Integration of Technical Trading Behaviour in Asset Pricing

Krassimir Vanguelov

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

2016

I, Krassimir Vanguelov, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis investigates methods applied to technical analysis based on particle filtering for detecting the presence of technical trading on foreign exchange and futures markets. The objective is to measure the intensity of that trading and its influence on short term price formation of the traded securities. Technical trading is a type of trading that is based on technical analysis. This is a method to form a view on the future development of the price of a traded security based on the currently observed pattern of the price itself. Technical analysis and trading strategies based on price patterns are not isolated phenomena. They are extremely popular among a very large group of financial markets professionals.

The experiments of this research rely on extensive amount of data. We use prices for a wide range of securities representing different markets, countries and liquidity profiles. More than ten years of data at different level of aggregation are processed in order to test the ideas during different economic cycles.

The research comprises an experimental software environment and three experiments:

1. Experimental software environment

   The experiments carried out in this project require a robust software environment and a significant amount of coding. An intuitive choice for this purpose is R[1], a language and environment for statistical computing. Most of the programming is done in R utilising the flexibility of the language and the environment. The code has been consolidated into an R package - a library module for the environment. Wherever better speed or interfacing to other systems are needed, additional modules have been developed.

2. Detection and tracking of technical trading

   We developed a method based on particle filtering for detecting and measuring the intensity of technical trading. We tested the methodology on a simulation framework created for this purpose. The technique has been used to test whether a set of technical price patterns are actively traded on the market.

3. Option pricing in the presence of technical trading

[1] see [1]

This experiment demonstrates that the intraday security price is not a Markov process when it is actively traded by technical traders. It then proposes a model for pricing intraday options on securities that are subject of technical trading. This is achieved by including the technical patterns parameters in the state space of the random process. The experiment is exemplified with price patterns that have been identified as actively traded.

4. Technical trading, prominence and liquidity

   This experiment introduces a new method for automation of technical patterns detection based on topographic prominence. It measures the prominence of the fluctuations on the price series and uses the result to linearize the series and detect technical patterns such as 'Head-and-shoulders' and support and resistance levels. The model from the first experiment combined with the method of this experiment is applied on a range of securities. The experiment extends the analysis by exploring the effect of different liquidity conditions on the intensity of technical trading.

The first contribution of this thesis is developing an environment for trading simulation and technical strategies identification and backtesting. The second main contribution of the thesis is developing a methodology, based on particle filtering, for identifying the presence of technical trading on the futures and foreign exchange markets. We applied the technique on a number of securities to measure the intensity of technical trading. We also investigated the effect of liquidity on the presence of technical trading. The next contribution is developing a new approach for technical pattern automation based on price prominence. The next contribution of the research is the application of the newly developed technical pattern automation method to identify and test the performance of four different technical price patterns on a range of futures and foreign exchange securities. Finally, we demonstrated how the information on specific technical patterns and their trading intensities can be used for pricing intraday options on the securities exposed to active intraday technical trading.

Acknowledgements

This thesis would not have been possible without the continuous guidance of my supervisor Professor Philip Treleaven. I am immensely grateful for his time, strategic advice and encouragement. The interesting discussions with my second supervisor Professor Piotr Karasinski helped me in the process of formulating my research ideas. I would like to thank him for the thought-provoking comments and questions.

I would also like to thank the University College London Department of Computer Science and especially the Centre for Financial Computing for giving me the opportunity to be part of its research community. I enjoyed the intellectual feast of meeting some of the brightest researchers in this field. In particular, I would like to thank Dr Gordon Ross for the engaging discussion and Dr Guido Germano for the constructive feedback.

This research links together quantitative finance, technical analysis and particle filtering. The idea for this thesis came as a result of the opportunities to discuss each one of these aspects with leaders in their fields. I am grateful to Dr Jessica James, Shyam Devani, John Noyce and Professor Simon Godsill for the inspirational conversations.

During the years that I spent in quantitative finance and trading I had the pleasure to work with many financial markets professionals. For this research I owe special thanks to Richard Adams, Daniel Lister and Dr Rene D'Hulst who have been not only valuable colleagues but also close friends.

It was my mum, who ignited in me the spark for learning and education many years ago. I want to use this opportunity to thank her once again for that. I hope that I will succeed in keeping this flame to share with my children one day. My sweetest Rali and Anton, thank you for all your patience and smiles. Rali, I promise you, we are going to finish all those projects we were planning to do.

Finally, I want to dedicate this to you, the one who has been always there to help me and hold my hand to every success in my life. My dearest Lu.

Contents

List of Figures

List of Tables

Industry Background and Publications

Industry Background

- More than twelve years of financial industry experience in research and trading on equity, foreign exchange and futures markets at Commerzbank and Citigroup
- Pioneering research in applying machine learning methods in quantitative finance
- Publication of research in variety of quantitative topics for internal and clients' use

Prepared Publications

Based on the research developed in this thesis we are preparing two papers for publication:

- 'Detection and Tracking of Technical Trading Using Particle Filtering'
- 'A New Method for Automated Technical Price Patterns Discovery Based on Topographic Prominence'

Chapter 1

Introduction

The main purpose of this chapter is to describe the research ideas and the motivation behind them. The chapter starts by defining technical trading and describing its role on the financial markets. It continues with presenting a brief overview of the different angles from which the subject has been approached in the past and how our approach differs from those. The methods, data and the software developed for the analysis are described. The chapter then presents the contributions of this research thesis. It concludes with a brief description of the chapters that follow.

1.1 Motivations for this Research

Trading professionals usually define themselves along the axis that goes from fundamentally driven to price driven. *Fundamental trading* is based on fundamental market analysis. This type of analysis studies the relationships between the security market prices and external factors that are not directly linked to these prices. These factors can range from fundamental economic or accounting variables, that can be observed and measured, to more discretionary ones, for which the interpretation by the trader or the analyst plays an important role.

At the opposite end of the trading strategies spectrum is the so called *technical trading*. Technical traders base their decisions on technical analysis. This is defined as the ability to form a view on the future development of the price using only the currently observed price history. The historical price charts are investigated for specific price patterns. These patterns could be derived from the price and the traded volume or from the price only. The time periods used for searching for those patterns could span from minutes to days, months or even years. It is a well known fact that thousands of traders constantly watch for occurrences of their favourite chart constellations and then execute trades based on the expectations they associate with those chart patterns.

Although ignored by most researchers as non-scientific, technical analysis has been used by traders and discussed by analysts since the time the first price charts were created. Nowadays sales and trading departments at every financial institution have their technical analysis teams who publish reports and forecasts used by traders and external clients. Many hedge funds and bank trading teams trade a broad range of technical strategies either on their own or in combination with other types of strategies. Out there millions of self-employed individuals and groups of people ranging from professionals to mere enthusiasts constantly watch the price charts of the moving asset prices to execute their trades when their favourite technical patterns occur. Technical analysis and price chart patterns are the topic of many books that can be found in the personal libraries of financial market professionals. Books such as Murphy [2], Colby [3] and Saettele [4] are considered to be the technical analysis bibles. Most of the online trading platforms offering trading opportunities to retail clients and web based financial charting engines contain tutorials explaining common technical trading strategies. Some of these tutorials (Knight [5]) evolve into well known books in technical trading strategies. These and other sources have been used in the myriads of offline and online courses offered by different organisations to teach the next generation of technical traders. The technical analysis and trading professionals have created their international regulatory body - The Market Technicians Association [1]. The organisation promotes the idea of technical analysis and charting to the financial industry and the general public.

Can technical strategies be profitable? For many years there has been a great divide between academics and financial market professionals on the forecasting power of technical analysis and the winning possibilities of trading strategies based on those. For a long time the classic academic approach has been based on ignoring them as lacking any scientific foundation and hence not able to produce anything but random outcomes. These results were based on the assumptions of information transparency, no arbitrage opportunities and efficient markets. The Efficient Market Hypothesis introduced by Fama Fama [6] has been one of the main angles that technical analysis have been approached by academics. Its main conclusion being that observed patterns on asset prices cannot be used for forecasting future price movements.

On the other hand the industry witnessed the success and thriving of not just one or two hedge funds using strategies based on technical patterns, traders smiling at the proofs that the profits they had generated are highly unlikely to occur. There were many cases when outstanding performance has been claimed without the possibility for confirmation by an intermediary or counterparty: paper traded strategies (trading without actually buying and selling securities but only keeping record of intended actions), analysts' recommendations, strategies backtests (trading performance generated by using historical asset prices). Most of the time however there is a way to get an official proof of the track record by an industry recognised institution.

Having observed the persistence with which technical analysis survived all ignorance and denials over time continuously increasing their presence in the financial industry the academic world started paying closer attention to that phenomenon. Three different aspects of the subject have been approached:

1. Automation of the pattern recognition process

[1] http://www.mta.org

One of the first challenges that price charts patterns offer to the academics is the ability to substitute human eye and simple ruler with a computer algorithm that produces similar results

2. Questioning the hypothesis and assumptions that had created the above mentioned views

   The efficient market hypothesis and the assumption of no arbitrage have been challenged and cases have been explored when their validity is limited. Recent researchers focus on the relationship between market efficiency and active investment management. A recent study (see Hurst et al. [7]) demonstrated that a relatively simple active trading strategy based on price momentum shows a positive performance for a period longer than a century. That is only one example that empirically justifies the existence of active money management. Further on Gârleanu and Pedersen [8] generalized the role of active money management starting from the obvious problems that firstly, if security markets were efficient then the market for active asset managers would be inefficient and secondly, security market efficiency must be achieved by market participants who, through their trading, move prices towards more efficient markets. The analysis of Gârleanu and Pedersen [8] and Pedersen [9] conclude that the security market is 'efficiently inefficient'. This is a new notion that differs significantly from the classic ideas of market efficiency. It outlines the role of active money managers as liquidity providers. This helps the other market participants execute their transaction when they need to and as a result they are prepared to compensate the money managers for this service.

3. Analysing the subject using concepts and methods from behavioural sciences and psychology

   The development of behavioural finance and recent studies on market psychology offered a totally innovative approach to technical analysis and trading. The assumptions of market participants' rationality and information transparency have been given different meanings. Agents are considered to be not only passive receivers of information but also active interpreters, see Tuckett [10].

From the observation that technical traders are significant participants on the financial markets follows naturally the assumption that their actions will contribute to the price formation of the financial assets and contingent claims on those assets. As a result of that a component of the price at any moment is defined by technical trading activity, i.e. by the history of the price itself. This introduces a non-Markovian property of the pricing process. Analysing this property is the main objective of this thesis.

1.2 Methods and Data Used

The market of a security is presented as a stochastic dynamic system. The technical traders are divided into groups based on the price pattern they monitor. Every group of traders is considered to drive a hidden state process of technical trading the effect of which is reflected on the security price via a known return impact function. The price observations that are recorded on the market are perturbed by the background trading done by all other market participants. The parameters of this background trading noise are assumed known. The non-linear state space model that such a system creates is analysed with the help of particle filtering. The methodology is initially developed on a series of simulations. It is then used to test the securities for evidence of technical trading. The particle filtering algorithm requires a prior knowledge about the presence of the pattern it explores. An automated procedure for detecting the patterns is developed and executed on the price series. It mimics real-life traders watching the price for pattern occurrences.

The experiments of this research use price and volume data of the futures markets from Tick Data[2]. These include equity indices, bonds and short term interest rates futures. We analyse also foreign exchange data that has been acquired from an online database Dukascopy[3]. The idea is to explore wide variety of markets, countries and liquidity profiles. We aggregate the data initially to minutely frequency. This is the level of aggregation used for the intraday trading analysis. Aggregation to lower frequencies: daily and weekly is also processed and saved. This enables us to model different types of strategies. We load the model with up to ten years of data so we are able to analyse boom and bust times. Some of the datasets include the 2008 credit crunch period that needs to be approached very carefully in some of the analysis.

The software design and implementation of the ideas is an important part of this research. All of the ideas have been prototyped in R during the initial phases of this research. Then the code has been fully refactored and consolidated into an R package. The data is stored into a database and a layer of abstraction that interfaces the analytical part with the database is developed. Some modules are compiled to achieve better speed.

1.3 Major Contributions

This thesis does not intend to promote any technical analysis methods or techniques to a more scientific level, it simply accepts the fact that they exist and many people do use them in their decision making. Ignoring this behaviour creates an opportunity to describe the asset price process in a way that appears logically sound but misses an important aspect of reality. This thesis aims to fill this gap by integrating such types of behavioural components into asset pricing processes. The main contribution of the thesis are as follows:

1. Design and implementation of a particle filtering based methodology for detection and tracking of technical trading. The technique is developed and tested over a series of simulations. It is then applied to measure the trading intensity of a set of technical price patterns based trading strategies on a series of securities.

2. Extend the idea of pricing contingent claims for a process that contains a non-Markovian

[2] www.tickdata.com
[3] www.dukascopy.com

component. The concept of no-arbitrage is extended to accommodate the properties of the model.

3. Develop a new method for automation of technical patterns detection that is based on topographic prominence. Use this method and the particle filtering technique described above to investigate technical patterns trading at different levels of liquidity.

The economic value of this research lies in using the developed methodology for analysing technical trading and more generally market participants' behaviour as well as using the empirical findings of the research into further analysis. In the case of technical trading this knowledge can be used for creating more realistic pricing and risk management models as well as models for forecasting market liquidity, which is an important building block of electronic market making and automated execution algorithms. In its generic form the method can be used by regulators for detecting illegal trading, such as "cartels" of traders. Traders and researchers can use the findings of the thesis directly or apply the suggested methodologies for technical patterns identification and trading detection on other price patterns or different parameter sets.

1.4 Outline and Structure

This thesis continues as follows:

Chapter 2 Background and Literature Review: We define formally technical analysis and technical trading. The most widely used technical patterns are described and charts of those are displayed. Classical academic views on the topic are presented, introducing and explaining concepts such as efficient market hypothesis, arbitrage opportunities and asset pricing. We introduce basic concepts of behavioural finance and financial psychology. In addition to that we review the concepts of sequential Monte Carlo and particle filtering and their applicability for solving non-linear non-Gaussian problems. We describe the basic algorithms and the main challenges of their usage and implementation. Difference between Markov and non-Markov processes are explained and demonstrated with examples.

Chapter 3 Detection and Tracking of Technical Trading Strategies: The focus of this chapter is to develop a particle filtering based implementation for detection and measuring the intensity of technical trading activity. An automated detection method for a small range of very widely used technical patterns is developed for the purpose of the experiment. It continues with developing a series of simulation to test and demonstrate the particle filtering implementation. A real data test is then carried out for each of the technical patterns on a range of securities. The chapter ends with a summary of the test results and conclusions.

Chapter 4 Option Pricing in the Presence of Technical Trading: This chapter integrates behavioural elements such as technical trading strategies into existing option pricing framework. To achieve this we extend the process describing the asset price movement by adding a non-Markovian part that reflects the technical trading patterns. We analyse it with respect to no-arbitrage requirements.

Chapter 5 Technical Trading, Prominence and Liquidity: We introduce a new model for price linearization and hence automated detection of price patterns based on topographic prominence. The model derived in chapter 4 is tested under different liquidity constraints and the effects are analysed. It is also tested for different period lengths and levels of information transparency.

Chapter 6 Assessment: This chapter analyses the effectiveness of the results presented in the previous chapters. We analyse the extent to which the problems stated at the beginning are solved. We comment on practical issues related to implementation and usage of the framework and the model.

Chapter 7 Conclusion and Future Work: We summarize the research presented in the thesis. The main results and contributions are outlined and briefly discussed. Ideas for future research in each topic are suggested.

Chapter 2

Background and Literature Review

This chapter introduces technical analysis and reviews the existing research on quantitative methods for identifying and automation of technical patterns. We explain how technical trading works and present some basic technical trading strategies that are used in the thesis. The chapter reviews the current research on particle filtering technique and specifics of its implementations. We also introduce the idea of asset pricing and no-arbitrage requirements with Markov and non-Markov processes.

2.1 Introduction to Technical Analysis and Trading

2.1.1 Definition of technical analysis

Technical analysis is defined as the process of forming an expectation for the future development of prices of financial assets based on their observed history. It is not uncommon in economics and finance to form expectations based on historical analysis. In its pure version technical analysis ignores the cause behind the events and concentrates only on the effect that is observed on the price, making the assumption that this effect consistently follows the price pattern. The primary sources for technical analysis research are the different types of charts visualising the prices. Technical analysts use charts to identify patterns that are associated with price movements. According to Kahn [11] traders have been analyzing price charts for more than two hundred years. The tools and methods used for the analysis progressed substantially moving from drawing lines on paper to using powerful computers with sophisticated charting software. The use of technical analysis and charting expanded massively their universe of users in the last twenty years. It is not only professional investors who use them today but also a multiple of trading individuals, many without any knowledge or formal education in finance or economics. The modern trading platforms, designed for the retail clients, analyse the price charts automatically and present the patterns to the users. Many of them base their trading decisions entirely on these patterns.

2.1.2 Supply, demand and types of market directions

In this section we are introducing first some basic economic concepts needed for understanding the idea of technical trading. Market demand of an asset is defined as the quantities that market participants are prepared to buy at a given range of prices. Similarly market supply of that asset would be the quantities that market participants are prepared to sell along that price range. Current market price is defined as the price where these two groups of market participants meet in terms of price and quantity. Expected changes of demand and supply define the expectation for price changes in the future. The difference between the current supply and demand defines the expected market direction. When current demand exceeds supply the price is expected to rise and the state is commonly known as "Bull market". The opposite is known as "Bear market" and the price is expected to fall. Armed with these definitions we are going to introduce the idea of technical analysis and trading within the framework of quantitative finance as seen by practitioners supporting these views. A common approach used by economists and finance specialists as well as traders is to examine the historical development of an asset or an economic variable of interest in order to derive principles and build hypotheses. Economic forecasts are then published or predictions of future asset price movements are made on the basis of those outcomes. The logic behind this approach is based upon the assumption that certain categories are stable in time. These vary from numerical variables and their interdependencies to notions and views on human and social behaviour. Technical analysis assumes that the way price charts reflect supply and demand as defined above, has stayed the same since the time people started recording market prices. Chartists claim that the reasons for increased demand or supply of an asset could vary, but the resulting values for that demand or supply will be reflected always by a similar range of patterns on the combination of price and volume charts. By correctly recognising these patterns it can be concluded, chartists believe, whether an asset is going through a bull or bear market.

2.1.3 Technical versus fundamental analysis

In trading and financial markets literature technical analysis is often taken as an alternative to fundamental analysis. Murphy [2] summarizes the two types of analysis into fundamental studying the cause of the price move and technical concentrating on the effect i.e. the market price move itself.

Fundamental analysis is based on the notion of an intrinsic value of the asset and on the assumption that this value can be derived by examining all external factors that influence it. Kahn [11] defines two additional types of analysis to add to the fundamental: economic and quantitative. Both use a set of input variables to forecast future prices with the former focused exclusively on economic variables whereas the value of the latter comes from exploiting advanced quantitative techniques. Once the intrinsic value of the asset is defined via the above technique, it is assumed that the price will gravitate towards this value, i.e. there is a causality effect from value to price. Then by comparing the current price with the intrinsic value the asset is defined as undervalued or overvalued. The current misvaluation defines the trading direction.

Technical analysts and traders, on the contrary, claim that all these factors are already reflected by the current price and its recent history. They assume that market price leads the fundamental factors (see Kahn [11]). Furthermore technical analysts believe that even if the fundamental factors do drive the price, most of the time the values of these factors remain hidden during the period they have an influence and become public after their effect expires. So by continuously watching the market price they expect to be able to observe the compound effect of all factors behind the market price move at the right time as well as make conclusions about the current supply and demand of the asset that would move the price in the future. The possibility to observe the factors that affect the price leads us to another category that is worth introducing at this point.

2.1.4 Market information transparency

Information transparency can be defined as the ability of market participants to observe the above mentioned factors and their developments as soon as those occur. Information transparency on the financial markets is a widely discussed topic. It has been studied by academics, taken into account by practitioners and considered by regulators as a condition for efficiently operating markets. Transparency for information based on data published by primary sources as companies, government institutions and data providers can be imposed by the regulators and achieved to a great extent. The range of information that is considered to influence the market prices is much wider. Different market agents manage to acquire different parts and sizes of it at different times. This can be due to a number of reasons including limited time and resources. This is why a situation where a hedge fund has a successful trading strategy as a result of spending a lot of time, effort and money to get and analyse pieces of information not easily available is part of the financial markets reality.

