}$$

It can be seen that the skewness of the GP distribution depends on the values of λ for any given value of θ. Any increase in the value of λ decreases the skewness of the GP distribution and the infinitely large value of λ results in zero skewness. Also, for any given value of λ, the skewness is infinitely large when θ is close to unity. The skewness is negative for $\theta < -\frac{1}{2}$.

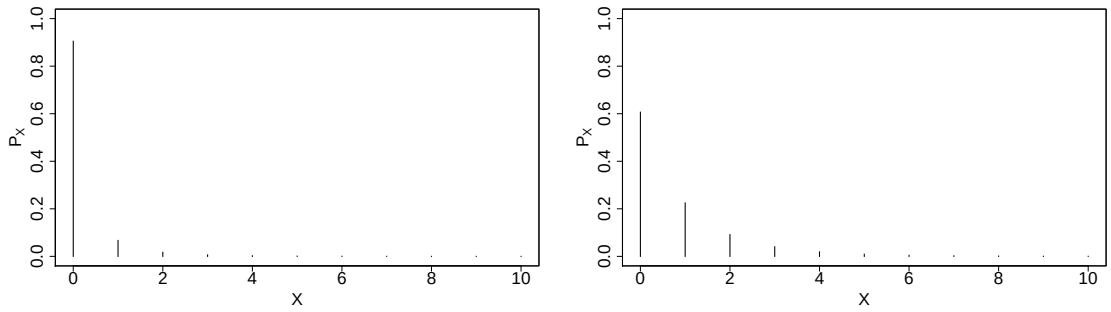
According to Consul and Famoye (2006), the GP distribution becomes leptokurtic (β_2 > 3) for all values of λ and $0 < \theta < 1$. The values of β_2 will be less than 3 resulting in a platykurtic GP distribution when values of θ vary between $-\frac{1}{6}\sqrt{10}-\frac{2}{3}$ and $\frac{1}{6}\sqrt{10}-\frac{2}{3}$; see Consul and Famoye (2006) for further details on the Lagrangian distributions and recent reviews.

5.2.3 Graphical presentation of GP distribution models

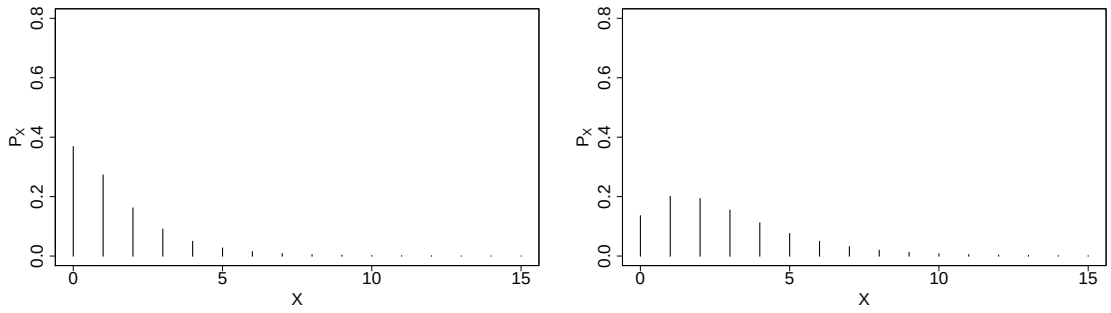
In this section we investigate the behaviour of the GP distribution for various values of λ and θ. For the purpose of comparison, we assume only one parameter is varying in each figure.

Figure 5.1 shows that for a given value of θ, when $\lambda < 1$ the GP has a small span and as λ increases the spread increases and forms a bell-shaped distribution. A comparison of the graphs in Figure 5.1, indicates that as λ decreases, the right tail shortens and results in a one-tailed distribution.

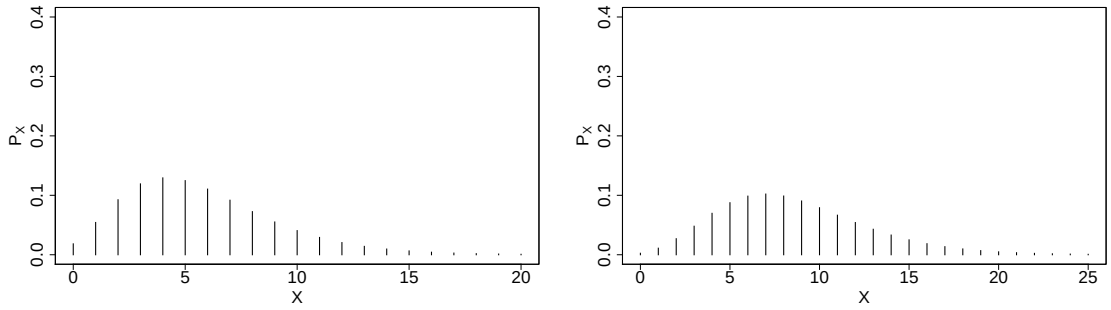
Figure 5.2 suggests that for any given value of λ, the variation in the value of θ substantially alters the form of the GP distribution. The increase in the value of θ increases the mean as well as the variance of the distribution. Also, the right tail of the GP distribution becomes very heavy, and the distribution tends to become flat as θ increases.



(a) $\lambda = 0.1$ *(b)* $\lambda = 0.5$



(c) $\lambda = 1$ *(d)* $\lambda = 2$

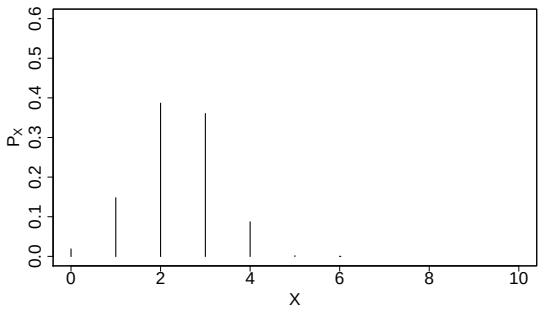


(e) $\lambda = 4$ *(f)* $\lambda = 6$

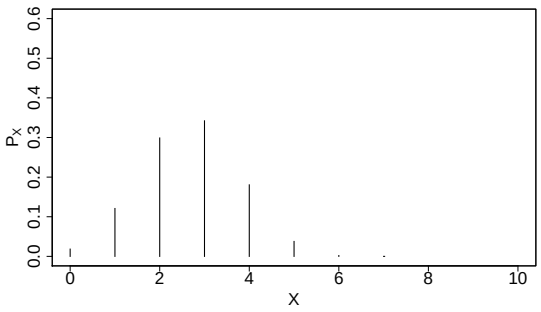
Figure 5.1: The GP distribution for $\theta = 0.3$

5.3 The generalised Poisson difference distribution

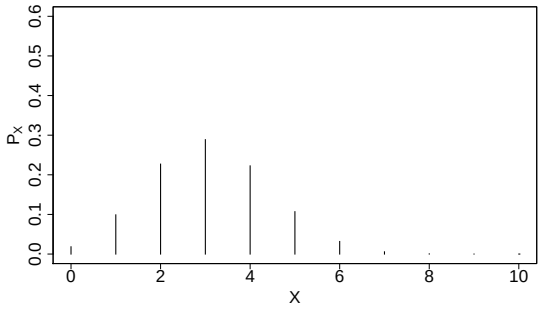
To overcome the underestimation of the tails in a PD model we might be able to use the distribution of the difference of the two random GP variables. That means the additional parameters give more control of the tails compare to the PD distribution. In this section we introduce the generalised Poisson difference (GPD) distribution with four parameters and investigate its properties. We then outline a possible application of the GPD distribution in modelling UHF data sets.



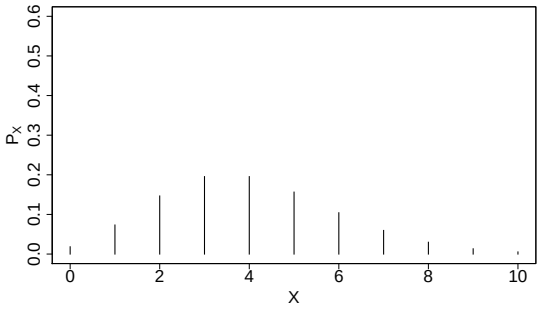
(a) $\theta = -0.7$



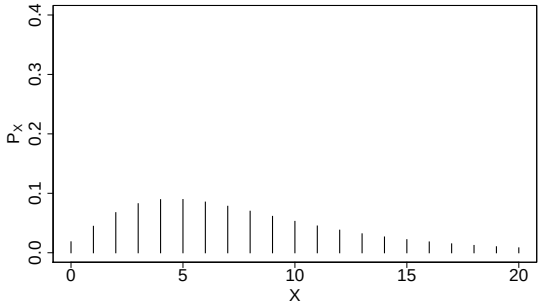
(b) $\theta = -0.5$



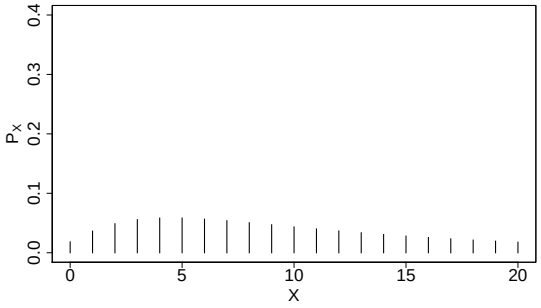
(c) $\theta = -0.3$



(d) $\theta = 0$



(e) $\theta = 0.5$



(f) $\theta = 0.7$

Figure 5.2: The GP distribution for $\lambda = 4$

5.3.1 Differences of two GP variates

Let us assume $X \sim \mathrm{GP}_X(\lambda_1, \theta_1)$ and $Y \sim \mathrm{GP}_Y(\lambda_2, \theta_2)$ are independently distributed. The joint distribution of X and Y is given by

$$f_{\mathrm{X,Y}}(x, y|\lambda_1, \lambda_2, \theta_1, \theta_2) = f_{\mathrm{X}}(x|\lambda_1, \theta_1)\, f_{\mathrm{Y}}(y|\lambda_2, \theta_2).$$

Therefore, we obtain the distribution of the difference $Z = X - Y$ as follows

$$\begin{aligned} f_{\text{GPD}}(Z = X - Y = z) &= \sum_{y=0}^{\infty} f_{\text{X,Y}}(z+y,\, y) \\ &= \sum_{y=0}^{\infty} \frac{\left[\lambda_1(\lambda_1 + \theta_1(z+y))^{z+y-1}\right]}{(z+y)!}\, e^{-\lambda_1 - \theta_1(z+y)} \times \\ &\quad \frac{\left[\lambda_2(\lambda_2 + \theta_2 y)^{y-1}\right]}{y!}\, e^{-\lambda_2 - \theta_2 y} \\ &= e^{-\lambda_1 - \lambda_2 - \theta_1 z} \sum_{y=0}^{\infty} (\lambda_1, \theta_1)_{z+y}\, (\lambda_2, \theta_2)_y\, e^{-(\theta_1+\theta_2)y}, \end{aligned} \tag{5.18}$$

for any value of $z \in \mathbb{Z}$, where

$$(\lambda, \theta)_x = \frac{\lambda(\lambda + x\theta)^{x-1}}{x!}.$$

Similar to Section 5.2, we set lower limits for θ_1 and θ_2 to ensure that there are at least five class of non-zero probabilities at both tails when $\theta_1 < 0$ or $\theta_2 < 0$. Thus, we set

$$\max(-1, -\lambda_1/m_1) < \theta_1 < 1,$$

$$\max(-1, -\lambda_2/m_2) < \theta_2 < 1,$$

where $m_1,\ m_2 \geq 4$ are the largest positive integer in which $\lambda_1 + m_1\theta_1 > 0$ and $\lambda_2 + m_2\theta_2 > 0$. Therefore, for any $z > m_1$ when $\theta_1 < 0$, or $z < -m_2$ when $\theta_2 < 0$, we have

$$f_{\text{GPD}}(z|\lambda_1, \lambda_2, \theta_1, \theta_2) = 0.$$

It is difficult to simplify the probability defined in (5.18) in a compact form. We generally denote this distribution by $\text{GPD}(\lambda_1, \lambda_2, \theta_1, \theta_2)$ in which it can have the following particular forms when some of the parameters are zero.

$$\text{GPD}(\lambda_1, 0, \theta_1, 0) = \text{GP}(\lambda_1, \theta_1) = \frac{\lambda_1(\lambda_1 + \lambda_1 z)^{z-1}}{z!} e^{-\lambda_1 - \theta_1 z},$$

$$\mathrm{GPD}(0,\lambda_2,0,\theta_2) = \frac{\lambda_2(\lambda_2-\lambda_2 z)^{-z-1}}{(-z)!}e^{-\lambda_2+\theta_2 z},$$

$$\mathrm{GPD}(\lambda_1,\lambda_2,0,0) = \mathrm{PD}(\lambda_1,\lambda_2) = e^{-\lambda_1-\lambda_2}(\frac{\lambda_1}{\lambda_2})^{z/2}\,\mathrm{I}_{|z|}(2\sqrt{\lambda_1\lambda_2}).$$

It can be seen that the third case is the PD distribution given in Chapter 3.

5.3.1.1 Cumulant generating function

Let us assume $\psi(\beta) = \sum_{i=1}^{\infty}\beta^k\, L_k/k!$, where L_k is the kth cumulant of the random variable Z, is the cumulant generating function of the GPD distribution. Using the moment generating function and probability generating function of the variable Z, we are able to obtain cumulants as follows. Consider the pgf of the random variable $X \sim G_1(u)$, and $Y \sim G_2(u^{-1})$, are given by

$$\mathrm{G}_1(u) = e^{\lambda_1(t_1-1)}, \qquad \text{where } t_1 = ue^{\theta_1(t_1-1)},$$

$$\mathrm{G}_2(u^{-1}) = e^{\lambda_2(t_2-1)}, \qquad \text{where } t_2 = u^{-1}e^{\theta_2(t_2-1)}.$$

Thus, the pgf of a GPD random variable is given by

$$\begin{aligned}\mathrm{G}(u) &= \mathrm{G}_1(u)\mathrm{G}_2(u)\\ &= e^{\lambda_1(t_1-1)}e^{\lambda_2(t_2-1)}\\ &= e^{\lambda_1(t_1-1)+\lambda_2(t_2-1)}. \qquad (5.19)\end{aligned}$$

By replacing t_1, t_2 and u by e^{T_1}, e^{T_2} and e^{β}, respectively in (5.19), and taking logarithm, the cumulant generating function, $\psi(\beta)$, of $Z = X - Y$ can be obtained as follows,

$$\begin{aligned}\psi(\beta) &= \lambda_1(e^{\mathrm{T}_1}-1)+\lambda_2(e^{\mathrm{T}_2}-1)\\ &= \frac{(\mathrm{T}_1-\beta)\lambda_1}{\theta_1}+\frac{(\mathrm{T}_2+\beta)\lambda_2}{\theta_2}, \qquad (5.20)\end{aligned}$$

where $T_1 = \beta + \theta_1(e^{T_1} - 1)$ and $T_2 = -\beta + \theta_2(e^{T_2} - 1)$. Using cumulant generating function, we are able to obtain cumulants for the GPD distribution by differentiating w.r.t β, θ_1 and θ_2.

Let us start by differentiating T_1 and T_2 with respect to β, θ_1 and θ_2 as follows

$$\frac{\partial T_1}{\partial \beta} = (1 - \theta_1 e^{T_1})^{-1} \quad \text{and} \quad \theta_1 \frac{\partial T_1}{\partial \theta_1} = -1 + (1 - \theta_1)\frac{\partial T_1}{\partial \beta}, \tag{5.21}$$

$$\frac{\partial T_2}{\partial \beta} = -(1 - \theta_2 e^{T_2})^{-1} \quad \text{and} \quad \theta_2 \frac{\partial T_2}{\partial \theta_2} = -1 - (1 - \theta_2)\frac{\partial T_2}{\partial \beta}. \tag{5.22}$$

Next, we partially differentiate $\psi(\beta)$ w.r.t β, θ_1 and θ_2. Furthermore, to simplify $\psi(\beta)/\partial\beta$, we can eliminate $\partial T_1/\partial\beta$ using (5.21)-(5.22) and obtain following equations

$$\theta_1 \frac{\partial \psi(\beta)}{\partial \theta_1} + \psi(\beta) = \lambda_1 \frac{\partial T_1}{\partial \theta_1} + \frac{(T_2 + \beta)\lambda_2}{\theta_2}, \tag{5.23}$$

$$\theta_2 \frac{\partial \psi(\beta)}{\partial \theta_2} + \psi(\beta) = \frac{(T_1 - \beta)\lambda_1}{\theta_1} + \lambda_2 \frac{\partial T_2}{\partial \theta_2}, \tag{5.24}$$

$$(1 - \theta_1)(1 - \theta_2)\frac{\partial \psi(\beta)}{\partial \beta} - (1 - \theta_2)\lambda_1 + (1 - \theta_1)\lambda_2 = (1 - \theta_2)\lambda_1 \frac{\partial T_1}{\partial \theta_1} - (1 - \theta_1)\lambda_2 \frac{\partial T_2}{\partial \theta_2}, \tag{5.25}$$

then add (5.23), (5.24) and (5.25), which results in

$$\begin{aligned}(2 - \theta_2)\lambda_1 \frac{\partial T_1}{\partial \theta_1} + \theta_1 \lambda_2 \frac{\partial T_2}{\partial \theta_2} = \theta_1 \frac{\partial \psi(\beta)}{\partial \theta_1} + \theta_2 \frac{\partial \psi(\beta)}{\partial \theta_2} \\ + (1 - \theta_1)(1 - \theta_2)\frac{\partial \psi(\beta)}{\partial \beta} \\ + \psi(\beta) - (1 - \theta_2)\lambda_1 + (1 - \theta_1)\lambda_2.\end{aligned} \tag{5.26}$$

The partial differentiations of $\psi(\beta)$ w.r.t λ_1 then θ_1, and λ_2, then θ_2 given in (5.27) and (5.28), are used to eliminate $\partial T_1/\partial\theta_1$ and $\partial T_2/\partial\theta_2$ in (5.26).

$$\frac{\partial^2 \psi(\beta)}{\partial \theta_1 \partial \lambda_1} = -\frac{1}{\theta_1}\frac{\partial \psi(\beta)}{\partial \lambda_1} + \frac{1}{\theta_1}\frac{\partial T_1}{\partial \theta_1}, \tag{5.27}$$

$$\frac{\partial^2\psi(\beta)}{\partial\theta_2\partial\lambda_2} = -\frac{1}{\theta_2}\frac{\partial\psi(\beta)}{\partial\lambda_2} + \frac{1}{\theta_2}\frac{\partial T_2}{\partial\theta_2}. \tag{5.28}$$

On the elimination of $\partial T_1/\partial\theta_1$ and $\partial T_2/\partial\theta_2$, we obtain the following relation

$$\begin{aligned}(2-\theta_2)\lambda_1\left[\theta_1\frac{\partial^2\psi(\beta)}{\partial\theta_1\partial\lambda_1} + \frac{\partial\psi(\beta)}{\partial\lambda_1}\right] + \theta_1\lambda_2\left[\theta_2\frac{\partial^2\psi(\beta)}{\partial\theta_2\partial\lambda_2} + \frac{\partial\psi(\beta)}{\partial\lambda_2}\right] = \\ \theta_1\frac{\partial\psi(\beta)}{\partial\theta_1} + \theta_2\frac{\partial\psi(\beta)}{\partial\theta_2} + (1-\theta_1)(1-\theta_2)\frac{\partial\psi(\beta)}{\partial\beta} + \psi(\beta) - (1-\theta_2)\lambda_1 + (1-\theta_1)\lambda_2.\end{aligned} \tag{5.29}$$

We substitute the value of the cumulant generating function by $\psi(\beta) = \sum_k \beta^k L_k/k!$ in (5.26), and obtain the following recurrence relation for the cumulants of the probability distribution of the random variable Z.

$$\begin{aligned}(1-\theta_1)(1-\theta_2)L_{k+1} =& (2-\theta_2)\lambda_1\theta_1\frac{\partial^2 L_k}{\partial\theta_1\partial\lambda_1} + (2-\theta_2)\lambda_1\frac{\partial L_k}{\partial\lambda_1} + \theta_1\theta_2\lambda_2\frac{\partial^2 L_k}{\partial\theta_2\partial\lambda_2} \\ &+ \theta_1\lambda_2\frac{\partial L_k}{\partial\lambda_2} - \theta_1\frac{\partial L_k}{\partial\theta_1} - \theta_2\frac{\partial L_k}{\partial\theta_2} - L_k,\end{aligned} \tag{5.30}$$

where $L_1 = [\lambda_1/(1-\theta_1)] - [\lambda_2/(1-\theta_2)]$. The expression for the other cumulants of the GPD distribution can be obtained by using (5.30) recursively for $k = 1, 2, 3, \ldots$. For example, the second cumulant, L_2, is obtained as follows

$$\begin{aligned}(1-\theta_1)(1-\theta_2)L_2 =& (2-\theta_2)\lambda_1\left[\frac{\theta_1}{(1-\theta_1)^2} + \frac{1}{1-\theta_1}\right] - \theta_1\lambda_2\left[\frac{\theta_2}{(1-\theta_2)^2} + \frac{1}{1-\theta_2}\right] \\ &- \frac{\theta_1\lambda_1}{(1-\theta_1)^2} + \frac{\theta_2\lambda_2}{(1-\theta_2)^2} - \frac{\lambda_1}{1-\theta_1} + \frac{\lambda_2}{1-\theta_2} \\ =& \frac{(2-\theta_2)\lambda_1}{(1-\theta_1)^2} - \frac{\theta_1\lambda_2}{(1-\theta_2)^2} - \frac{\lambda_1}{(1-\theta_1)^2} + \frac{\lambda_2}{(1-\theta_2)^2} \\ =& \frac{(1-\theta_2)\lambda_1}{(1-\theta_1)^2} + \frac{(1-\theta_1)\lambda_2}{(1-\theta_2)^2} \\ =& \frac{(1-\theta_2)^3\lambda_1 + (1-\theta_1)^3\lambda_2}{(1-\theta_1)^3(1-\theta_2)^3}\end{aligned}$$

$$=\frac{\lambda_1}{(1-\theta_1)^3}+\frac{\lambda_2}{(1-\theta_2)^3}.$$

Now, consider the case of $\theta_1=\theta_2=\theta$. We can show that the GPD distribution discussed by Consul and Famoye (2006) is a special case of the GPD distribution we introduced here. By substituting θ in (5.20), the cumulant generating function is obtained as

$$\psi(\beta)=\frac{(T_1-\beta)\lambda_1}{\theta}+\frac{(T_2+\beta)\lambda_2}{\theta}. \tag{5.31}$$

In this case, to obtain cumulants for the random variable Z, we only need to differentiate T_1 and T_2 w.r.t β and θ which are given as follows

$$\frac{\partial T_1}{\partial \beta}=(1-\theta e^{T_1})^{-1} \quad \text{and } \theta\frac{\partial T_1}{\partial \theta}=-1+(1-\theta)\frac{\partial T_1}{\partial \beta}, \tag{5.32}$$

$$\frac{\partial T_2}{\partial \beta}=(1-\theta e^{T_2})^{-1} \quad \text{and } \theta\frac{\partial T_2}{\partial \theta}=-1-(1-\theta)\frac{\partial T_2}{\partial \beta}. \tag{5.33}$$

Therefore, the differentiation of $\psi(\beta)$ w.r.t β and θ in (5.23)- (5.25) reduces to

$$\theta\frac{\partial \psi(\beta)}{\partial \theta}+\psi(\beta)=\lambda_1\frac{\partial T_1}{\partial \theta}+\lambda_2\frac{\partial T_2}{\partial \theta}, \tag{5.34}$$

$$(1-\theta)\frac{\partial \psi(\beta)}{\partial \beta}-\lambda_1+\lambda_2=\lambda_1\frac{\partial T_1}{\partial \theta}-\lambda_2\frac{\partial T_2}{\partial \theta}. \tag{5.35}$$

In the next step, (5.34) and (5.35) are added and result in

$$2\lambda_1\frac{\partial T_1}{\partial \theta_1}=(1-\theta)\frac{\partial \psi(\beta)}{\partial \beta}+\theta\frac{\partial \psi(\beta)}{\partial \theta}+\psi(\beta)-\lambda_1+\lambda_2. \tag{5.36}$$

To eliminate $\partial T_1/\theta$, we then differentiate $\psi(\beta)$ first w.r.t λ_1 and then θ and obtain

$$\frac{\partial^2 \psi(\beta)}{\partial \theta \partial \lambda_1}=-\frac{1}{\theta}\frac{\partial \psi(\beta)}{\partial \lambda_1}+\frac{1}{\theta}\frac{\partial T_1}{\partial \theta}, \tag{5.37}$$

and substitute the resulting equation in (5.36),

$$2\lambda_1\theta\frac{\partial^2\psi(\beta)}{\partial\theta\partial\lambda_1}+2\lambda_1\frac{\partial\psi(\beta)}{\lambda_1}=\theta\frac{\partial\psi(\beta)}{\partial\theta}+(1-\theta)\frac{\partial\psi(\beta)}{\partial\beta}+\psi(\beta)-\lambda_1+\lambda_2. \quad (5.38)$$

Similarly, by replacing $\psi(\beta)$ by its definition in (5.38), the following recurrence is obtained.