2.2 Views on Technical Analysis in Theoretical and Applied Finance

In this section we summarize the views on technical analysis and trading based on it. The review follows a time line that starts almost two centuries ago and continues to present times. The reason for including the older research is to make the thought process that led to current developments of the subject more transparent to the reader.

2.2.1 Technical analysis versus market efficiency hypothesis

Given that technical trading has been used for more than two hundred years, it does not come as a surprise that it attracts the attention of the quantitative finance authors in the early 1960s. We start our review with Fama [6] that dates back to 1965. It was one of the first research that targeted technical analysis and attracted a great deal of attention. It would not be a great exaggeration if we say that for a long time the paper has been considered the classic academic research on technical analysis. They demonstrated that there was no evidence of statistical relationship between past and current prices. Price processes, they concluded, are much more like random walks rather than experiencing any deterministic behaviour. Put in trading terms, they implied that no profit could be made by using historical prices to predict current or by using available history to predict future prices. Many analysts would consider it only as a work that uses statistical approach to reject the possibility of a technical strategy being able to outperform a simple buy-and-hold one. However the research actually had more generic objectives. One of the main objectives of the research was to demonstrate empirically that equity returns or successive price changes behaviour is very close to a random walk. Furthermore they attacked the widely accepted at that time, and within many research and professional circles for many years later, hypothesis that the distribution of these returns is approximately Gaussian. Running tests on a great amount of equities data they demonstrated that price changes exhibit behaviour that cannot be fit within the tail constraints of a normal distribution. The distribution of the returns, they demonstrated, look much more like family of distributions called stable Paretian that had been introduced two year before their research by Benoit Mandelbrot, see Mandelbrot [12]. The normal distribution belongs to that family too and shares the additivity property. Demonstrating that the price changes are serially independent and follow a random walk described by heavy tailed stable Paretian distribution the research concluded that a profitable technical strategy could not be created. The authors highlighted the fact that their research is focused on purely statistical tests exploring the distribution of the returns. Only one family of actual trading rules was tested, the so-called Alexander's filter technique. The authors realised that this was not sufficient and many more trading rules could have been tested. It is also of great importance that the trading environment evolves since the time the research had been done. This has been reflected in subsequent research on the topic, see Fama [13] and Fama [14].

2.2.2 Technical analysis within behavioural and social economics

The idea that markets often are irrational and driven by crowd psychology has been introduced long time ago by Mackay [15]. From the 50s to the 80s of the last century the academic views on portfolio and finance theory were dominated by the idea of the rational investor and efficient markets. But at the end of the 70s (see Kahneman and Tversky [16]) and in the following decades the idea of irrationality and herding behaviour came back as part of the field of behavioural finance which quickly became very popular. The economics sciences Nobel Prize in 2002 was the highest recognition of this new field. The ideas of technical analysis were reviewed from the aspect of the new paradigm and were extended with additional interpretations. Kahn [11] pointed out that behavioural finance and socionomics contributed to further development of the idea of technical analysis. Saettele [4] among others described the technical patterns as a way to observe collective psychology. Their research claimed that technical analysis worked because they successfully registered the optimistic and pessimistic extremes of market participants.

Technical analysis have always been part of the more generic question, how predictable asset prices are. The research on this topic resulted in another economic sciences Nobel Prize in 2013. Eugene Fama, Lars Peter Hansen and Robert Shiller shared the prize for their research in asset pricing, efficient markets, behavioural finance and statistical methods for researching these issues.

2.3 Anatomy of technical analysis

2.3.1 Main variables of technical analysis: Price and Volume

In order to be able to explore the universe of technical patterns we need to introduce the building blocks these patterns are built with. The most basic of these building blocks are the time series variables that this type of analysis operates with. In this section we introduce the two most important technical analysis variables: security price and traded volume.

The most important market variable that technical analysis operates with is the security price. This does not come as a surprise provided that technical analysts are also known as price chartists. Charts are the basic tools of technical analysis and almost all of the charts that are used plot the price of one or more than one security for a certain period of time and level of aggregation. The enormous variety of charts is possible because technical analysis approaches the security price as a very complex category. It is considered to contain different components. The price also becomes the subject of a range of transformations. The ideas behind technical analysis and the price patterns they explore cannot be appreciated without understanding the specifics of the price. There are two different meanings that are related to the notion of price on the financial markets. The first one reflects the level of available demand or supply of a certain amount of a given security. These are known as bid and ask price respectively. They are used to identify a potential opportunity for a deal not a deal itself. The second meaning identifies a price level at which a trade has been executed. This is usually called a traded price or "last traded". The latter is an industry jargon that even if called "last" is not restricted to the most recently executed trade but includes the whole time series prices of recorded transactions for the analysed security. Both meanings can be considered a fair reflection of the current price of the security. The advantage of the bid and ask prices is considered to be the fact that the asset can be traded for the quoted price until the quote is changed. That is not necessarily true for the last traded. On the other hand the last traded is more representative as a part of the price history since trades at that price really occurred at that particular moment in time. Because of these specifics and the fact that the technical analysis focuses on the historical development of the price, the last traded price is usually the time series that is used.

In this research we used only traded prices rather than quotes. For the securities and the time periods for which traded prices were not available we used a proxy calculated as the average of the best bid and ask quotes. There are three major reasons for using the traded prices: relevance, availability and reproducibility. Firstly, the majority of the technical analysts and traders create price charts and identify price patterns using traded prices series. Therefore, if we want to track their behaviour, we have to use the same type of price data. Secondly, the traded prices can be uniformly collected across exchange traded securities such as futures and interbank traded such as foreign exchange. Lastly, if a third party wishes to reproduce our analysis, the traded prices are most widely available.

Once the price series to be analyzed is selected, it is aggregated to a certain frequency. The longer term technical analysis operates with daily price aggregates, while the intraday use hourly, minutely and higher frequency prices. The aggregated price data are stored in a four dimensional vector series that preserves certain features of each period. These are the price at the beginning of the period (Open), the one at the end of the period (Close) and the highest and lowest observed prices during the period (High and Low). The most frequently observed charts produced and analyzed by technical analysts are the OHLC bars and the candle sticks. Both of them plot all four above mentioned values for each data point of aggregated price. Most of the technical patterns are created either directly on these types of charts or are based on input that is derived from these values.

The actual process of plotting a price chart by a technical analyst requires two additional parameter choices. The first one is the plotting window. That is the time horizon that the chart is required to display. The plot horizon had been of a critical importance in the past when the traditional technical analysis had been carried out by naked eye observation and manual line drawing. Nowadays most of the technical price patterns are automatically identified and the horizon is one of the input parameters of the pattern detection algorithms. The plotting horizon defines the time axes or the abscissa of the price charts, the type of scaling defines the relative size of the price axis or the ordinate of the chart. Depending on the range of prices that has to be displayed on the chart, the analyst selects whether to use arithmetic or logarithmic scale for the price axis. According to Murphy [2] the logarithmic scale is used mostly for the stock markets and arithmetic scale is used on the futures markets. Our own experience shows that the choice is defined mainly by the range that needs to be displayed.

The second major variable that technical analysis uses is the trading volume. Murphy [2] defines the volume as the number of security entities traded during the time period that is used as the unit of the current price aggregation. As stated above, there are many technical patterns that are based only on the price history. Some of those patterns use the volume to measure the significance or the intensity of the pattern. There are also a number of patterns that are based on both the price and the volume. For these patterns the technical analysts watch both the price and the volume history in order to identify the formation and the occurrence of the pattern.

The technical patterns that we analysed in this research are based only on security prices, they do not use the traded volume. In order to keep the analysis at a very generic level we selected types of technical patterns that use only the security price series. There is a version of the 'Head-and-shoulders' patterns that includes the traded volume (see Murphy [2]) but we chose to keep the analysis to the pattern's generic definition. The traded volume is used only in our liquidity analysis in chapter 6, section 6.5.

2.3.2 Fundamental concepts and categories of technical analysis

The theory and practice of technical analysis defines a range of categories that support the main concepts introduced by the field. There are two major concepts that technical analysts use most of the time to explain their ideas. These are the trend and the momentum of the market. The momentum, as the physics term implies, measures the speed of price changes. We are not going to review the momentum further in this review, since the patterns that this research focuses on are based on the trend only. By the “market trend” technical analysts imply the current persisting direction of the market price as opposed to its constant up and down fluctuations. Technical analysis recognize three types of trend, according to Murphy [2] among others and also to industry experience and observations. There are upward, downward and sideways trend. The third type is used for all the periods when a distinct direction of the price cannot be identified. The trend also depends on the time horizon. It can vary from long term to very short term. A technical analyst can explore different term trends on the same chart and create justification for each one of them.

The basic building blocks of the trend, as defined by technical analysis, are the so-called support and resistance levels. These are the peaks and troughs that a price plot would form along its path. Technical analysts explain the meaning behind the support and resistance level with the help of supply and demand forces. The logic summarizes as follows: if the price is currently in an uptrend the new peaks are the resistance levels, since above these levels the desire of the investors to sell at the current peak dominates the market. The opposite occurs at the opposite end of the scale: once the price reaches a certain low level the number of investors willing to buy at that local minimum reaches a level that stops further decrease of the price, a new support level is created. If the trend is considered downward the identical logic but reversed top to bottom is applied. The peaks on the way of price development become support levels and the troughs become resistance levels. The line that is drawn through two consecutive support levels becomes a candidate for the trend. It is not considered a real trend until it is confirmed by at least one more support level in the same direction as the first two. The breaks through the resistance levels are interpreted as continuation and further confirmation of the trend. On the other hand a break through a support level is interpreted as a potential end of the current trend. The more significant the support level that has been broken is, the more it is considered likely that the trend is not present any more The significance of a support level is based on both its price and its time distance from the last resistance level.

2.3.3 Types of technical patterns

The objects that build the technical patterns are the variables and the new categories that have been introduced in the previous section. There are patterns that can be based on each one of the components or on a combination of them. For example, a particular formation of the price line over a certain horizon can be known as a pattern. A sequence of trends that fulfil certain conditions can also be considered a technical pattern. The two major groups that technical patterns divide into are continuation and reversal. As the names suggest the first group includes patterns that are believed to predict price move in the current direction whereas, the second group comprises of such patterns for which, having observed them, the price is believed to start moving in the opposite direction. According to technical analysts' views the reversal patterns do not simply mean that the move in the current direction will not continue but that the price will move over a significant distance in the opposite direction. A subgroup of the continuation patterns are the so-called break-out patterns. In these patterns the price is expected to continue in the direction in which its recent significant move was registered. The idea is that this significant move breaks a predefined range that has been defined by the pattern. An example of a reversal pattern, that we use in our experiment (chapter 4) is the head-and-shoulders pattern. All patterns that are used in chapter 4 are explained in 2.5.2.

2.4 Automating Detection of Technical Patterns

One area where human brains have been better than computers, is pattern recognition and comparison of similar patterns. Computational techniques for image analysis have made immense progress recently but the human eye still has a distinctive advantage over computers. It is a very natural process for the human brain to define the level of approximation needed for recognition of a pattern on an image while at the same time it is capable of quite detailed analysis of the image at the chosen level. Computers on the contrary are capable of analyzing any image at an extremely deep level but their ability to assess 'on the fly' how much details are needed are still evolving. What we described above is also valid in the field of technical analysis. The traditional method of technical patterns recognition used to be a manual process whereby a technical analyst drew lines connecting local minima and maxima on the chart. The major challenge when automating this process is to define the minima and maxima that a human analyst would consider as peaks and troughs rather than intermediate fluctuations. A common approach is to employ a smoothing algorithm that smooths out the fluctuations and leaves only the relevant extrema. This approach is used by Lo et al. [17]. They introduce a smoothing estimator $m(x)$ on a variable x as:

$$\hat{m}(x) = \frac{1}{T}\sum_{t=1}^{T}\omega_t(x)P_t, \tag{2.1}$$

where the price P_t at time t is assumed to be an observation of a hidden state variable X_t transformed by an unknown non-linear function $m(X_t)$ and perturbed by some Gaussian noise ε_t. The above estimate for $m(x)$ from equation (2.1) expresses the smoothed value as the weighted average of the prices P_t over the whole observation window $\{1,\dots,T\}$. There are different approaches to specify the rate the weights decrease moving further away from the current time t. The approach they take is a kernel regression: the weights are derived from a probability density function $K(x)$. An additional parameter $h > 0$ that controls the effective spread of the kernel is added. That combined with a

Gaussian kernel as their choice becomes:

$$K_h(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{x^2}{2h^2}} \tag{2.2}$$

Once the kernel is defined the challenge is to come up with an adequate value for the bandwidth parameter h. Lo et al. [17] derive the value for h via the cross-validation method minimizing a cross-validation function:

$$CV(h) = \frac{1}{T} \sum_{t=1}^{T} (P_t - \hat{m}_{h,t})^2 \tag{2.3}$$

The values derived by the above method were too big and needed further validation by technical analysts. Once the smoothing function is fully specified, every technical pattern, that the research tests, is described as a sequence of local minima and maxima. A procedure that searches for these sequences of extrema along the smoothed price line identifies the locations of the patterns.

Shaw [18] revises the kernel regression method of Lo et al. [17] highlighting once again the importance of a good choice for the bandwidth value. A problem with the weights close to the boundaries of the smoothing window is detected that distorts significantly the values for acceptable values of the bandwidth. Shaw [18] introduces the idea of modal smoothing. A window size M is selected and the last M prices are projected onto the first k orthogonal eigen vectors derived from the M values. The research claims that in most cases an effective choice for the first three modes is deriving orthogonal polynomials from the discrete data points starting with a simple moving average. The second one is a straight line and the third one is a quadratic. Since the resulting smoothed estimates are too noisy, Shaw [18] proposes a combination of the two models within the framework of local polynomial regression (see Fan and Gijbels [19]). The local minima and maxima derived by the smoothed line are good approximations for the support and resistance levels that a technical analyst would work with. There is an important difference between this model and a real technical analysis. Both use some degree of smoothness to locate the local extrema used as support and resistance levels. The real technical analysis then uses the actual price chart not the smoothed line for any further analysis that use these levels.

Not all attempts for automation of technical patterns have been based on price smoothing. Academic researchers have recognized the fact that technical analysts use sequences of straight lines to identify the patterns (see Neftci [20]). There have been different approaches in selecting the price peaks and troughs to build these lines. Osler and Chang [21] identify the significant minima and maxima that are used for spotting Head-and-shoulders patterns by measuring the height of every peak relative to its preceding trough and selecting only the peaks that are above certain pre-defined height. The same method is used for the troughs to define the local minima to be used. This approach, although appearing to imitate the human technical analysts, suffers an obvious disadvantage. It is not only the preceding troughs that a technical analyst uses to define whether to include certain peak.

The significance of the price fluctuations is defined by including a broader part of the surrounding price movements. The human technical analysts achieve this by analyzing the price series as a whole rather than focusing on the price moves one by one.

In their research on technical trading and price memory effects Garzarelli et al. [22] automate the process of identifying support and resistance levels. Their analysis uses tick data. They aggregate the data by taking every τ^{th} tick and ignoring the rest of the data. Using different values for τ, they experiment with different levels of aggregation. This approach, although very different in appearance, is similar to a classic time interval based aggregation that uses the close values for each interval. The fact that it is not properly based on time intervals can make it very different from what technical analysts' data look like. Therefore their approach creates results that are to a great extent irrelevant.

Technical patterns that are based less on visual appearance and more on quantitative rules are much easier to automate. These include comparison between current price and a predefined range, calculation of simple or more sophisticated moving averages and so on. The patterns in this group can be based on quantitative rules that use not only the price but also returns for certain periods, price volumes and any other variables from the technical analysts' repertoire. Most market data and financial software platforms offer a combined approach for identifying technical patterns. They provide the data and the visualisation tools. In addition to that they provide a toolbox of technical patterns elements that the user can combine to create patterns. Patterns that are based on widely accepted conventions would be created automatically but the user has also the ability to create custom modifications the existing ones or create completely new ones.

2.5 Technical trading

2.5.1 Introduction

Technical trading is a process that in general consists of the following phases: preparation and actual trading. During the first phase the traders select the securities to be traded. They also decide which technical patterns they will be following. The first phase also includes any analysis on historical price data and performance of the patterns. This is the so-called backtesting. It can also include testing the patterns under some simulated scenarios including extreme cases not necessarily observable in the historical data. The above described processes can be done explicitly as a part of a well-defined systematic process or as a thought process in traders' minds.

The second phase is when the trading strategy is launched. It consists of a continuous repetition of price observation, trade execution, position and risk management for as long as a strategy is traded. The observation can be done visually or as part of an automated process. It can be done in real time or at some intervals, either predefined or ad hoc. Usually it is not the raw price updates that are being watched. In most of the cases they would undergo some processing as required for the particular pattern to be identified. Examples are aggregation to a required frequency, computing of running statistics. Once the pattern is considered present the trader would enter a position by buying or selling the security. This is the trade execution. It can be done automatically as a part of a computerized process that detects the pattern and executes the trade, but it can be a manual process where the trader executes having observed the pattern or having being prompted by an automated process. Once a trade is executed the short or long position that is created as a result of this trade becomes a part of the trader's portfolio. As such the position is an object of their position and risk management that can be done at individual trades level and/or any higher aggregated up to overall portfolio level. The most obvious actions that can be taken based on this are closing the position when the security price moves as expected - 'take profit'; closing the position when it starts generating losses due to price moving further than anticipated in the opposite direction - 'stop loss' or change the position based on more recent occurrence of the traded pattern.

The time it takes for a pattern to evolve can vary widely. It depends on the pattern frequency. That is the level of aggregation of the price needed to build that pattern or, in other words, is the price observed as a minutely, daily or weekly process. It also depends on other parameters that define that pattern. For example, how far back in time it looks or processes data. In comparison to that, the actual execution of the trade, once the pattern is observed, can also vary but usually within a smaller range. The time intervals imbalance that is created as a result of this needs to be accounted for when these processes are researched.

2.5.2 Technical Trading Strategies

In 2.3.3 we explained the two basic categories of technical patterns. In this section we present trading strategies based on such patterns. The strategies that are described in this sections are not necessarily meant to be traded as stand-alone. In most of the cases they would be only one part of a trading strategy, being most of the time the criteria for entering into a short or long position on the traded security. Other rules define the time or price at which the position is exited. These rules could be based on the current gain or loss of the position, they are called "take profit" and "stop loss" levels respectively. These gain and loss levels could be set on each positions, a group of positions or the whole portfolio. The exit criteria could also be governed by current or future expectations for the risk of the portfolio. Another exit criteria could be the trader's or the technical analysts' view on the lifespan of the price move caused by the technical pattern that had been used to enter into the existing trade.

The strategies selection that are presented contain continuation and reversal patterns as their basis. The head-and-shoulders is taken as a very popular example of a reversal pattern. On the other hand, the range break is a typical member of the continuation family. The other two strategies presented are based on moving averages and trend lines based on support and resistance levels.

2.5.3 Range Breaks

Trading signals based on breaking a predefined range are probably the most common technical trading signals. The ranges that are monitored for being broken can vary in complexity. The simplest range is defined as two static values that are equally distant above and below an initial price. The distance is usually defined in return (simple or logarithmic) terms. In their essence the range breaks are continuation patterns since the following rule applies: when the price breaks the upper bound of the range a buy signal is generated and when it breaks downward a sell signal follows. An example of a day when such a signal is observed is displayed in figure 2.1. Once a new range is defined, it is usually set to allow only one signal during its lifetime. Once a signal is generated, it effectively turns off the pattern and no more signals are generated until the range is reset around a new price level. At intraday trading this type of range is set at the beginning of the trading day and is kept until the market closes. The position that is created as a result of a signal trigger can be kept until the next day or longer, depending on the overall strategy.

Figure 2.1: *Sell signal generated by a 1% range broken downwards.*

A more generic version of the above strategy is described by Bessembinder and Chan [23] and BROCK et al. [24]. Instead of defining the range around a single value at the start of the period, it is considered to be confined between the minimum and maximum values that the price reached over a period of historical observations. The number of observations taken vary within the range of 20, 50, 100, 200, that is roughly equivalent to time periods from one month to one year. A buy signal is generated when the price breaks above the recorded maximum and a sell signal is generated when it falls below the minimum. The range can be extended by adding a one percent band around it.

Another, more complex type of ranges, is created using the recently observed support and resistance levels as introduced in section 2.3.2. Either the maximum and minimum of series of resistance and support levels or the most recently observed ones are used to define the borders of the range. The price is considered having broken the range when it goes beyond the support or resistance levels that define it. Trading range break-out buy signal is generated when price goes above the resistance level and sell signal when the price goes below the support level. Depending on its definition, this strategy can be very similar to trading a trend defined around observed support and resistance levels as described in the next section 2.5.4.

2.5.4 Support, Resistance and Trend

Trend is one of the major concepts in technical analysis and technical trading. There are a number of methods that technical analysts use to identify the current trend of the security price. The analysis of support and resistance levels formed by the price chart is one of the primary approaches for defining a trend in the price movement. The idea of support and resistance levels and trend based on those has been explained earlier in section 2.3.2. The trading strategy follows the trend by opening a position in the direction of the trend when the trend line is confirmed by three consecutive support levels and closing the position when the trend is broken by crossing one of the support levels that has been used to define it.

2.5.5 Moving Averages

There is a very broad range of trading strategies that are based on moving averages of the price. The moving average indicator itself is not a uniquely defined notion. In this review we introduce only the concept and present a basic strategy that uses this concept. The review follows Murphy [2] and Saettele [4] and reflects our own experience from the industry. In its essence the moving average is a relatively straight-forward indicator. As its name suggests, it is produced by computing the average of a number of historical price observations. The basic algorithm uses a simple arithmetic average. Once the frequency of the data and the window length of the moving average are decided upon, the indicator is easy to automate and built in any trading system. Traditionally the moving average has been calculated on daily prices. We have seen also technical analysis reports using weekly and monthly periods. For the purposes of intraday and high frequency trading this indicator is also computed over shorter periods: hourly, minutely and in the case of ultra high frequency even tick-by-tick data. The window length over which the average is calculated varies between one, reflecting just the current price, and higher numbers such as 5, 10, 20 or more. The numbers are usually selected to reflect a longer period. With daily data that would be: 5 for a week, 10 for 2 weeks, 20 for a month, 250 for a trading year.

The concept behind the idea of the moving average is that it is a very easy to use smoothing algorithm. It presents to the analysts a long term price movements where all short-term fluctuations have been removed. As such it suffers a series of disadvantages, probably the major one being that it is lagged. Technical analysis respond by creating different types of moving average calculations that compensate for this by trying to put higher weightings on the more recent price observations. The most commonly used methods for this include linearly and exponentially weighted moving averages. The former uses weights inversely proportional to the age of the observation in number of periods and the latter adds a fixed weighted new observation to the current calculation.

The trading signals that use moving averages can be based on one or more indicator values. The simplest moving average strategy would be to compare a longer and shorter window moving average values in order to produce a trading signal when these two cross. The idea is that the moving of the more recent indicator above the longer term one indicates that the trend takes an upper direction. The reverse move is considered as an indication for a downtrend. The strategy can introduce bands around the longer term moving averages. In that case the shorter term indicator needs to cross the upper band or the lower band in order to initiate a buy or respectively a sell signal. In figure 2.2 an example of such strategy is shown. The choice of the window lengths for both the short and the long moving average varies too. According to BROCK et al. [24] the most often used combination is 200 for the long period and 1 for the short period for a strategy that is based on daily data. Similarly, a common choice at intraday trading, is to use the current price as the short moving average. The choice for the long one can be within minutes or hours of the current day but it can also look further back and compare with a moving average derived from the last few days.

2.5.6 Head and Shoulders

The ‘Head and shoulders’ pattern is one of the best known technical analysis pattern. It is one of the first patterns that comes to mind when technical analysis is mentioned. Directionally it is considered a reversal pattern, i.e. the price is expected to change its current trade after the pattern has been observed. Murphy [2] presents it as one of the major reversal patterns and an important implementation of the technical analysis idea of trend. The pattern is based on the idea of support and resistance levels that define a current trade on the price as described in section 2.5.4. When during an upward price move the last observed price peak is followed by a lower peak with a height very similar to the one of the peak preceding the highest point and the troughs on both sides of the highest peak are of similar depth a ‘Head and shoulders’ top pattern is about to occur. The two smaller peaks are called the shoulders of the pattern and the one in the middle is the head. The line that is connects the troughs that separate the shoulders from the head is called a neckline. The technical analysts expect the height of the shoulders to be about two thirds of the head height. The neckline should be close to horizontal and clearly below the shoulders. An important part of the pattern completion is that once the price descends from the right shoulder and crosses the neckline

Figure 2.2: *Comparison of 20-days vs. 5-days moving averages on SP500 futures. The trading signals of a simple trading strategy based on those are indicated by blue and red triangles for buy and sell signals respectively.*

it does not cross it back up during its immediately following upward moves. The crossing of the line back up is called 'breaking' the neck line and is considered to void the pattern. The 'Head-and-shoulders' that is formed on an upward trending price with peaks pointing in positive direction is called a top pattern and after its completion the price is expected to change its trend into a downward one. The opposite situation when initially the price is trending downwards and the head and the shoulders are created by the troughs with a neckline out of the intermittent peaks is called 'Head-and-shoulders' bottom pattern (Osler and Chang [21]) or inverse head-and-shoulders (Murphy [2]). The top and the bottom patterns are considered to have identical properties and differ only by sign.

The concept behind the 'Head-and-shoulders' pattern lies on the change of the balance of the supply and demand forces as described by Murphy [2]. The pattern contains two important signs, according to technical analysts, for the changed demand relative to current supply. The first is that the price retreats back to its previous trough after the most recent peak. The second sign is the unsuccessful attempt to reach the height of that peak on its following upward move, that forms the right shoulder. Murphy [2] highlights also the role of the traded volume for a more advanced interpretation of the pattern.

The 'Head-and-shoulders' pattern is considered stronger when the above supply and demand picture is confirmed by the traded volume. The first noticeable change is a lighter volume during the formation of the second peak, i.e. the 'head' of the pattern. This, according to Murphy [2], is not a necessary requirement but the technical analysts with whom we discussed the issue, considered it as important. The second part of this feature is a much lighter volume during the formation of the right 'shoulder' of the pattern. That is considered a confirmation sign for the decreased demand (respectively supply on a reverse pattern). The third part is an increasing volume while the trading pushes the price through the 'neckline'. The last part is when the volume keeps the same type of behaviour, decreasing on the way back to the 'neckline' and increasing moving away from it, while the price is reaching the next significant levels on its way in the opposite direction from the previous trend.

A trading strategy that is based on the 'Head-and-shoulders' pattern will emit a trading signal when the pattern is fully formed. That includes the price reversing from the right shoulder and crossing the neckline in the direction opposite to the initial trend. The direction of the signal corresponds to the reversal type of the pattern, thus a negative signal would trigger at a top pattern and a positive on at a bottom pattern. Often the trader or the automated model would observe the price movements after the signal has been traded. The strategy allows the price to retreat back to the trending line but crossing it substantially might trigger an early closing trade of a stop-loss type. The pattern is traded each time it occurs.

2.6 Sequential Monte Carlo Methods

2.6.1 Introduction

This section reviews the sequential Monte Carlo signal processing methodology. It focuses particularly on its implementation for solving filtering problems within an environment of stochastic dynamic systems and discrete time state space models. When used for solving this kind of problems, the technique is widely known as particle filtering. Similarly to other adaptive filtering methods, sequential Monte Carlo works by updating and improving its view on the system at every point in time an update arrives. That makes it particularly useful for problems where results are needed while the system is running. These outcomes are crucial for managing the system in an optimal working order. Similarly to other sampling based techniques, the sequential Monte Carlo methods rely on randomly generated samples to produce their results. Like most of these methodologies it originated from the world of engineering but finds a much wider area of implementation nowadays. It has been used for different kinds of tracking of manoeuvring targets and classification, see Avitzour [25], Angelova and Mihaylova [26] and [27] among others. Mihaylova and Boel [28] develop a particle filtering based solution for freeway traffic estimation. The particle filtering has also been used extensively in speech recognition and processing, defence industry, and many other areas of signal processing. Recently the methodology has been making its way through to the financial industry and especially high-frequency finance. Examples are Jasra and Moral [29] and Christensen et al. [30] on high-frequency returns forecasting.

A state space model is described with the help of two equations: a state equation that describes the process of transition of the hidden variable x from moment $t-1$ to moment t, and a measurement equation describing the relationship between the hidden variable x and an observable variable y. In its general form the state space model can be presented as:

$$x_t = F_t(.|x_{1..t-1}, \eta_t) \tag{2.4}$$

$$y_t = Z_t(.|x_t, \varepsilon_t) \tag{2.5}$$

$$t = 0, 1, \ldots$$

Equation (2.4) is the state equation, also known as transition equation. Equation (2.5) is the measurement equation. The variable x is the hidden state variable that can be either one dimensional or multidimensional. In the former case x_t is the outcome at time t. In the latter case x_t is a vector of outcomes with length the dimension of x. The function $F(.|x_{1..t-1}, \eta_t)$ is the known state transition function that defines the rules by which the state x_{t-1} evolves to x_t. If the state process is first order Markov, which is generally the case, the state x_t depends only on the last outcome x_{t-1} instead of the whole sequence of outcomes $x_{1..t-1}$ and the state transition function simplifies to $F(.|x_{t-1}, \eta_t)$. The random process η adds serially uncorrelated innovation to the state process. The process η is not necessarily stationary but its parameters are known. The variable y from the measurement equation (2.5) is the observable variable. It can be one or higher dimensional. The known function $Z(.|x_t, \varepsilon_t)$ transforms the state x into the noisy observation y. In the case of x and y having different dimensions the function $F()$ transforms also the vectors from the state dimension to the observation dimension. The observation noise is described by the random process ε. The two random processes are uncorrelated.

In the case when both $F()$ and $Z()$ are linear functions and the random processes are uncorrelated Gaussian processes the model can be solved analytically. That means that the posterior probability of x_t given the current observation y_t, $p(x_t|y_t)$, can be computed from the prior $p(x_t|x_{t-1})$, and the likelihood $p(y_t|x_t)$. The analytical formula for the optimal estimates is given by the celebrated Kalman filter, see Kalman and Others [31] and Kalman and Bucy [32]. The proof of this technique is summarised in Durbin and Koopman [33] and Harvey [34]. The idea is based on the fact that the posterior of the new state x_t conditioned on the prior and the newly arrived observation is also normally distributed with mean and variance that can be computed analytically applying a lemma from multivariate normal distribution theory.

If the system is not fully linear and Gaussian the above filtering technique for computing an optimal analytical solution cannot be applied. A range of methods have been developed to deal with such problems. One approach, that some of the methods take, is to modify the system in order to enable the usage of the Kalman filter. This is achieved by "linearising" the system using a Taylor series expansion. This has been known as extended Kalman filter (see Jazwinski [35]).

Gordon et al. [36] introduced a new method for recursive Bayesian state estimation of a non-linear system. In their research the method was called bootstrap filtering. Later on with the research of Kong et al. [37], Avitzour [25], Hürzeler and Künsch [38], Liu and Chen [39], Arulampalam et al. [40] the technique evolved as sequential Monte Carlo and when used for filtering it is known as particle filtering.

The basis of the sequential Monte Carlo method is the concept of importance sampling. Following Chen [41] and Doucet and Johansen [42] if we have a function $h(X)$ and a distribution $\pi()$, we can make an inference of $E_\pi(h(X))$ using a set of samples that have been generated from a distribution $g()$, called the proposal distribution, since:

$$E_\pi(h(X)) = \int h(x)\pi(x)dx = \int h(x)\frac{\pi(x)}{g(x)}g(x)dx = E_g(h(X)w(x)) \tag{2.6}$$

In the above equation 2.6 $w(x) = \frac{\pi(x)}{g(x)}$ is the set of sample weights.

2.6.2 Particle filtering algorithm

When we have a hidden process x_t that evolves in a dynamic system as the one introduced with the state space model, equations 2.4 and 2.5, where y_t is the observation process, we can employ the importance sampling idea in a sequential mode. In this case at every step t we generate a sample $x_t^{(j)}$ from a sampling distribution $g_t(x_t|X_{t-1},Y_t)$. Where $j = 1,\ldots,m$ is the particle index for m number of simulated particles. Only the incremental weight $u_t^{(j)}$ needs to be computed and used to update the particle weights $w_{t-1}^{(j)}$ evolved from the last samples. Lin et al. [43] show that the incremental weight is given as:

$$u_t^{(j)} = \frac{\pi_t(X_{t-1}^{(j)},x_t^{(j)}|Y_t)}{\pi_{t-1}(X_{t-1}^{(j)}|Y_{t-1})g_t(x_t^{(j)}|X_{t-1}^{(j)},Y_t)} \tag{2.7}$$

When we express the above equation substituting with the transition function $F_t(\cdot)$ and the measurement function $Z_t(\cdot)$ and take into account the independence relationships, the incremental weights become:

$$u_t^{(j)} \propto \frac{F(x_t^{(j)}|x_{t-1}^{(j)})Z(y_t|x_t^{(j)})}{g_t(x_t^{(j)}|X_{t-1}^{(j)},Y_t)} \tag{2.8}$$

The current weight $w(j)_t$ then becomes: $w_t^{(j)} = w_{t-1}^{(j)}u_t^{(j)}$

The two important choices that one needs to make for the above algorithm are the sampling distribution $g_t(\cdot)$ and when to resample the whole X_t. The two most common alternatives for the sampling distribution are:

- $g_t(x_t^{(j)}|X_{t-1}^{(j)},Y_t) = g_t(x_t^{(j)}|X_{t-1}^{(j)})$, this is called a bootstrap filter and was introduced by Gordon et al. [36]. In this case every particle evolves in its own stream.

- $g_t(x_t^{(j)}|X_{t-1}^{(j)}, Y_t) = g_t(x_t^{(j)}|Y_t)$, this is known as an independent particle filter (see Lin et al. [43])

The choice of a sampling distribution depends on different factors. That also includes how easy it is to generate samples from the selected distribution.

When the above sequential algorithm is run for a multiple of steps most of the particles can end up with zero weights with just a few particles having a non-zero effective weight. This is called particle degeneracy and can happen at different extent and speed depending on the particular situation. The way to deal with it is to initiate an additional step in the particle filtering algorithm called 'Resampling step'. The idea of this step is to resample the whole stream of particles $X_t^{(j)}$. Some implementations of the algorithm employ the resampling on every step. Another common approach is to introduce a measure for the degeneracy based on the percentage of particles that have weights above certain threshold.

If the function $h(X)$ needs to be evaluated during the update process one can insert an inference step. This step is placed after the weights of the new samples are calculated. The estimation of $E_\pi(h(X_t))$ is computed using the particles $X_t^{(j)}$ and the weights $w_t^{(j)}$.

Chapter 3

Experimental Software Environment

This chapter introduces the software framework that has been developed for the needs of this research. It describes the challenges related to optimal storage and processing of time series data. We explain the design and the implementation of the environment used for trading simulation, identification of technical patterns and testing of trading strategies.

3.1 Introduction

3.1.1 Technical Requirements of the Research

The experiments that we present in this thesis consist of three components: a theoretical framework, real or simulated datasets for testing the ideas and software implementation to carry out the experiments. The theoretical background, the researched ideas and the datasets are described separately in each of the experiments. The software implementation has been designed and implemented as a unified framework that operates across different layers. These are the data layer for storing and processing time series data, application layer for trading simulation, strategies and backtesting and reporting layer for automating the process of analyzing and visualizing the results.

3.1.2 The problem

The nature of this research requires processing of large volumes of securities price data. If this is approached in a suboptimal way, the project can easily become infeasible. The optimal solution is a combination of an efficient way to store and process the data, and a module based architecture that automates the process in a way that can be easily maintained and further developed. In addition, backtesting trading strategies is a challenging task. It requires a solution that is not only fast but also safe against strategies that have been specified incorrectly.

3.1.3 Our Solution

Our solution to the technological challenges of this research is to design and build a framework that satisfies the above requirements. The time series data are stored as vectors within a relational data base management system. This offers structure and fast access at the same time. The core time series functionality has been developed within the database to minimize the data transfer between the different layers of the framework. The trading simulation, the strategies and the backtesting engine have been developed in R using the flexibility of the language.

3.1.4 Contribution

This chapter presents two main contributions. The first one is an R-based framework for trading simulation, technical trading strategies identification and backtesting. The second is a vector based time series processing library.

3.1.5 Structure of the Chapter

The structure of this chapter is as follows. In the next section we introduce the main challenges that the storage and the processing of large volumes of time series data present. Section 3.3 explains our approach for simulating technical trading and asset price development. This is followed by section 3.4 that introduces the concept of backtesting and the ways to program trading strategies efficiently. Section 3.5 summarizes the architecture of the framework presenting the different modules and how they are connected. Section 3.6 lists the choices that we have made for the implementation of that design. The last section 3.7 concludes this chapter and discusses possible extensions.

3.2 Time Series Data

By far the biggest part of the data that we use for this research are prices of financial securities. We operate with prices aggregated to minutely frequency for a range of securities. Our dataset contains real market prices for historical periods with lengths between five and eight years. The price data for each instrument is represented as a time series. These are specific data types that fit optimally neither the relational nor the hierarchical database paradigms. In essence, the time series Y_t is defined as a data vector Y with a time index t. Depending on the dimension of the data vector, the time series are univariate or multivariate. If we separate the data from the time, the time series can be represented as a data matrix with a vector of time indices attached to it. This representation is used by some implementations and helps processing the data using numerical libraries. The two major problems that one needs to solve when working with time series are data storage and data processing. In the following sections we summarize the choices and the challenges related to each one of these problems.

3.2.1 Time Series Data Storage

The specifics of the time series and their usage define the optimal requirements for storage. The three major operations of a database are inserting, updating and retrieving data. In database parlance they are called ‘insert’, ‘update’ and ‘select’. The first one is used to fill the database with new data. When operating with financial time series, we either insert the data for a whole historical period or add the most recently observed values. It is relatively rare to insert records between already inserted timestamps. The ‘update’ operation, that for other type of data is used to change the values of one or more fields defined by certain criteria, in the case of time series is simplified to changing the data values defined by a specific time or a time window. Similarly, the data retrieval when operating with time series is also time based. The values are extracted either for a whole period or sequentially.

The two most common alternatives for storing data are a relational database management system (RDBMS), also known as an SQL type database or a non-relational, 'NoSQL' database solution. The former is the traditional way of storing data that benefits from a solid and reliable structure, fixed schema and a data query language (SQL), that is (almost) common across different platforms and implementations. Probably the most important characteristic of the RDBMS is that the data are stored row-wise using 'horizontal' records. The structure of the data is optimised by normalization of the tables holding the data. The RDBMS also support transactions: if an insert or update operation or sequence of operations fail, the data can be restored to its previous state. Storing time series in a traditional RDBMS, using a separate record for each time stamp, is far from optimal. The data retrieval both for bulk data and single time stamps can be very slow.

The alternative is to store the data column-wise. In this case the whole time series is stored sequentially in the memory and can be retrieved much faster. There are different database solutions that implement this model. Most of them are commercial and very expensive. These systems are usually implemented in a structureless way. Some of them lack even basic protection against incorrect usage of the built-in functionality.

The approach that we took shares the benefits of the two alternatives described above without suffering their disadvantages. There are DBMS, such as PostgreSQL[1], that support array data types. We use this RDBMS and store the time series data and the time indices inside separate arrays. This creates the possibility for extremely fast retrieval of the whole time series. Time based search is optimized by maintaining a common time index across all time series. This translates time stamp search into array indexing which is extremely fast. Keeping the data in a proper database management system allows us to use all the additional features that come with it. These include, transactions, automatic backup and replication, structure and many others.

3.2.2 Time Series Data Processing

The data processing and manipulation that our research requires could be done in three different ways. The first one is to use built-in database functionality, in this case this would be a set of SQL queries. This approach benefits from using all the properties of the RDBMS. Unfortunately, the functionality that is available, especially for array data types and time series data, is too limited.

The most obvious alternative would be to transfer the data to another environment that is optimized for data processing and can be easily extended with all the required functionality. This would minimize the processing time but the overall time will increase significantly since large amounts of data need to be transferred from and back to the database.

The optimal solution is to process the data within the database by extending its functionality

[1] www.postgresql.org

beyond the limitations of the database queries. We achieved this by developing and attaching to the database a numerical library that provides all the necessary functionality. Since the library is accessed from within the database it operates within the same memory space as the database itself and does not need to transfer the data arrays.

3.2.3 Time Series in R

Since the application layer of our framework has been developed in R, we need to choose a way of representing the time series data. R offers a range of alternatives for that. Initially the choice was limited to built-in regular time series objects. Recently many new alternatives have been developed. The representation that is conceptually and programmatically closest to the one we use in the database is delivered by the timeSeries package. The timeSeries objects are built in a similar way, keeping the data in a matrix form with a vector of time indices attached to it. We developed the necessary functionality to transfer time series objects between the data and the application layers of the framework.