$$(1-\theta)L_{k+1}=2\theta\lambda_1\frac{\partial^2 L_k}{\partial\theta\partial\lambda_1}+2\lambda_1\frac{\partial L_k}{\partial\lambda_1}-\theta\frac{\partial L_k}{\partial\theta}-L_k, \quad (5.39)$$

where $L_1=(\lambda_1-\lambda_2)/(1-\theta)$. Therefore, the first four cumulants are given by

$$L_1=\frac{\lambda_1}{1-\theta}-\frac{\lambda_2}{1-\theta},$$

$$L_2=\frac{\lambda_1}{(1-\theta)^3}+\frac{\lambda_2}{(1-\theta)^3},$$

$$L_3=\frac{(\lambda_1-\lambda_2)(1+2\theta)}{(1-\theta)^5},$$

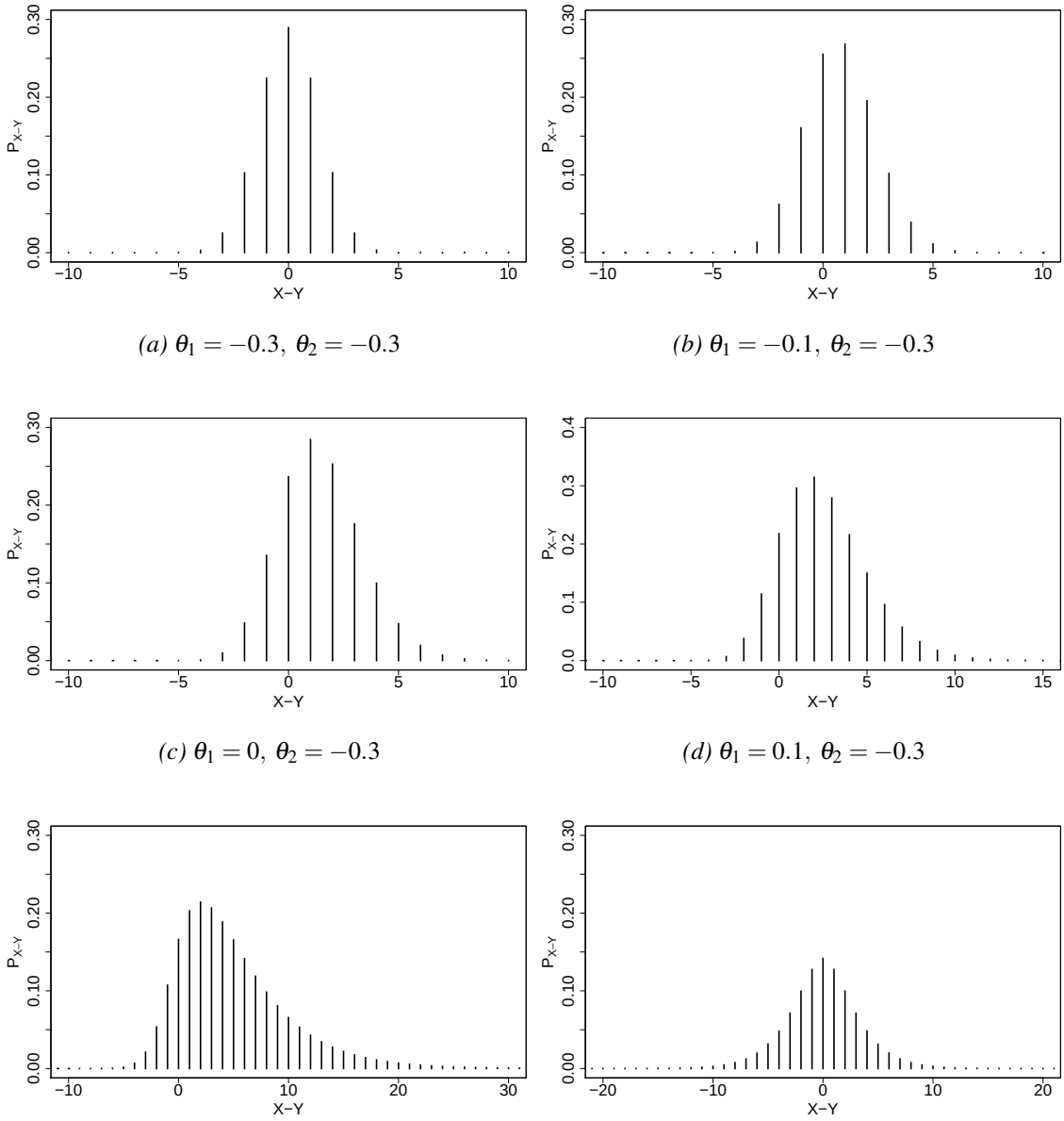
$$L_4=\frac{(\lambda_1+\lambda_2)(1+8\theta+6\theta^2)}{(1-\theta)^7},$$

Consul and Famoye (2006, Chapter 9) has obtained the coefficients of skewness and kurtosis for the random variable Z, for the case of $\theta_1=\theta_2=\theta$ as follows

$$\beta_1=\frac{(\lambda_1-\lambda_2)^2}{(\lambda_1+\lambda_2)^3}\frac{(1+2\theta)^2}{(1-\theta)} \quad \text{and} \quad \beta_2=3+\frac{1+8\theta+6\theta^2}{(\lambda_1+\lambda_2)(1-\theta)}. \quad (5.40)$$

5.3.2 Graphical presentation of GPD distribution models

The behaviour of the GPD distribution for a random variable Z with different values of the λ_1, λ_2, θ_1 and θ_2 is illustrated in this part. To study only the effect of θ_1 and θ_2 on the GPD distribution, first, we assume $\lambda_1=\lambda_2=\lambda$ and change the values of θ_1 and



(a) $\theta_1 = -0.3,\ \theta_2 = -0.3$

(b) $\theta_1 = -0.1,\ \theta_2 = -0.3$

(c) $\theta_1 = 0,\ \theta_2 = -0.3$

(d) $\theta_1 = 0.1,\ \theta_2 = -0.3$

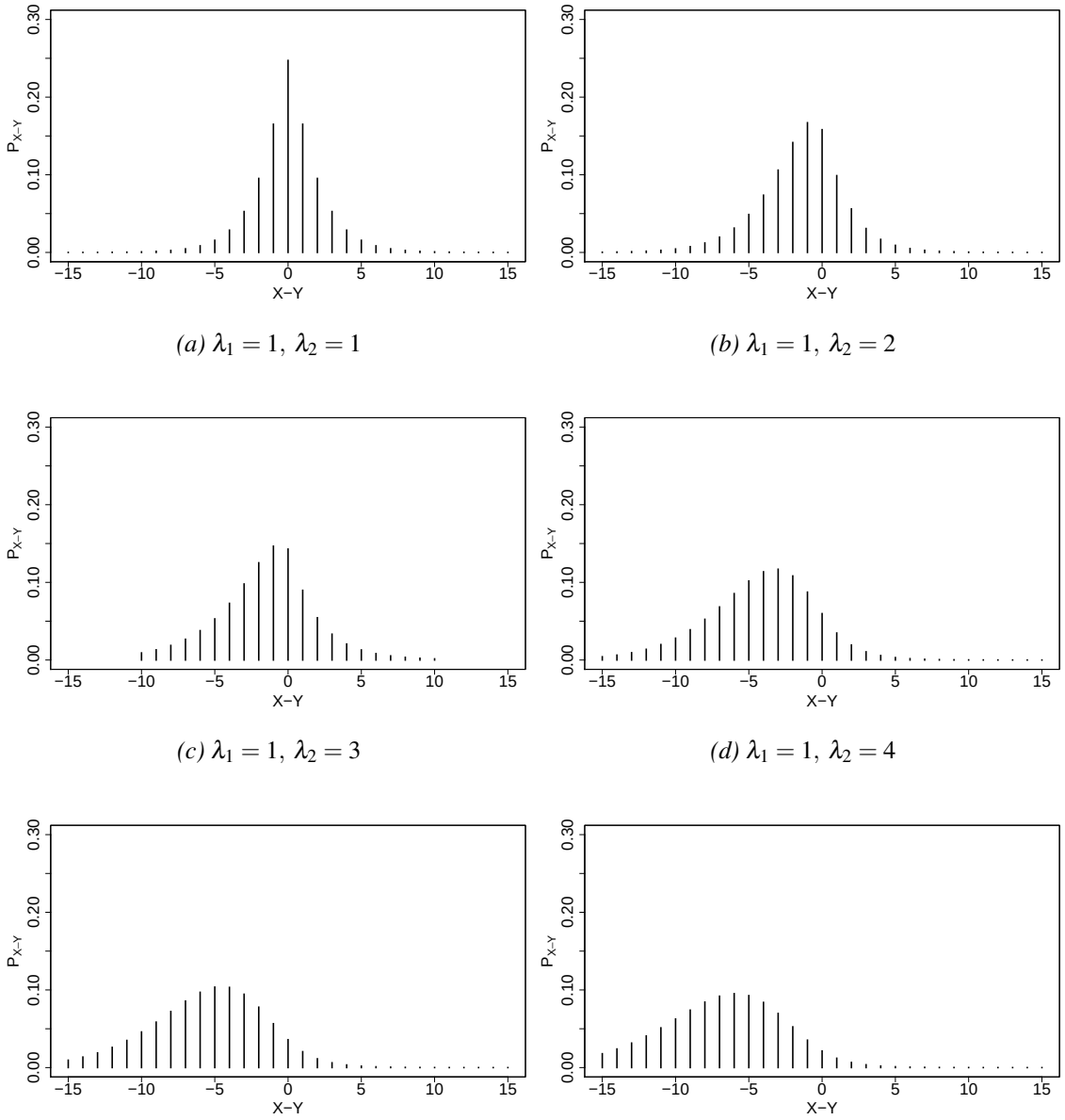
(e) $\theta_1 = 0.3,\ \theta_2 = 0$

(f) $\theta_1 = 0.3,\ \theta_2 = 0.3$

Figure 5.3: The GP distribution for $\lambda_1 = \lambda_2 = 2$

θ_2 (Figure 5.3). In general, the distribution has a wider span over Z as either θ_1 or θ_2 increases. The distribution also becomes short tailed as either of the parameters decrease. For a fixed λ, the sign of the of $[\lambda/(1-\theta_1)] - [\lambda/(1-\theta_2)]$ determines the skewness of the GPD distribution.

Now, let us assume $\theta_1 = \theta_2 = \theta$. It can be seen from Figure 5.4 that when $\lambda_1 < \lambda_2$, the GPD distribution become a skewed distribution to the left. For any given value of θ, when λ_1 decreases, the GPD tends to have a shorter tail. Assuming the same type of



(a) $\lambda_1 = 1,\ \lambda_2 = 1$

(b) $\lambda_1 = 1,\ \lambda_2 = 2$

(c) $\lambda_1 = 1,\ \lambda_2 = 3$

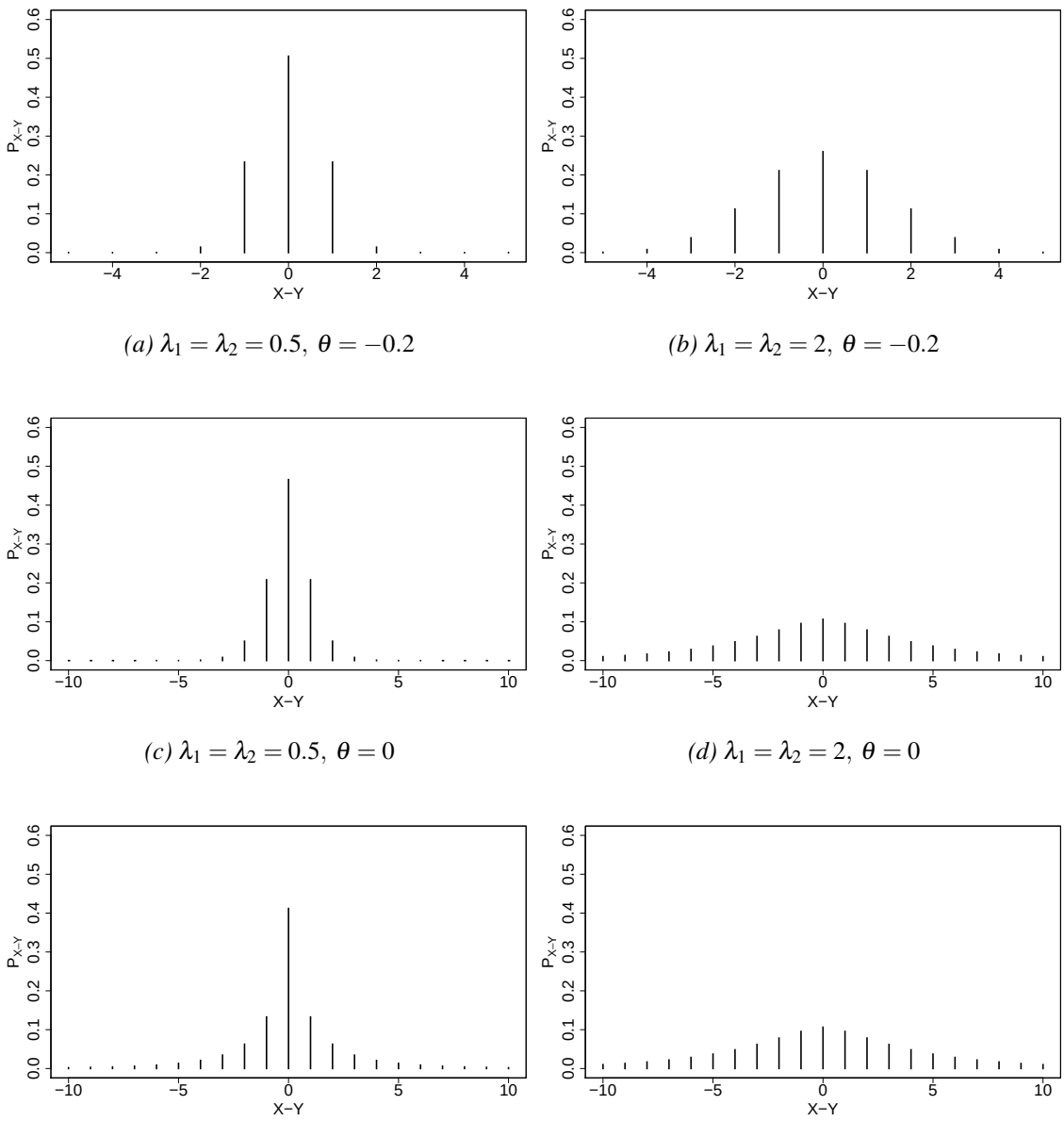
(d) $\lambda_1 = 1,\ \lambda_2 = 4$

(e) $\lambda_1 = 1,\ \lambda_2 = 5$

(f) $\lambda_1 = 1,\ \lambda_2 = 6$

Figure 5.4: The GP distribution for $\theta = 0.4$

symmetry in (3.4) in Section 3.2.1 holds for the GPD distribution, thus for a given value of θ the GPD distribution tends to be skewed to the right when $\lambda_1 > \lambda_2$ (not illustrated). Similar to the GP distribution, for any given values of λ_1 and λ_2, the form of the GPD is substantially altered by varying θ. Figure 5.5 suggests that an increase in the value of θ increases the variance of the distribution and both the tails of the GPD distribution become long and thin. For any value of λ, when θ becomes smaller, most of the mass becomes concentrated on $Z = 0$.



(a) $\lambda_1 = \lambda_2 = 0.5,\ \theta = -0.2$

(b) $\lambda_1 = \lambda_2 = 2,\ \theta = -0.2$

(c) $\lambda_1 = \lambda_2 = 0.5,\ \theta = 0$

(d) $\lambda_1 = \lambda_2 = 2,\ \theta = 0$

(e) $\lambda_1 = \lambda_2 = 0.5,\ \theta = 0.5$

(f) $\lambda_1 = \lambda_2 = 2,\ \theta = 0.5$

Figure 5.5: The GPD distribution for different values of $\lambda_1 = \lambda_2 = \lambda$ and θ

5.3.3 Application: a model for index change

The GPD distribution is useful in fitting over-dispersed as well as under-dispersed integer data. We outline below a potential application of the GPD distribution for modelling UHF data. The application in this chapter is based on the assumption that only the difference of two variables are observed.

Let $Z_i \sim \text{GPD}(\lambda_{1,i}, \lambda_{2,i}, \theta_{1,i}, \theta_{2,i})$ index change, where $i = 1, \ldots, n$ and n is the number of index changes in a given trading day, be the response variable. We start by introducing

the zero inflated version of the GPD (ZGPD) distribution, to capture the excess of zeros in the FTSE100 data set, as follows

$$f_{\text{ZGPD}}(z|\lambda_1,\lambda_2,\theta_1,\theta_2,p) = \begin{cases} p+(1-p)f_{\text{GPD}}(z \mid \lambda_1,\ \lambda_2,\ \theta_1,\ \theta_2) & \text{if } z=0, \\ (1-p)f_{\text{GPD}}(z \mid \lambda_1,\ \lambda_2,\ \theta_1,\ \theta_2) & \text{if } z\neq 0, \end{cases} \tag{5.41}$$

for $z \in \mathbb{Z}$, where $f_{\text{GPD}}(z|\lambda_1,\lambda_2)$ is defined by (5.18), $\lambda_1,\lambda_2,\theta_1$ and θ_2 are given by (5.42)-(5.45), and $p \in (0,1)$ is the proportion of extra zeros. The model parameters, $\lambda_1,\ \lambda_2,\ \theta_1$ and θ_2, may adopt a similar structure to the ZPD model in Chapter 3. Therefore, we may set

$$\log(\lambda_1) = \mathbf{x}\alpha, \tag{5.42}$$

$$\log(\lambda_2) = \mathbf{x}\beta, \tag{5.43}$$

$$\log(1-\theta_1) = \mathbf{x}\gamma, \tag{5.44}$$

$$\log(1-\theta_2) = \mathbf{x}\phi, \tag{5.45}$$

where $\alpha,\ \beta,\ \gamma$ and ϕ are vectors of parameters of the ZGPD model of size $p\times 1$, $\mathbf{x}$ a the matrix of covariates of size $n\times p$. Thus, the likelihood for the ZGPD distribution in terms of the model parameters, (5.42)-(5.45), can be obtained as follows

$$\begin{aligned} \mathrm{L}(\lambda_1,\lambda_2,\theta_1,\theta_2,p) &= \prod_{i=1}^{n}\left(p\, I_{\{0\}} + (1-p)f_{\text{GP}}(z_i|\lambda_{1,i},\ \lambda_{2,i},\ \theta_{1,i},\ \theta_{2,i})\right) \\ &= \prod_{i=1}^{n}\left(p\, I_{\{0\}} + (1-p)f_{\text{GPD}}\left[z_i|\ \exp\left(\mathbf{x}_i\,\alpha\right),\ \exp\left(\mathbf{x}_i\,\beta\right),\ 1-\exp\left(\mathbf{x}_i\,\gamma\right),\ 1-\exp\left(\mathbf{x}_i\,\phi\right)\right]\right), \end{aligned} \tag{5.46}$$

where $\mathbf{x}_i$ is the ith row of the covariance matrix and $I_{\{0\}}$ is given by (3.7) in Section 3.3.1. The model can be fitted to the FTSE100 index changes using Bayesian methods exactly in the same way as in Section 3.3.1. Similar to the case of the ZPD model, the posterior distribution will not be explicitly known, therefore we may need to use MCMC methods in order to generate samples from the posterior distribution. However, in order to evaluate the likelihood in (5.46), currently there are no computational algorithms available. Thus,

the development of an efficient and suitable algorithm for this purpose may be considered as a possible extension.

5.4 Summary

The Poisson distribution has been widely used in different fields of applications such as finance and medicine. The underlying assumption for application of the Poisson distribution is the independency of the successive events. However, this assumption might be violated in the real life, and therefore, the use of the Poisson model leads to the overestimation or underestimation of the reality. The generalised Poisson (GP) distribution may be a possible way to handle this problem.

In this chapter we looked at the GP distribution proposed by Consul (1989) with a second parameter which can account for the over- or under-dispersion in the data and discussed its characteristics. In order to overcome the problem of underestimation of the Poisson difference model, we introduced the generalised Poisson difference (GPD) distribution with four parameters. We obtained the probability generating function and the cumulant generating function for the GPD distribution and explored the theoretical characteristics of such distribution. We established a recurrence relation for all the cumulants of a GPD random variable and derived the first two cumulants. Moreover, we showed that the GPD distribution suggested by Consul and Famoye (2006) is a special case of our GPD distribution.

Furthermore, we suggested a possible application of the GPD distribution for modelling UHF data and also introduced a zero inflated version of the GPD distribution. Due to computational difficulties the model was left at the formal level. There is, however, a great potential for using this distribution to model UHF data. The development of suitable algorithms for this purpose may be considered as a possible future work.

Chapter 6

Conclusion

6.1 Summary

The availability of (ultra) high-frequency (UHF/HF) data on transactions, in the recent decade, has revolutionized data processing and statistical modelling techniques in finance and introduced new theoretical and computational challenges. Research in a variety of issues related to trading process and market microstructure studies has become possible by the advent of UHF/HF data sets. However, the unique characteristics of UHF data, such as the discrete structure of price change and unequally spaced time intervals, has introduced a new challenge to both statistical and financial studies.

Our concern in this study was to develop an appropriate Bayesian methodology for modelling the discrete price change. Two days of the FTSE100 index futures contract were available for this study (March 25 and 26, 2008). One day of the two trading days was used for the purpose of modelling and the other was used in order to validate the fitted models.

We started by looking at the ADS model suggested by Rydberg and Shephard (2000, 2003) in Chapter 2. The ADS model considered price change as a combination of three components: an indicator for price activity (A), the direction of price change (D) and the size of price change (S). This chapter adopted a Bayesian framework for inference and prediction and implemented using MCMC methods. Furthermore, we investigated the effect of the previous index change, the volume of the previous transaction and the time

duration between two consecutive transactions (time duration) on the probability of index change at the next transaction for each of the components of price change. Our results from the analysis of one day's transactions of FTSE100 suggested the existence of high liquidity in the market.

However, the ADS model has some limitations. For example, our interpretation from the model is only valid when results from all the three components are available, or modelling each component of price change separately leads to us ignoring the simultaneous effect of one component on another. Thus, our main concern in this study was dealing with price change as a single process.

In Chapter 3 we introduced the zero inflated Poisson difference (ZPD) distribution and its Bayesian application for modelling the FTSE100 index change. The Poisson difference distribution provided an integer range for index change and, by considering its zero inflated version, we were able to consider the excess of zeros in the model. We assessed the effect of the previous index change, the volume of the previous transaction and the time duration between two consecutive transactions on the index change for the next transaction. Our results indicated that index change was significantly affected by the covariates in the model.

In order to assess the performance of the MCMC fitting algorithm for both the ADS and the ZPD models in Chapters 2 and 3, Gelman and Rubin's convergence diagnostics was implemented and suggested that we could assume the convergence of the MCMC chains. In addition, one-step ahead predictions for the next trading day were obtained by assuming that time duration and volume were known. Based on the probability integral transform (PIT) modified for the case of discrete random variables, we show that the ADS and ZPD models are both capable of explaining well the observed distribution of index change. However, a comparison between the deviance information criterion (DIC) of the ADS and the ZPD models suggested that the ZPD model performed significantly better than the ADS model.

In Chapter 4 we introduced a dynamic zero inflated Poisson difference model with parameters varying over time. In this context, we assumed that index change was influ-

enced by latent processes. Unlike the (off-line) ZPD model in Chapter 3, the DZPD model enabled us to update our inference from the data as new transactions took place. We referred to this type of analysis as online analysis data processing. MCMC methods were not computationally feasible for this purpose. Instead, the analysis was carried out using sequential Monte Carlo methods. For the purpose of model comparison we introduced a sequential form of the DIC, which suggested that the model performs better if time duration is included as a covariate in the model. Finally, we obtained one-step ahead predictions. Results from the PIT suggested that the marginal predictive distribution is able to explain the behaviour of the data well. However, the model slightly underestimated the number of zero index changes.

In our modelling, we noticed that the ZPD distribution had a tendency to underestimate the extreme values in the tail of the distribution of index change. Therefore, to add more flexibility to the tails of the ZPD distribution, we introduced the GPD distribution with four parameters and its zero inflated version in Chapter 5. We obtained the probability generating function and the cumulant generating function for the GPD distribution and established a recurrence relation for all the cumulants. Furthermore, we outlined a possible model for index change with the ZGPD distribution. However, we left the model at the formal level due to computational difficulties.

6.2 Further developments and outlook

In this thesis we investigated the problem of modelling price change as a single process. While the ZPD distribution has enabled us to model price change as an integer-valued variable and provided the first step of analysing price change as a single process, there are still opportunities for extending our study. We list below some of the potential extensions. Sections 6.2.1-6.2.3 provide possible extensions that apply to both off-line and online modelling. Section 6.2.4 suggests alternative choices for the state equation in the dynamic model. Improvements for some of the computational algorithms implemented in this thesis has been provided in Section 6.2.5, followed by possible extensions from a financial perspective in Section 6.2.6.

Let us point out that the ZPD model in this work was fitted to index change using the same framework as suggested by Karlis and Ntzoufras (2009). However, it would be interesting to investigate more general frameworks, such as Generalised additive models for location, scale and shape (GAMLSS) (Rigby and Stasinopoulos, 2005) in which the model is characterised by shape parameters, in addition to the location and scale parameters. Developing an appropriate methodology for setting the ZPD distribution in a GAMLSS class can be considered as an alternative modelling framework for a future study.

6.2.1 Bivariate models for price change and time duration

Joint modelling of price change and time duration has been the main interest of many studies (Russell and Engle, 1998; Darolles et al., 2000; McCulloch and Tsay, 2001). However, earlier studies either limited price change over a finite range or had to decompose price change. An extension to the ZPD model (in off-line as well as online applications) is to jointly model price change and time duration. Let P_{t_i} be the transaction price, $Z_i = P_{t_i} - P_{t_{i-1}}$ the ith price change, and $\Delta t_i = t_i - t_{i-1}$ the time duration between two consecutive transactions, for $i = 1, \ldots, n$, where n is the number of price changes in a given trading day.

A ZPD duration model will be concerned with the joint analysis of $(\Delta t_i,\, Z_i)$. We may propose a decomposition[1] of the conditional joint distribution of $(\Delta t_i,\, Z_i)$ as follows

$$f(\Delta t_i, Z_i | F_{i-1}) = f(Z_i | \Delta t_i, F_{i-1}) f(\Delta t_i | F_{i-1}). \tag{6.1}$$

where F_{i-1} denotes all information available up to time t_{i-1} which may contain all possible covariates. The decomposition in (6.1) simplifies the modelling task by allowing us to specify suitable models for the conditional distribution of Z_i and the distribution of Δt_i separately. There are many ways to specify models for distributions in (6.1). A proper specification might depend on the data under study. In the two following sections we

[1] A similar decomposition has been proposed by McCulloch and Tsay (2001) in which price change was decomposed into two components: direction and size. In that model other components, such as number of transactions within Δt_i, were also considered.