3.3 Trading Agents Simulation

The simulation module is the first part of the application layer of the software framework. Its purpose is to generate market returns and prices, as well as specific trading agents' behaviour. The market conditions that this module simulates are used as inputs for testing the methodologies developed in chapter 4 to detect the presence of technical trading.

3.3.1 Simulation of Asset Returns based on Background Trading

Throughout this thesis we work under the assumption that at any moment in time the return increments of a security price come as a result of two separate trading processes. These are the actions of a particular group of traders who act at the occurrence of a set of trading signals and the rest of the trading that is happening in the background. Our approach to simulating the returns and following from that the prices of a security follows the same logic. The first part of the simulation module consists of the functionality for simulating the background return process.

The basic version of the background trading simulation is based on serially independent normally distributed return increments with zero mean and constant volatility. The returns are generated for the simulated time period and sampling frequency. The resulting security price process follows a discretized GBM. An enhancement of this model is implemented by using intraday market prices to compute input parameters in a dynamic way. This produces more realistic simulations. In this case the desired distribution of the return increments and the parameter vector for each simulation step are given as inputs to the simulation algorithm. This approach allows generating simulated return series that feature the main properties of the real market returns, such as leptokurtosis and heteroskedasticity.

3.3.2 Simulation of Technical Trading

The behaviour of technical traders has been simulated by an agent based system. This section explains the implementation details of the system. The theoretical framework is presented in chapter 4, section 4.2.1

Each agent simulates a specific type of traders. In the case of technical trading an agent is associated with a group of traders who trade a certain technical pattern. The behaviour of a trading agent is described by its trading times and trading intensity. Since the agents represent technical trading, they are activated by the occurrence of the price patterns that their trading is bound to. The actual process of automated detection of technical patterns is not part of this module. It is abstracted into a separate module that we introduce in section 3.4.

Given the above specification of the trading agents the simulation of the technical trading is based on the following random processes. The first one generates the times of price patterns occurrences. These times can be simulated either randomly or taken from the actual patterns detection. The former are used for simulating a random or unknown price pattern and is also the way to abstract this system to other types of trading. The latter provides the opportunity to simulate specific patterns. The second feature of the trading agents is their trading intensity. It is simulated by a random process using the underlying intensity value as an input parameter. The last part of the trading agent simulation is reflecting the trading agents' activity onto the return increments of the traded security. This is done with the help of an impact function. We introduce market impact functions in chapter 4, section 4.2.1. The trades stream that is the output of the trading simulation process is converted into a return stream via the impact function.

3.4 Trading Strategies and Backtesting

This section introduces the strategies and the backtesting modules of the software environment. The trading strategies are implemented as classes that hold the parameters and describe the logic of each strategy. The strategy objects created from these classes execute the automated pattern detection logic and record the details of the registered patterns and the intended trades associated with these patterns. The backtesting engine runs the trades generated by the strategies through the security price streams and evaluates the trading performance of each of the strategies and their combination into an overall portfolio.

3.4.1 Anatomy of a Trading Strategy

The generic design of a technical trading strategy follows three major principles:

- The strategy properties and methods should be enclosed within their own space that is not mixed up with the R global environment;

- The strategy must provide functionality to act on price updates;

- The strategy must be able to preserve its state between consecutive updates.

We implement the above design using the utilities that the R language provides. Our implementation of the technical strategies follows logic similar to the recently introduced R reference classes. Each strategy is created within its own environment. The strategy constructor creates the environment, populates it with the properties and the methods of the strategy and initializes the strategy parameters. The properties of the strategies can be accessed or changed by generic accessor and mutator functions. Within their own space the strategies hold container objects to store the technical patterns that they identify and the trades that the strategies triggered on those patterns.

Every strategy implements a default *watch(quote)* method. This is the method that is automatically called when the strategy object is used as a function. This method processes the price updates received by the strategy. This implementation allows the strategy to be launched in an identical way with historical data and with live price streams. A very important advantage that this approach brings is the protection against look-ahead bias. Since the quotes are revealed sequentially the strategy decisions are guaranteed to be based only on the data available at that time. The strategies process the price updates independently, thus the implementation allows parallelisation across the strategies.

In appendix A we list an example R code of a strategy template that implements the above requirements.

3.4.2 Backtesting Engine

The backtesting engine uses the trades generated by the strategies to create a time series of positions for each strategy object. It matches these positions with the price streams to calculate the trading returns. Once the returns time series are generated we create functions to evaluate all the performance statistics that are needed to assess and compare the strategies.

The strategies are processed independently by the backtesting engine, thus the process can be run in a parallel or distributed mode. The current implementation uses the native R parallelisation to achieve better performance.

3.5 Design

The architecture of the framework identifies three separate layers. These are the data layer, the application layer and the reporting layer. The data layer consists of the database holding the time series data and the data processing module developed within the database.

The application layer contains the simulation, the strategies and the backtesting module. The communication between the database and the application modules is provided by a set of functions within the application layer that abstract the database access for time series objects. These functions provide also the ability to load price data from external sources.

The reporting layer contains functions for summarizing the results. It is also used for automated creation of charts and tables.

Figure 3.1 illustrates the design of the software environment with the three separate layers and the modules in each of the layers.

3.6 Implementation

Both the application and the reporting layers have been implemented in R. This has been our natural choice for a research platform for many years. During the last fifteen years it has evolved as a stable open-source language and environment with a large and ever growing community of users. R attracts users and contributors from academia as well as from a wide range of industries including finance, biochemistry, engineering and many others.

Most of the functions that build the two layers have been enclosed in an R package. The functionality is then available simply by loading the package into a running R session. Some of the functions are designed in a way that allows internal parts to be substituted with native code, developed in C/C++ for speed and efficiency.

In order to implement the views introduced in sections 3.2 and 3.5 we needed a relational database (RDBMS) that supports the necessary data types and offers the required flexibility and extensibility. Taking these considerations into account we selected PostgreSQL to hold the data layer of our software environment. This is a reliable, open-source platform that is used by a large number of academic, commercial and non-profit organisations. It supports relatively standard SQL flavour. It also supports a range of built-in programming languages. Unfortunately, when we started this project the R support was not at a mature enough stage and we had to use another alternative to develop our time series processing library. Python with its numerical programming extensions Numpy and Scipy proved to be a good choice for that. We used Python built into PostgreSQL to develop a library for array based time series processing.

3.7 Conclusions

The feasibility limits of any computationally intensive research are defined by the software environment on which it is implemented. The software for this project has been developed on stable open-source solutions. Both R and PostgreSQL provide on one side, stability, and on the other, flexibility for further extensions.

All modules have been designed to be extended easily in functionality and capacity. For example, following the existing templates, the trading strategies module can be populated with new strategies outside of the technical trading space. Further improvements of speed and capacity can be achieved by exporting more functionality to native code.

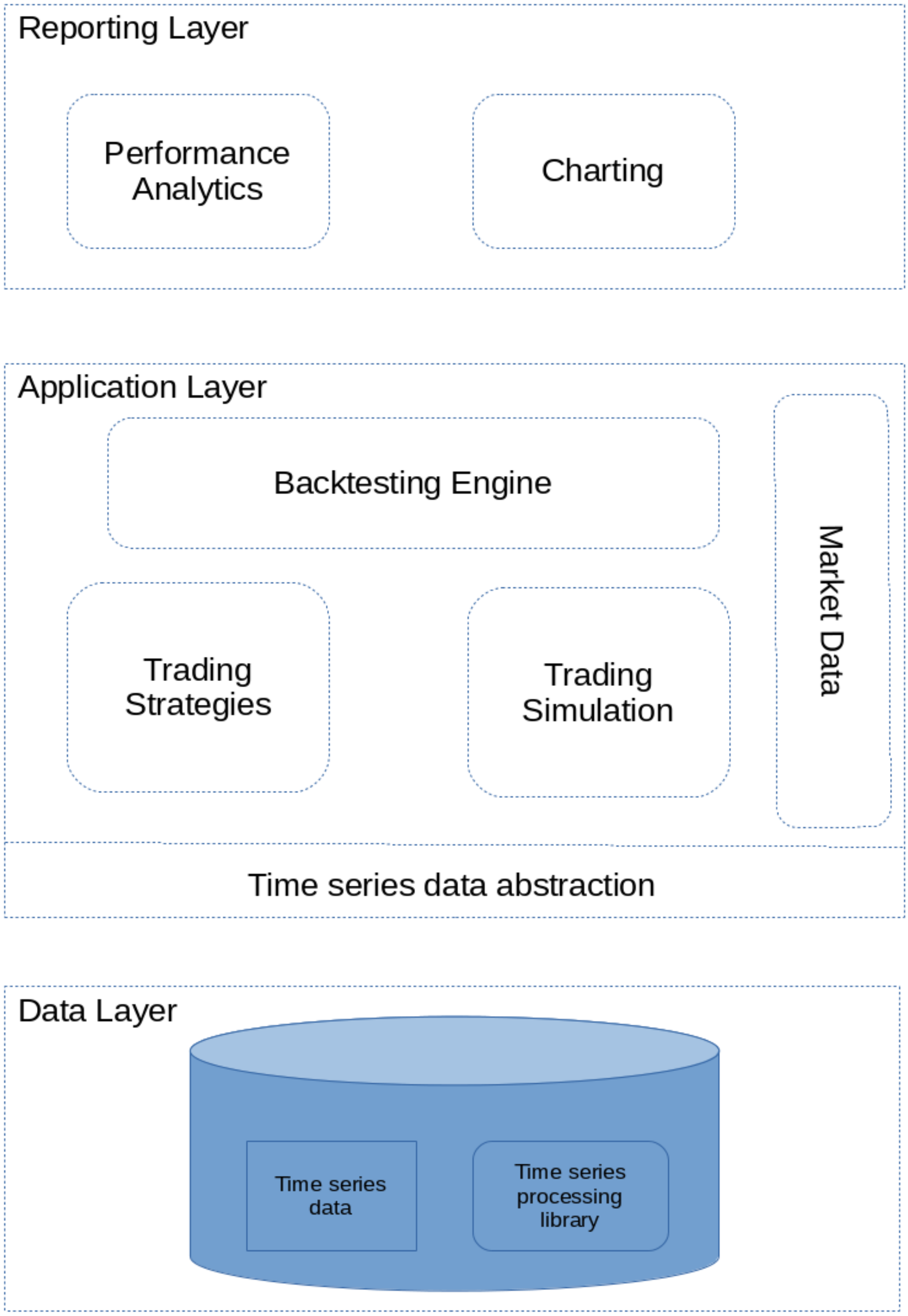


Figure 3.1: *Software environment architecture*

Chapter 4

Detection and Tracking of Technical Trading

This chapter contains the first experiment of the thesis. The chapter starts by describing the experiment. It continues by introducing the theoretical framework. A design and an implementation of the validation of the concept follow. Once the idea is demonstrated with the simulation the experiment is implemented on real market data. It is then tested with a range of asset prices. In the last section we analyse the result of the tests.

4.1 Introduction

4.1.1 Introduction to Technical Analysis and Trading

There are two major categories of traders on the financial markets. We will refer to the first one as "fundamental traders" and to the second one as "technical traders". These names may create some ambiguity since such names especially "fundamental" are used in various contexts. We will try to clear this ambiguity by explaining the criteria on which our classification has been made. A fundamental trader is anyone who makes trading decisions based on external factors. These traders include speculators trading discretionary views or systematic models derived by observing and analysing accounting, economic or any other type of information. In addition, we include all traders that execute trading decisions based on non-speculative factors, i.e. hedging, financial transactions needed by the core business of non-financial institutions or individuals. The characteristics that all the above market participants share is that their trading decision is based on a factor not related to the price development of the traded asset. The current market price can be taken into consideration by the trader either as part of the trading model or decision making process, e.g. price/earning ratio, price to book ratio of an equity, or as part of the execution strategy. However, in those cases the main reason for the trade is still based on external factors. Technical traders, on the contrary, base their trading decisions solely on the development of the asset price by watching and interpreting its movements. Trading decisions are generated by closely watching the price charts trying to detect certain patterns that are believed to reveal the direction of future price moves. These technical patterns have been evolving for many years. They have been described in multiple of books and technical trading manuals. Nowadays on the trading floors of all major financial institutions teams of technical analysts watch the prices and prepare a daily or weekly summary of the patterns observed. These pieces of research are distributed to internal desks and external clients and undoubtedly many of these patterns are traded upon. We can confirm from our own experience in the industry that trades based on technical patterns are part of the financial markets reality. We not only witnessed it but we were actively involved in technical trading.

4.1.2 The Problem

The certainty of the existence of technical traders is not enough for receiving the full benefit of this information. The ability to track their behaviour would also be of great importance. Therefore if we have a way to identify when and what intensity they trade with, we would be able to form a more informed view about the processes that define the market price of an asset. In other words if we can establish which technical patterns are traded on different markets, we can benefit from this knowledge in a variety of ways. For example, this information can be used to improve the pricing models for traded securities as well as the risk management procedures when these securities are part of a portfolio.

4.1.3 Research on Technical Analysis

Technical trading and technical patterns have drawn the attention of the academic world long time ago. There is a lot of existing research on the topic. Although enormous in volume, there are only few questions that have been creating the main wave of excitement within academic circles. By far the most heavily researched one remains the profitability of technical trading strategies. From the time this type of trading attracted academic interest the question of profitability has been given both theoretical and empirical treatment. As we mentioned earlier one of the classics is Fama [6] but also BROCK et al. [24], Bessembinder and Chan [23], Lo et al. [17] and more recently Garzarelli et al. [22]. In addition to exploring the price forecasting abilities of the trading patterns some research efforts concentrate on how to use those patterns into more complicated models. For example Zabbah and Partovi [44] introduce additional sophistication to improve the performance of existing technical strategies.

Technical patterns and technical trading have been object of research aiming to model the behaviour of real markets. Cristelli et al. [45] discuss agent based model that includes technical trading. This research has been based on the Santa Fe artificial stock market developed by Arthur et al. [46]. They introduce an artificial market model that has two different regimes: one based on rational expectations and the other on complex psychological behaviour. The former is a slower and fully transparent one that accommodates the idea of efficient markets. The latter is faster and a core idea of the model is that the information reaches different market agents at different times. As a consequence, the market participants act on different expectations. Realistic features such as price bubbles and crashes become possible and a number of technical trading patterns are demonstrated to work.

A multi-agent simulation without real information input from outside has been demonstrated to resemble closely foreign exchange markets, see Caldarelli et al. [47]. The authors use this result as an indirect proof that these markets are populated mainly by technical traders.

In [48] Feng et al. present a model that simulates technical traders. They assume that this type of traders dominate the market and that they trade at static intensity with their opinions varying randomly at any point in time. This research is related to Duffy and Feltovich [49] where the agents' behaviour is analyzed, with a particular focus on their reactions when they are informed about decisions made by their peers. Both papers make the assumption that technical traders are always present on the market and do not take into account the fact that their actions are prompted by the occurrence of certain events, i.e. technical patterns.

A substantial amount of research on technical analysis and trading explores the profitability of technical patterns. We discuss these research developments in 2.2. Their major objective is either to identify and demonstrate profit opportunities or to present a theoretical proof why these opportunities cannot exist.

4.1.4 Is This Another Research on Technical Analysis

Having penetrated so deep in the world of technical analysis and trading, it is now the time to answer an obvious question: Is this another technical analysis research? The short answer is 'No'. As briefly introduced above the objective of this research focuses on a different problem. We aim to investigate ways to identify the existence of technical traders on the market at any point of time. Technical trading, being such a widely spread phenomenon, offers the opportunity to implement the ideas of this research.

4.1.5 Our Solution

The objective of this chapter is to create a technique to identify the existence and track the behaviour of technical traders. We do not treat technical traders as one whole group, unlike most of the reviewed papers where this is usually the case. We divide the technical traders into separate groups based on the individual patterns they trade. Then we explore some of the most common patterns and use machine learning methods to identify existence of traders following that particular patterns. The methods utilise the sequential Monte Carlo technique or particle filtering. Since its introduction more than twenty years ago by Gordon et al. [36] the technique has been used in many areas mainly for solving engineering problems, see section 2.6 for more details. We apply the technique to make an inference of the technical trading activity since it cannot be observed directly. We create a dynamic system with an unobservable state process and directly observable noisy measurements on the hidden process. The hidden process models the technical trading. The measurements are taken from the observable market data: prices and trading volumes. Prior knowledge and assumptions how traders' actions affect the market are used to model the interactions within the system.

4.1.6 Contributions

The direct contribution of this research is identifying the existence and size of technical trading on a particular market. This knowledge can be used for creating more realistic pricing and risk management models. The ability to anticipate technical trading can be very useful in forecasting market liquidity, which is an important building block of electronic market making and automated execution algorithms. In addition, the method that we develop can be used for identifying any other types of trading activity, based on its expected behaviour. For example, identifying a broker front running clients can be one application of this model. Similarly, by applying our methodology regulators can detect illegal trading, such as "cartels" of traders.

4.1.7 Structure of the Chapter

In the next section we introduce the dynamic system model that describes the technical trading process. In section 4.3 we create a simulation environment to demonstrate how the technique works within the model settings. We create a range of simulations starting from a simple one that is easy to implement however not including all of the model features.Then we move to more sophisticated simulations that reflect all important characteristics of the model. In section 4.4 we apply the technique on real data and discuss the issues related to that and the decisions made in order to overcome those issues. Section 4.5 presents the results of the experiment and discusses their meaning. The last section concludes and presents suggestions for improvement and future work.

4.2 Technical Trading Strategies within a Probabilistic Framework

Following Feng et al. [48], Shaw and Schofield [50] among others we describe the market of a security as populated by two major groups of trading agents. We refer to the first one as *fundamental traders*. Their trading decisions are based on information that is not related to current or historical price levels. The second group is represented by the *technical traders*. This group includes traders who follow a single one or several price patterns and trade when they consider these patterns to be present. In the current section we create an agent based system that models the behaviour of those traders. We commence with a description of the environment, the rules and the assumptions on which our model is based. Then we present the model that we propose and explain it in details.

4.2.1 Agent Based Model of Technical Trading

Technical traders are a very large group of traders known for making their trading decisions on the basis of observed price patterns. We build an agent based model that mimics this behaviour. We concentrate on the most popular technical patterns. It would be very challenging to try to include all known patterns and even then, there will still be many others that are traded behind closed doors. By choosing only the most commonly traded patterns we aim to demonstrate whether our model is successful in detecting these patterns. Our agents are based on technical patterns rather than actual traders. Technical traders usually consider a range of patterns but since each pattern can be traded independently it allows us to associate every pattern with a unique group of agents. Thus a real world technical trader is represented in our model by one or more types of agents where each agent trades only one pattern.

Based on research and practical experience we define the following set of rules that create the necessary requirements for our model. This is followed by detailed explanations for each point in the list:

Model Environment and Rules

1. For each technical strategy there exist a number of agents who monitor the price movements for the occurrence of a particular pattern.

2. The agents following different patterns act independently of each other.

3. At any moment t the number of agents trading the current pattern is x_t:

$$x_t = \begin{cases} n, & \text{if the pattern is observed.} \\ 0, & \text{if the pattern is not currently observed.} \end{cases} \tag{4.1}$$

   where n is based on the number of agents following that pattern.

4. Trading times are aggregated into regular periods. This is defined as trading frequency.

5. Technical patterns and the trading rules associated with them are described with the help of a set of parameters θ.

6. Traders' actions are reflected onto security price and log-return via a return impact function

$$\Delta r = \Im(n) \tag{4.2}$$

7. Trades generated by all other market agents unrelated to the current technical pattern also contribute to security price and return fluctuations.

Agents Trading Different Patterns

We introduce the assumption that agents following different patterns act independently. This assumption creates the impression of moving away from reality. We explain why this is not the case. Let us explore three separate cases. In the first one the different types of patterns, that trades are based upon, occur at completely different points in time. In that case the above assumption is realistic. The second case is when price patterns overlap in time. If a group of agents is focused on one particular pattern, these agents' trading decisions still depend on that pattern only. Another pattern can occur while the first one is being traded but the decision processes of both groups of agents will still be confined within the pattern each group follows. Thus for each group the trading of other groups can be considered part of the background trading that moves the security price. The third case is when agents' strategies are based on more than one single pattern. This case can be integrated in the previous ones by representing the combination of patterns that a strategy uses into one composite pattern with parameters the merge of the parameters of the simple patterns.