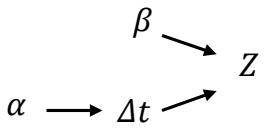


Figure 6.1: A ZPD duration model jointly models the duration and index change.

briefly outline model specification for off-line and online frameworks.

6.2.1.1 Off-line modelling

The ZPD duration model in an off-line framework may be set by assigning a generalised linear model (GLM) to Δt_i and a ZPD model to Z_i. In the ZPD duration model let α and β denote the parameter vectors of appropriate sizes, corresponding to time duration and price change, respectively. Figure 6.1 illustrates the conditional structure of price change on time duration and the parameter vector β. Depending on the data under study, for example, we may consider

$$\Delta t_i \sim \mathrm{P}(\theta_i), \tag{6.2}$$

$$\log(\theta) = \mathbf{X}\alpha, \tag{6.3}$$

where $\mathrm{P}(\theta_i)$ is a Poisson distribution with parameter θ_i such that $\theta = (\theta_1, \ldots, \theta_n)^{\mathrm{T}}$, $\mathbf{X}$ is a matrix of covariates. Then, we may fit a ZPD model to the price change as described in Chapter 3.

Furthermore, the Bayesian analysis of the joint distribution may be undertaken by obtaining the posterior distribution. Using the decomposition in (6.1), the posterior distribution also can be decomposed as follows

$$\begin{aligned} f_{\text{post}}(\alpha,\beta|Z,\Delta t) &\propto f_{\text{like}}(\Delta t, Z|\alpha,\beta)\, f_{\text{prior}}(\alpha)\, f_{\text{prior}}(\beta) \\ &= f_{\text{like}}(Z|\Delta t,\beta)\, f_{\text{like}}(\Delta t|\alpha)\, f_{\text{prior}}(\alpha)\, f_{\text{prior}}(\beta) \\ &\propto f_{\text{post}}(\beta|Z,\Delta t)\, f_{\text{post}}(\alpha|\Delta t), \end{aligned} \tag{6.4}$$

where $f_{\text{post}}(\alpha,\beta|Z,\Delta t)$ is the joint posterior distribution, $f_{\text{like}}(\Delta t, Z|\alpha,\beta)$, $f_{\text{like}}(Z|\Delta t,\beta)$

and $f_{\text{like}}(\Delta t|\alpha)$ are the joint likelihood for price change and time duration, the likelihood function for price change and the likelihood function for time duration, respectively. Therefore, further work will be involved in obtaining the posterior distribution of each of the components, Δt and Z, and developing an MCMC algorithm in order to sample from the posterior distributions.

6.2.1.2 Online modelling

In order to jointly model $(\Delta t_i, Z_i)$ in a dynamic structure, we may assign a dynamic model to Δt_i. Consider the hidden Markov model described in Chapter 4 and the decomposition shown in (6.1). Let $\{\Delta t_i\}_{i\geq 1}$ be the observed process and $\{Y_i\}_{i\geq 0}$ be a hidden Markov process with its initial density $Y_0 \sim g(y_0)$. Then we can set

Observation equation

$$\Delta t_i|(Y_i = y_i) \sim \mathrm{P}(\theta_i), \tag{6.5}$$

State equation

$$\log(\theta_i) = h_\psi(y_i), \tag{6.6}$$

$$Y_i|(Y_{i-1} = y_{i-1}) \sim g_\psi(y_i|y_{i-1}), \tag{6.7}$$

where $\mathrm{P}(\theta_i)$ is a Poisson distribution with parameter θ_i, $h(.)$ is any function of y_i, $g(y_i|y_{i-1})$ denotes the probability density associated with moving from y_{i-1} to y_i and ψ is a vector of unknown static parameters of the model. Consequently, we can use the DZPD model in order to characterise price change. Choosing an appropriate state equation and a suitable particle filter algorithm in order to update both models for time duration and price change can be considered as the next stage of dynamic modelling of price change.

6.2.2 Modelling the proportion of excess of zeros

We assumed in both off-line and online models in Chapters 3 and 4, the probability of the excess of zero values, p, as a fixed parameter over a trading day. In practice, p might be influenced by some of the covariates (e.g. price change, the volume of transaction and

time duration). Therefore, p may be considered as a function of covariates in both off-line and online applications as follows

$$\log\left(\frac{p_i}{1-p_i}\right) = h_\psi(\mathbf{x}_i), \quad \text{for } i = 1,\ldots,n, \tag{6.8}$$

where $\mathbf{x}_i$ is the ith row of a matrix of covariates, ψ is a vector of parameters and n is the number of index changes within a given trading day.

6.2.3 Extending the analysis to more than one trading day

In this study we modelled the index changes of one trading day. In practice, we would be interested in extending our analysis to more than one day's transactions. In this case, we may model each trading day separately or put the data from all trading days together into one long series and then fit a single model. The first option does not combine information from all trading days and also may not be computationally feasible over a long period of time. On the other hand, in the second option we fit the same model to several days of data, thus ignoring the dynamics between trading days. We can extend our analysis in order to address this problem using off-line or online models.

6.2.3.1 Hierarchical models

For models with static parameters, such as the (off-line) ZPD model in Chapter 3, hierarchical modelling can be used in order to extend the analysis to more than one trading day. Consider the decomposition of the joint distribution of $(\Delta t_i, Z_i)$ shown in (6.1). In a Bayesian hierarchical modelling framework we may set the posterior distribution as follows

$$\begin{aligned} f_{\text{post}}(\alpha,\, \beta,\, \gamma,\, \vartheta|Z,\, \Delta t) \propto & f_{\text{like}}(\Delta t,\, Z|\alpha,\, \beta,\, \gamma,\, \vartheta)\, f_{\text{prior}}(\alpha|\, \gamma)\, f_{\text{prior}}(\gamma)\, f_{\text{prior}}(\beta|\vartheta)\, f_{\text{prior}}(\vartheta) \\ = & f_{\text{like}}(Z|\beta,\, \vartheta,\, \Delta t)\, f_{\text{prior}}(\beta|\, \vartheta)\, f_{\text{prior}}(\vartheta) \times \\ & f_{\text{like}}(\Delta t|\alpha,\, \gamma)\, f_{\text{prior}}(\alpha|\, \gamma)\, f_{\text{prior}}(\gamma) \\ \propto & f_{\text{post}}(\beta, \vartheta|Z,\, \Delta t)\, f_{\text{post}}(\alpha,\, \gamma|\Delta t), \end{aligned} \tag{6.9}$$

where γ and ϑ are hyperparameters corresponding to α and β, respectively, in an hierarchical ZPD duration model. Therefore, by conditioning the parameter vectors of the model for n trading days on another set of hyperparameters, we allow for variation in the parameters of the model from one trading day to another.

6.2.3.2 Dynamic models

Having a dynamic structure, the DZPD model in Chapter 4 can be easily extended to the next trading day. In order to avoid the weight degeneracy problem mentioned in Section 4.4.2.3 and initialise the algorithm for the next trading day, we may run an MCMC sampler overnight when the data flow stops. Alternatively, we may implement a smoothing algorithm. The smoothing problem refers to the case when we aim to obtain estimates of the previous states given a block of observations $y_{1:T}$. Further investigation of how to use an MCMC algorithm for this purpose or finding a smoothing algorithm that may be appropriate for our analysis provides another scope for further developing the dynamic modelling of price change.

6.2.4 Alternative state space processes

The dynamic modelling of index change can be extended by considering more complicated processes for the state equation in (4.49)-(4.52) than the nonlinear AR(1) model that we have used. Diffusion and Dirichlet processes have been suggested for analysing high-frequency data in other applications, e.g. physics and genetics, (Fearnhead et al., 2010; Yau et al., 2011). We can consider similar processes in the DZPD model for modelling UHF financial data.

6.2.5 Improvements for computational algorithms

The computational algorithms implemented in this study can be also developed further. The LWF algorithm implemented in Chapter 4 can be replaced by other sequential learning algorithms. There are some new algorithms, for example, the sequential parameter learning algorithm by Polson et al. (2008) which claims that the weight degeneracy is prevented by avoiding reweighting particles. In addition, by considering the above de-

velopments of the DZPD model (modelling time), our algorithm can be improved and made accessible to researchers as well as traders for the purpose of online prediction of the market.

In Chapter 5, in order to apply the GPD distribution, we needed to develop a computational algorithm to evaluate the likelihood function. To obtain a better understanding of the likelihood function, we may need to set up a simulation study. The summation in the probability distribution function of the GPD distribution in (5.18) may be considered as a generalisation of the modified bessel function in (3.3) when $\theta_1, \theta_2 \neq 0$. Therefore, we may be able to evaluate the likelihood function in (5.18) by modifying the algorithm used for computing the modified Bessel function. Furthermore, it would be useful to check the derivatives in a symbolic algebraic package e.g. *Mathematica*.

6.2.6 General remarks

Other interesting open issues relating to various financial aspects of our study may include some of the following:

(i) Assessing the effect of bid-ask spread on price change. Chapter 1 pointed out that in futures markets there is no market maker. Therefore, bid and ask prices may influence the transaction price process in a different way compared to other markets. For this study, bid and ask quotes were not available. However, in the case of the availability of bid and ask quotes, investigating the effect of their dynamics on the price process in the FTSE100 futures market may be considered an interesting area of study in its own right.

(ii) Studying price change process in other futures markets (e.g. DAX futures (German stock index), WTI (West Texas Intermediate) futures).

(iii) Finally, applying the analysis proposed in this study to other types of UHF data such as share prices and foreign exchanges.

Appendix A

ADS model: Gelman and Rubin's convergence diagnostics, and traces of the MCMC chains

Potential scale reduction factors:		
	Point est.	97.5 quantile
β_0	1.00	1.00
β_1	1.00	1.00
β_2	1.00	1.00
Multivariate psrf		1.00

Table A.1: Gelman and Rubin's R statistics for the parameter vector β of model M_{A} in (2.3). Values less than 1.1 suggest that we could assume the convergence of the MCMC chains.

Potential scale reduction factors:		
	Point est.	97.5 quantile
γ_0	1.00	1.00
γ_1	1.00	1.00
Multivariate psrf		1.00

Table A.2: Gelman and Rubin's R statistics for the parameter vector γ of model M_{D} in (2.4). Values less than 1.1 suggest that we could assume the convergence of the MCMC chains.

Potential scale reduction factors:		
	Point est.	97.5 quantile
θ_0	1.00	1.00
θ_1	1.00	1.00
θ_2	1.00	1.00
θ_3	1.01	1.01
Multivariate psrf		1.00

Table A.3: Gelman and Rubin's R statistics for the parameter vector θ of model M_{S}. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains.

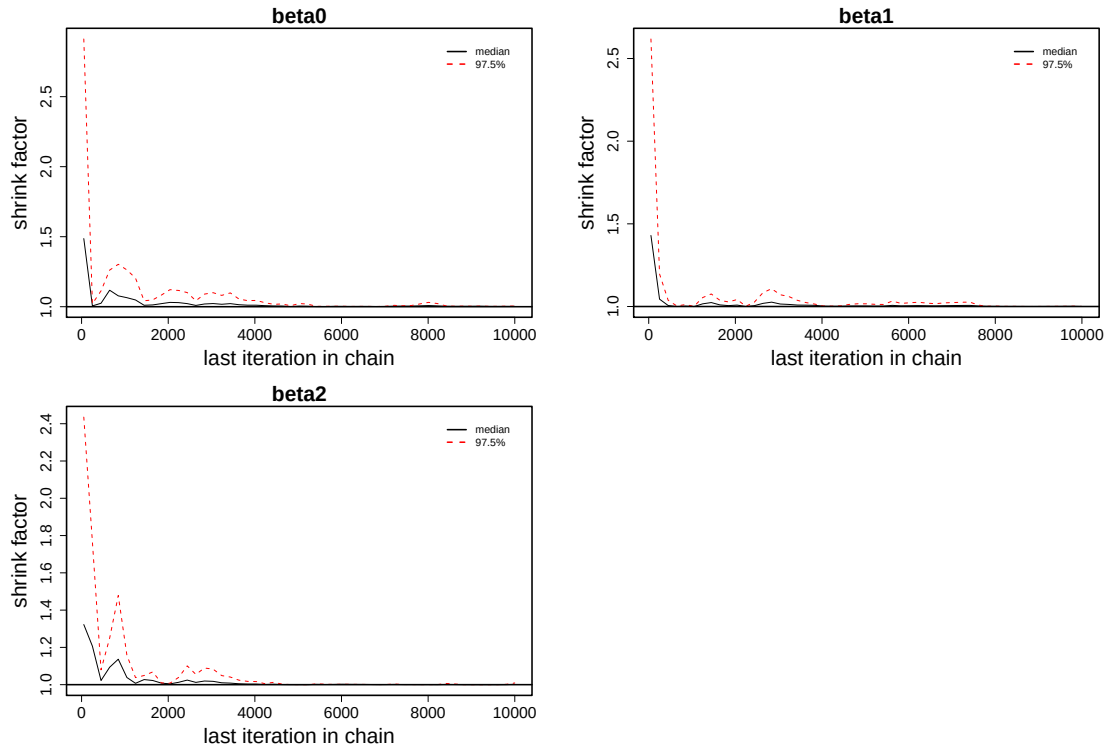


Figure A.1: Gelman and Rubing's R statistics over 10000 iterations for the parameter vector β of model M_{A} in (2.3). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistics.

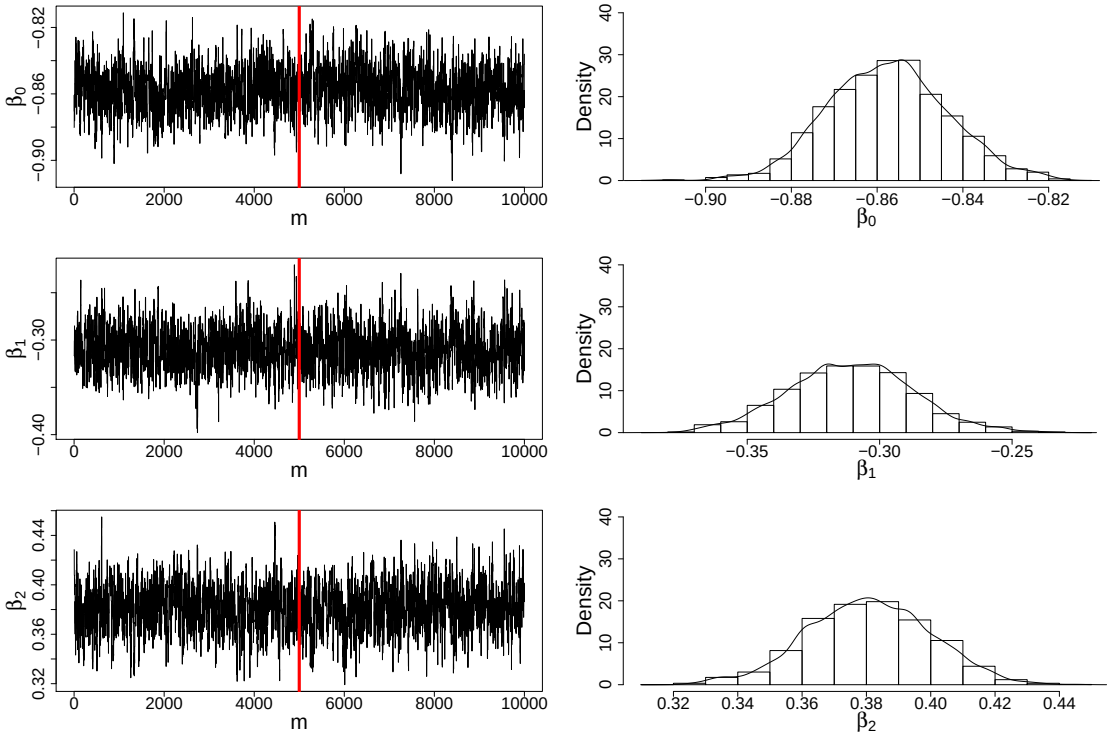


Figure A.2: Traces of the sampled values of the vector of parameters β (left) and the estimates of their posterior distributions (right) for model M_{A} in (2.3) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in.

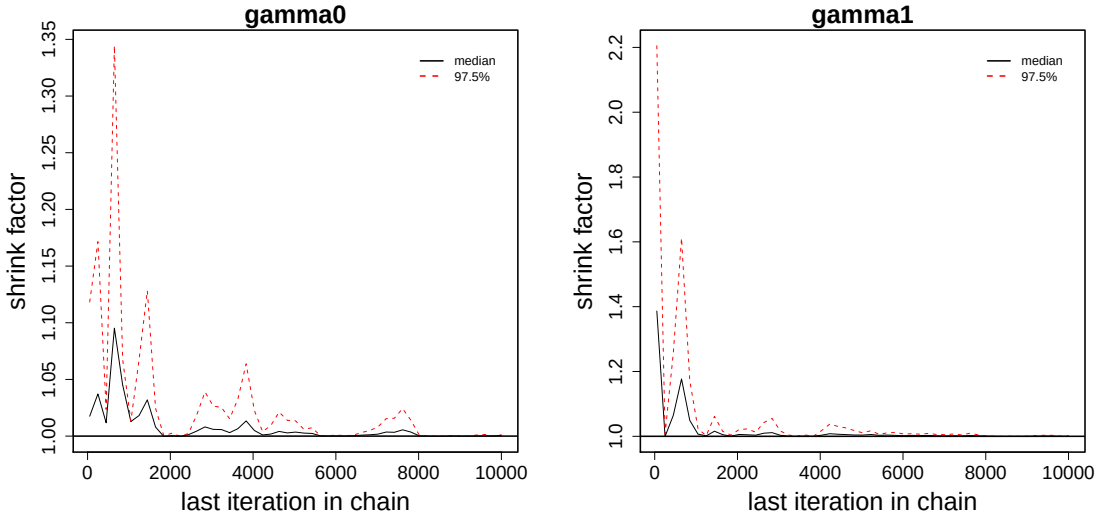


Figure A.3: Gelman and Rubing's R statistics over 10000 iterations for the parameter vector γ of model M_{D} in (2.4). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistic.

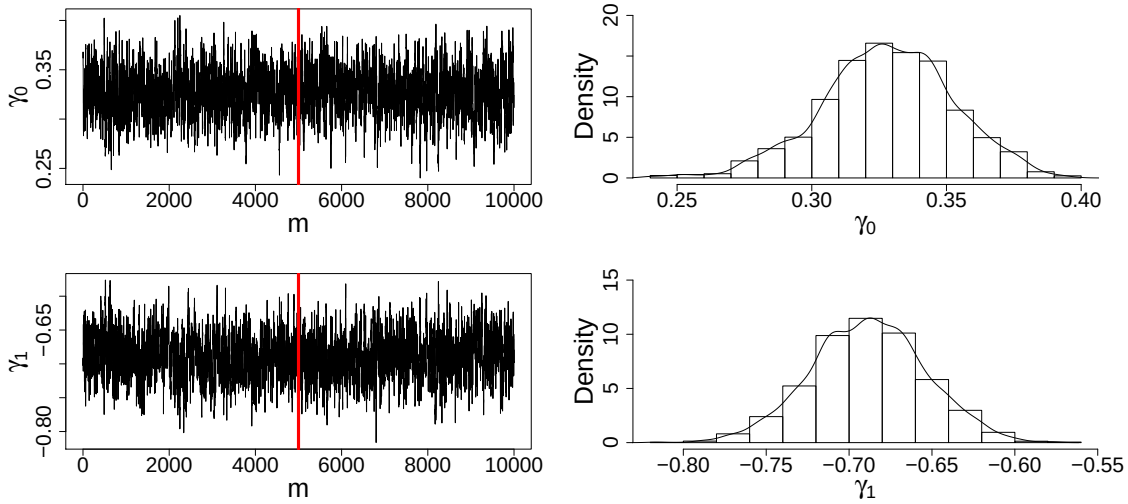


Figure A.4: Traces of the sampled values of the vector of parameters γ (left) and the estimates of their posterior distributions (right) for model M_{D} in (2.4) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in.

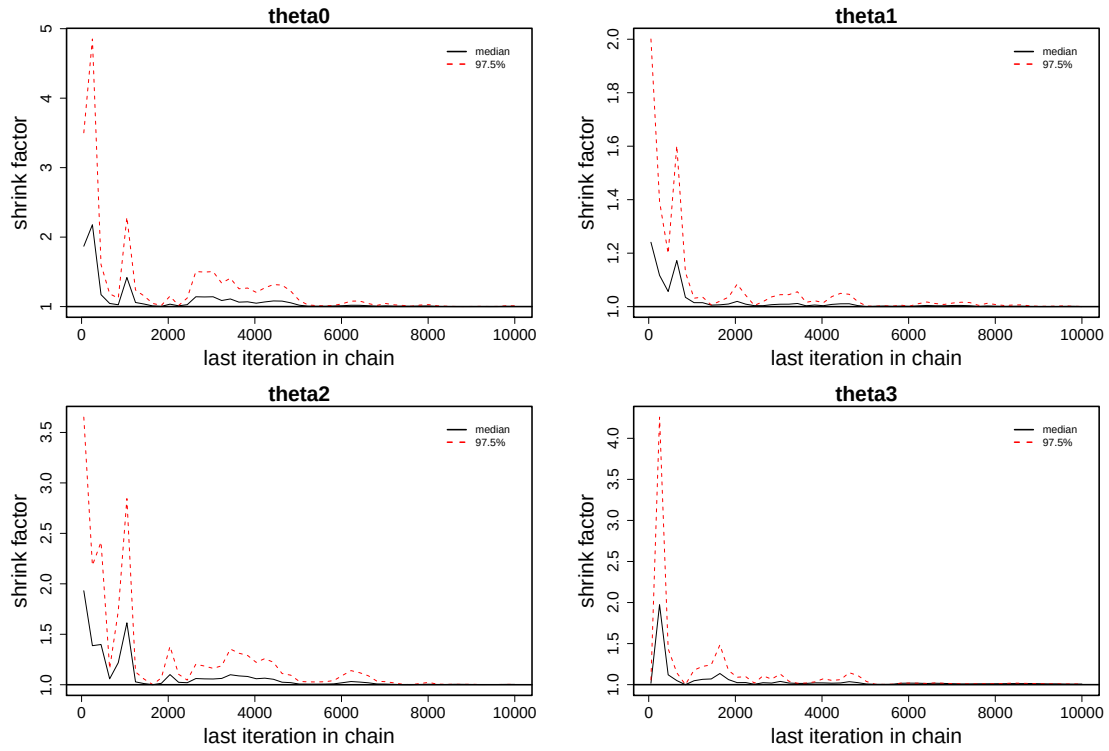


Figure A.5: Gelman and Rubing's R statistics over 10000 iterations for the parameter vector θ of model M_S in (2.5). Solid black line is the median and dashed red line represents the 95% credible intervals for R statistic.

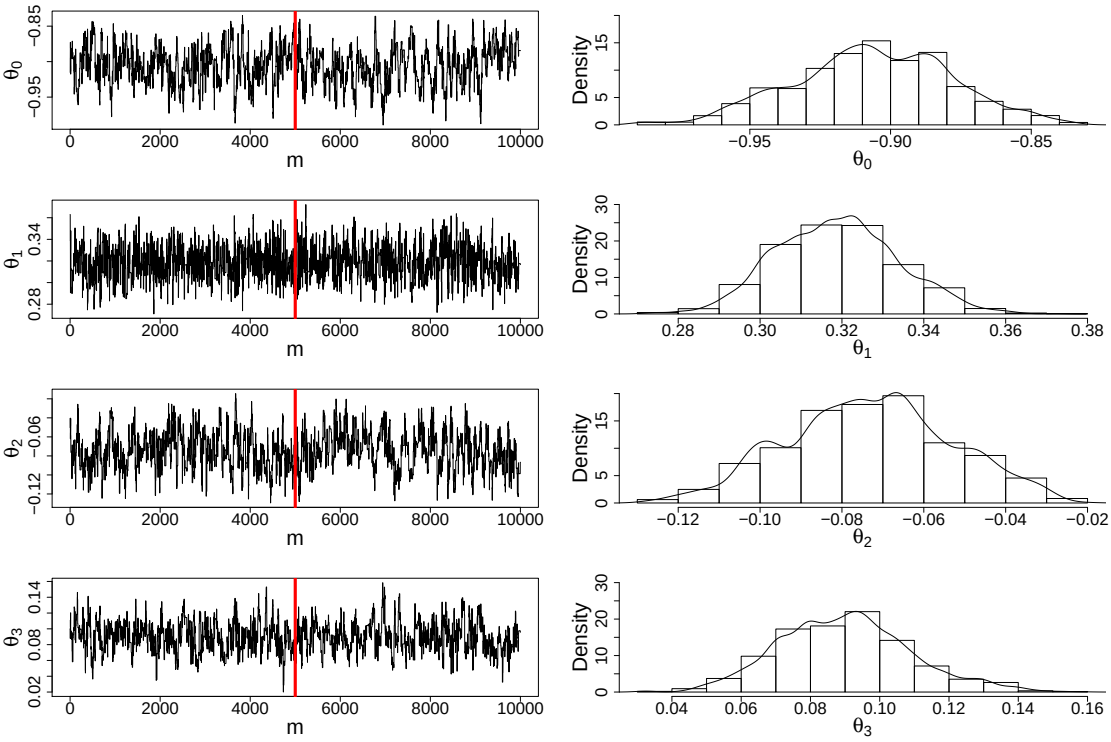


Figure A.6: Traces of the sampled values of the vector of parameters θ (left) and the estimates of their posterior distributions (right) for model M_S in (2.5) fitted to the FTSE100 index change. We set $m = 10000$ and discarded the first 5000 samples as burn-in.

Appendix B

ZPD model: Gelman and Rubin's convergence diagnostics, and traces of the MCMC chains

Coefficients	Simulated data	FTSE100 index change
α_0	0.59	0.45
α_1	0.60	0.41
α_2	0.60	0.44
α_3	0.60	0.46
β_0	0.63	0.59
β_1	0.48	0.30
β_2	0.63	0.51
β_3	0.60	0.65
p	0.58	0.44

Table B.1: Acceptance rate of the MCMC chains for the ZPD model in (3.10) fitted to the simulated data and the FTSE100 index change.

Potential scale reduction factors:			Potential scale reduction factors:		
	Point est.	97.5 quantile		Point est.	97.5 quantile
α_0	1.00	1.00	β_0	1.00	1.00
α_1	1.00	1.00	β_1	1.00	1.00
α_2	1.00	1.00	β_2	1.00	1.00
α_3	1.00	1.00	β_3	1.00	1.00
p	1.00	1.01			
Multivariate psrf					1.00

Table B.2: Gelman and Rubin's R statistics for the parameter vectors, α and β, of the ZPD model in (3.10) for the simulated data set. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains.

Potential scale reduction factors:			Potential scale reduction factors:		
	Point est.	97.5 quantile		Point est.	97.5 quantile
α_0	1.01	1.02	β_0	1.01	1.02
α_1	1.00	1.02	β_1	1.01	1.04
α_2	1.01	1.02	β_2	1.01	1.01
α_3	1.01	1.02	β_3	1.01	1.01
p	1.01	1.02			
Multivariate psrf					1.01

λ_1 λ_2

Table B.3: Gelman and Rubin's R statistics for the parameter vectors, α and β, of the ZPD model in (3.10) for FTSE100 data set. Values less than 1.1 suggest that we could assume the convergence of the MCMC chains.

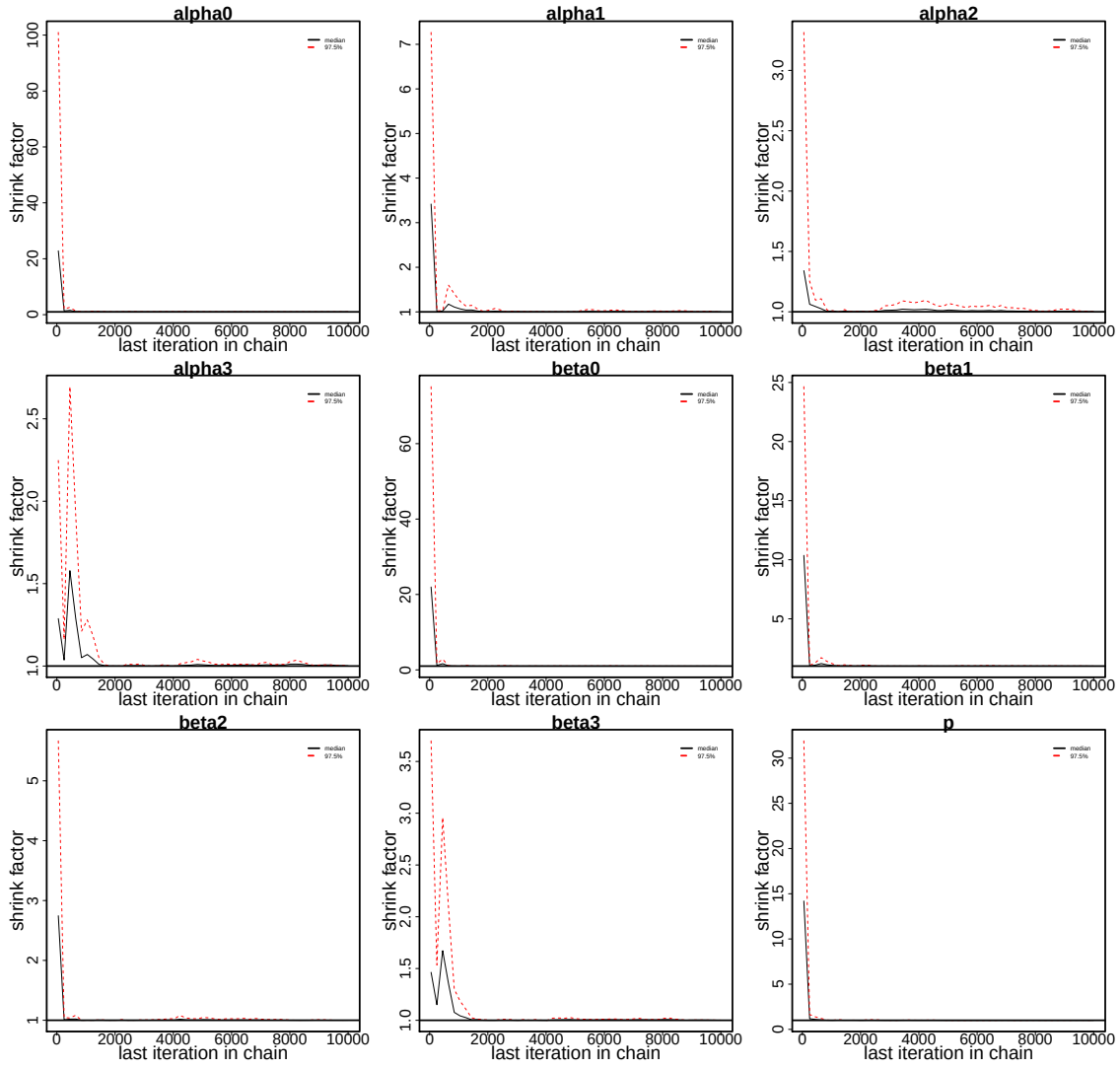


Figure B.1: Gelman and Rubing's R statistics over 10000 iterations for the model parameter vector ψ, for simulated data set. Solid black line is the median and dashed red line represents the 95% credible intervals for R statistics.

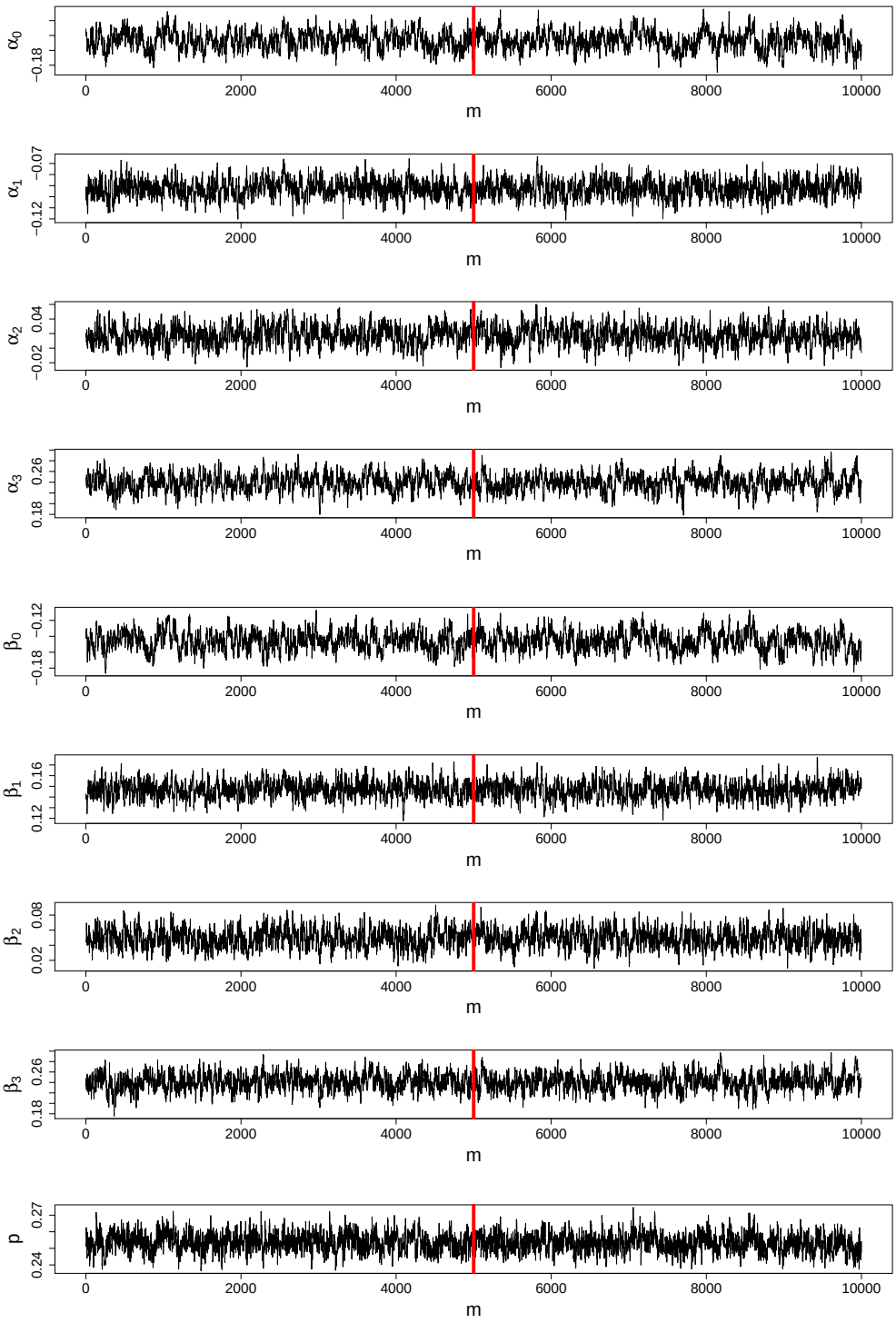


Figure B.2: Traces of the sampled values of the vector of parameters ψ of the ZPD model fitted to a set of simulated data. We set $m = 10000$ and discarded the first 5000 samples as burn-in.

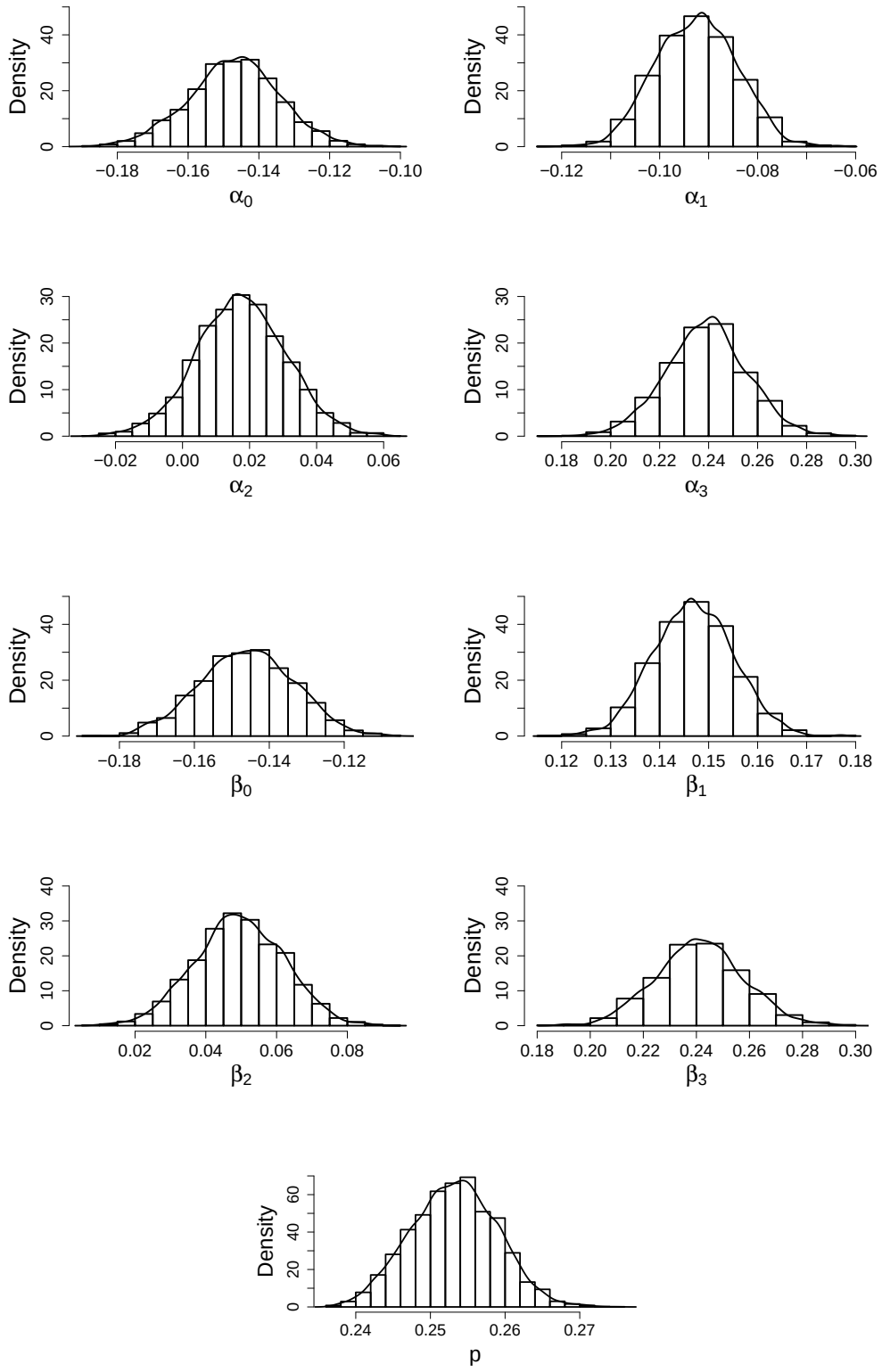
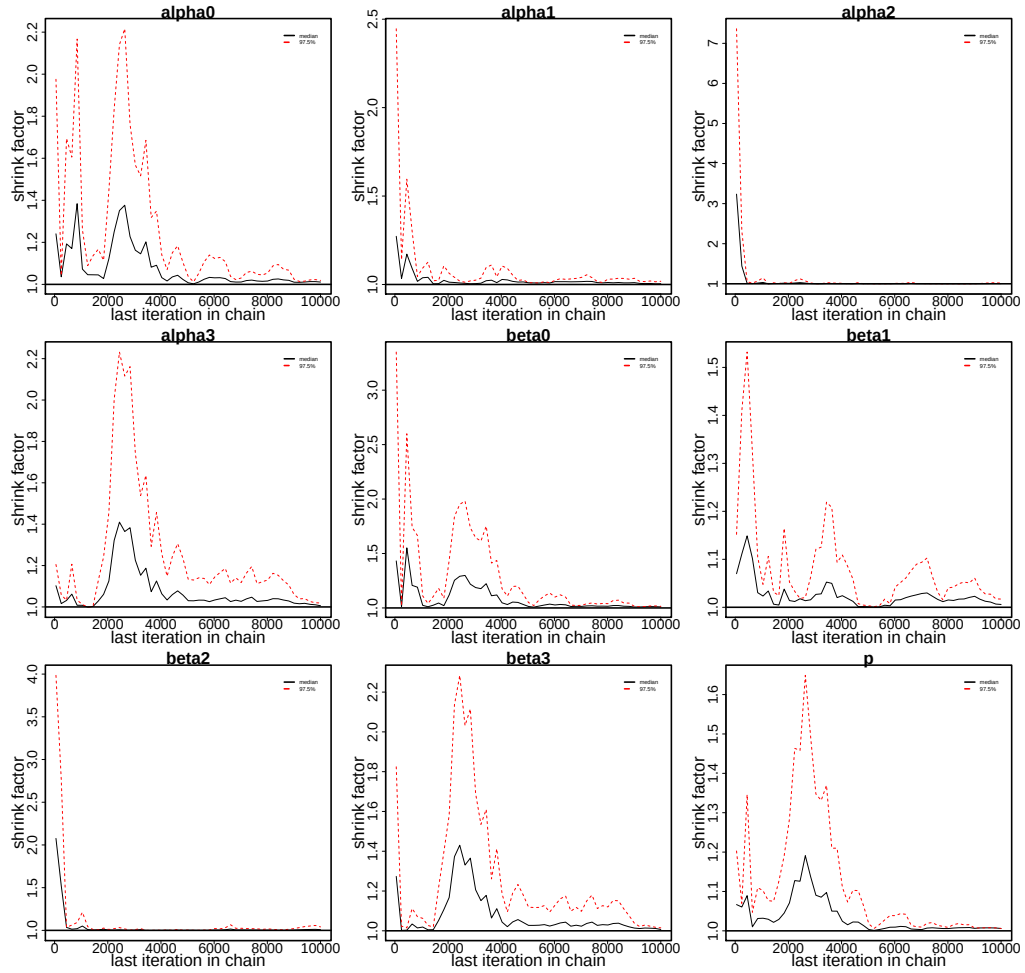


Figure B.3: The the estimates of the posterior distributions of the vector of parameters ψ of the ZPD model fitted to a set of simulated data after discarding the first 5000 samples as burn-in.



(a) α

Figure B.4: Gelman and Rubing's R statistics over 10000 iterations for the model parameter vector ψ, for FTSE100 data set. Solid black line is the median and dashed red line represents 95% credible intervals for R statistics.

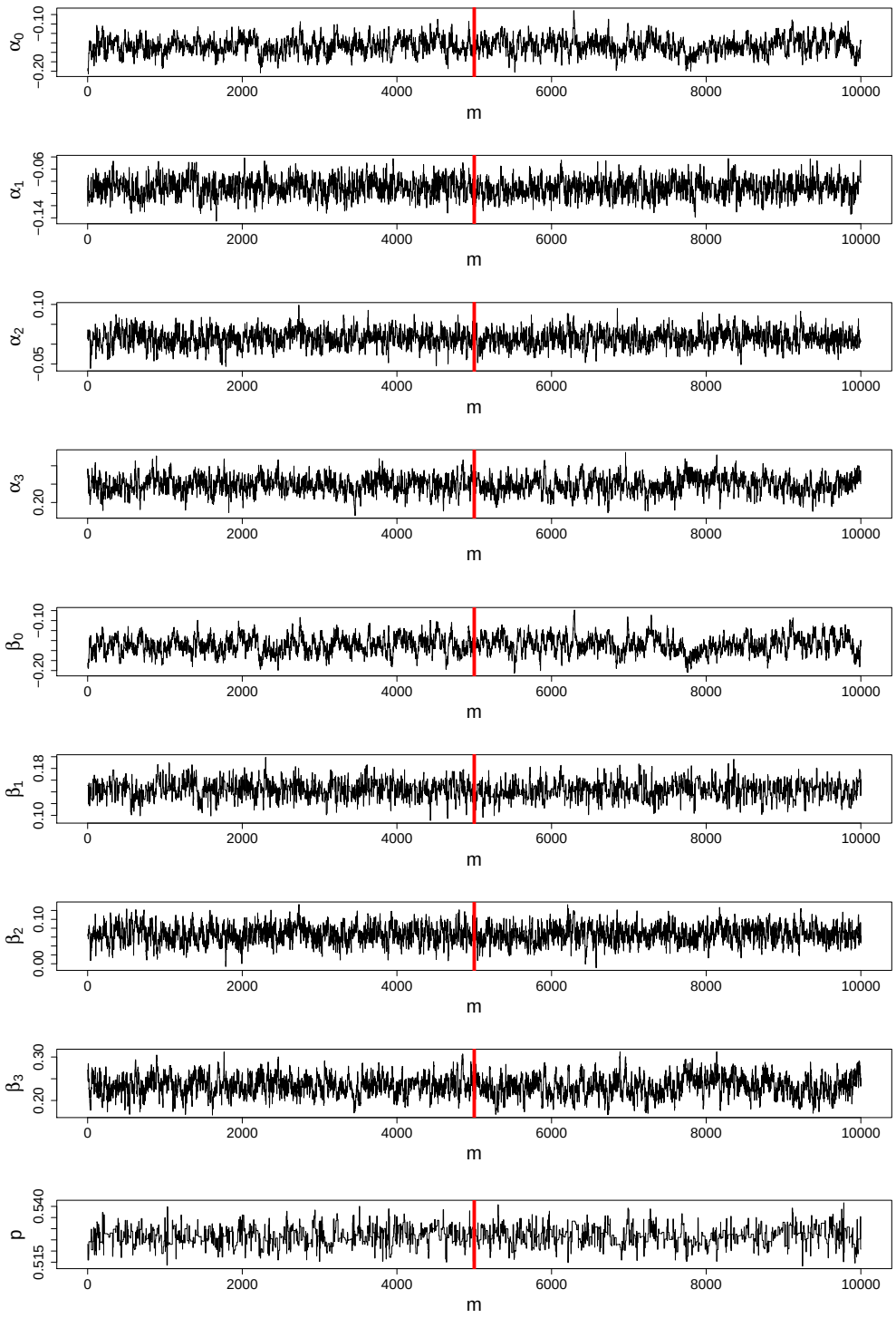


Figure B.5: Traces of the sampled values of the vector of parameters ψ of the ZPD model fitted to the FTSE100 index change. We set $m = 10000$ and discard the first 5000 iterations samples as burn-in.

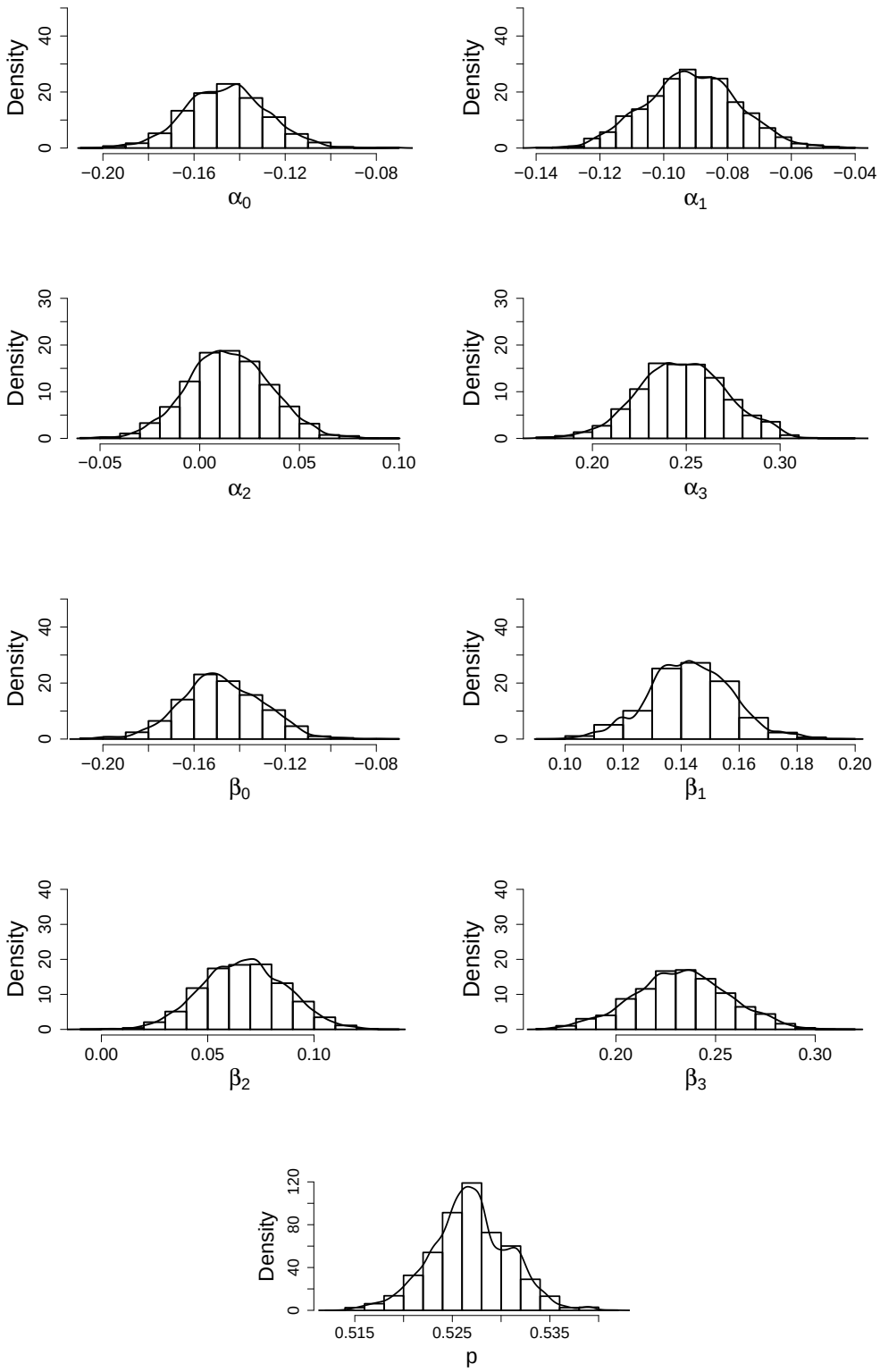


Figure B.6: The estimates of the posterior distributions of the vector of parameters ψ of the ZPD model fitted to the FTSE100 index change after discarding the first 5000 samples as burn-in.