Number of Agents Trading a Technical Pattern

The parameter n represents the number of agents trading the analyzed pattern during the current trading period. It is important to clarify that n is not the total number of agents following that pattern. It is the number of agents trading the pattern at the moment t given that the pattern is present. We feel that this explanation is needed because in reality for some of the patterns the situation is not binary as one could conclude from above. Even if the trades peak straight after the occurrence of a pattern some traders can act in advance and some can delay their actions. In a scenario where technical analysis users want to take into account a herding effect they would try to utilize the pattern before others do. Trades appearing to be executed before the pattern is complete could be due to a group of traders that trade the pattern with slightly different parameters. For example, a 'range break' pattern (see section 4.4.2) for a certain security is mostly considered to have triggered at a 1% return move, however there could be players entering the trade at 0.9%. On the other hand reasons for a delay could vary amongst: achieving better execution price, sticking to a range of pre-specified trading times or routines or in some cases not monitoring the price continuously. A technical trader with a substantial amount to trade could also slice their trade into smaller chunks in order to decrease the market impact and by doing so, get a better total execution price.

Trading Frequency

Trading frequencies in general can vary significantly. The shortest period can be seconds or even milliseconds, this is known as high frequency. On the long and the periods can extend from minutely up to daily and weekly. Using this feature the technical traders are divided into two major groups: trading intraday patterns or trading lower frequency patterns - daily or weekly. In this research we concentrate on the former group. The latter has been dominating the markets in the past but in the recent years with the advances of electronic execution the intraday trading is a substantial part of all technical trading. Using the terms introduced in section 2.5, the technical patterns that are traded are also based on minutely (or multiple of minutes) intervals similar to the ones trading decisions are executed within.

Technical Trading Strategies

Price patterns used by technical traders are combined with sets of rules that describe the way those patterns are traded. The combinations of patterns and rules define *technical trading strategies*. Every strategy is codified by a set of parameters that remain constant or vary insignificantly.

The Impact Function

One of the major functions of any security market is to provide liquidity to the trading agents i.e. to supply liquidity. Agents on the other hand consume liquidity by executing market orders. Liquidity is quite complicated notion and there exists many definitions for it. A widely accepted one can be summarized as follows: Liquidity measures the change in price as a result of a trade. The smaller the change is the higher the liquidity or the "deeper" the market is. Liquidity depends on a range of factors. First of all, the security which is traded - there are securities in every asset class that are more liquid than others. A G10 currency pair is, for example, more liquid than other currency pairs, e.g. emerging markets countries' currency pairs. Other factors also affect liquidity: the time of day (more liquidity during daily hours); the time of year (limited liquidity during the last week of December); market and economy turbulence (liquidity was affected by the credit crunch in 2008). How does the idea of liquidity work? When a trader approaches the market to execute an order, the order is matched with the so called limit orders which are a part of a trading counterparty order book. This counterparty could be an automated exchange or another agent acting as a market maker. Some of the limit orders will be at the current market price which represents the top of the order book. If the market order is executed by being matched with a subset of these limit orders, the current price would not change. However, if all limit orders at that level are wiped out, the price will move up (when buying) or down (when selling) to reflect the new level. All that means that any market order has the potential to push the current price and the return computed on it in the direction of that trade. From the above description there are two major factors that define the grade of the impact - the size of the trade and the current liquidity conditions on the market. The former is available immediately but the latter is notoriously difficult to quantify. The mapping between the trade size and liquidity conditions on one side and the price change on the other is known as *Price Impact Function*. If we use log-returns instead of price changes it becomes *Return Impact Function*.

The price impact of a trade has been analysed by many researchers both in academia and in the financial industry. The price move created by a trade is directly related to the cost of execution of any trading or hedging strategy. This is usually the context in which the subject appears in former studies. In Kyle [51] a parameter λ is defined to reflect the cost incurred by trading. It reflects the price change per small amounts of shares traded. $1/\lambda$ is considered as an indicator for the depth of the market. λ is derived via a regression. Later Almgren and Chriss [52] introduce the separation between temporary and permanent market impact. They use linear functions to represent both notions. Shortly after that in another research Almgren [53] relax the linearity assumption on the temporary impact function admitting that it is quite unrealistic. The temporary impact is described by power law with the possibility for being convex or concave. In another research prompted by the development of automated electronic trading Iori et al. [54] further develop the idea of the price impact and supported by empirical evidence demonstrate that it is a concave function on the amount traded in short term. It is shown that the impact function can be simplified to $\Delta p \propto \omega^{\beta}$ where ω is the order size traded at certain time and $\beta \leq 1$ is a coefficient that appears to be relatively stable. Following a similar line of reasoning Smith et al. [55] demonstrate the same kind of results.

Fundamental Traders

All other trading agents on the market are referred to as fundamental traders. This is a term that describes a very large group of traders that can follow a variety of strategies. These strategies cover the whole spectrum from ones based on high level macroeconomics to others concentrating on specifics of individual securities. The strategies can be purely systematic, that means based on predefined and tested algorithms, but they can also be completely discretionary, i.e. a trader makes decisions based on their current views and trading experiences. Based on this, we assume that fundamental trades are being executed on the market at all times. Thus their impact can be modelled as a continuous random process that exists in the background of the asset price formation.

4.2.2 State Space Model

Overview

Once we have defined the problem, there are two fundamentally different approaches for its solution. The first one would be to treat it as a missing data problem. In that case we first collect a data sample and then try to infer the missing data, i.e. the technical trading intensity, from the existing data. This solution is static and it significantly limits the application of the methodology. What we need is a dynamically updated model that is able to adjust to changes in real time. We can achieve this by representing the problem as a state space model. This gives us the opportunity to update our inference on the system at every new observation. Following Guo et al. [56] and also Gordon et al. [36], Arulampalam et al. [40], Liu and Chen [39] and Kendall et al. [57] we introduce the following system of equations:

$$
\begin{aligned}
x_t &= f(x_{t-1}, \theta_{t-1}, \eta_t), && (4.3)\\
y_t &= h(x_t) + \varepsilon_t, && (4.4)\\
t &= 0, 1, \ldots
\end{aligned}
$$

This is a state space model in a relatively generalized form. It can be generalized further: see Liu and Chen [39], Carpenter et al. [58] and Gordon et al. [36]. Many authors, including Harvey [34], Guo et al. [56] and Hürzeler and Künsch [38], present the model in a more specific form with the white noise processes separated from the main functions in both equations. We start with this particular form since it is the closest representation of the problem that we are solving.

Equation (4.3) describes the transition of the system from its previous state to its current state. It is called *state equation*. The states of the system are hidden i.e. not directly observable. They are driven by a first order Markov process. The Markov property is described by the relation $p(x_t|x_{0:t-1}) = p(x_t|x_{t-1})$. The function $f(x, \theta)$ can be non-linear, taking as arguments the previous state x_{t-1} and an external parameter θ. θ identifies one or more parameters (a parameters vector) that can be time dependent but is known at time $t-1$. η represents the innovation in the state update and its distribution is known.

Equation (4.4) is called *measurement equation* and represents a noisy measurement of the current state of the system. The measurement noise is represented by the stationary process ε with expectation $E(\varepsilon_t) = 0$ and known variance $Var(\varepsilon_t) = V_t$. The current value of y_t is defined only by the last state x_t as $p(y_t|x_{0:t}, y_{0:t-1}) = p(y_t|x_t)$.

The problem we are analyzing can be represented by a state space model as the one introduced above. The next two sections elaborate on the specifics of the formulation: the state process and the state measurements.

The State Process

At every discrete time interval t a number of agents execute trades in the direction dictated by a technical pattern. This determines the amount of technical trading on the analysed pattern during that period. The most obvious choice to represent the number of executed trades is to use a Poisson distributed random variable with intensity λ. Following the rules defined in section 4.2.1 we have to define that the intensity is close to 0 when a pattern is not on sight. However, if the pattern is currently active it is a positive number and its value represents the average number of agents trading that pattern whenever this pattern occurs. If λ is the unobserved state variable with expectation L, the expected value for λ_t would be:

$$E(\lambda_t) \begin{cases} = L, & \text{if the pattern is currently observed} \\ \approx 0, & \text{if the pattern is not currently observed} \end{cases} \tag{4.5}$$

That leads to a state equation of the model defined as:

$$\lambda_t \begin{cases} = \lambda_{t-\tau}, & \text{if the pattern is currently observed} \\ \approx 0, & \text{if the pattern is not currently observed} \end{cases} \tag{4.6}$$

where the last time the pattern had been observed was at time $t-\tau$.

The process for λ in the above representation is not Markov, however to use the suggested technique the Markov property is required. There are two ways to convert it to a Markov type process. The first one would be to ignore the estimated value for λ when a pattern is not present and carry forward only the value when the strategy is active. A better way would be to swap the current scalar variable λ with a two dimensional vector x that holds the updated values for λ in both cases - when the strategy is traded and when it is not.

Another difficulty that this formulation presents is the discontinuity between the active and non-active states. This is due to the choice of distribution that we made. We could improve the situation by switching to a distribution that "behaves" better close to zero or even we could employ a continuous distribution. Would such a change make the model too unrealistic? In order to answer this question we look back to what our objective is, indeed it is to identify the intensity of technical trading for a certain pattern. The intensity is measured relative to the periods when the pattern is not present. This does not necessarily restrict the choice to an integer variable and a continuous valued measure can be as good and even better.

An alternative development of the above line would be to model the actual realizations of the trading of the analysed pattern as the state variable. In that case every state x_t is a random variable drawn from an underlying distribution:

$$x_t = f(x_{t-1}, \lambda) \tag{4.7}$$

We could still use a Poisson distribution for $f(x)$ with the intensity as a parameter or take the previous considerations into account and model the effect of the trading with a different distribution.

This approach appeals with its proximity to common intuition. It naturally separates the trading from the effect on the security price that it creates. The former, as described above, is modelled by the state process and the latter by the measurement equation. On the other hand we are interested to make an inference of technical trading as one of the processes that contribute to the asset price and return movements. This consideration makes the above separation of the two processes unnecessary. Although quite intuitive when a Poisson distribution is used, the same intuition cannot be preserved with a different choice of representation.

In the currently introduced model we use λ to identify the trading intensity that is associated with a technical price pattern. We preserve the same meaning for λ within the simulations, that follow in the next section, and the empirical tests further on. When a technical strategy is traded by a significant number of trading agents every time the price pattern is observed, the measured intensity, λ is expected to be higher than the one measured for a technical pattern that is traded less consistently. We do not expect any particular values for the trading intensities neither in absolute nor in relative terms. One of the major objectives of this research is to produce estimates for the trading intensities of the tested technical strategies.

The State Measurements

The measurement equation (4.4) models the relationship between the state process and a series of observations. The variables that can be observed vary depending on the data available to the analysis. The minimum amount of data that can be obtained easily and at a affordable price are security prices with a relatively high level of aggregation. In addition, the least expensive option is time delayed - prices are published with some time lag. The closer the system (like the one we are developing) is to the data sources or the crossing points of the data streams, the more variables it will have available for observation. Thus within a financial or trading institution that subscribes directly to trading channels, all trading data will be available in real time. In the case when the system is within an exchange or an alternative type of a market making system, it will also have access to the full order book. The order book is the collection of all limit orders - the price levels and quantities on bid and ask side at which trading agents, both market makers and traders, are prepared to respectively buy or sell.

The measurement equation contains two major parts: a transformation function and a noise process. The function $h(x_t)$ transforms the hidden state variable x into the observable one y. As explained in section 4.2.2, the trading activity of the technical traders cannot be directly observed and therefore it is modelled by the state process. If that process is restricted to following only the intensity λ of the trading activity, the transformation function reflects the realization of a random process parameterized by its intensity λ and the effect that the realised outcome creates on the market. If these realizations are described by the state process, the transformation function is limited only to converting the trading outcomes into contributions to the changes of the security return.

The technical trading that our model analyzes is happening at the same time as all other trading on the market of that security. What we can observe is the blended effect of all that trading reflected by the observable variables. Examples for such variables are the security price and the volume traded. In our solution we consider both the fundamental trading and technical trading on patterns different from the currently analyzed one to be a background noise. A similar approach is taken by Shaw and Schofield [50]. They also consolidate a type of trading to be represented by a random process. Our approach is different. In our solution we combine the rest of the technical trading with the fundamental trading into a single source of randomness. This approach is justified by the observation that for a given strategy all the trading that is not part of that strategy creates the same effect independent of its source. Given these observations the noise can be modelled as a Gaussian process with a known variance $\varepsilon_t \sim N(0, V_t)$.

4.3 Simulations

This section presents a series of simulations to demonstrate the methodology for solving the problem introduced in the previous section. For each case we define a dynamic model with noisy observations. We explain the intuition behind the choice of such a model. After that we describe how particle filtering can be used to make an inference about one or more unknown variables of the model. There are specific choices to be made in each one of the cases and we explain the reasons behind our decision for making each one of these choices. Having decided on the settings, we coded them into the framework, developed for the simulations, and executed a series of runs for each one of them. The results of each simulation are also presented in this section. After running the simulations with the initial settings we explored the sensitivities of the results and the method efficiency to the exogenous parameters of the model. The simulations progress from a very basic one, to a more complicated one and finally to an exact copy of the analyzed situation. The first one is simple to implement but lacks important features of the problem being solved. The ones that follow gradually introduce new features and improve the efficiency.

4.3.1 Stationary point process with Gaussian noise

Our first simulation is of a point process with a constant intensity that is observed in the presence of Gaussian noise. There are two major reasons that defined our choice to start with this particular simulation. Firstly, it is a good, even if simplified, reflection of the reality that we want to model. Secondly, it is relatively straightforward to implement.

The world that we want to simulate with our model contains the following elements: trading agents, that are the objective of our analysis; background noise, that comes from all additional trading; dynamic price of the traded security; a law of return impact, that relates the amount of trading to the changes in price. That appears to be quite complicated but a series of intuitive assumptions can convert it into a much simpler form. We assume that the agents, that we are analyzing, are active during all times with a constant intensity. In addition to that we accept a trivial law of return impact and bring all the magnitudes down to the same scale. What we are left with can be described by the model that the current simulation uses.

At the same time we need an easy to implement simulation to start with. This will give us the needed confidence that the techniques we employ works as expected. Also the code is much more manageable and allows for a gradual enhancement and adding new features.

The setup that this simulation is based upon, has already been explored for solving problems in engineering. A dynamic model with Poisson based observations has been analyzed by Hernandez and Teal [59]. They extend the bearings-only tracking problem of Gordon et al. [36] by introducing dynamically changing number of targets. Another application that focuses on spatial tracking of multiple targets is presented by Gilholm et al. [60].

The current simulation is described by the following state space model:

$$x_t \sim Pois(\lambda) \tag{4.8}$$

$$y_t = x_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, V_t) \tag{4.9}$$

The state equation (4.8) describes the realizations of the random Poisson events with a constant unknown intensity λ. The observations y_t are perturbed by the Gaussian noise ε with known variance V. The noise process is not necessarily stationary. The interpretation of the model variables in the case of technical trading is as follows: y_t is the security return at time t; x_t is the unobservable part of that return that is created by the price changes caused by the technical trading at that time; ε_t is the effect on the security return contributed by all the rest trading on the market of that security at time t.

We develop a particle filter implementation for the above state space model with the objective to make an inference on the unknown variables x_t and λ. The implementation is as follows:

- **Simulation**: For the purpose of the simulation we sample a range of series X drawn from a Poisson distribution with predefined intensity λ. For each of the series we produce a series Y that is created when X is overlaid with a sample from Gaussian distribution with a predefined variance V.

- **Initialization**: The known parameter V is initialized with its real value used for the simulation. The unknown intensity λ is initialized by sampling it from a gamma distribution with shape and rate respectively $\alpha = 1$ and $\beta = 1$.

- **Sampling distribution**: We have a range of choices that we can refer to when deciding on a sampling distribution $q(x_t|x^{(j)}_{1:t-1}, y_{1:t})$ to draw the particles from. A straight forward choice is to base the proposal distribution on the state process or in other words sampling from the prior:
$$q(x_t|x^{(j)}_{1:t-1}, y_{1:t}) = p(x_t|x^{(j)}_{t-1}), \tag{4.10}$$
that equates to:
$$p(x_t|x^{(j)}_{t-1}, \lambda) = \frac{\lambda^{x_t} e^{-\lambda}}{x_t!} \tag{4.11}$$

- **Particle weights**: With the above choice of sampling distribution the weights are calculated as: $w^j_t = w^j_{t-1} p(y_t|x^j_t) \propto w^j_{t-1} e^{-\frac{(y_t - x^j_t)^2}{2V^2}}$

- **Inference**: The inference step is not a part of the actual implementation loop. It is needed only if, at each time step t, the value of a function $f(x_t)$ is estimated. The estimate for $f(x_t)$ is calculated using the sampled particles and the weights at step t:
$$E[f(x_t)] = \frac{\sum_{j=1}^{m} f(x^{(j)}_t) w^{(j)}_t}{\sum_{j=1}^{m} w^{(j)}_t} \tag{4.12}$$
The denominator in the above expression guarantees normalization. If the weights are normalized within the loop the denominator will be equal to one.

- **Resampling rule**:The degeneracy of the weights that inevitably occurs after a number of updates creates the need of "refreshing" the weights or a resampling step. Our decision, when to run a resampling, is based on the effective sample size as introduced by Kong et al. [37] and Liu and Chen [39] and described later by Kendall et al. [57] and Guo et al. [56]. At the end of each step we calculate first the coefficient of variation v^2_t which we use to calculate the effective sample size $\bar{m}_t$:

$$v_t^2 = \frac{1}{m}\sum_{j=1}^{m}(\frac{w_t^{(j)}}{\bar{w}_t} - 1)^2 \tag{4.13}$$

$$\bar{m}_t = \frac{m}{1+v_t^2} \tag{4.14}$$

If the effective sample size is smaller than a predefined proportion of the initial sample size m, we execute a resampling step. It is done by drawing a random sample of size m from the existing $x^{(j)}$ pool. The likelihood of drawing a particular x_j is equal to its current weight $w_t^{(j)}$. After creating the new sample we reset the weights back to equal: $w_t^{(j)} = \frac{1}{m}$

The implemented algorithm produces good results within the simulated environment. It manages to approach the underlying intensity value, that has been set to 10, after a reasonable number of steps. In figure 4.1 we show the progress in estimating λ in a simulation run with 1000 observations and 2000 particles.

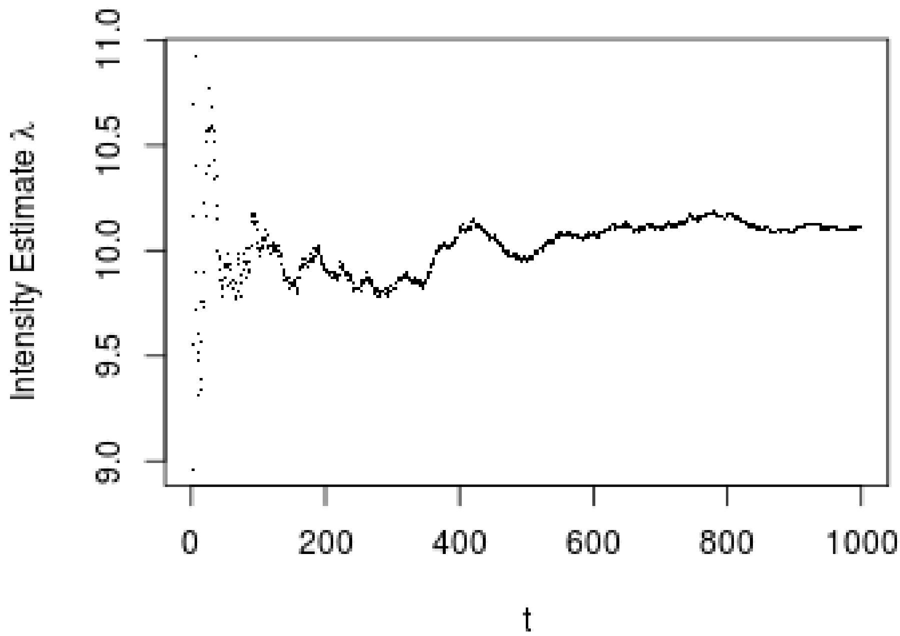


Figure 4.1: *Update for the intensity λ at every step of the simulation. The vertical axis displays the trading intensity λ in units of trading activity. The simulation used 1000 observations displayed on the horizontal axis and 2000 particles. The underlying value of λ is* 10*. The variance of the observation noise is 4.*

The efficiency of the technique depends on the relative strength of the noise. The stronger the background noise is, the more difficult it is to get to the underlying process. We ran simulations with a range of values for the volatility parameter varying from 1 to 16. Values for volatility lower or equal to the underlying intensity of 10 did not affect the convergence rate. Volatility levels above 12 required higher number of observations or more particles to achieve the similar convergence rate for the λ estimates.

4.3.2 Continuously distributed state process with Gaussian noise

The purpose of our second simulation is to present the state development as a continuous process. During the first simulation we modelled the state dynamics as a point process. This choice was dictated by a common intuition about trading a security. That is, the security is traded on a stock or futures exchange and every market order that a dealer sends is for an integer multiple of standardized amounts such as contracts for futures. A point process representation can present very good modelling opportunity when the research is focused on the order book dynamics. An order book is any system that can match buyers' and sellers' orders. In many cases the order book will be integrated in a much bigger market making or trades execution system but even then it still can be considered as a separate logical unit. The flow of orders seen by an order book side consists of discrete amounts arriving at discrete, although very close to each other, points in time. The ability to observe each order at the moment it arrives, or "hits" the order book in market makers' parlance, creates the opportunity to make a decision at each of these time points. Unsurprisingly a point process comes as a primary choice for modelling these events by many researchers in that field.

Our research does not assume a direct access to the order book. In addition to that we see the market events as aggregated during some time interval rather than occurring at specific points in time. Our objective is to identify the intensity of the trading that happens at and around a technical pattern occurs. Even in the case of a relatively high frequency price observations, the time periods we are analysing are longer than just discrete moments. Moreover the concept of intensity that we promote in this research is more general than just number of units traded. The idea of the intensity is to serve as a higher level aggregated measure of the amount of trading done. As such it makes much more sense the intensity to be modelled as a continuous rather than a discrete variable. These considerations become even more valid when we include the effect that the trading has on the security price in the next simulation.

The system that we simulate in this section is based on the logic we developed in the previous simulation. The extensions described above extend it to the current one. We want a state process that behaves similarly to the point process from section 4.3.1 but we allow it to take non-integer states. A straightforward way to represent this is to draw the states from a distribution with density $p(x) \propto \frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)}$. This is achieved by substituting the factorial with a gamma function $\Gamma(x), x \in \Re_0^+$. Although this is the representation that conceptually is the closest to the model from section 4.3.1, a more standard approach would be to model the state process with a gamma distribution. Indeed, it can be demonstrated that a gamma distribution with a shape parameter equal to the intensity parameter

λ of the above distribution and an appropriate rate β can approximate the above distribution very well. An example plot of both distributions' density functions is shown in figure 4.2. On the other hand the method that we employ allows us to work with the former distribution without too much additional effort.

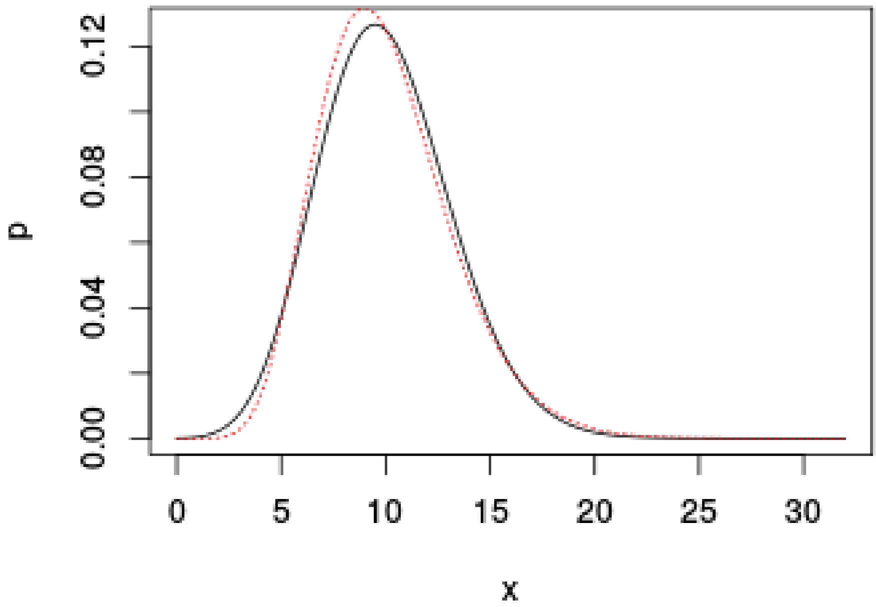


Figure 4.2: *Comparison between two probability density functions: the black line is the modified pdf of section 4.3.1* $p(x) \propto \frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)}$ *and the red dotted line is a gamma density* $Gamma(\alpha, \beta)$ *with shape* $\alpha = \lambda$ *and rate* $\beta = 1$

The variable y, via which the system states are observed, is exposed to Gaussian noise following the same logic as in our first simulation (see 4.3.1). Thus the current simulation is modelled with the following state space model:

$$p(x_t) \propto \frac{\lambda^x e^{-\lambda}}{\Gamma(x+1)} \tag{4.15}$$

$$y_t = x_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, V_t) \tag{4.16}$$

Our objective, similarly to the previous simulation, is to develop a technique for making an inference on the unknown state variables x_t and the distribution parameter λ that corresponds to the intensity of the process. We create another particle filtering implementation that follows the same ideas as our first one. Here is a summary of the main new points:

- **Simulation**: The time series data that we create for this simulation, are drawn from a gamma distribution with a predefined parameters shape $\alpha = \lambda$ and rate $\beta \approx 1$. The observation series Y are simulated the same way as in the previous simulation, section 4.3.1.

- **Initialization**: The known parameter for the variance of the noise process V is initialized with its actual value. We could derive a conjugate prior distribution for the intensity parameter, following Fink [61] and use that distribution to draw initial values for λ. Since the observations are more informative than the prior, we prefer to initialize λ from a uniform distribution. The parameter space needs to be broad enough to guarantee that the range of the generated values is not completely out of reach. The algorithm could work even in that case but it will introduce unnecessary complication.

- **Sampling distribution**: This implementation could repeat the approach in the previous section 4.3.1 and use the prior as a sampling distribution. As explained above, we consider the observations as our most informative source. We can utilize this knowledge by generating the particles from a sampling distribution based on the observations:

$$q(x_t|x^{(j)}_{1:t-1}, y_{1:t}) = p(x_t|y_t) \tag{4.17}$$

This has been introduced as *independent particle filter* by Lin et al. [43] and later summarized by Kendall et al. [57].

- **Particle weights**: The incremental particle weights for the above choice of sampling distribution are calculated as:

$$u^j_t = p(x^j_t|x^j_{t-1}, \lambda^{(j)}) \propto \frac{\lambda^{(j)x^j_t} e^{-\lambda^{(j)}}}{\Gamma(x^j_t + 1)} \tag{4.18}$$

The above calculations are applied for a series of permutations since, as explained by the above authors, the complete detachment of the particles from the state process allows them to be evolving from more than one stream. We generate L random permutations of the particles sequence. Using the above notation this means L permutations of the sequence of the j index. We denote the permutations by $q_1, \ldots, q_L$, where $q_l(1), \ldots, q_l(J)$ is a permutation of the particle indices $1, \ldots, J$. We use these permutations to calculate a set of weights for each permutation where the original sequence is preserved for the current sample x_t and the permuted sequence is used for the previous weights w_{t-1} as well as λ:

$$w^{j,l}_t = w^j_{t-1} u^{j,l}_t = w^j_{t-1} p(x^{q_l(j)}_t|x^j_{t-1}, \lambda^{(j)}) \propto \frac{\lambda^{(j)x^{q_l(j)}_t} e^{-\lambda^{(j)}}}{\Gamma(x^{q_l(j)}_t + 1)} \tag{4.19}$$

The resulting weights are then calculated as the normalised sum across these permutations:

$$w^j_t = \frac{\sum_{l=1}^{L} w^{j,l}_t}{L} \tag{4.20}$$

In fact the individual particles generated at a time step are never attributed to a particle stream. The permutations procedure described above is used only to calculate the particle weights. Using the particles and their weights, one can calculate an estimate for x_t or for any function $f(x_t)$ by adding an inference step to the algorithm.

- **Inference**: As in the previous simulation, inference about any functions of x_t is created by a weighted sum of the function of the particles.

- **Resampling rule**:Resampling decisions are, as in the previous simulation, based on the effective sample size (see section 4.3.1).

We created a series of simulations and ran the particle filtering algorithm that we described for each one of them. We demonstrated that the implementation works as expected. It produced a good estimate for the state process intensity λ and its realizations x_t. Furthermore we explored the sensitivity to the measurement noise variance and once again we demonstrated that the noisier the observations, the longer the algorithms needs to run to produce good estimates.

4.3.3 Scalable continuous state process with Gaussian noise

The purpose of our third simulation is to introduce the idea of scaling. The first two simulations, sections 4.3.1 and 4.3.2, were based on a state process and the observation values were assumed to be from a similar order of magnitude. The problem that our research is aiming to solve does not fulfil this assumption. The state process models the technical trading measured in number of trades executed with some intensity at a pattern occurrence. The notion of 'number of trades' is used in a less strict meaning as explained in section 4.3.2. We can summarize it as the number of trades smoothed to a continuous value representation. On the other hand, the observed variables are the security prices and more importantly the returns derived from these prices for the period and frequency the observations are taken. The two sets of variables, the trading with its intensity on one side and the prices and returns on the other, are not expected to be in similar orders of magnitude. The impact function, introduced in section 4.2.1, provides the map between the two sets.

The previous simulations presented two isolated cases that did not have enough features to simulate a real financial market situation. In the current simulation we aim to mimic a real market case within certain limitations. Following from one of the main assumptions that we introduced in this chapter, we have two groups of traders. The first one is the technical traders that trade an active pattern and the second group is represented by all the other traders on the market of that security. As in our main analysis the latter group contributes to the background noise of the system. The main difference between this simulation and the main analysis is that in this simulation we assume that the pattern is constantly present, i.e. the primary group of traders, whose trading intensity we are trying to detect and monitor, is constantly active. This assumption can be considered to be too restrictive for the current research but we want to draw the attention to the fact that these settings can be used for solving real life problems. These include any trading behaviour that stays permanently active.

We create the dynamic model for the current simulation by further evolving the model of section 4.3.2. The state process represents the trading intensity λ and the return movements x_t as a result of the technical trades. That includes the actual realizations of the trading events and their effect on the price of the security via the impact function. The state process outcomes x_t are assumed to be drawn from a gamma distribution with parameters λ and β:

$$x_t \sim \Gamma(\lambda, \beta) \tag{4.21}$$

We do not follow the usual notation for a gamma distribution by using λ for the name of the shape parameter but since it corresponds to the trading intensity from our previous simulations we prefer to stay consistent with that notation. The parameter β controls the scale of the distribution. In our case it provides the ability to move from number of trades to return scale. In the case of a linear impact function, as in Shaw and Schofield [50] or Almgren and Chriss [52], β is equivalent to the linear impact coefficient. If the effect on the price is represented by a power law, then β corresponds to the power coefficients. In that case λ represents the log-intensity of the trading. Both cases present a workable solution, since our objective is to produce a measure of the trading intensity. It can be the number of trades or any known transformation of it as long as it is kept consistent throughout the experiment.

The state space model that we use to describe the system is :

$$p(x_t) \propto \frac{\beta^{-\lambda} x^{\lambda-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda)} \tag{4.22}$$

$$y_t = x_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, V_t) \tag{4.23}$$

We upgrade the simulation generator and the particle filtering algorithm from the previous simulation in order to accommodate the added features of the current model. The details of the current implementation can be summarized as follows:

- **Simulation**: We simulate the state process by drawing from a gamma distribution with predefined values for the shape λ and the scale β. The latter corresponds to return contributions with values in the range of 0.0001 to 0.01. As in the previous simulations, sections 4.3.1 and 4.3.2 we simulate the disturbance by adding a Gaussian overlay with mean 0 and a predefined variance V. In order to make the simulation more realistic we select V using the following logic: we define a range of annual return variances that have been observed in the history of the analyzed securities; the values used in the simulations are derived by dividing the annual variance by the square root of the observation frequency.

$$V = \frac{V_{ann}}{\sqrt{\tau}} \tag{4.24}$$

where τ is the number of observation periods per year or annualisation factor.

- **Initialization**: The return impact factor and the market volatility are assumed known and are initialized with the values that were used for the simulations. The initial intensity λ_0 is initialized from a uniform distribution sampled over a broad enough range following the ideas from the previous simulation, section 4.3.2. The values for x_0 are sampled from a gamma distribution with parameters λ_0 and β:

$$x_0 \sim Gamma(\lambda_0, \beta) \tag{4.25}$$

- **Sampling distribution**: We keep the independent particle filter based sampling distribution from section 4.3.2. The new particles are created by sampling from a distribution based on the observations:

$$q(x_t|x_{1:t-1}^{(j)}, y_{1:t}) = p(x_t|y_t) = p(N_+(y_t, V_t)) \tag{4.26}$$

where N_+ is a Gaussian distribution truncated to the positive side. The values produced by that sampler represent the security returns generated by the technical trading.

- **Particle weights**: The weight increments for the particles are computed from the prior probability:

$$u_t^j = p(x_t^j|x_{t-1}^j, \lambda^{(j)}) \propto \frac{\beta^{-\lambda^{(j)}} x_t^{\lambda^{(j)}-1} e^{-\frac{x_t^j}{\beta}}}{\Gamma(\lambda^{(j)})} \tag{4.27}$$

As in the previous simulation we generate values for a series of permutations and create the weight increments as the sum of the values across the permutations.

- **Resampling rule**: Following the same principles as in the previous simulations we compute the coefficient of variation v_t^2 and then the effective sample size $\bar{m}_t$. If the latter falls below the threshold level we run a resampling step. We draw samples from the current particle streams that are built of $x^{(j)}$ and their corresponding λ_t.

- **Inference**: As described in the previous simulations the inference step allows the algorithm to estimate a function of x_t using the particles $x_t^{(j)}$ and their weights $w_t^{(j)}$.

At time T, when all observations are processed by the algorithm, we can calculate maximum likelihood estimation for the intensity λ using the values $x_{1..T}$ generated by the above described loop. We follow the logic demonstrated by Rice [62] for maximizing the log-likelihood function. The derivation in the case of a gamma distribution with known scale parameter β, unknown shape parameter λ and a vector of observation X with length T follows:

$$f(x|\lambda,\beta) = \frac{\beta^{-\lambda} x^{\lambda-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda)} \tag{4.28}$$

$$l(\lambda|\beta,X) = \log(\prod_{t=1}^{T} f(x_t|\lambda,\beta)) = \sum_{t=1}^{T} \log f(x_t|\lambda,\beta)) \tag{4.29}$$

$$l(\lambda|\beta,X) = \sum_{t=1}^{T} -\lambda \log(\beta) + (\lambda-1)\log(x_t) - x_t\beta - \log(\Gamma(\lambda)) \tag{4.30}$$

The last expression simplifies to:

$$l(\lambda|\beta,X) = -T\lambda\log(\beta) + (\lambda-1)\sum_{t=1}^{T}\log(x_t) - \beta\sum_{t=1}^{T} x_t - T\log(\Gamma(\lambda)) \tag{4.31}$$

The above log-likelihood function is maximized by finding the value of λ that solves the equation:

$$\frac{\partial l}{\partial \lambda} = 0 \tag{4.32}$$

The above derivative is:

$$\frac{\partial l}{\partial \lambda} = -T\log(\beta) + \sum_{t=1}^{T}\log(x_t) - T\frac{\Gamma'(\lambda)}{\Gamma(\lambda)} \tag{4.33}$$

where $\frac{\Gamma'(\lambda)}{\Gamma(\lambda)} = \psi(\lambda)$ is the digamma function.

The equation that we need to solve becomes:

$$\psi(\lambda) = \frac{1}{T}\sum_{t=1}^{T}\log(x_t) - \log(\beta) \tag{4.34}$$

We use a numerical procedure to find the solution of the above equation. In figure 4.3 we demonstrate how the estimate for the intensity λ improves with the number of observations.

We tested the simulation by creating a series of runs over a broad range of input values. An important difference from the previous simulations is that the current one uses realistic values for some of the input parameters such as the market volatility. The approach that we used for selecting these values has been explained earlier in this section when we introduced the simulation. As demonstrated in figure 4.3 the simulation manages to locate the underlying intensity within reasonable number of steps.

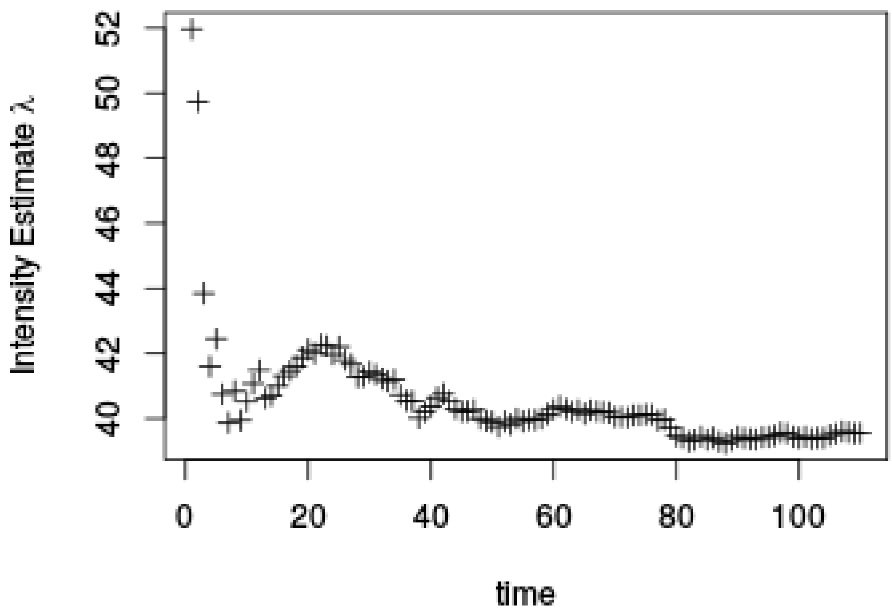


Figure 4.3: *The estimate of the trading intensity* λ *improves with more observations included in the maximum likelihood estimation. The simulated data for the experiment has been created with intensity* $\lambda = 40$ *and volatility 0.00065 that corresponds to an annual market volatility of* 40%

4.3.4 Non-stationary smooth process with Gaussian noise

This is the main simulation of the current experiment. The purpose of this simulation is to demonstrate that the implementation of the research works with a dataset with known parameters. The model that is employed here is exactly the same as the one we use in the next section to run the real market data.

Compared with the previous simulation from section 4.3.3, the current one enables the model to operate in two regimes: the pattern is active and inactive. When the pattern is active the technical traders who trade this pattern are assumed to be executing trades with an intensity λ, unknown to the observer. When the pattern is not active, sporadic appearances of trades trading the same pattern could exist but not intensive enough to create a significant stream of trades and respectively price and return moves. We assume the intensity during that mode to be $\lambda^{(p)} \sim 0$, where the superscript p identifies a "passive" mode. Such settings of the model imply that one can recognize correctly the periods when the technical pattern is in an active state. We can either assume this with certainty or introduce some probability level at which the system can assess whether the pattern is active. Therefore a parameter p is introduced to quantify the probability of identifying an active state correctly. Whenever a price pattern is detected, the system is in "pattern active" mode with probability

p and in an "inactive" mode with probability $1-p$. This parameter is initialized based on our prior knowledge and experience on technical price patterns.

The dynamic system that we model for this simulation and in fact for our main experiment shares the same basic elements and parametrization with the previous one from section 4.3.3 with the addition of the new features described above. The state space model underlying the system is described as:

$$p(x_t) \propto \begin{cases} \frac{\beta^{-\lambda} x^{\lambda-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda)} & : \text{pattern active} \\ \frac{\beta^{-\lambda^{(p)}} x^{\lambda^{(p)}-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda^{(p)})} & : \text{pattern inactive} \end{cases} \tag{4.35}$$

$$y_t = x_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, V_t) \tag{4.36}$$

The measurement equation (4.36) is kept the same as in the previous simulation. The logic on the background noise that has been introduced before is still valid. The state process (4.35) is extended to accommodate the time periods when the pattern is absent. As a result the state vector contains the return outcome x_t of the current technical trading activity as well as the current intensity values λ and $\lambda^{(p)}$ for two alternative pattern states, active and inactive.