Appendix C

Dynamic zero inflated Poisson difference model: the graphical presentation

C.1 DPD model: Graphical presentation of the results for the six simulated sets

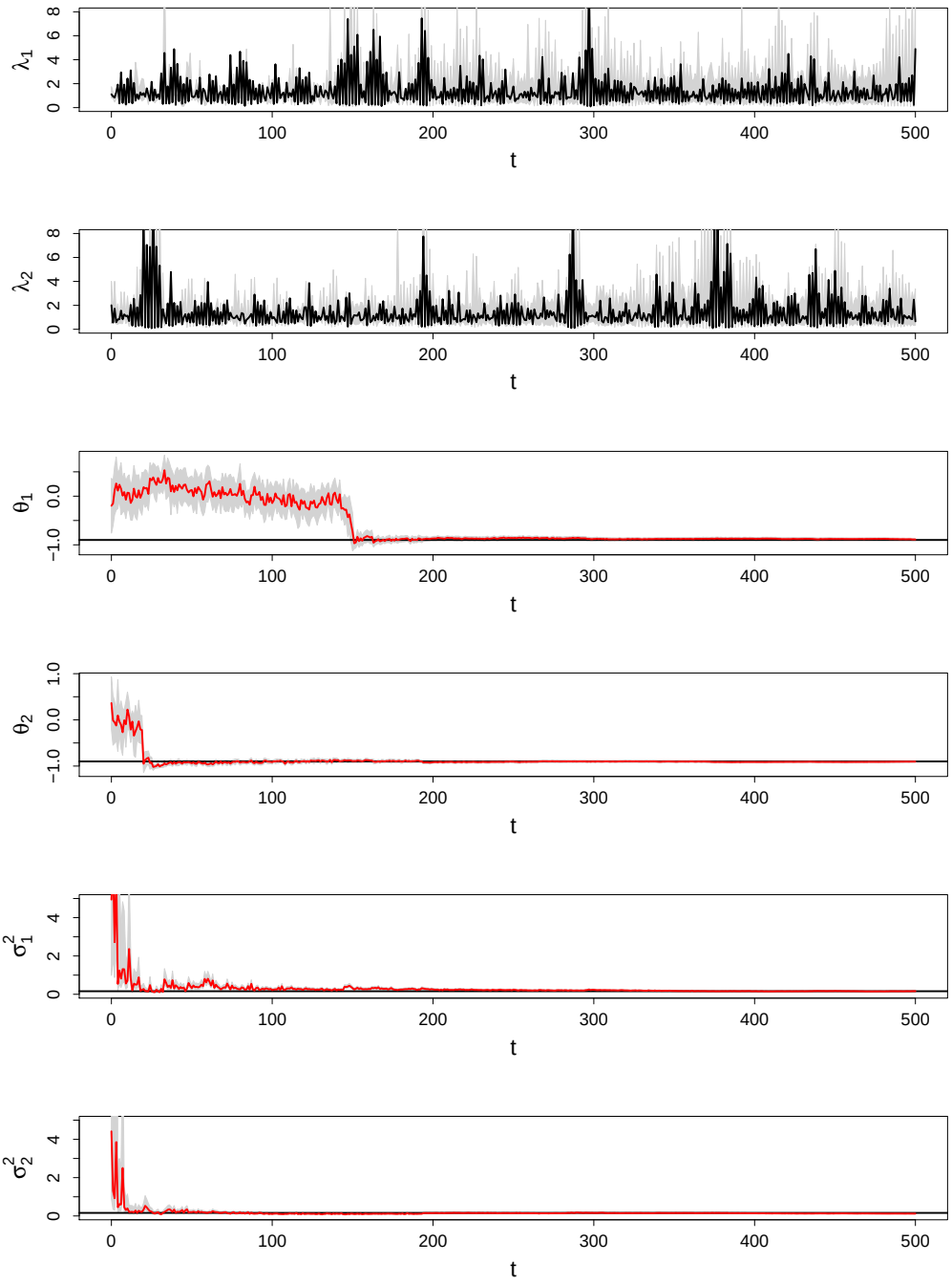


Figure C.1: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 1

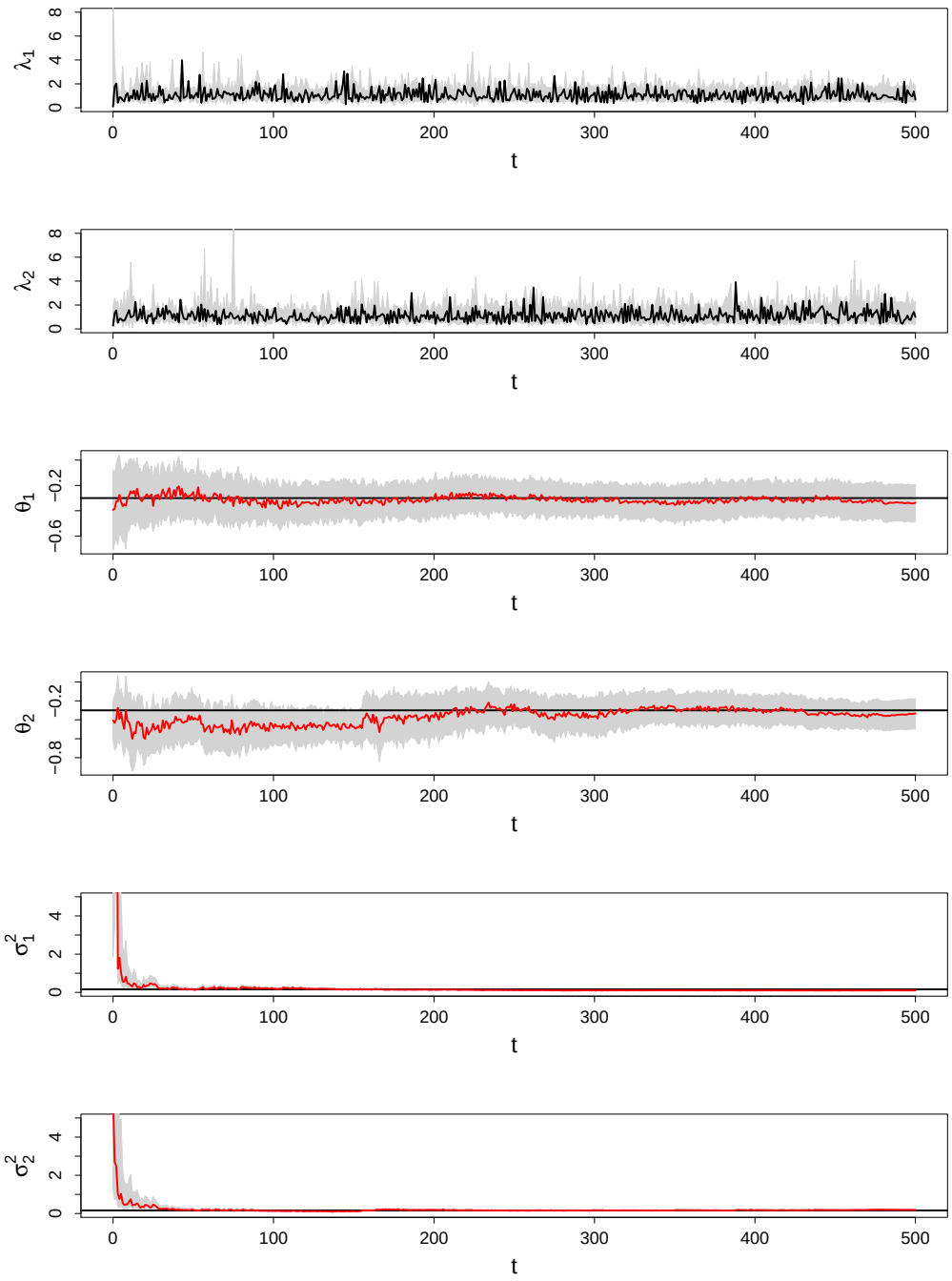


Figure C.2: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 2

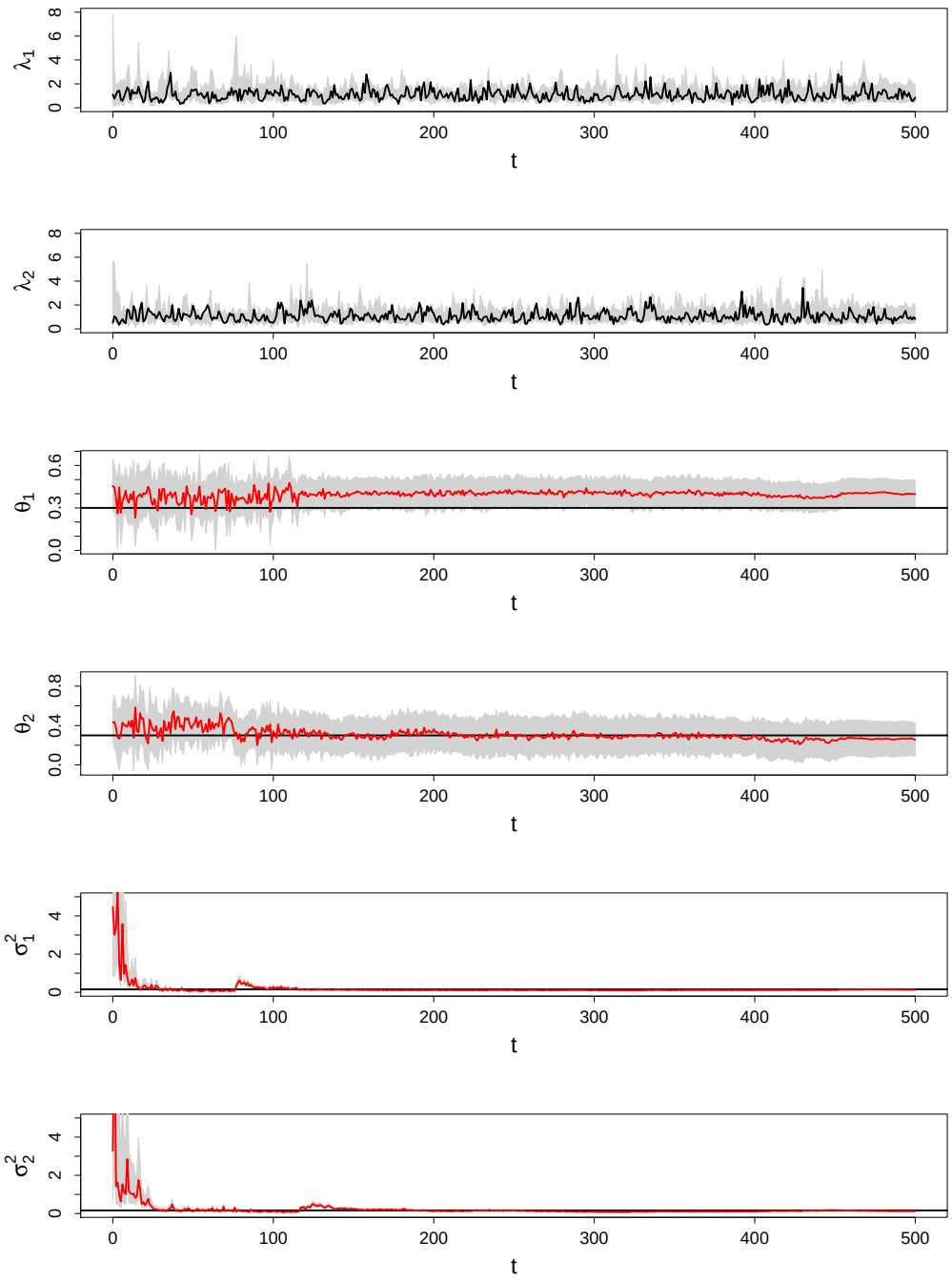


Figure C.3: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 3

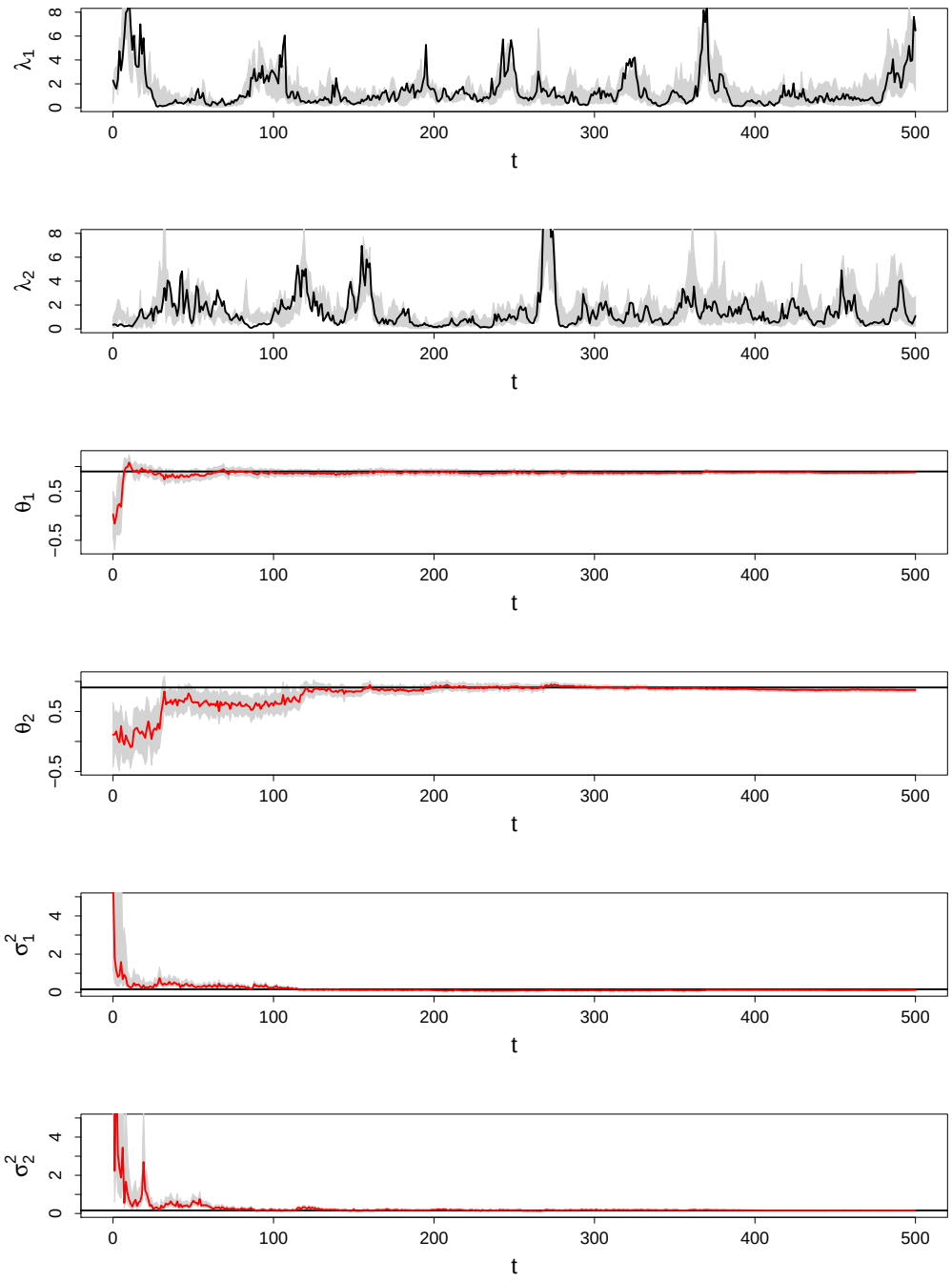


Figure C.4: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 4

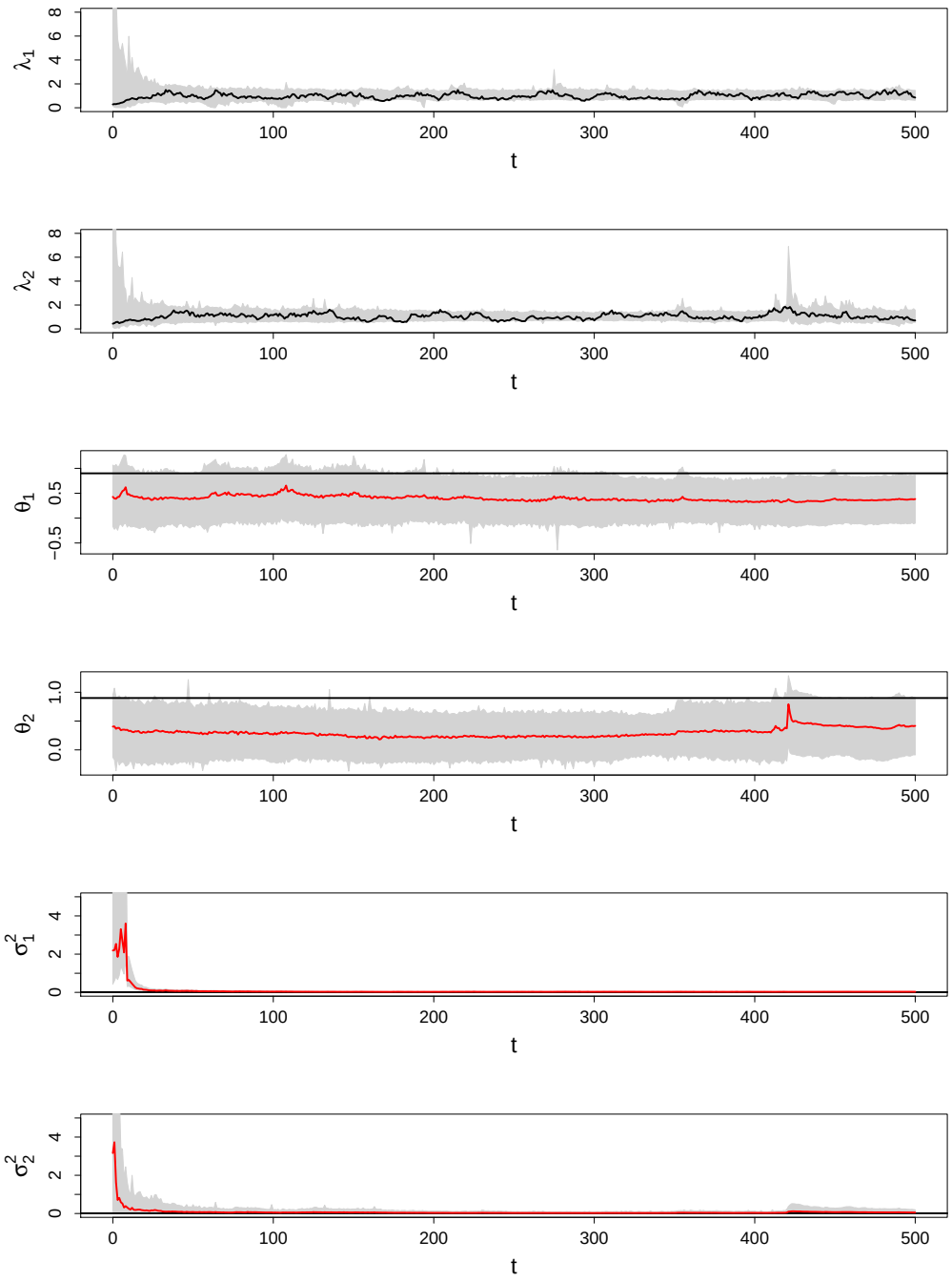


Figure C.5: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 5

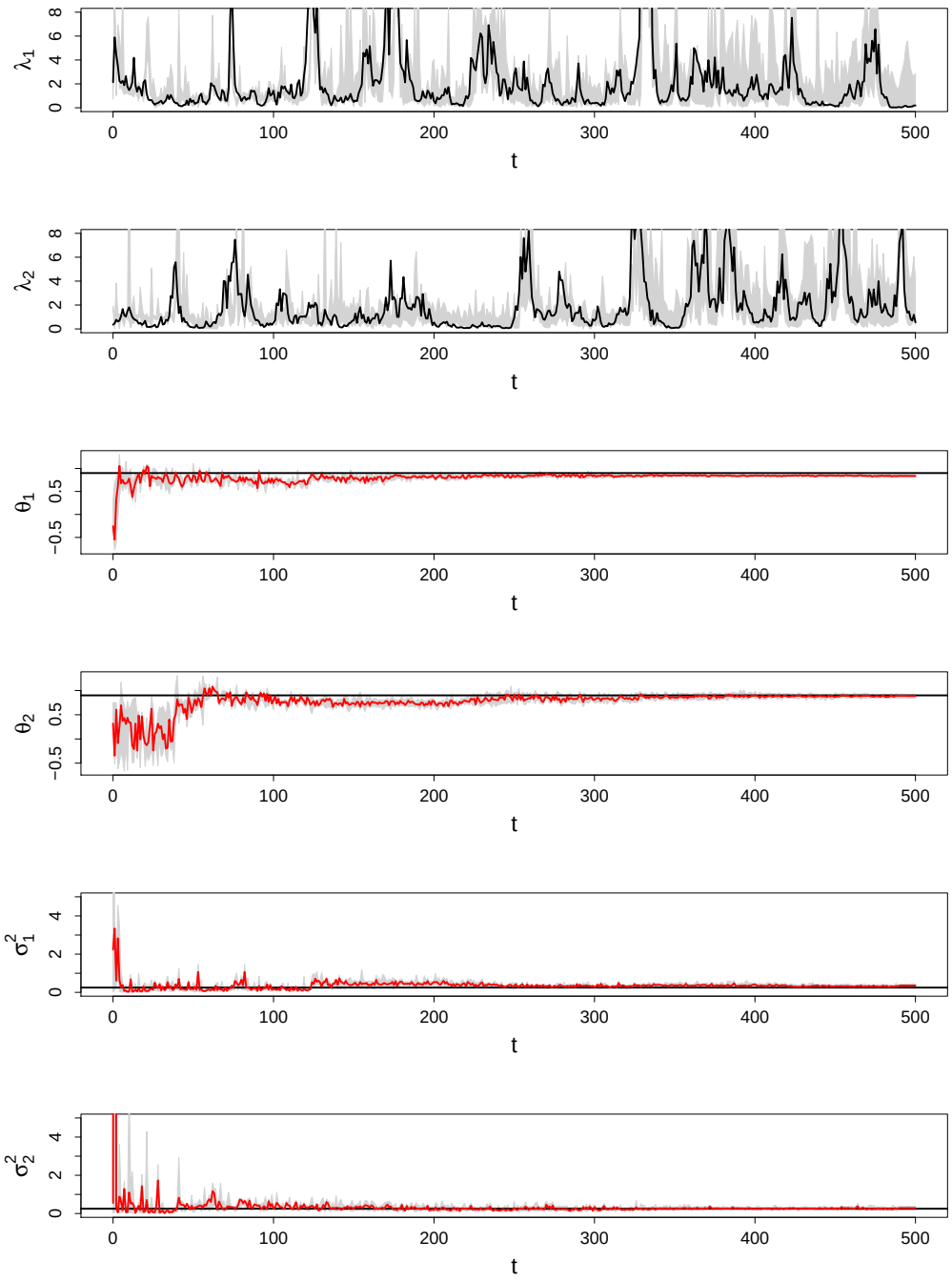


Figure C.6: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model for Dataset 6

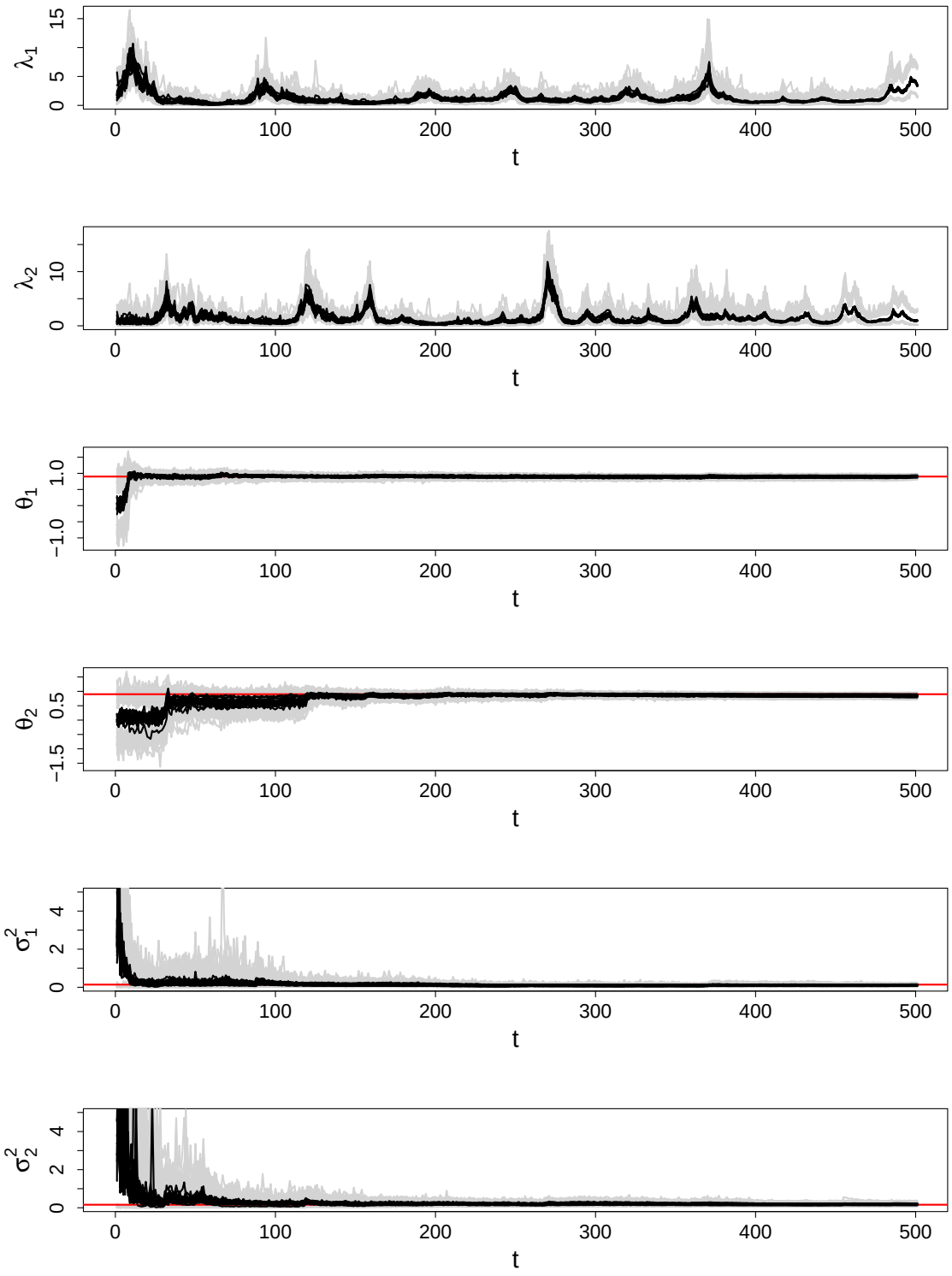


Figure C.7: The DPD model, a total of 20 replications of the LWF algorithm, each one based on $N = 5000$

C.2 DZPD model: Graphical presentation of the results for the three simulated data sets

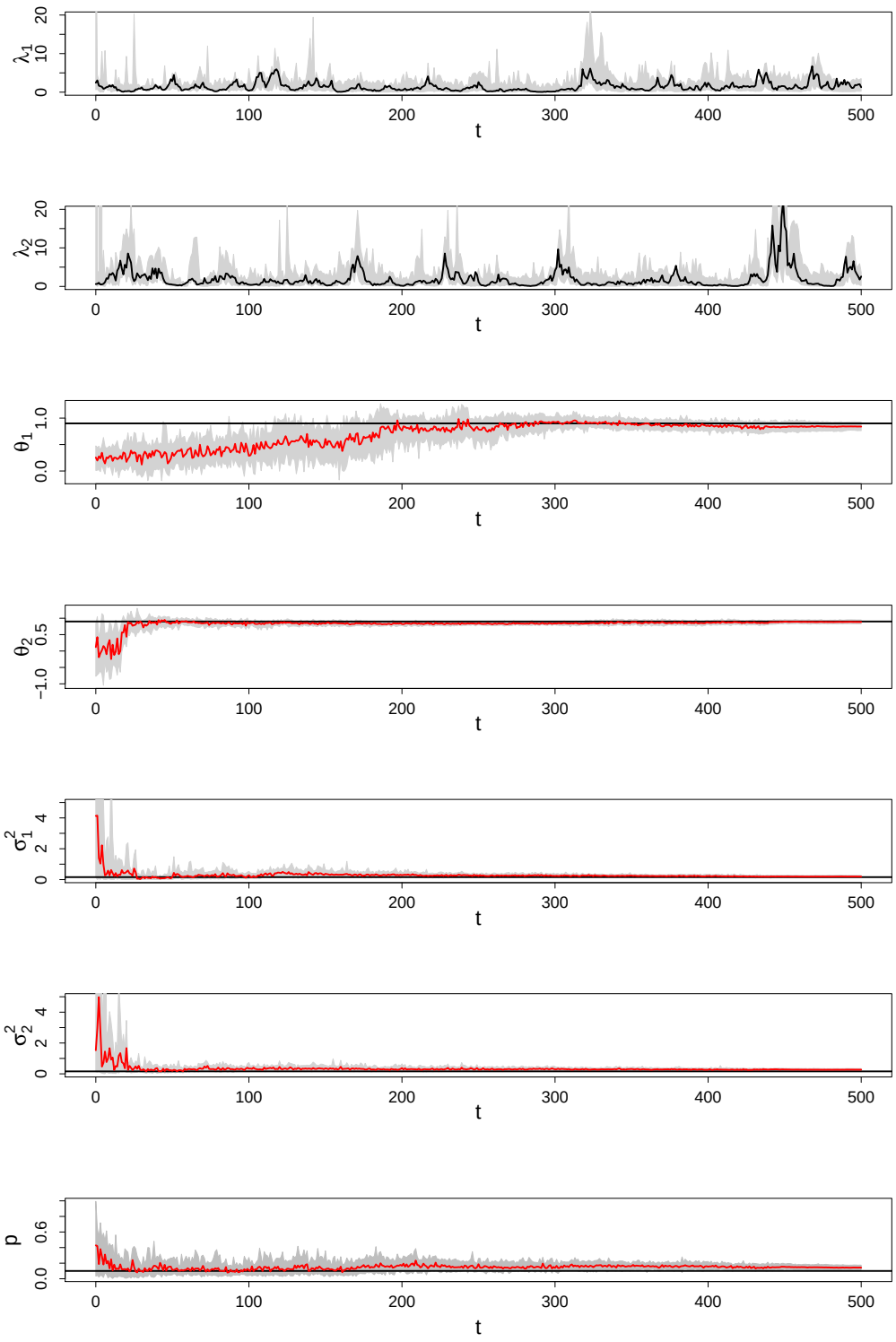


Figure C.8: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 7.