The implementation of this simulation (as well as the real data experiment) required some further development of the framework and the functionality created for the previous simulations. These new features and extensions of both the simulation framework and the particle filtering implementation are summarized and explained below:

- **Simulation**: We start the simulation by generating time series of values that are independent and identically normally distributed with a predefined variance V and mean 0. These time series correspond to the observation noise of our model and as such they represent the trading activity that is running in the background by all other market agents except the ones trading the technical pattern that our model currently analyzes.

  The second phase of the simulation is to generate the time points $t_i^{(a)}$ of the occurrences of the technical pattern and the number of periods these patterns will be traded for. The former values are generated as samples are drawn from a binomial distributions with parameters T, the number of time periods for the number of experiments and q for the probability of success. The latter values are drawn from a Poisson distribution with an intensity l. For each of the pattern occurrences we randomly define whether the pattern is in positive or negative direction that corresponds to buying or respectively selling the security.

  Following the above procedure we create a set of time period vectors of technical pattern trading activity. For each of these periods we use the method described in the previous simulation, section 4.3.3 to simulate the technical trading, i.e. trades are drawn from a gamma distribution with shape λ and scale β. The simulated trading activity is then combined with the return stream generated for the whole period.

- **Initialization**: As with the previous simulations the return impact factor and the market volatility are assumed known and are initialized with the values that were used for the simulations. We give both intensity parameters λ and $\lambda^{(p)}$ initial values from a uniform distribution with a broader range for the former and narrower for the latter. The algorithm is notified when a technical pattern moves into an active state. p_a is the probability that the pattern is in active state when the algorithm is notified. Following from that $1 - p_a$ is the probability that the algorithm is prompted to move to the active state but the pattern is not being traded. p_a has been initialized with 0.9. The values for x_0 are sampled from a gamma distribution with scale β and shape λ_0 or $\lambda_0^{(p)}$ depending on the pattern presence.

- **Sampling distribution**: We demonstrated in the previous simulations that the independent particle filter performed well within these settings so we keep the same method in this simulation. Since the new particles are created by sampling from a distribution based on the observations, the logic is the same in both active and passive pattern state :

$$q(x_t|x_{1:t-1}^{(j)}, y_{1:t}) = p(x_t|y_t) = p(N_+(y_t, V_t)) \tag{4.37}$$

where N_+ is a Gaussian distribution truncated to the positive side. In both cases the values produced by the sampler correspond to price changes created as a result of technical trading with values higher when the pattern is present and close to zero otherwise.

- **Particle weights**: With the above choice of sampling distribution the weightings must be based on the prior distribution as demonstrated by Lin et al. [43] and Kendall et al. [57]. In addition to the considerations that were valid in the previous simulation we need to include the knowledge about the technical pattern presence :

$$u_t^j = p(x_t^j|x_{t-1}^j, \lambda^{(j)}) \propto \begin{cases} p_a \frac{\beta^{-\lambda^{(j)}} x^{\lambda^{(j)}-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda^{(j)})} + (1-p_a) \frac{\beta^{-\lambda^{(jp)}} x^{\lambda^{(jp)}-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda^{(jp)})} & : \text{active} \\ \frac{\beta^{-\lambda^{(jp)}} x^{\lambda^{(jp)}-1} e^{-\frac{x}{\beta}}}{\Gamma(\lambda^{(jp)})} & : \text{inactive} \end{cases} \tag{4.38}$$

where p_a is the probability of correctly identifying the presence of a pattern and $\lambda^(j)$ and $\lambda^{(jp)}$ hold the active and the passive trading intensities respectively. The logic for executing the above algorithm on a series of permutations and deriving the weights from the results of those permutations is preserved and the process is followed as in section 4.3.3.

- **Resampling rule**: The resampling decision as in the previous simulations is based on the effective sample size $\bar{m}_t$ that is computed with the help of the coefficient of variation v_t^2. Once a resampling step is executed, the $x^{(j)}$ streams are built as before. The difference is with the intensity λ streams. Depending on the pattern presence phase only the corresponding intensities are resampled with the other set preserved.

- **Inference**: Given the algorithm described in section 4.3.3, equation (4.34) one can calculate maximum likelihood estimation for each of the intensities at any point in time t using the inferred state process values up to that time: $x_{1..t}$.

  In addition to the above result we use Gibbs sampling following the approach in Pradhan and Kundu [63] using prior probability for the shape λ given the scale β and all the observations $x_{1..t}$:

$$p(\lambda_t|x_{1..t},\beta) \propto \prod_{i=1}^{t} \frac{(\frac{x_i}{\beta})_t^{\lambda} e^{-\beta^{-x_i}}}{x_i \Gamma(\lambda_t)} \tag{4.39}$$

  We derive the above expression by following Fink [61] and rearranging the result for more efficient programming.

  In figure 4.4 we plot the likelihoods for λ and $\lambda^{(p)}$ at $t = 400$ for a simulation with $m = 5000$ particles.

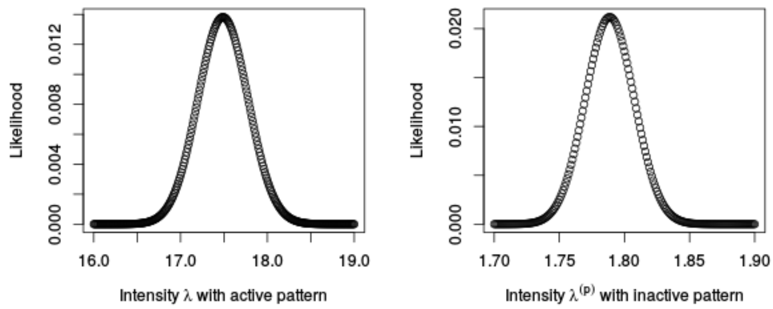


Figure 4.4: *The maximum likelihoods computed from the results after 4000 time steps for the intensities λ and $\lambda^{(p)}$ identifying technical trading with the pattern active and inactive respectively.*

We tried to select the range of input values of the simulation in a way that will produce simulated values that look like real market data. The results of the simulations that we ran demonstrated that the implementation can produce a good estimate for the trading intensity and provides the flexibility needed to accommodate a test with price data of the securities we want to test. This is the subject of the next section 4.4.

4.3.5 False positive trading detection

In this section we carry out a series of simulations that aim to identify how likely it is for our trading detection approach to produce positive outcome when technical trading is not taking place. We call these signals ‘false positive’. This is a way to measure the reliability of the outcomes of the empirical tests that we intend to use our approach for.

Test settings

We simulate 100 sets of security price data. For each simulated dataset we identify the technical price patterns using the automated method developed for the empirical tests. Then for each of the explored technical patterns and dataset we employ the technical trading detection approach. Each technical pattern is specified with exactly the same parameters as in the empirical test in Chapter 6. For the moving average pattern we tested only the version with a long period of 30 minutes. The test for each case consists of the following steps:

1. On the simulated day when the pattern was detected, specify an active trading period starting immediately after the pattern occurrence and lasting for 5 minutes for the moving average pattern and 30 minutes for the other patterns.

2. Specify a passive trading period that is one hour long and ends at a randomly selected time before the occurrence of the pattern.

3. Run the trading detection algorithm to measure the trading intensity during active and passive periods.

4. Compute the active and passive intensity estimates and the KL-divergence between the active and passive distributions of the returns that have been estimated to be contributed by the technical trading.

Result analysis

Figure 4.5a depicts the ranges of the Kullback-Leibler divergences that have been generated for each technical pattern. The Range Break pattern ended up with the smallest KL-divergence values. The moving average pattern on the other hand produced the broadest range of KL-divergences reaching above 1 in 7.5% of the cases and above 1.25 in 2% of the cases.

The Kullback-Leibler divergence is only a measure of the difference of the distributions, it does not directly compare the means of the two distributions. Figure 4.5b plots the distributions of the active trading intensities generated at the occurrence of the technical patterns. These results show that trading intensities values above 4, and in the case of the moving average pattern above 5, are relatively less likely to be false positive. These results deliver benchmark values for the empirical tests that we present in chapter 6 with a brief preview at the end of this chapter.

4.3.6 Programming design and implementation considerations

The simulation framework and each of the simulations are designed and implemented within the statistical environment R (see [1]). Most of the functions are coded using R which is the native for the environment programming language. The design is created with the idea to make adding of new types of simulations that utilize the same framework relatively easy. In order to achieve that, we decouple the framework functionality from the code that is specific for one or more simulations.

Any functions that are self-sufficient, i.e. can be used on their own within another simulation, are grouped together in a utility module that is loaded together with the framework.

The majority of the mathematical and probability functions were relatively straight-forward to code, simply by translating the logic into R. A small part required special treatment in order to keep the intermediate calculations within ranges that are supported by the language and the environment. For example, a product including power function, exponentials and gamma functions. These were dealt with by splitting a power into a product of powers, plugging a division between the two parts:

$$\lambda^x e^{-\lambda} = \lambda^{\frac{x}{2}} e^{-\lambda} \lambda^{\frac{x}{2}} \tag{4.40}$$

or similarly:

$$\Gamma(x)e^{-\lambda} = \Gamma(x-k)\prod_{i=1}^{k}(x-i)e^{-\lambda} = \Gamma(x-k)e^{-\lambda}\prod_{i=1}^{k}(x-i) \tag{4.41}$$

Whenever the speed of execution was an issue or the language did not allow an efficient implementation of the logic, we coded it in C and attached it as a dynamic module.

4.4 Market data test

This section presents a set of tests that are carried out using real market data. The tests utilize the framework that was built and tested in the previous section. The objective of this section is limited to demonstrating the implementation of the trading detection methodology with real market data. An extensive empirical test over a broad range of securities is the focus of the third experiment of this thesis. We explore a number of technical trading strategies and for each one of them we run a test that aims to estimate the intensity level of trades that are executed following from the occurrence of that pattern. At the beginning we introduce the challenges that need to be dealt with during the initialization of the testing framework and how we approached each one of them. Some of the choices that need to be made are relevant for all the tests. At the same time each of the analyzed technical patterns has its own specifics that need to be approached individually.

4.4.1 Test environment

- **Trading frequency** Technical trading is considered to be concentrated on the shorter end of the horizon spectrum. Technical trading strategies generate trades with higher frequency than the fundamental trading. Nevertheless it often happens in the world of technical analysis that price patterns are analyzed over price data that had been aggregated to very high levels including weekly and monthly. In this research we restrict our analysis to security prices aggregated to minutely time intervals. The trading strategies that we want to detect are assumed to be traded intraday. That means the price patterns are monitored during the course of the trading day and once they are observed the trading strategy initiates the execution of trades. Our experience and observations on the foreign exchange and financial futures markets show that intraday trading makes a significant part of overall trading for speculative purposes. The so-called day traders can be found in most of the hedge funds and investment management houses trading systematic or discretionary strategies. There are also a great number of stand-alone individuals whose major occupation is day trading.

The methodology that this research presents and the framework that we created for it do not restrict it to intraday trading on minutely data. The same or similar settings can be used to analyze strategies with various horizons and frequencies.That includes high frequency and ultra high frequency trading, the latter being the play area of trading and market making algorithms optimized for speed and low latency.

- **Securities** The securities that we selected for our tests represent two major markets: foreign exchange and financial futures. Foreign exchange and equities are the markets where we have witnessed the highest level of technical trading activity. The major part of it is concentrated on the most liquid instruments on these markets. For that reason we selected the most liquid instruments for our tests. From the foreign exchange market we carried out our test on EURUSD, USDJPY and GBPUSD and from the equity index futures we select the futures on SP500, FTSE and DAX. Each one of these instruments offers an uninterrupted stream of trading data for the time periods that this experiment analyzes.

- **Exogenous market data parameters** Some of the parameters that the model uses need to be initialized on the basis of exogenous analysis. These parameters include the return impact factor β, the return background process variance V and any additional inputs that are not trading strategy specific.

  The most effective way to estimate the return impact coefficient β is to use order book data. Results are demonstrated in Smith et al. [55] and Cont et al. [64]. The precision of the estimate of the return impact coefficient is of particular importance for a market making algorithm. A market making entity has at least its own order book to analyze although in many cases it will have access to a range of order books across securities and collaborating institutions. In our research the value of the return impact coefficient is only of relative importance, since we need only to identify a measure of the trading intensity and it does not necessarily need to correspond to the actual number or volume of trades. What we need is a consistent way for changing the scale between return fluctuations and trading intensity. This can be achieved by taking values from the ranges derived by the market impact research. As Cont et al. [64] explains and our experience confirmed the return impact is not constant through the day and shows some clear seasonality features. To minimize the effect of these fluctuations we analyse the data only during the times of day that can be defined as normal. We exclude periods of very low trading activity, such as night times as well as time periods during and immediately before major economics and market announcements. Any time periods when sporadic market events occurred that can be considered extreme have also been excluded from the analysis.

The randomness that the model experiences by the observation process is parametrized by the noise variance V. Since this noise reflects the overall background trading, we need an estimate of the volatility of the returns of the traded security that is not caused by the trading of the analyzed technical pattern. The return volatility is a variable that is notoriously unstable. That means that in order to have a reliable estimate it needs to be updated frequently enough. There are two major sources for producing market volatility estimates. These are the observed historical returns and the implied volatilities from the market of options on that security. The former is easily available as long as the return data are available but it reflects only the history lacking the current and future information that the market participants hold. The advantage of the latter is that it reflects these information and views about the future but its availability in higher frequencies is limited. A working solution is to model the market volatility by using both sources taking into account the specifics of each one of them. Contributions from the historical returns are limited to the periods when the analyzed technical pattern is not present. In the case of minutely or higher frequencies we consider it safe to use the volatility estimates from just before the pattern appeared or in other words to consider the volatility constant for the time period the pattern is being traded. With daily and lower frequency modelling implied volatility based estimates can be used while the pattern is considered active.

4.4.2 Strategy settings and parameters

In chapter 2, section 2.5.2 we describe four different types of technical trading strategies. The methodology developed in this experiment is applied to each one of these strategies. First an algorithm that automatically detect the patterns identifies the periods when they occur. The price data, combined with the results from the pattern presence identification, is fed into the particle filtering implementation with the objective to measure the intensity of trading generated by the pattern. The technical price patterns can be successfully identified only if they are clearly defined. That includes the parameter set that defines the pattern and values assigned to each of these parameters. For each of the analyzed patterns we present below our choices for parameters and their values. We also explain when the strategy is assumed to start trading and for how long.

Moving Averages

There is a very large family of technical trading strategies that are based on moving average calculations of a security price. Our own experience and conversations with other traders and technical analysts show that these types of strategies use mainly daily data. The moving average statistics is computed from the closing daily prices. Murphy [2] describes some additional ways of selecting the values to use for the calculations in addition to simply using the close values. As explained above all the strategies we tested for this experiment are based on minutely data and intraday trading. In order to create sensible intraday strategies that are based on the moving averages idea we have to stick to the following basic principles: first, both moving averages must use the same time units, and second, the right limit of both moving average inputs must be the same. The first rule means that if we decide to use the strategy intraday we have to operate only with intraday time units, in our case minutes. The second rule guarantees that the strategy cannot move into a state in which it would be constantly losing money. It would be unrealistic to assume that such a strategy is actively traded.

Range Breaks

The pattern is defined by three parameters. These are the start time, the end time and the range that the price is monitored for breaking. We base the pattern on European market day, monitoring the price between 8:00 and 19:00 BST. The values for the range vary between 0.8% and 1%.

Support, Resistance and Trend

It is easy to specify this pattern conceptually but a challenge to transfer the concept into a unique set of parameters. Firstly, the algorithm tries to identify a trend by building a trend line supported by at least three levels. Once the trend is defined, it monitors for confirmation of the trend that is identified by breaking further resistance levels or for reverse of the trend by breaking an important support level. We allow both the trend building process and the monitoring after it has been created to stretch to more than one day.

Head and Shoulders

The Head and Shoulders pattern is monitored during the active trading hours of the day. We combine the active periods of European and US markets by turning on the monitoring at 8:00 BST and turning it off and resetting at 22:00 BST. The pattern is built on minutely data as this is the finest granularity we are operating with. The automation of the pattern is applied using the prominence based method developed in chapter 6 and following the set of rules described in chapter 6, section 6.2.5. As explained in the introduction to the pattern in section 2.5.6 some of its definitions contain also a requirement to the volumes at different sections of the pattern. We decided to ignore the volume in our tests for the sake of simplicity.

4.5 Test results

The effectiveness of the test depends on two major factors. The first one is based on the quality of the estimates that the user provides for the market environment parameters. Initially we ran our tests with very simple and static estimates for the market volatility. That resulted in significantly lower quality outcomes compared to our further attempts when we tried to feed the system with more precise volatility estimates. The other factor that affects the test effectiveness is how narrowly an analyzed strategy can be defined. If the parameters of a technical pattern that different traders use vary broadly then it is difficult to trace their behaviour using only one set of values for these parameters. We observed this when testing the trend based strategies.

We carried out our tests on two different asset classes: foreign exchange and futures. The experiment with the futures data produced clearer results. In order to simplify the interpretation of the results we introduced a binary set of criteria. It is based on the estimated trading intensity at the occurrence of the price patterns and the security returns attributed to that trading activity. Technical trading test has been marked as positive if the active trading intensity is more than twice the passive one and the security return contributed by the pattern trading during active times is more than two standard deviations of the returns contributed by the same type of trading during passive times. At the same time, having observed the false positive results in the previous section, we require the active trading intensities of the strategies to be above the 90th percentile of the false positive intensities. Using the above criteria two of the tested strategies: Head-and-shoulders and range breaks showed significant levels of trading intensity when the patterns were present. Both strategies showed clearer results on the futures than on the currency pairs. The results for the moving average strategy with a two hours longer period appeared to be slightly better than the ones where the longer period was 30 minutes but both strategies failed to clearly demonstrate active technical trading. The support and resistance based trend strategy produced better results for the exchange traded futures but failed to meet the table entry criteria. The tests on the foreign exchange pairs produced weaker results for all strategies. All results by strategies and securities are summarized in table 4.1

	Foreign Exchange			Futures		
Strategy	EURUSD	USDJPY	GBPUSD	SP500	DAX	FTSE
Trend	✗	✗	✗	✗	✗	✗
Range Breaks	✗	✗	✗	✓	✓	✓
Mean Rev 2h	✗	✗	✗	✗	✗	✗
Mean Rev 30 min	✗	✗	✗	✗	✗	✗
Head & Shoulders	✗	✗	✗	✓	✓	✓

Table 4.1: Test results summary: technical trading has been detected (✓) or not (✗) in the strategy-security grid

4.6 Conclusions

In this chapter we developed a simulation framework and a particle filtering based methodology for detecting trading activity caused by executing trades based on a range of technical trading strategies on the foreign exchange and futures markets. We demonstrated that there are trading strategies based on intraday technical patterns that might be followed by technical traders in a relatively consistent way. The strategies that showed the clearest results are based on following the current intraday trend, breaking a pre-defined range and comparing the current price with the mean price computed from daily closing prices.

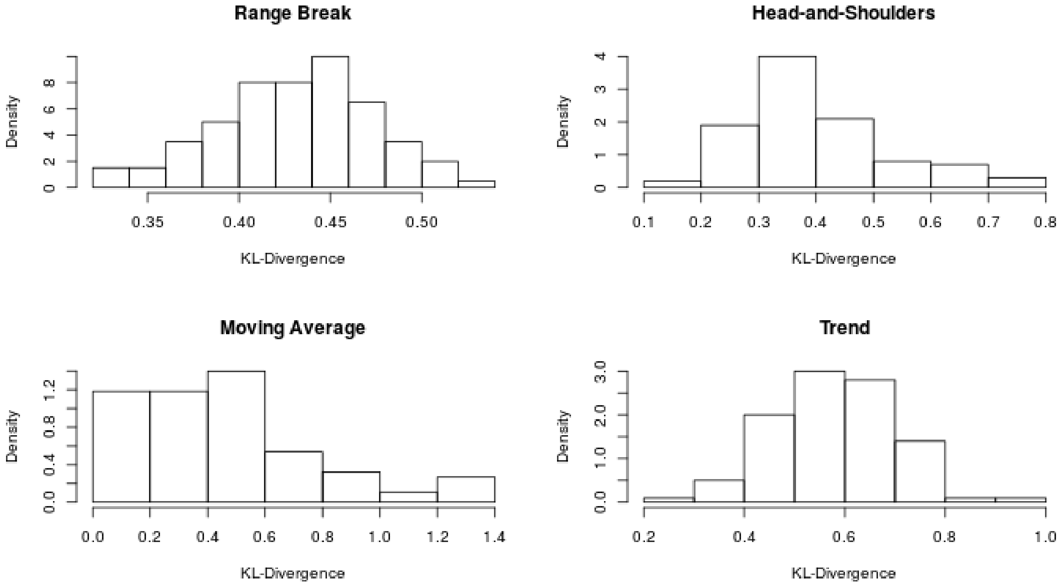
The tests in this experiment have been carried out with a small range of strategies but neither the methodology nor the implementation is limited to these strategies or this type of trading. Any trading strategy or traders' behavioural pattern with known parameters can be analyzed using the system that we developed for this experiment.