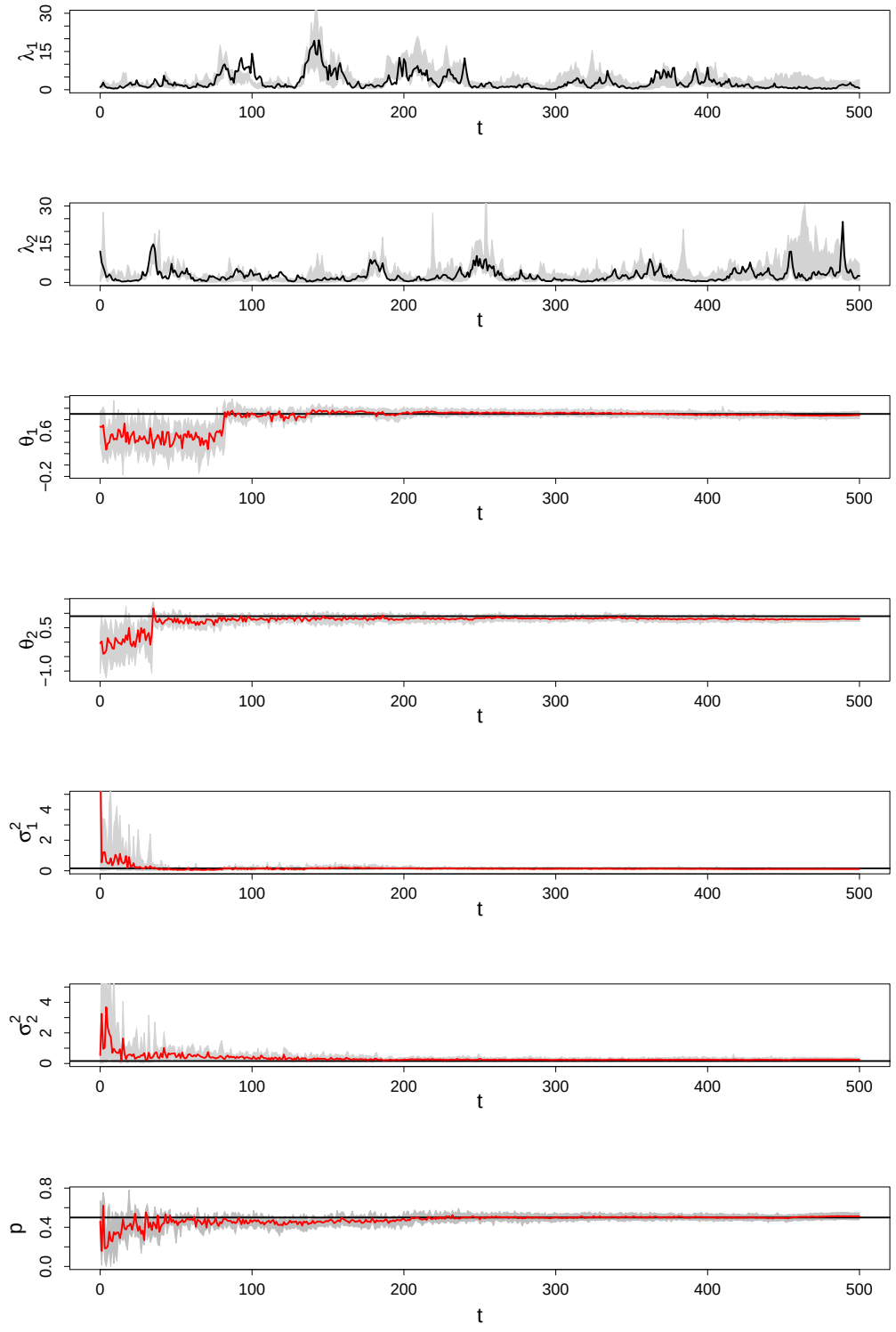


Figure C.9: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 8.

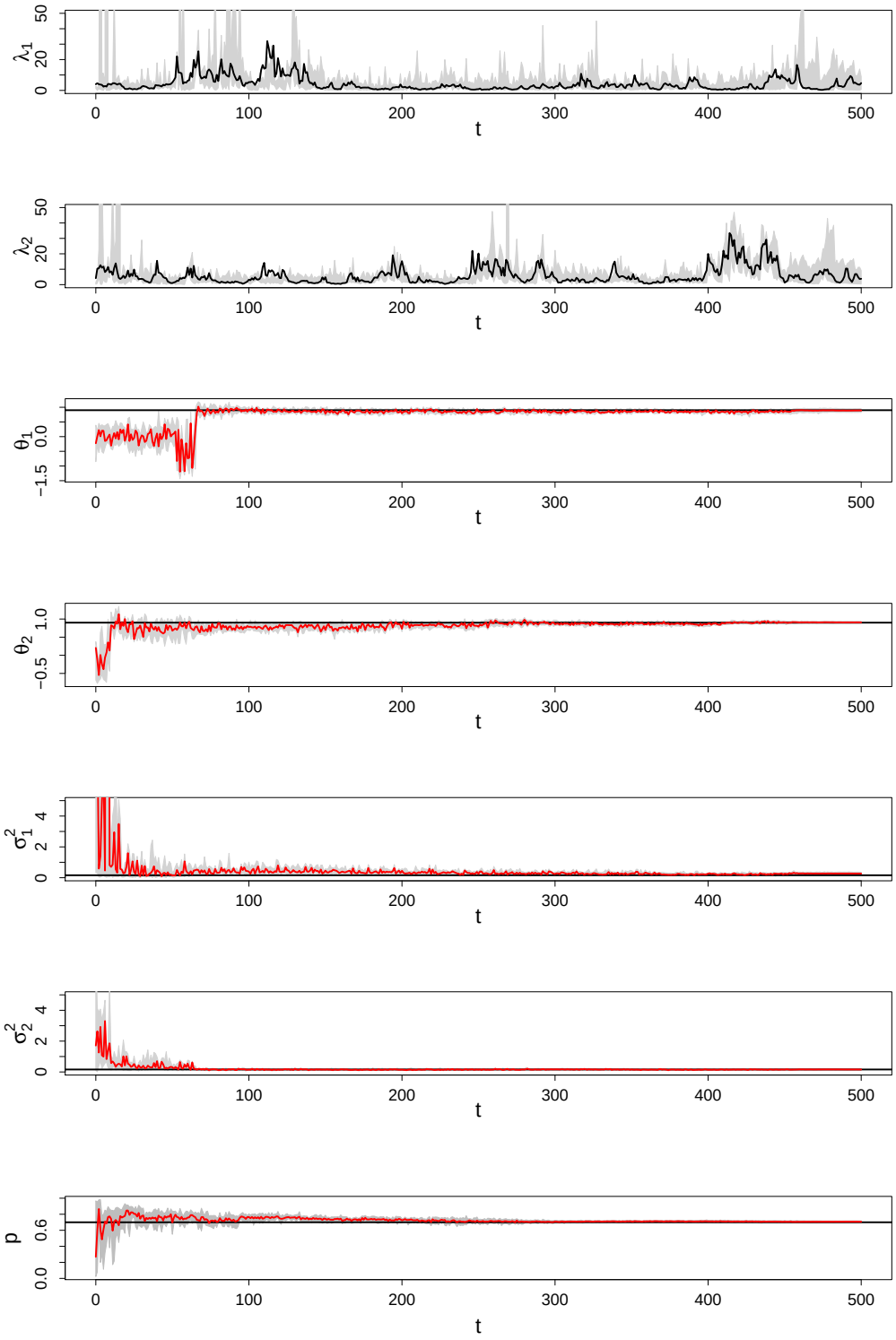


Figure C.10: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DZPD model for Dataset 9.

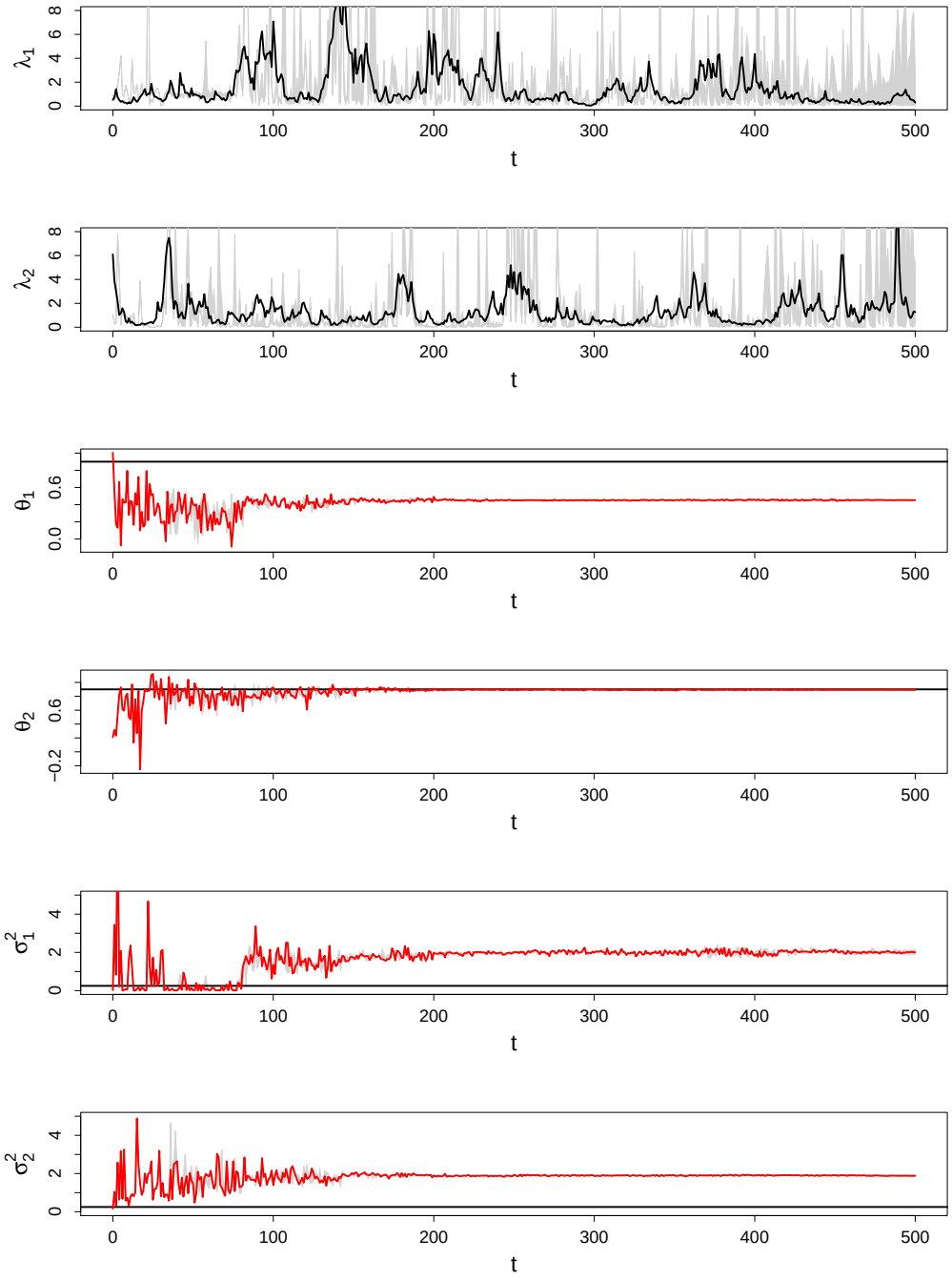


Figure C.11: True value (black line), the posterior mean (red line) and the 95% credible intervals (grey area) of the unobserved processes (the first and the second rows) and the parameters of the DPD model fitted to the Dataset 8.

C.3 DZPD model: Graphical presentation of the results for the the FTSE100 data set

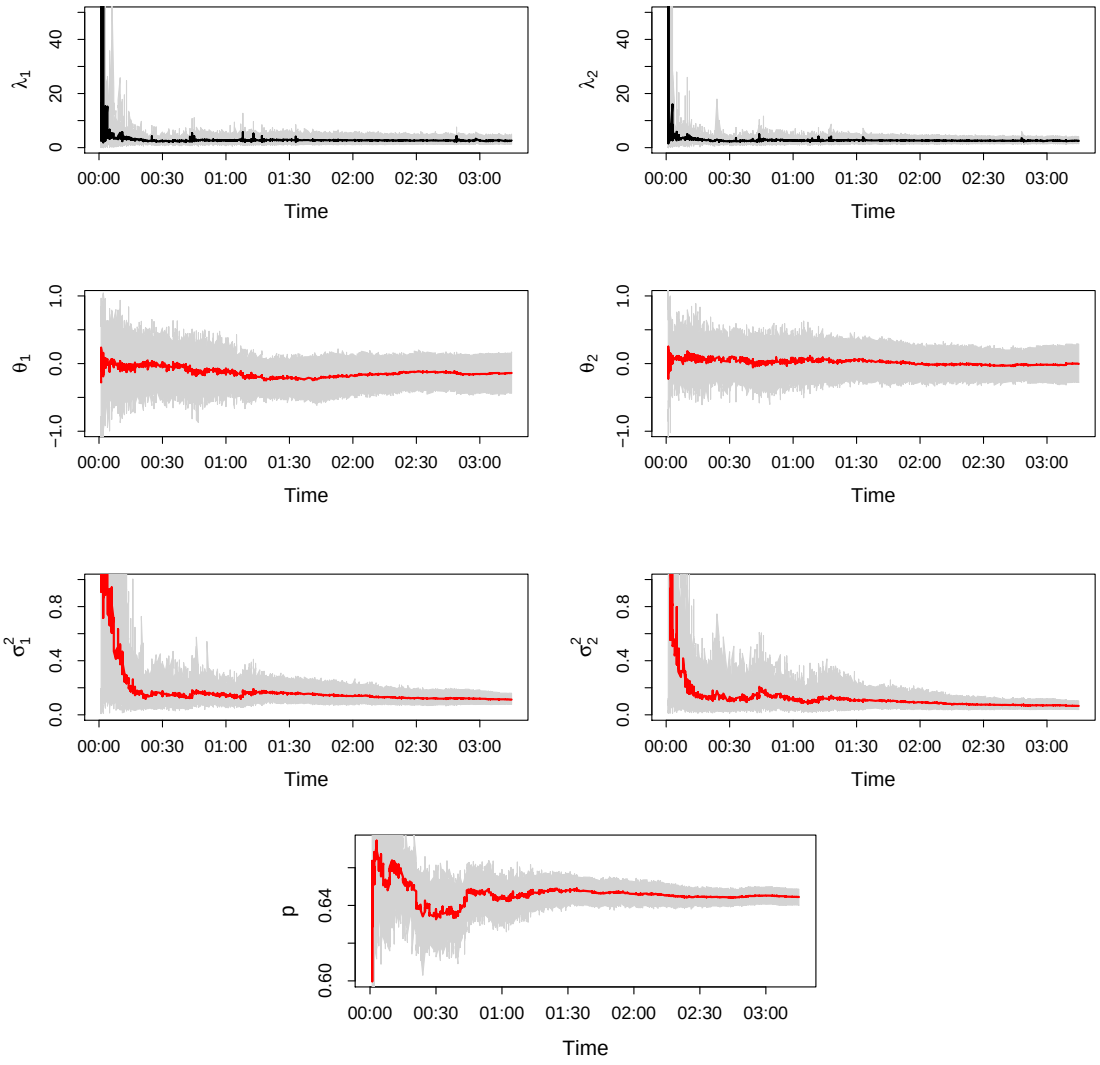


Figure C.12: The posterior mean, the 95% credible intervals (grey area) of the unobserved processes (the first row) and the DZPD model parameters for the first 1000 index changes of the FTSE100 data set.

Appendix D

R codes

D.1 ADS model

Following R codes (R.Code 2) are presented for the M-H algorithm for the activity of index. Same R codes were applied for the case of the direction of price change with two parameters. Also, for the size of index change same algorithm have been implemented except codes in R.Code 1 were replaced for the log-likelihood function and the posterior distribution.

```
## Likelihood function  for the size of price change.
## We use a same function for the Metroplis-Hastings algorithm for the
## size f price change as the activity and direction. We only need
## to replace the following codes in "R Code2", lines 5-15.
f<- function(t0, t1, t2, t3,t4){
   sum(-(t0 + t1*x.var + t2 * t.var + t3 * v.var + t4 * d.var))-
   sum((y.var+1)*
   log(1+exp(-(t0 + t1*x.var + t2 * t.var + t3 * v.var + t4 * d.var))))
  }

## Posterior distribution
posterior<- function(beta0, beta1, beta2, beta3, beta4){
                    betas<-cbind(beta0, beta1, beta2, beta3, beta4)
                    f(beta0, beta1, beta2, beta3, beta4) + dmvnorm(betas,
                    rep(0,5), diag(100, 5), log = TRUE )
}
```

R.Code 1: The log-likelihood and posterior distribution for the size of index change.

```r
## Bayesian ADS model, model for the Activity of price and direction of
price change.
### First step is to evaluate the posterior distribution
## Note: MCMCpack results in very similar values, but we construct our own
# functions
## Posterior distribution
f<- function( beta0, beta1, beta2){
sum(y.var*(beta0 + beta1 * x.var + beta2 * t.var)-
log(1 + exp(beta0 + beta1 * x.var + beta2 * t.var)))
   }

posterior<- function(beta0, beta1, beta2){
                    betas<-cbind(beta0, beta1, beta2)
                    f(beta0, beta1, beta2) + dmvnorm(betas
                    , rep(0,3), diag(100, 3), log = TRUE )
}
## R function to run the M-H algorithm
m.h.function<- function(m,s1,  beta01, beta11, beta21, tune){
beta0<- numeric(m)
beta1<- numeric(m)
beta2<- numeric(m)
## Initialise the algorithm
beta0[1]<-beta01
beta1[1]<-beta11
beta2[1]<-beta21
mu<- rbind(beta01, beta11, beta21)

### Set the counter to zero
k<-0
### For i = 2, ...,n repeat the following steps
for(i in 2:m){
  betas<- c(beta0[i-1],beta1[i-1],beta2[i-1])
### Sample candidates from the normal proposal distribution
## centred at the values from the previous interation
## tune is the tuning parameter which 2.4/sqrt(3)
  y<- rmvnorm(1, betas, s1 * tune)
  a<-posterior(y[1],y[2], y[3])-
  posterior(beta0[i-1],beta1[i-1], beta2[i-1] )
  d<- dmvnorm(betas,y,s1, log = TRUE) - dmvnorm(y,betas,s1, log= TRUE)
## Obtain the log acceptance ratio
  ratio1<- exp(d + a)
  ratio1<- min(1, ratio1)
  u1<-runif(1)
## Accept or reject the move
  if (ratio1 <= u1) {
    beta0[i]<- beta0[i-1]
    beta1[i]<- beta1[i-1]
    beta2[i]<- beta2[i-1]
    }
    else{
    beta0[i]<- y[1]
    beta1[i]<- y[2]
    beta2[i]<- y[3]
## Count the number of accepted moves
    k1<- k1+1
   }
print(i)
}
print(k1)
return(cbind(beta0, beta1,beta2))
}

```

R.Code 2: The M-H algorithm R codes, modelling index activity *A*.

```r
### One-step ahead predictions for the whole next trading day's index changes
## This function has the following arguments:
## m: number of preditions; a, b, c: vector of parameters of A, D and S models;
## a.v, a.dt: coresponding covariates for index activity and,
## d.v, d.dt: direction and size of index change.
simu<- function(m, a,b,c, a.v, a.dt, d.v, d.dt){
### "Index" represents the predicted index change of the next trading day
Index<- rep(0,m)
### "active" is the preditced index activity
active<- rep(0,m)
active[1]<- 0
size<- 0
## We set counter in order to keep track on the active indices
k<-0
direction<- 0
#pp.d<- NULL
nn<- length(d.v)
for(i in 2:m){
          p.a<- exp(a[1] + a[2] * active[i-1] + a[3]* a.v[i-1]  + a[4]* a.dt[i-1] )/(1+exp(
                    a[1] + a[2] * active[i-1] + a[3] * a.v[i-1] + a[4]* a.dt[i-1]))
          active[i] <- rbinom(1,1,p.a)
######## First we decide wether the next index change is active or not:
          if(active[i] == 1){
##If a change occurs, repeat the following steps
                    k<- k+1
                    if(k > nn) k<- 1
                    ## Obtain the p.d = p(D = 1| a = 1)
                    p.d<- exp(b[1] + b[2] * direction+ b[3] * d.v[k]+ b[4]* d.dt[k] )/(1+exp(
                              b[1] + b[2] * direction+ b[3] * d.v[k]+ b[4]* d.dt[k]))
                    ## Sample "direction" with prbability p.d from a Bernoulli distribution
                    direction<- rbern(1,p.d)
                        ifelse(direction== 0, direct<- -1, direct<- 1)
                    ## Obtain p.s = p(S' = s')
                    p.s<- exp(- c[1] - c[2] * size- c[3] * d.v[k] - c[4]* d.dt[k])/(1+exp(
                              - c[1] - c[2] * size- c[3] * d.v[k] - c[4]* d.dt[k]))
                    ## Sample "size - 1" with prbability p.s from a geometric distribution
                    size<- rgeom(1,p.s)
                    ## Obtain the value of index change
                    Index[i]<- (size+1) * direct
          }
           else{
                    Index[i]<- 0
           }
}
return(Index)
       }
```

R.Code 3: One-step ahead prediction for the next trading day using the ADS model.

D.2 Zero inflated Poisson difference model

```
## Bayesian ZPD model with 9 parameters
## Version 9, modified on 21/10/2010
##
### We start by evaluating the Log-likelihood function in two parts,
## 1 - Non zero index changes
PD.l<- function(a0, a1, a2,  a3, b0, b1, b2, b3,  p){ #
                  y.var1<- y.var[y.var!= 0]
                  if(p <1 & p >0){
                      n<- length(y.var1)
                      mu1 <- exp(a0 + a1 *( x.var[y.var!= 0]-x.bar) + a2*(log(v.var[y.var!= 0]) - v.bar) + a3*(t.var[y.var!= 0] - t.bar))
                      mu2<-  exp(b0 + b1 * ( x.var[y.var!= 0]-x.bar) + b2*(log(v.var[y.var!= 0]) - v.bar) + b3*(t.var[y.var!= 0] - t.bar))
                      bessel<- besselI(2*sqrt(mu1 * mu2),abs(y.var1))
                      n * log(1-p) + sum(-(mu1 + mu2) + (y.var1)/2 * (log(mu1) - log(mu2)))+
                                 sum(log(bessel))
                  }
      }


## 2 - Zero index changes
ZPD.l<- function(a0, a1, a2,a3, b0, b1, b2, b3, p ){  #
      y.var2<- y.var[y.var== 0]
      if(p <1 & p >0){
                      mu1 <- exp(a0 + a1 * ( x.var[y.var== 0]-x.bar) + a2*(log(v.var[y.var== 0]) - v.bar) + a3*(t.var[y.var== 0]- t.bar))
                      mu2<-  exp(b0 + b1 * ( x.var[y.var== 0]-x.bar) + b2*(log(v.var[y.var== 0]) - v.bar) + b3*(t.var[y.var== 0] -t.bar))
               bessel<- besselI(2*sqrt(mu1 * mu2), 0 )
              f_0PD <- exp(-(mu1 + mu2)) * bessel
              sum(log( p + (1-p) * f_0PD))
      }
   }
################################################################################
#####
### The posterior distribution distribution is obtained as follows:
###
posterior<- function(a0,a1,a2,a3,b0,b1,b2, b3, p){
## log likelihoods
        PD.l(a0,a1,a2,a3,b0,b1,b2,b3,p) + ZPD.l(a0,a1,a2,a3,b0,b1,b2,b3,p)+
## log of prior distributions, normal for the parameters of "lambda1" and
# "lambda2" and uniform for the porportion of excess of zeros "p"
        dnorm(a0, 0,100, log = TRUE) + dnorm(a1, 0,100, log = TRUE) +
        dnorm(a2, 0,100, log = TRUE) + dnorm(a3, 0,100, log = TRUE) +
        dnorm(b0, 0,100, log = TRUE) + dnorm(b1, 0,100, log = TRUE) +
        dnorm(b2, 0,100, log = TRUE)+ dnorm(b3, 0,100, log = TRUE)+dunif(p, 0,1,log = TRUE)

    }
posterior(a0,a1,a2, a3,b0,b1,b2,b3, p)#
## The following function evaluate the ratio in order to decide accepting
## or rejecting draws at "i"th  iteration . It just save some lines within the
## the M-H algorithm.
###
### Ratio function for the M-H algorithm
ratio<- function(post.cand, post.old, x1, y, k){
                  ratio1 <-  post.cand - post.old
                            ratio1<- min(0 , ratio1)
                            u<- log(runif(1))
                            if (ratio1 <= u) {
```

```
 55:                                x2<- x1
 56:                              }
 57:                              else{
 58:                                x2<- y
 59:                                k<- k+1
 60:                              }
 61:                 return(c(x2,k))
 62:               }
 63: #
 64: ##
 65: ### The M-H algorithm
 66: ##
 67: #
 68: Bayesian.Skellam2<- function(m, vc.matrix, tune, a, b, p.i, Var.p){
 69: ## We update one parameter at the time, thus for each of the parameters
 70: # we set one counter
 71:       k1<-0;k2<- 0  ;k3<-0 ;k4<-0
 72:       k5<-0;k6<- 0 ; k7 <- 0 ;k8<-0 ;k9<-0
 73:
 74:       alpha0<- numeric(m)   # ### Define 9 vectors for the model
    parameters,
 75:       alpha1<- numeric(m)   # ##  each with length m
 76:       alpha2<- numeric(m)   #
 77:       alpha3<- numeric(m)   #
 78:       beta0<- numeric(m)    ##
 79:       beta1<- numeric(m)    #
 80:       beta2<- numeric(m)    #
 81:       beta3<- numeric(m)    #
 82:       p <-  numeric(m)      #
 83:
 84:       alpha0[1]<- a[1]      # ### Initialise the algorithm
 85:       alpha1[1]<- a[2]      #
 86:       alpha2[1]<- a[3]      #
 87:       alpha3[1]<- a[4]      #
 88:       beta0[1] <- b[1]      #
 89:       beta1[1] <- b[2]      #
 90:       beta2[1] <- b[3]      #
 91:       beta3[1] <- b[4]      #
 92:       p[1] <- p.i           #
 93: ### tune is the tuning parameter in our case is 2.4/sqrt(1) for each
 94: ##  of the parameters
 95: for(i in 2:m){
 96: y10 <- rnorm(1,alpha0[i-1], vc.matrix[1,1]* tune)     ## Sample from
    candidate distribution
 97: y11 <- rnorm(1,alpha1[i-1], vc.matrix[2,2]* tune)
 98: y12 <- rnorm(1,alpha2[i-1], vc.matrix[3,3]* tune)
 99: y13 <- rnorm(1,alpha3[i-1], vc.matrix[4,4]* tune)
100: y20 <- rnorm(1,beta0[i-1], vc.matrix[5,5]* tune)     ## Sample from
    candidate distribution
101: y21 <- rnorm(1,beta1[i-1], vc.matrix[6,6]* tune)
102: y22 <- rnorm(1,beta2[i-1], vc.matrix[7,7]* tune)
103: y23 <- rnorm(1,beta3[i-1], vc.matrix[8,8]* tune)
104: x.j1<- log(p[i-1]/(1 - p[i-1]))
105: ### The next 9 parts perform updates one parameter at the time.
106: ## For example we sample for "p", obtain the ratio and accept or reject the
107: ## sampled value and move to alpha1, and so.
108: ## Each part provide a separate acceptance ratio.
109: ################################################################################
110: x.j<- rnorm(1, x.j1, Var.p*tune )
111: p.cand<- exp(x.j)/(exp(x.j) + 1)#rbeta(1, beta.a, beta.b)
112: post.cand5<- posterior(alpha0[i-1], alpha1[i-1], alpha2[i-1],alpha3[i-1]
113:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
```

```
114:            ,p.cand) + log(p.cand*(1- p.cand)) +
115:            dnorm(x.j1, x.j,Var.p *tune, log = TRUE )
116: post.old5 <- posterior(alpha0[i-1], alpha1[i-1], alpha2[i-1],alpha3[i-1]
117:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
118:            ,p[i-1]) + log(p[i-1]*(1- p[i-1]))+
119:             dnorm(x.j,x.j1,Var.p, log = TRUE )
120: x1<-  ratio(post.cand5, post.old5, p[i-1], p.cand, k5)
121: p[i]<- x1[1]
122: k5<- x1[2]
123: ##########################        lambda1     ###################
124: post.cand1<- posterior(y10, alpha1[i-1], alpha2[i-1], alpha3[i-1]
125:            , beta0[i-1],beta1[i-1], beta2[i-1], beta3[i-1]
126:            ,p[i]) + dnorm(alpha0[i-1],y10, vc.matrix[1,1] *tune, log =
     TRUE )
127: post.old1<- posterior(alpha0[i-1], alpha1[i-1], alpha2[i-1],alpha3[i-1]
128:           , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
129:           ,p[i]) + dnorm(y10,alpha0[i-1], vc.matrix[1,1] *tune , log =
     TRUE  )
130: x2<- ratio(post.cand1, post.old1, alpha0[i-1], y10, k1)
131: alpha0[i]<- x2[1]
132: k1<- x2[2]
133: ####################################
134: post.cand2<- posterior(alpha0[i], y11, alpha2[i-1],alpha3[i-1]
135:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
136:            ,p[i]) + dnorm(alpha1[i-1],y11, vc.matrix[2,2] *tune, log =
     TRUE )
137: post.old2<- posterior(alpha0[i], alpha1[i-1], alpha2[i-1],alpha3[i-1]
138:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
139:            ,p[i]) + dnorm(y11,alpha1[i-1], vc.matrix[2,2] *tune , log =
     TRUE  )
140: x3<- ratio(post.cand2, post.old2, alpha1[i-1], y11, k2)
141: alpha1[i]<- x3[1]
142: k2<- x3[2]
143: ####################################
144: post.cand3<- posterior(alpha0[i], alpha1[i], y12,alpha3[i-1]
145:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
146:            ,p[i])+ dnorm( alpha2[i-1],y12, vc.matrix[3,3] *tune, log =
     TRUE )
147: post.old3<- posterior(alpha0[i], alpha1[i], alpha2[i-1],alpha3[i-1]
148:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
149:            ,p[i]) + dnorm(y12,alpha2[i-1], vc.matrix[3,3] *tune , log =
     TRUE  )
150: x4<- ratio(post.cand3, post.old3, alpha2[i-1],  y12, k3)
151: alpha2[i] <- x4[1]
152: k3<- x4[2]
153: ####################################
154: post.cand4<- posterior(alpha0[i], alpha1[i], alpha2[i]
155:            ,y13, beta0[i-1],beta1[i-1], beta2[i-1]
156:            ,beta3[i-1],p[i]) + dnorm(alpha3[i-1],y13, vc.matrix[4,4] *tune,
      log = TRUE )
157: post.old4<- posterior(alpha0[i], alpha1[i], alpha2[i]
158:            ,alpha3[i-1], beta0[i-1],beta1[i-1], beta2[i-1]
159:            ,beta3[i-1],p[i]) + dnorm(y13,alpha3[i-1], vc.matrix[4,4] *tune,
      log = TRUE  )
160: x5<- ratio(post.cand4, post.old4, alpha3[i-1], y13, k4)
161: alpha3[i]<- x5[1]
162: k4<- x5[2]
163:
164: ##########################   lambda2     ######################
165: post.cand6<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
166:            , y20,beta1[i-1], beta2[i-1],beta3[i-1]
167:            ,p[i]) + dnorm(beta0[i-1],y20, vc.matrix[5,5] *tune, log = TRUE
     )
```

```
168: post.old6<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
169:            , beta0[i-1],beta1[i-1], beta2[i-1],beta3[i-1]
170:            ,p[i]) + dnorm(y20,beta0[i-1], vc.matrix[5,5] *tune , log =
     TRUE  )
171: x6<- ratio(post.cand6, post.old6, beta0[i-1], y20, k6)
172: beta0[i]<- x6[1]
173: k6<- x6[2]
174: ####################################
175: post.cand7<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
176:            , beta0[i],y21, beta2[i-1],beta3[i-1]
177:            ,p[i]) + dnorm( beta1[i-1],y21, vc.matrix[6,6] * 2.4, log =
     TRUE )
178: post.old7<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
179:            , beta0[i],beta1[i-1], beta2[i-1],beta3[i-1]
180:            ,p[i]) + dnorm(y21,beta1[i-1], vc.matrix[6,6] * 2.4 , log =
     TRUE  )
181: x7<- ratio(post.cand7, post.old7, beta1[i-1], y21, k7)
182: beta1[i]<- x7[1]
183: k7<- x7[2]
184: ####################################
185: post.cand8<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
186:            , beta0[i],beta1[i], y22 ,beta3[i-1]
187:            ,p[i])+ dnorm( beta2[i-1],y22, vc.matrix[7,7] *tune, log = TRUE
     )
188: post.old8<- posterior(alpha0[i], alpha1[i], alpha2[i],alpha3[i]
189:            , beta0[i],beta1[i], beta2[i-1],beta3[i-1]
190:            ,p[i]) + dnorm(y22,beta2[i-1], vc.matrix[7,7] *tune , log =
     TRUE  )
191: x8<- ratio(post.cand8, post.old8, beta2[i-1],  y22, k8)
192: beta2[i] <- x8[1]
193: k8<- x8[2]
194: ####################################
195: post.cand9<- posterior(alpha0[i], alpha1[i], alpha2[i]
196:            ,alpha3[i], beta0[i],beta1[i], beta2[i]
197:            ,y23,p[i]) + dnorm(beta3[i-1],y23, vc.matrix[8,8] *tune, log =
     TRUE )
198: post.old9<- posterior(alpha0[i], alpha1[i], alpha2[i]
199:            ,alpha3[i], beta0[i],beta1[i], beta2[i]
200:            ,beta3[i-1],p[i]) + dnorm(y23,beta3[i-1], vc.matrix[8,8]*tune,
     log = TRUE  )
201: x9<- ratio(post.cand9, post.old9, beta3[i-1], y23, k9)
202: beta3[i]<- x9[1]
203: k9<- x9[2]
204:
205:
206: ###
207: print(c(i,alpha0[i],alpha1[i], alpha2[i],alpha3[i]
208:     ,beta0[i],beta1[i],beta2[i],beta3[i], p[i]))
209:  }
210:
211: print(c(k1/m,k2/m,k3/m,k4/m,k5/m, k6/m,k7/m,k8/m,k9/m))
212: return(cbind(alpha0, alpha1,alpha2,alpha3,beta0,beta1,beta2,beta3, p))
213: }
```

R.Code 4: The log-likelihood, the posterior distribution and the M-H algorithm for the ZPD model with 9 parameters.

D.3 Dynamic zero inflated poisson models

```r
### Probability distribution function for ZPD distribution
library(VGAM)
dzskellam<- function(z, mu1, mu2, p, log =FALSE){
          ifelse(z == 0, u <- 1, u<- 0)
          dzpd <- p * u + (1 - p) * dskellam(z, mu1, mu2)
          if(log) dzpd <- log(dzpd)
          return(dzpd)
}
################################################################################
#####
#####
#mu<- (1-p) * (lambda_1 - lambda_2)
### Simulation codes where z_i ~ ZPD(lambda_1, \lambda_2, p), when p = 0 we
## we similuate from a PD distribution
zpd.lambda<- function(n,theta1, theta2, sigma1 , sigma2, p){
             x1<- numeric(n+1)
             x2<- numeric(n+1)
             index<- numeric(n+1)
             l.lambda1<- numeric(n+1)
             l.lambda2<- numeric(n+1)
             x1[1]<- rnorm(1, 0,1)
             x2[1]<- rnorm(1, 0,1)
             l.lambda1[1] <- x1[1]
             l.lambda2[1] <- x2[1]
            for(i in 2:(n+1)){
             x1[i]<- rnorm(1, theta1 * x1[i-1], sd = sigma1)
             x2[i]<- rnorm(1, theta2 * x2[i-1], sd = sigma2)
             l.lambda1[i] <- x1[i] + my.const
             l.lambda2[i] <- x2[i] + my.const
             u<- rbinom(1,1,p)
             ifelse(u == 1, index[i] <- 0 , index[i]<- rskellam(1,
    exp(l.lambda1[i]), exp(l.lambda2[i])))

             }

return(list(x1 =x1, x2 = x2,l.lambda1 = l.lambda1, l.lambda2 = l.lambda2,
    index = index,theta1 = theta1
            ,theta2 = theta2, sigma2 = sigma2, sigma1 = sigma1))
}
```

R.Code 5: R codes for simulating from the DZPD model.

```
## Sequential Monte Carlo methods: Liu and West's particle filter algorithm
## Version 12, last modified: 30/09/2011
## z is our observed, X1 and X2 are unobserved processes
###A DZPD model  z ~ ZPD(lambda1, lambda2) where
## log(lambda) = X_t + h(p)     and X_t = X_{t-1} + sigma^2
n<- length(z) - 1       ## number of time points
N<- 5000                ## number of particles
a<- 0.975               ## Shrinkage constant
N0<- N/2                ## Threshold for the number of effective sample
size
#
# Initialise values
# NOTE: logp denotes -log(1-p) > 0
ess<- rep(NA, n+1)
X1<- matrix(NA_real_, nrow = N, ncol = (n+1))
X2<- matrix(NA_real_, nrow = N, ncol = (n+1))
## l.sigma1 = log(sigma1), l.sigma2 = log(sigma2)
l.sigma1<- matrix(NA_real_, nrow = N, ncol = (n+1))
l.sigma2<- matrix(NA_real_, nrow = N, ncol = (n+1))
t1<-  matrix(NA_real_, nrow = N, ncol = (n+1))
t2<-  matrix(NA_real_, nrow = N, ncol = (n+1))
## P.est   is the logit(p)
p.est<- matrix(NA_real_, nrow = N, ncol = (n+1))
## Weights
w<- matrix(NA_real_, nrow = N, ncol = (n+1))
## One-step ahead redictions
Preds<- matrix(NA_real_, nrow = N, ncol = (n+1))
#
# Initilise the algorithm
X1[,1]<- rnorm(N, 0, 1)
X2[,1]<- rnorm(N, 0, 1)
l.sigma1[,1]<- rnorm(N, 0,2)
l.sigma2[,1]<- rnorm(N, 0,2)
t1[,1]<- rnorm(N, 0,0.25)
t2[,1]<- rnorm(N, 0,0.25)
p.est[,1]<- rnorm(N, 0,1)
w[,1]<- rep(1/N, N)
Preds[,1]<- z[2]

i<-2
#
# set i = 2 to n+1
#
for(i in 2:(n+1)){
### Obtain plug-in extimates of theta1 and theta2
meant1<- weighted.mean(t1[,i-1], w[,i-1])
vart1<- weighted.mean((t1[,i-1] - meant1)^2, w[,i-1])
meant2<- weighted.mean(t2[,i-1], w[,i-1])
vart2<- weighted.mean((t2[,i-1] - meant2)^2, w[,i-1])

mut1<- a * t1[,i-1] + meant1 * (1 - a)
sigma2t1 <- (1 - a^2) * vart1
mut2<- a * t2[,i-1] + meant2 * (1 - a)
sigma2t2 <- (1 - a^2) * vart2
### Obtain plug-in extimates of l.sigma1 and l.sigma2
meansigma1<- weighted.mean(l.sigma1[,i-1], w[,i-1])
varsigma1<- weighted.mean((l.sigma1[,i-1] - meansigma1)^2, w[,i-1])
meansigma2<- weighted.mean(l.sigma2[,i-1], w[,i-1])
varsigma2<- weighted.mean((l.sigma2[,i-1] - meansigma2)^2, w[,i-1])
##
musigma1<- a * l.sigma1[,i-1] + meansigma1 * (1 - a)
sigma2sigma1 <- (1 - a^2) * varsigma1
```

```
musigma2<- a * l.sigma2[,i-1] + meansigma2 * (1 - a)
sigma2sigma2 <- (1 - a^2) * varsigma2


###  Obtain plug-in extimates of  logit(p)
meanp <- weighted.mean(p.est[,i-1], w[,i-1])
varp <- weighted.mean((p.est[,i-1] - meanp)^2, w[,i-1])

mup<- a * p.est[,i-1] + (1 - a) * meanp
sigma2p<- (1 - a^2) * varp
########
#
# Draw auxilary indicator variable "aux"
#
p.hat<- exp(mup)/(1 + exp(mup))
const.est <- -log(1 - p.hat)
lambda1.hat<- exp(mut1 * X1[,i-1] + const.est)
lambda2.hat<- exp(mut2 * X2[,i-1] + const.est)#(f2 * theta2 * X2[,i-1])^2
prob1<- w[,i-1] * dzskellam(z[i], lambda1.hat, lambda2.hat, p.hat)
aux<-  sample(1:N, N, replace = TRUE, prob = prob1)

#
#
## Draw samples from proposal distributions
#
p.est[,i]<- rnorm(N, mup[aux], sqrt(sigma2p))
x.ests<-  exp(p.est[,i])/(1 + exp(p.est[,i]))
l.sigma2[,i] <- rnorm(N, musigma2[aux], sqrt(sigma2sigma2))
l.sigma1[,i] <- rnorm(N, musigma1[aux], sqrt(sigma2sigma1))
t1[,i] <- rnorm(N, mut1[aux], sqrt(sigma2t1))
t2[,i] <- rnorm(N, mut2[aux], sqrt(sigma2t2))
const.new<- -log(1 - x.ests)
X2[,i]<- rnorm(N, mean = t2[,i] * X2[aux,i-1], sd = sqrt(exp(l.sigma2[,
    i])))
X1[,i]<- rnorm(N, mean = t1[,i] * X1[aux,i-1], sd = sqrt(exp(l.sigma1[,
    i])))
lambda1.h<- exp(X1[,i] + const.new)
lambda2.h<- exp(X2[,i] + const.new)
## Obtainun-normalised weights
w[,i]<- exp(dzskellam(z[i], lambda1.h, lambda2.h,x.ests, log = TRUE) -
       dzskellam(z[i], lambda1.hat[aux], lambda2.hat[aux],p.hat[aux], log
    = TRUE))
## Normalise weights
w[,i] <- w[,i] / sum(w[,i])
#
## Calcualte the effective sample
#
N.eff<- 1/crossprod(w[,i])
ess[i]<- N.eff
## Perform resampling if ess < N0
if(N.eff < N0) {
        resample<- sample(1:N, N, replace = TRUE, prob  = w[,i])
        X1[,1:i]<- X1[resample, 1:i]
        X2[,1:i]<- X2[resample, 1:i]
        l.sigma1[,1:i]<- l.sigma1[resample,1:i]
        l.sigma2[,1:i]<- l.sigma2[resample,1:i]
        t1[,1:i] <- t1[resample,1:i]
        t2[,1:i] <- t2[resample,1:i]
        p.est[,1:i]<- p.est[resample,1:i]
        w[,i]<- rep(1/N, N)

}
```

```
121: ### One-step ahead predictions
122: p.new<-  exp(p.est[,i])/(1 + exp(p.est[,i]))
123: c.new<- -log(1 - p.new)
124: lambda1.new<- exp(X1[,i] * t1[,i] + c.new)
125: lambda2.new<- exp(X2[,i] * t2[,i] + c.new)
126: pred<- ZPD.sims(lambda1.new, lambda2.new, p.new)
127: Preds[,i]<- pred
128: print(c(i, ess[i]))
129: }
```

R.Code 6: R codes the LWF algorithm for the basic DZPD model. Same codes can be implemented for the DPD model when $p = 0$.

```
## Sequential Monte Carlo methods: Liu and West's particle filter algorithm
## Version 4, last modified: 30/09/2011
## z is our observed, X1 and X2 are unobserved processes
##A DZPD model  z ~ ZPD(lambda1, lambda2) where
##
## log(lambda) = X_t + h(p) + alpha * log(d.t + 1)
## and X_t = gamma * X_{t-1}+ v * X_{t-1} + sigma^2
##
n<- length(z) - 1       ## number of time points
N<- 5000                ## number of particles
a<- 0.975               ## Shrinkage constant
N0<- N/2                ## Threshold for the number of effective sample
size
## The ff package provides data structures that are stored on disk but
behave
# (almost) as if they were in RAM by transparently mapping only a section
# (pagesize) in main memory
library(ff)
#####################
## In order to define a data set with "ff" format we need first to save it
as an "ff" file:
X1<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/X1-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
X2<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/X2-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
Preds<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/oneCovPre
,vmode="double", dim=c(N,Total.n))
gamma1.est<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/g1-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
gamma2.est<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/g2-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
l.v1<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/v1-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))     ## variance for the second state
equation
l.v2<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/v2-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))      ## variance for the second state
equation
l.sigma1<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/s1-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
l.sigma2<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/s2-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
p.est<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/p-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
```

```
cov1.est<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/cov1-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
cov2.est<- ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/cov2-
oneCov.ff"
,vmode="double", dim=c(N,Total.n))
w<-ff(filename="C://Documents and
Settings/gshahtahmassebi/Desktop/OneDayAnalysis/Codes/SMC/Results/weightsCo
,vmode="double", dim=c(N,Total.n))

#
# Initilise sampling from the prior
#
# Initilise sampling from the prior
Preds[,1]<- rep(z[2], N)
k<- seq(0,44800, by = 200)
d <- rev(k)
#
for(j in 1:length(k)){
## After k = 200 observations initialise the PF algorithm ,
## "aa" and "dd" are counters
 aa<- (2+ k[j])
 dd<-((n-d[j]))
X1[,1+k[j]]<- rnorm(N, X1.mean, X1.sd)
X2[,1+k[j]]<- rnorm(N, X2.mean, X2.sd)
p.est[,1+k[j]]<-  rnorm(N,p.mean, p.max)
##
gamma1.est[,1+k[j]]<-  rnorm(N, g1.mean, g1.sd)
gamma2.est[,1+k[j]]<-  rnorm(N, g2.mean, g2.sd)
## l.v1 = log(v1), l.v2 = log(v2), l.sigma1 = log(sigma1), l.sigma2 =
log(sigma2)
l.v1[,1+k[j]]<- rnorm(N, v1.mean, v1.sd)
l.v2[,1+k[j]]<- rnorm(N, v2.mean, v2.sd)
l.sigma1[,1+k[j]]<- rnorm(N, s1.mean,s1.sd)
l.sigma2[,1+k[j]]<- rnorm(N, s2.mean,s2.sd)
## Coeficients for the covariates in the model
cov1.est[,1+k[j]]<- rnorm(N, cov1.mean, cov1.sd)
cov2.est[,1+k[j]]<- rnorm(N, cov2.mean, cov2.sd)
w[,1+k[j]]<- rep(1/N, N)

for(i in aa:dd){
## The following codes obtain plug-in estiamtes of the model parameters
#### coefficients  for lambda1
meancov1 <- weighted.mean(cov1.est[,i-1], w[,i-1])
varcov1 <- weighted.mean((cov1.est[,i-1] - meancov1)^2, w[,i-1])

mucov1<- a * cov1.est[,i-1] + (1 - a) * meancov1
sigma2cov1<- (1 - a^2) * varcov1

cov1.hat<-  mucov1 *  log.time[i]  #   log.volume[i]
#### coefficients  for lambda2
meancov2 <- weighted.mean(cov2.est[,i-1], w[,i-1])
varcov2 <- weighted.mean((cov2.est[,i-1] - meancov2)^2, w[,i-1])

mucov2<- a * cov2.est[,i-1] + (1 - a) * meancov2
sigma2cov2<- (1 - a^2) * varcov2

cov2.hat<- mucov2 * log.time[i]
#### gamma1 and gamma2
meang1 <- weighted.mean(gamma1.est[,i-1], w[,i-1])
```

```
 95: varg1 <- weighted.mean((gamma1.est[,i-1] - meang1)^2, w[,i-1])
 96:
 97: mug1<- a * gamma1.est[,i-1] + (1 - a) * meang1
 98: sigma2g1<- (1 - a^2) * varg1
 99:
100: meang2 <- weighted.mean(gamma2.est[,i-1], w[,i-1])
101: varg2 <- weighted.mean((gamma2.est[,i-1] - meang2)^2, w[,i-1])
102:
103: mug2<- a * gamma2.est[,i-1] + (1 - a) * meang2
104: sigma2g2<- (1 - a^2) * varg2
105: ### v1 and v2
106: meanv1<- weighted.mean(l.v1[,i-1], w[,i-1])
107: varv1<- weighted.mean((l.v1[,i-1] - meanv1)^2, w[,i-1])
108: meanv2<- weighted.mean(l.v2[,i-1], w[,i-1])
109: varv2<- weighted.mean((l.v2[,i-1] - meanv2)^2, w[,i-1])
110:
111: muv1<- a * l.v1[,i-1] + meanv1 * (1 - a)
112: sigma2v1 <- (1 - a^2) * varv1
113: muv2<- a * l.v2[,i-1] + meanv2 * (1 - a)
114: sigma2v2 <- (1 - a^2) * varv2
115: ###sigma1 and sigma2
116: meansigma1<- weighted.mean(l.sigma1[,i-1], w[,i-1])
117: varsigma1<- weighted.mean((l.sigma1[,i-1] - meansigma1)^2, w[,i-1])
118: meansigma2<- weighted.mean(l.sigma2[,i-1], w[,i-1])
119: varsigma2<- weighted.mean((l.sigma2[,i-1] - meansigma2)^2, w[,i-1])
120:
121: musigma1<- a * l.sigma1[,i-1] + meansigma1 * (1 - a)
122: sigma2sigma1 <- (1 - a^2) * varsigma1
123: musigma2<- a * l.sigma2[,i-1] + meansigma2 * (1 - a)
124: sigma2sigma2 <- (1 - a^2) * varsigma2
125: ### p.hat
126: meanp <- weighted.mean(p.est[,i-1], w[,i-1])
127: varp <- weighted.mean((p.est[,i-1] - meanp)^2, w[,i-1])
128:
129: mup<- a * p.est[,i-1] + (1 - a) * meanp
130: sigma2p<- (1 - a^2) * varp
131: alphap <- (((1 - mup)*mup^2)/sigma2p) - mup
132: betap <- (((1 - mup)^2*mup)/sigma2p) -(1- mup)
133: ########    Here we draw auxiliary variable with  the following
     probability:
134: ##   w[,i-1] * dzskellam(z[i], lambda1.hat, lambda2.hat, p.hat)
135: #
136: # Draw auxilary indicator variable
137: #
138: const.est <- mup#-log(1-p)
139: lambda1.hat<- exp(mug1 * X1[,i-1] + const.est + cov1.hat)
140: lambda2.hat<- exp(mug2 * X2[,i-1] + const.est + cov2.hat)
141: p.hat<- exp(mup)/(1 + exp(mup))
142: prob1<- w[,i-1] * dzskellam(z[i], lambda1.hat, lambda2.hat, p.hat)
143: aux<-  sample(1:N, N, replace = TRUE, prob = prob1)
144:
145: #
146: #
147: ## Draw the samples from proposal densities for each of the model
     parameters:
148: #
149: cov1.est[,i]<- rnorm(N, mucov1[aux], sqrt(sigma2cov1))
150: cov2.est[,i]<- rnorm(N, mucov2[aux], sqrt(sigma2cov2))
151: p.est[,i]<- rnorm(N, mup[aux], sqrt(sigma2p))
152: l.sigma1[,i] <- rnorm(N, musigma1[aux], sqrt(sigma2sigma1))
153: l.sigma2[,i] <- rnorm(N, musigma2[aux], sqrt(sigma2sigma2))
154: l.v1[,i] <- rnorm(N, muv1[aux], sqrt(sigma2v1))
```

```
155: l.v2[,i] <- rnorm(N, muv2[aux], sqrt(sigma2v2))
156: gamma1.est[,i] <- rnorm(N, mug1[aux], sqrt(sigma2g1))
157: gamma2.est[,i] <- rnorm(N, mug2[aux], sqrt(sigma2g2))
158: const.new<- p.est[,i]
159: var.est1<- exp(l.sigma1[,i]) + exp(l.v1[,i])*( X1[aux,i-1])^2
160: var.est2<- exp(l.sigma2[,i]) + exp(l.v2[,i])*( X2[aux,i-1])^2
161: X1[,i]<- rnorm(N, mean = gamma1.est[,i] * X1[aux,i-1], sd = sqrt(var.est1))
162: X2[,i]<- rnorm(N, mean = gamma2.est[,i] * X2[aux,i-1], sd = sqrt(var.est2))
163: coef1.new<- cov1.est[,i] * log.time[i]
164: coef2.new<- cov2.est[,i] * log.time[i]
165: lambda1.h<- exp(X1[,i] + const.new + coef1.new)
166: lambda2.h<- exp(X2[,i] + const.new + coef2.new)
167: p.hat2<- exp(p.est[,i])/(1 + exp(p.est[,i]))
168: ###
169: # Calculate the un-normalised weight
170: w[,i]<- exp(dzskellam(z[i], lambda1.h, lambda2.h, p.hat2 , log = TRUE) -
171: dzskellam(z[i], lambda1.hat[aux], lambda2.hat[aux], p.hat2, log = TRUE))
172: #
173: w[,i] <- w[,i] / sum(w[,i])     ## Normalise the weight
174: #
175: ## effective sample
176: #
177: N.eff<- 1/crossprod(w[,i])
178:
179: ess[i]<- N.eff
180:
181: if(N.eff < N0) {
182: resample<- sample(1:N, N, replace = TRUE, prob  = w[,i])
183: X1[,(aa):i]<- X1[resample, (aa):i]
184: X2[,(aa):i]<- X2[resample, (aa):i]
185: gamma1.est[,(aa):i]<- gamma1.est[resample,(aa):i]
186: gamma2.est[,(aa):i]<- gamma2.est[resample,(aa):i]
187: l.sigma1[,(aa):i]<- l.sigma1[resample,(aa):i]
188: l.sigma2[,(aa):i]<- l.sigma2[resample,(aa):i]
189: l.v1[,(aa):i]<- l.v1[resample,(aa):i]
190: l.v2[,(aa):i]<- l.v2[resample,(aa):i]
191: p.est[,(aa):i]<- p.est[resample,(aa):i]
192: cov1.est[,(aa):i]<- cov1.est[resample,(aa):i]
193: cov2.est[,(aa):i]<- cov2.est[resample,(aa):i]
194: w[,i]<- rep(1/N, N)
195:
196: }
197: print(c(i, ess[i]))
198: ### one-step ahead prediction
199: const.n<- p.est[,i]
200: cov1.n<- cov1.est[,i] *  log.time[i+1]
201: cov2.n<- cov2.est[,i] *  log.time[i+1]
202: lambda1.new<- exp(gamma1.est[,i] *X1[,i] + const.n + cov1.n)
203: lambda2.new<- exp(gamma2.est[,i] *X2[,i] + const.n + cov2.n)
204: p.new<- exp(p.est[,i])/(1 + exp(p.est[,i]))
205: Preds[,i]<- ZPD.sims(lambda1.new, lambda2.new, p.new)
206: }
207: ### After k = 200 time points, summarize the latest sampled values  and
     start the
208: ## algorithm again
209: X1.mean  <- weighted.mean(X1[,i-2], w[,i-2])
210: X2.mean  <- weighted.mean(X2[,i-2], w[,i-2])
211: g1.m       <- weighted.mean(gamma1.est[,i-2], w[,i-2])
212: g2.m       <- weighted.mean(gamma2.est[,i-2], w[,i-2])
213: g1.s      <- sqrt(wt.var(gamma1.est[,i-2], w[,i-2]))
214: g2.s      <- sqrt(wt.var(gamma2.est[,i-2], w[,i-2]))
215: g1.sd  <- g1.s
```

```
216: g2.sd  <- g2.s
217: g1.mean  <-  g1.m
218: g2.mean  <-  g2.m
219: cov1.mean       <- weighted.mean(cov1.est[,i-2], w[,i-2])
220: cov2.mean       <- weighted.mean(cov2.est[,i-2], w[,i-2])
221: cov1.sd      <- sqrt(wt.var(cov1.est[,i-2], w[,i-2]))
222: cov2.sd      <- sqrt(wt.var(cov2.est[,i-2], w[,i-2]))
223: v1.mean  <- weighted.mean(l.v1[,i-2], w[,i-2])
224: v2.mean  <- weighted.mean(l.v2[,i-2], w[,i-2])
225: s1.mean  <- weighted.mean(l.sigma1[,i-2], w[,i-2])
226: s2.mean  <- weighted.mean(l.sigma2[,i-2], w[,i-2])
227: p.mean  <- weighted.mean(p.est[,i-2], w[,i-2])
228: p.sd  <- sqrt(wt.var(p.est[,i-2], w[,i-2]))
229: X1.sd  <- sqrt(wt.var(X1[,i-2], w[,i-2]))
230: X2.sd  <- sqrt(wt.var(X2[,i-2], w[,i-2]))
231: v1.sd  <- sqrt(wt.var(l.v1[,i-2], w[,i-2]))
232: v2.sd  <- sqrt(wt.var(l.v2[,i-2], w[,i-2]))
233: s1.sd  <- sqrt(wt.var(l.sigma1[,i-2], w[,i-2]))
234: s2.sd  <- sqrt(wt.var(l.sigma2[,i-2], w[,i-2]))
235: }
236:
```

R. Code 7: R codes for the LWF algorithm for the DZPD model with a non-linear state equation and covariate.

D.4 Generalised Poisson difference distribution

```r
## Generalised Poisson distribution: following codes evaluate the density of the
# GP distribution and draw random sample from it.
## Version 4, modified on 23/03/2011
##
G.p<- function(x, lambda, theta){
(lambda * (lambda + x * theta)^(x-1) * exp(-lambda - x * theta))/factorial(x)
}
### Function "rgpoisson" draws samples from the GP distribution, with the
##  following arguments:
## "n" the size of the random verctor, "lambda" and "theta" parameters of the
## GP distribution
rgpoisson <- function(n , lambda, theta){
        rands<- rep(0, n)
        lambda <- lambda
        theta <- theta
        for( i in 1:n){
              N<- 0
              p<- G.p(N, lambda, theta)
              F.p <- p
              U<- runif(1, 0, 1)
              while(U > F.p){
                      N<- N+1
                      p<- G.p(N, lambda, theta)
                      F.p <- F.p + p
                }
            rands[i]<- N
          }
        return(rands)
  }
```

R.Code 8: R codes for evaluating the density of the GP distribution and draw random samples from it.

```r
## Generalised Poisson difference distribution (GPD)
#  Version 4, modified on 24/07/2011, modified for thesis on 30/11/2011
#
## The following function evaluates the GPD density :
##  at point "d" over an infinit range of "aa"
## with parameters of lambda1 , theta1 and lambda2, theta2
d.GPD1<- function(d, aa, lambda1, lambda2, theta1, theta2, log = FALSE){
## Note for negative values of "d", "y" cannot start from 0.
# Thus we set the following condition.
                ifelse(d < 0, y <- abs(d):aa[length(aa)], y<-aa)
## The density of the GPD distribution at point "d" is calculated in the
## following four parts
                part1<- (lambda1 *(lambda1 + (y+d) * theta1)^(y + d -
1))/gamma(y+d+1)
                part2<- (lambda2 * (lambda2 + y * theta2)^(y-1))/ gamma(y+1)
                part3<-  exp(-y * (theta1 + theta2))
                Total<- sum(part1 * part2 * part3) * exp(-lambda1 - lambda2
- d *  theta1 )
## Resturn the value of GPD density
                ifelse(log = TRUE, GPD<- log( Total),
                GPD<- Total )
                return(GPD)
         }

## A function for evaluating the GPD density over a range of "d"
# Input "d" as a vector e.g. d<- -10:10
GPD.d<- function(d, aa, lambda1, lambda2, theta1, theta2, log = FALSE){
## let length(d) = nd
      nd<- length(d)
      f.ds<- rep(0,nd)
      for(i in 1:nd){
        f.ds[i]<- d.GPD1( d[i], aa, lambda1, lambda2, theta1, theta2, log =
FALSE)
      }
      return(f.ds)
      }
```

R.Code 9: R codes for evaluating the density of the GPD distribution.

Glossary of terms

This glossary of terms was constructed using definitions from O'Hara (2006), Tsay (2005), Hull (2008) and Pass et al. (2005).

Ask Price The price that a dealer is offering to sell an asset.

Bid Price The price that a dealer is prepared to pay for an asset.

Bid-Ask Spread The amount by which the ask price exceeds the bid price.

Futures Contract A contract that obliges the holder to buy or sell an asset at a predetermined delivery price during a specified future time period. The contract is settled daily.

Futures Price The delivery price currently applicable to a futures contract.

Hedge Fund A pool of capital which fund managers (for example, international banks) used to speculate on the foreign exchanges, stock and commodity markets. Fund managers aim to make windfall profit by "correctly" guessing future price movements. Their activities, which become increasingly global and largely unsupervised by national regulatory frameworks, have on occasion, served to destabilise the financial markets.

Hedge A trade design to reduce risk.

Index Futures A futures contract on a stock index or other index.

Market Liquidity The ability to buy or sell significant quantities of a security quickly, anonymously, and with little price impact.

Market Maker A trader who is willing to quote both bid and offer prices for an asset.

Market Microstructure The study of the process and outcomes of exchanging assets under explicit trading rules.

Maturity Date The end of the life of a contract.

Open Outcry System of trading where traders meet on the floor of the exchange.

Option The right to buy or sell an asset.

Tick The minimum change in price that a security can have, either up or down. Price changes in high-frequency data only happen at multiple of tick.

List of references

Abramowitz, M. and I. A. Stegun (1964). *Handbook of mathematical functions with formulas, graphs, and mathematical tables* (ninth Dover printing, tenth GPO printing ed.). New York: Dover.

Adler, D., C. Glæser, O. Nenadic, J. Oehlschlägel, and W. Zucchini (2010). *ff: memory-efficient storage of large data on disk and fast access functions*. R package version 2.1-2.

Agarwal, D. K., A. E. Gelfand, and S. Citron-Pousty (2002). Zero-inflated models with application to spatial count data. *Environmental and Ecological Statistics 9*, 341–355. 10.1023/A:1020910605990.

Albert, J. (2007). *Bayesian computation with R*. New York: Springer.

Alzaid, A. A. and M. A. Omair (2010). On the Poisson difference distribution inference and applications. *Bulletin of the Malaysian Mathematical Sciences Society 33*(1), 17–45.

Andrieu, C., N. de Freitas, and A. Doucet (1999). Sequential MCMC for bayesian model selection. In *IEEE Higher Order Statistics Workshop, Ceasarea*, pp. 130–134.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(3), 269–342.

Bauwens, L. and P. Giot (2000). The logarithmic ACD model: an application to the bid-ask quote process of three NYSE stocks. *Annales d'Economie et de Statistique 60*, 117–149.

Berzuini, C., N. G. Best, W. R. Gilks, and C. Larizza (1997). Dynamic conditional independence models and Markov chain Monte carlo methods. *Journal of the American Statistical Association 92*, 1403–1412.

Bien, K., I. Nolte, and W. Pohlmeier (2011). An inflated multivariate integer count hurdle model: an application to bid and ask quote dynamics. *Journal of Applied Econometrics 26*(4), 669–707.

Bollerslev, T. and M. Melvin (1994). Bid–Ask spreads and volatility in the foreign exchange market: an empirical analysis. *Journal of International Economics 36*(3-4), 355 – 372.

Brooks, S. P. and A. Gelman (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics 7*(4), 457–472.

Calamia, A. (1999). Market microstructure: Theory and empirics. LEM papers series, Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, Pisa, Italy.

Cameron, A. C. and P. K. Trivedi (1996). Count data models for financial data. In *Handbook of Statistics*, pp. 363–391. Elsevier, North-Holland.

Campbell, J. Y., A. W. Lo, and A. C. MacKinlay (1996). *The econometrics of financial markets*. New Jersey: Princeton University Press.

Carvalho, C. M. and H. F. Lopes (2007). Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis 51*(9), 4526 – 4542.

Chatfield, C. (2001). *Time-series forcasting*. London: Chapman & Hall/CRC.

Consul, P. (1989). *Generalized Poisson distributions: properties and applications*. Statistics, textbooks and monographs. M. Dekker.

Consul, P. C. and F. Famoye (2006). *Lagrangian probability distributions*. Boston: Birkhäuser.

Consul, P. C. and G. C. Jain (1973). A generalization of the Poisson distribution. *Technometrics 15*(4), 791–799.

Cont, R. (2011). Statistical modeling of high-frequency financial data. *Signal Processing Magazine, IEEE 28*(5), 16 –25.

Creal, D. (2009). A survey of sequential Monte Carlo methods for economics and finance. Serie Research Memoranda 0018, VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics.

Czado, C. and A. Kolbe (2004). Empirical study of intraday option price changes using extended count regression models.

Darolles, S., C. Gourieroux, and G. L. Fol (2000). Intraday transaction price dynamics. *Annales d'Economie et de Statistique* (60), 10.

Doucet, A., N. de Freitas, and N. Gordon (2001). An introduction to sequential Monte Carlo methods. In A. Doucet, N. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo in practice*, Chapter 1, pp. 3–13. New York: Springer-Verlag.

Doucet, A. and A. M. Johansen (2009). A tutorial on particle filtering and smoothing: fifteen years later. In *Handbook of Nonlinear Filtering (eds)*. Oxford University Press.

Engle, R. F. (2000). The econometrics of ultra-high frequency data. *Econometrica 68*(1), 1–22.

Engle, R. F. and J. R. Russell (1998). Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica 66*(5), pp. 1127–1162.

Engle, R. F. and J. R. Russell (2005). Analysis of high frequency financial data. In Y. Ait-Sahalia and L. Hansen (Eds.), *Handbook of Financial Econometrics*. North Holland.

Famoye, F. (2010). A new bivariate generalized Poisson distribution. *Statistica Neerlandica 64*(1), 112–124.

Fearnhead, P. (2002). Markov chain Monte Carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics 11*(4), 848–862.

Fearnhead, P., O. Papaspiliopoulos, G. O. Roberts, and A. Stuart (2010). Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 497–512.

Gamerman, D. and H. F. Lopes (2006). *Markov chain Monte Carlo: stochastic simulation for bayesian inference*. Boca Raton: Chapman&Hall.

Gelman, A., J. B. Carlin, H. S. Stern, and R. B. Donald (2003, July). *Bayesian Data Analysis* (second ed.). London: Chapman & Hall/CRC.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*(4), 457–472.

Gendron, R. and W. J. McCausland (2005). Analyse Bayèsienne d'un modèle pour les changements de prix de titres échangés à haute fréquence sur les marchés financiers. Master's thesis, Département des Sciences Économiques.

Gilks, W. R. and C. Berzuini (2001). Following a moving target–Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(1), 127–146.

Goodhart, C. A. E. and M. O'Hara (1997). High frequency data in financial markets: Issues and applications. *Journal of Empirical Finance 4*(3), 73–114.

Gourieroux, C., A. Monfort, and A. Trognon (1984, May). Pseudo maximum likelihood methods: Applications to Poisson models. *Econometrica 52*(3), 701–20.

Greene, W. H. (1994). Accounting for excess zeros and sample selection in Poisson and Negative Binomial regression models. Working papers ec-94-10, New York University, Leonard N. Stern School of Business, Department of Economics.

Harrison, P. J. and C. F. Stevens (1976). Bayesian forecasting. *Journal of the Royal Statistical Society: Series B (Methodological) 38*(3), 205–247.

Harvey, A. C. and C. Fernandes (1989). Time series models for count or qualitative observations. *Journal of Business & Economic Statistics 7*(4), 407–417.

Hausman, J. A., A. W. Lo, and A. MacKinlay (1992). An ordered probit analysis of transaction stock prices. *Journal of Financial Economics 31*(3), 319 – 379.

Hull, J. C. (2008, May). *Options, Futures, and Other Derivatives (7th Edition)*. New Jersey: Prentice Hall.

Hwang, Y., J.-S. Kim, and I.-S. Kweon (2007, June). Sensor noise modeling using the Skellam distribution: application to the color edge detection. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8.

Irwin, J. O. (1937). The frequency distribution of the difference between two independent variates following the same Poisson distribution. *Journal of the Royal Statistical Society 100*(3), 415–416.

Ismail, N. and A. A. Jemain (2007). Handling overdispersion with Negative Binomial and generalized Poisson regression models. *Casualty Actuarial Society Forum*.

Jasra, A. and P. D. Moral (2011). Sequential Monte Carlo methods for option pricing. *Stochastic Analysis and Applications 29*(2), 292–316.

Jiang, W., T. E. Murphy, and K.-L. Tsui (2006). *Springer handbook of engineering statistics*, Chapter 10, pp. 173 – 193. Springer.

Johannes, M. S., A. G. Korteweg, and N. Polson (2008). Sequential learning, predictive regressions, and optimal portfolio returns. Working paper, Graduate School of Business, Stanford Uuniversity.

Jr, F. E. H. and with contributions from many other users. (2010). *Hmisc: Harrell Miscellaneous*. R package version 3.8-0.

Karlis, D. and I. Ntzoufras (2006). Bayesian analysis of the differences of count data. *Statistics in Medicine 25*(11), 1885–1905.

Karlis, D. and I. Ntzoufras (2009). Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference. *IMA J Management Math 20*(2), 133–145.

Kim, S., N. Shephard, and S. Chib (1998). Stochastic volatility: likelihood inference and comparison with arch models. *The Review of Economic Studies 65*(3), 361–393.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics 5*(1), 1–25.

Li, R., M. Emmerich, J. Eggermont, E. Bovenkamp, T. Bäck, J. Dijkstra, and J. Reiber (2009). Optimizing a medical image analysis system using mixed-integer evolution strategies. In S. Cagnoni (Ed.), *Evolutionary Image Analysis and Signal Processing*, Volume 213 of *Studies in Computational Intelligence*, pp. 91–112. Berlin / Heidelberg: Springer.

Liesenfeld, R., I. Nolte, and W. Pohlmeier (2006). Modelling financial transaction price movements: a dynamic integer count data model. *Empirical Economics 30*(4), 795–825.

Liu, J. and M. West (2001). Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo in practice*, Chapter 10, pp. 271–293. New York: Springer-Verlag.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. Springer-Verlag.

Lopes, H. F. and R. S. Tsay (2011). Particle filters and Bayesian inference in financial econometrics. *Journal of Forecasting 30*(1), 168–209.

Luca, G. D. and G. M. Gallo (2009). Time-varying mixing weights in mixture autoregressive conditional duration models. *Econometric Reviews 28*(3).

Martin, A. D., K. M. Quinn, and J. H. Park (2010). *MCMCpack: Markov chain Monte Carlo (MCMC) Package*. R package version 1.0-6.

McCulloch, R. and R. Tsay (2001). Nonlinearity in high-frequency financial data and hierarchical models. *Studies in Nonlinear Dynamics & Econometrics 5*(1).

Mwalili, S. M. (2008). The zero-inflated Negative Binomial regression model with correction for misclassification: an example in caries research. *Statistical Methods in Medical Research 17*(2), 123–139.

Nikoloulopoulos, A. K. and D. Karlis (2008). On modeling count data: a comparison of some well-known discrete distributions. *Journal of Statistical Computation and Simulation 78*(3), 437–457.

Ntzoufras, I. (2009). *Bayesian modeling using WinBUGS*. Computational Statistics. New Jersey: Wiley.

O'Hara, M. (2006). *Market microstructure theory*. Oxford: Blackwell Publishing Limited.

Pacurar, M. (2008). Autoregressive conditional duration models in finance: A survey of the theoretical and empirical literature. *Journal of Economic Surveys 22*(4), 711–751.

Pass, C., B. Lowes, A. Pendleton, L. Chadwick, D. O'Reilly, and M. Afferson (2005). *Collins dictionary of business* (3rd ed.). Glasgow: HarperCollins.

Petris, G., S. Petrone, and P. Campagnoli (2009). *Dynamic Linear Models with R*. Use R! London: Springer.

Pitt, M. K. and N. Shephard (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association 94*(446), 590–599.

Polson, N. G., J. R. Stroud, and P. Müller (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(2), 413–428.

Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 54*(3), 507–554.

Russell, J. R. and R. F. Engle (1998). Econometric analysis of discrete-valued irregularly-spaced financial transactions data using a new autoregressive conditional multinomial model. *SSRN eLibrary: http://ssrn.com/paper=106528*.

Rydberg, T. H. and N. Shephard (1999). Dynamics of trade-by-trade price movements: Decomposition and models. *SSRN eLibrary: http://ssrn.com/paper=146189*.

Rydberg, T. H. and N. Shephard (2000). A modelling framework for the prices and times of trades made on the new york stock exchange. In W. J. Fitzgerald, R. L. Smith, A. T. Walden, and P. C. Young (Eds.), *Nonlinear and nonstationary signal processing*, Chapter 7, pp. 217–246. Cambridge: Cambridge University Press.

Rydberg, T. H. and N. Shephard (2003). Dynamics of trade-by-trade price movements: Decomposition and models. *Journal of Financial Econometrics 1*(1), 2–25.

Skellam, J. G. (1946). The frequency distribution of the difference between two Poisson variates belonging to different populations. *Journal of the Royal Statistical Society 109*(3), 296.

Smith, J. Q. (1979). A generalization of the Bayesian steady forecasting model. *Journal of the Royal Statistical Society: Series B (Methodological) 41*(3), pp. 375–387.

Smith, R. L. and J. E. Miller (1986). A non-Gaussian state space model and application to prediction of records. *Journal of the Royal Statistical Society: Series B (Methodological) 48*(1), pp. 79–88.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing 50*, 281–289.

Tanner, M. A. (1998). *Tools for statistical inference: models for the exploration of posterior distributions and likelihood functions*. New York: Springer.

Team, R. D. C. *R: a language and environment for statistical computing*.

Triantafyllopoulos, K. (2009). Inference of dynamic generalized linear models: On-line computation and appraisal. *International Statistical Review 77*(3), 430–450.

Tsay, R. S. (2005). *Analysis of financial time series* (second ed.). New Jersey: Wiley-Interscience.

Tse, Y. (1999). Market microstructure of FT-SE 100 index futures: An intraday empirical analysis. *Journal of Futures Markets 19*(1), 31–58.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.

West, M. (1993a). Approximating posterior distributions by mixture. *Journal of the Royal Statistical Society: Series B (Methodological) 55*(2), pp. 409–422.

West, M. (1993b). Mixture models, Monte Carlo, Bayesian updating and dynamic models. In J. H. Newton (Ed.), *Computing Science and Statistics:proceedings of the 24th symposium on the interface*, Virginia, pp. 325–333. Interface Foundation of North America.

West, M., P. J. Harrison, and H. S. Migon (1985). Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association 80*(389), 73–83.

Winkelmann, R. (2008). *Econometric analysis of count data* (fifth ed.). Berlin: Springer-Verlag.

Wood, R. A. (2000). Market microstructure research databases: History and projections. *Journal of Business & Economic Statistics 18*(2), 140–45.

Yan, B. and E. Zivot (2005). Analysis of high-frequency financial data with S-PLUS. Working papers, University of Washington, Department of Economics.

Yau, C., O. Papaspiliopoulos, G. O. Roberts, and C. Holmes (2011). Bayesian non-parametric hidden Markov models with applications in genomics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(1), 37–57.

Yee, T. W. (2010). The VGAM package for categorical data analysis. *Journal of Statistical Software 32*(10), 1–34.

Yee, T. W. (2011). *VGAM: Vector Generalized Linear and Additive Models*. R package version 0.8-4.

Yee, T. W. and C. J. Wild (1996). Vector generalized additive models. *Journal of Royal Statistical Society, Series B 58*(3), 481–493.

Zhang, M. Y., J. R. Russell, and R. S. Tsay (2001). A nonlinear autoregressive conditional duration model with applications to financial transaction data. *Journal of Econometrics 104*(1), 179 – 207.