An integral part of the developed methodology are the algorithms for automated detection of technical patterns. The choice of the algorithm is of a major importance for successful implementation of the process. These algorithms are usually straight forward for the subset of technical patterns that are easy to express in numerical terms. It could be a significant challenge to automate technical patterns that are based on the 'chart reading skills' of the technical analyst or trader. In chapter 6 we present a new method, based on topographic prominence, for automating these price patterns. The analysis of the Trend and the Head-and-Shoulders pattern use that method.

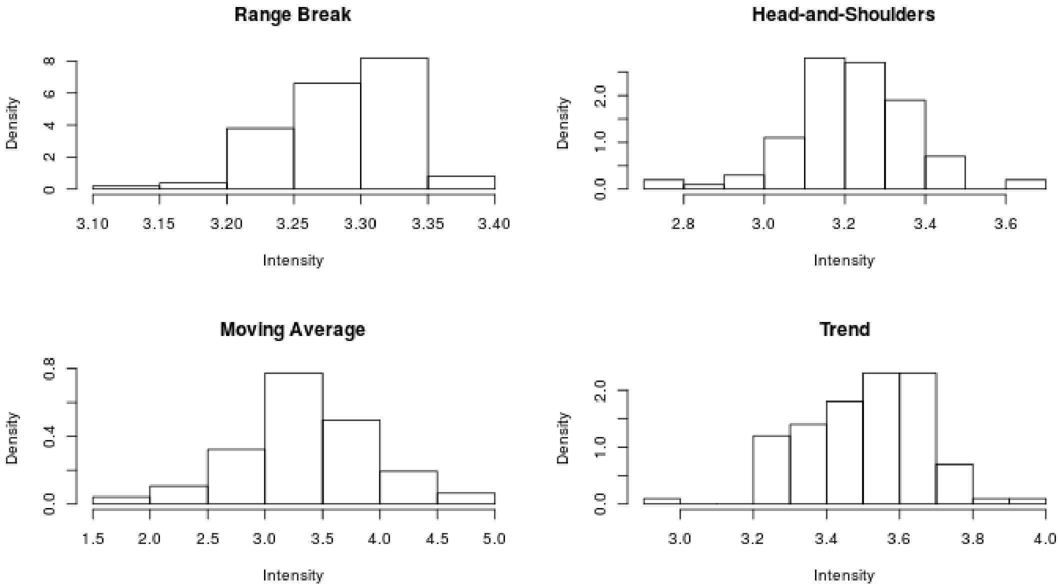
The range of instrument can also be extended to a much larger group as long as they are of comparable level of liquidity. A potentially interesting direction for extension of the instruments variety would be to include individual equities of the most liquid companies. Commodity futures are also a regular object of technical analysis and trading, especially natural gas, oil and metals.

The key factors that define the effectiveness of this experiment is the choice of strategies and the parameters used to specify these strategies. A very useful extension to this research would be to enhance the methodology with the ability to search and find the parameter set that is most actively traded for each strategy.

One of our main assumptions was that the background noise is Gaussian. It would be really interesting to relax this assumption by allowing other distributions for this process. Distributions that imply heavy tails would be the natural choice. These include Student distributions and distributions of the stable family.



(a) *Kullback-Leibler divergence of active trading intensities for the false positive simulation for each technical pattern.*



(b) *False positive simulations results for the trading intensities estimates of the active period for each technical pattern.*

Figure 4.5: *The above charts summarise the results of the false positive simulations. They depict two different measures of the false positive signals: (a) shows the KL-divergence of the active trading intensities of the false positive signals and (b) shows the values measured for the trading intensities during these active periods.*

Chapter 5

Option Pricing in the Presence of Technical Trading

This chapter describes the second experiment of the thesis. It investigates how the technical trading of a security influences the pricing of intraday options on that security. The chapter starts by presenting the experiment. It continues by introducing the theoretical framework of option pricing. Thereafter we explain the reason why the price cannot be described as a Markov process when the security is actively traded by technical traders. The experiment introduces a model that creates a Markov process by combining the price and a pattern specific set of additional variables. At the end of the chapter we present an example with a specific price process and one of the analyzed patterns.

5.1 Introduction

5.1.1 Introduction to Asset pricing and no-arbitrage

The idea of using the principles of dynamic hedging and no arbitrage requirements for pricing of options on financial securities has been introduced by Black, Scholes and Merton, see Black and Scholes [65], Merton [66]. The approach has been generalized for all contingent claims on securities in the papers Harrison and Kreps [67] and Harrison and Pliska [68] at the beginning of 1980s showing the mathematical justification. These authors described a general framework for pricing derivative securities and postulated the conditions for no arbitrage. They also introduced the link between martingales theory and risk neutral pricing. This served as a fundamental for many pricing models developed in the last thirty years. Duffie [69] and later Shreve [70] consolidated the ideas of no arbitrage asset pricing together with the mathematical apparatus that they are based upon. One major assumption that all these models utilize is that the security price is a Markov process. The intuition behind this assumption is based on the prevailing view within academic world that the current price of a security contains all the information available up to the current moment. This assumption is also an important requirement for the pricing models that use risk neutral pricing. In probability terms this assumptions can be presented as follows: a system is defined by a sample space Ω with a sigma algebra of events $\mathscr{F}$ and a filtration process $\mathscr{F}_t$ defined on it $(\Omega, \mathscr{F}, \mathscr{F}_t)$. The process X_t created by a random variable X is defined as Markov if at time t,

$$E(X_T|\mathscr{F}_t, \mathscr{F}_{t-1}, \ldots, \mathscr{F}_0) = E(X_T|\mathscr{F}_t) \tag{5.1}$$

for any $T > t$.

The Markov property assumption creates a convenient setting for building option pricing models. Unfortunately this does not always reflect the reality as described in the next section.

5.1.2 The Problem

The classic option pricing theory that we introduced in section 5.1.1 assumes that the security price is a Markov process. This assumption has been fundamental for the derivatives pricing models. In reality it has been demonstrated that this assumption does not always hold. Garzarelli et al. [22] investigated a range of stock price series and showed evidence for longer memory in the prices. In another research (see chapter 4) we presented a methodology for detecting the presence of technical trading (see chapter 2, section 2.5). Using that technique we demonstrated that some liquid securities are frequently traded based on the occurrence of some well known technical price patterns. The effect that this kind of trading creates on the security prices is another reason to consider the Markov assumption as unrealistic.

It is important to highlight the fact that the Markov assumption is not totally unrealistic. There exist a number of settings for which the assumption is valid and reflects the reality well enough so models based on it would produce reasonable results. The existence and the significance of Non-Markovian characteristics are defined by a number of factors. The origin of the factors can be specific for the security, the market it is traded on or the behaviour of the agents who trade the security.

The Markov property of the asset prices is closely related to the transparency of information. If we assume that all the information affecting the price is widely available on the market and reaches all market agents at the same time, then the assumption for a Markov process is justifiable. Most of the time this would be the case when we are dealing with long term trading horizons and prices quoted in very low frequency. The shorter the trading horizon or the maturity of the traded derivatives is, the more difficult it is to achieve the information transparency. At shorter horizons and higher frequencies the existence of technical trading, agents who initiate trades based on the previously observed price development, also affects the information flow and hence the Markov property assumption. Technical trading has been described in chapter 2, section 2.5. Thus the effect of the problem that we describe is concentrated on derivatives with a very short time to maturity which exist in high frequency environment. An example for that are the intraday options that we introduce in the next section.

5.1.3 Intraday options

Overview

This research focuses on options with a lifespan limited within a trading day. They are usually known as intraday options. This type of very short term options is a relatively new phenomenon on the financial markets. Historically options have been known as long term instruments used mainly for hedging and also for speculative purposes. Recently the development of high frequency finance created the need of very short term hedging. The extremely advanced low-latency technology, that high frequency finance uses, provides the necessary base for the creation and maintenance of intraday options. Similar to the longer term options, the intraday ones can vary in their characteristics. They can have a different payment profile. A number of platforms trade binary intraday options that pay a fixed amount or zero at maturity. The time point at which the options expire can be defined at pre-specified round day times or selected by the trader. In our definition of intraday options we include also the ones that have maturities longer than one day but their maturity time is during a trading day. The idea is that during the day when they are due to expire, they are exposed to the same risk factors as the intraday options that existed only during that day.

Types of Intraday Options

A broad range of online trading platforms offer different types of intraday options. Most often the options that are offered are based on foreign exchange but some platforms extend the underlying assets range to financial and commodities futures. The options also vary on other characteristics. Most of the trading platforms are focused on the so-called binary options that are in essence European style digital options. OANDA[1], one of the biggest foreign exchange trading platforms used to offer the so-called 'box options'. Similar to digital options, the box options paid a fixed amount if the foreign exchange pair registered any trades within a predefined price range during a fixed time interval. The options have also been offered as a 'miss' version, that paid a predefined amount if the price did not enter the time-price box described above. Currently the platforms offer to the traders only to buy these type of options but they claim that once they are ready they will offer the opportunity to the traders to write options too.

Regulatory Initiatives and Other Issues

High frequency finance and automated trading and market making developed during the last ten to fifteen years into a whole new industry within the financial services sector. This new technology extensive industry gained a great number of supporters mainly from the trading, financial engineering and information technology circles. At the same time a small number of errors and misuses became the reason for its notoriety and attracted the attention of the regulators of many of the developed market countries. There were ideas to limit and even ban ultra high frequency trading in order to protect ordinary investors. In the recent years the regulators' focus has been on demanding from the

[1] see Oanda [71]

banks and the trading companies to ensure that they have implemented solutions for monitoring and controlling all the risks that ultra high frequency fully computerized trading is susceptible to.

Intraday options are considered to belong to the high frequency world and as such they have also been affected by the regulatory response that the whole industry caused. The trading platforms offering intraday options have to constantly adjust to the regulatory requirements. In 2012 OANDA suspended their box options in response to the US "Dodd-Frank Wall Street Reform and Consumer Protection Act.

5.1.4 Research on price processes with technical trading and non-Markovian characteristics

Technical Trading

The effect of technical trading on derivatives pricing has been researched by a number of authors. Feng et al. [48] introduce an agent base model to analyse the long memory of the prices. They demonstrate how the existence of certain types of traders especially the ones using technical rules contribute to the formation of fat tails in the distributions of asset returns. the authors describe a stochastic model that is based on the agent-based models and show that the model can be parametrized using market data only.

Shaw and Schofield [50] divide market participants into two types: fundamental and technical. Similar to this research the authors assume that the two groups of traders operate independently. They proceed by describing the buy and sell orders of each of the two types. In the case of fundamental trading the authors assume that the number of arriving trades for a given time interval is independent of the size of the trades and that the trade sizes are independent and identically distributed. Following from these assumptions they derive the expectation and the variance of Z, the number of lots of size L, traded as a result of buy orders as:

$$E(Z) = E(Y)E(N) = E(Y)\bar{n} \tag{5.2}$$

$$Var(Z) = Var(Y)\bar{n^2} + E(Y)Var(N) \tag{5.3}$$

where N_i is the number of lots in a buy order, $E(N) = \bar{n}$ is the average number of lots across all buy orders and Y is the number of orders arriving during a time interval. The authors assume a general renewal process with arrival rate λ_B for the intertrade arrival times Y and based on that they derive the distribution of the inter-trade times P. If $\lambda_P = \frac{1}{E(P)}$ they show that

$$E(Y) \sim \lambda_P \Delta t \tag{5.4}$$

and that

$$Var(Y) \sim Var(P)\lambda_P^3\Delta t \tag{5.5}$$

Shaw and Schofield [50] show that the above relationship can also be represented as:

$$\begin{aligned} E(Y) \sim& \lambda_P \Delta t && (5.6)\\ Var(Y) \sim& \gamma_P \lambda_P \Delta t && (5.7) \end{aligned}$$

where γ_P relates the inter-trade time distribution to a Poisson structure. With the help of λ_P and γ_P relabelled as λ_B and γ_B to denote the fundamental buy orders, the previous expressions for the expectation and the variance of the number of lots become:

$$\begin{aligned} E(Z) =& \lambda_B \Delta t E(N) = \lambda_B \Delta t \bar{n} && (5.8)\\ Var(Z) =& \lambda_B \Delta t(\gamma_B \bar{n^2} + Var(N)) = \lambda_B \Delta t(E(N^2) + (\gamma_B - 1)\bar{n^2}) && (5.9) \end{aligned}$$

From the above the total number of shares traded in buy orders M_B can be expressed as:

$$\begin{aligned} E(M_B) =& L\lambda_B \Delta t \bar{n} && (5.10)\\ sd(M_B) =& L\sqrt{\lambda_B \Delta t(E(N^2) + (\gamma_B - 1)\bar{n^2})} = L\sqrt{\lambda_B \Delta t(\gamma_B \bar{n^2} + Var(N))} && (5.11) \end{aligned}$$

Where L is the lot size, as defined above.

Following the same logic, using λ_P and γ_P this time as λ_S and γ_S, the authors derive the expressions for the fundamental sell orders volume M_S:

$$\begin{aligned} E(M_S) =& L\lambda_S \Delta t \bar{n} && (5.12)\\ sd(M_S) =& L\sqrt{\lambda_S \Delta t(E(N^2) + (\gamma_S - 1)\bar{n^2})} = L\sqrt{\lambda_S \Delta t(\gamma_S \bar{n^2} + Var(N))} && (5.13) \end{aligned}$$

When all the fundamental orders are aggregated together the total net volume for fundamental orders M_F becomes:

$$\begin{aligned} E(M_F) =& L(\lambda_B - \lambda_S)\Delta t \bar{n} && (5.14)\\ Var(M_F) =& L^2 \Delta t[\lambda_B(\gamma_B \bar{n^2} + Var(N)) + \lambda_S(\gamma_S \bar{n^2} + Var(N))] && (5.15) \end{aligned}$$

In a similar fashion the authors introduce the trades by technical traders using the trade arrival rates μ_B and μ_S for buy and sell respectively. It is important to highlight the fact that this model allows periods of inactivity for the technical traders by setting $\mu_i = 0$. This is a major difference from models where technical traders are assumed to be trading all the time. The equations for the expectation and variance of the technical trading when the correlation between buy and sell trades is ρ^T are presented as:

$$E(M_T) = L(\mu_B - \mu_S)\Delta t \bar{n_T} \tag{5.16}$$

$$Var(M_T) = L^2 \Delta t(\mu_B(\gamma_B^T \bar{n_T^2} + Var(N_T)) + \mu_S(\gamma_S^T \bar{n_T^2} + Var(N_T)) \tag{5.17}$$

$$- 2\rho_T \sqrt{\mu_B \mu_S(\gamma_B^T \bar{n_T^2} + Var(N_T))(\gamma_S^T \bar{n_T^2} + Var(N_T)))} \tag{5.18}$$

After introducing a return impact function in a general form the authors continue by linearizing the functions and presenting the derived processes in continuous time in the form of stochastic differential equations. The volatility of the combined return is sourced by two independent Brownian motions W_1 and W_2. The SDE becomes:

$$dX_t = (\mu_1 - \mu_2 X_t)dt + \sigma_1 dW_{1t} + \Sigma_2 dW_{2t} \tag{5.19}$$

The indices 1 and 2 serve the fundamental and technical trading respectively. Shaw and Schofield [50] investigate the characteristics of the derived SDE and suggest different approaches for finding a solution.

Although the research of Shaw and Schofield [50] allows technical trading to occur only at certain times, it does not try to define those times. The research stays at generic level without going into the details of relating the technical trading occurrences to the price patterns that caused them.

Non-Markov Price Process

The assumption that security prices possess the Markov property has already been challenged by a number of researchers and practitioners, including analysts and traders. Kholodnyi [72] researches non-Markovian features of the power markets. The author describes the price spikes, a phenomenon typical for the power markets, and analyzes how the pricing of contingent claims are affected by the existence of the spikes. The price process is represented as a two state Markov process. The two states are formulated as 'spike state' and 'inter-spike state' and are defined based on whether the price is running through a spike or is between spikes. The author uses this formulation to describe the probabilities of the price staying in its current state or moving to the alternative one as a classic Markov chain in continuous time. The transition matrix for the process in times $0 <= t <= T$:

$$P(T,t) = \begin{pmatrix} P_{ss}(T,t) & P_{sr}(T,t) \\ P_{rs}(T,t) & P_{rr}(T,t) \end{pmatrix}$$

where s denotes a spike state and r denotes a regular, non-spike state.

The generator for the Markov process based on a

$$L(t) = \frac{d}{dT} P(T,t) \mid_{T=t} \tag{5.20}$$

becomes the matrix:

$$L(t) = \begin{pmatrix} L_{ss}(t) & L_{sr}(t) \\ L_{rs}(t) & L_{rr}(t) \end{pmatrix}$$

Following the theory of Markov processes the above transition matrix can be created by the generator using the following exponentiation:

$$P(T,t) = e^{\int_t^T L(\tau) d\tau} \tag{5.21}$$

The author highlights the time-homogeneous case when the transition probability becomes a function of simply the time interval $(T-t)$. In this case the above matrices simplify to:

$$P(T-t) = \begin{pmatrix} P_{ss}(T-t) & P_{sr}(T-t) \\ P_{rs}(T-t) & P_{rr}(T-t) \end{pmatrix}$$

and for the generator:

$$L = \begin{pmatrix} L_{ss} & L_{sr} \\ L_{rs} & L_{rr} \end{pmatrix}$$

and then equation (5.21) becomes:

$$P(T-t) = e^{(T-t)L} \tag{5.22}$$

In either cases the transition probabilities can be decomposed with the help of (5.21) and (5.22) respectively (see Kholodnyi [72]).

Using the above settings Kholodnyi [72] defines a spike process λ_t that identifies the presence of a spike and also its magnitude in the case of the price being in a spike state. The process λ_t achieves that by accepting the value of 1 if the price is not in a spike state and the value of a process $\xi_t > 1$ during a spike state. The process ξ_t is created by independent random variables ξ with the probability density function $\Xi(t,\xi)$. The spike process λ_t is actually a Markov process when it is defined as above. As such its transition probabilities can be derived using the transition probabilities of the Markov process M_t defined further above.

Modelling the spike occurrences of the price as a Markov process is the most important step for including the spikes into the price process for the purposes of pricing contingent claims on that security. The next step is to combine the spike process with a standard price process. Kholodnyi [72] uses a general SDE process:

$$d\hat{\Psi}_t = \mu(\hat{\Psi}_t, t)dt + \sigma(\hat{\Psi}_t, t)dW_t \tag{5.23}$$

where μ and σ identify the drift and the volatility of the process following the widely accepted convention. As a special case Kholodnyi [72] looks at a geometric mean-reverting process in the form of:

$$d\hat{\Psi}_t = \eta(t)(\mu(t) - \log\hat{\Psi}_t)\hat{\Psi}_t dt + \sigma(t)\hat{\Psi}_t dW_t \tag{5.24}$$

where $\eta(t) > 0$ is the mean reversion rate, $\mu(t)$ is the equilibrium mean and $\sigma(t) > 0$ is the volatility. The process $\hat{\Psi}_t$ formulated this way is non-Markovian when price spikes are possible. The author suggests to convert the process to Markov by extending its state space, i.e. by creating a new process out of the possible pairs of the values of the processes $(\hat{\Psi}_t, \lambda_t)$. The intuition behind this is that at any point in time the knowledge of the price level and its current characteristics together with the information about the existence of a spike and the probabilities of new ones occurring consolidates everything that has been acquired of the price history. The history itself cannot contribute any further. The author demonstrates the usage of this model for pricing European contingent claims on power with spikes.

Intraday Options

The development of high-frequency finance and the automation of trading attracted a lot of research interest in the last decade. Another factor that contributed to this development was the big increase of the availability of price data of much higher frequency than the one used in the past. The fast development of data storage technologies created the possibility to store enormous amount of data at a very affordable price. Banks, trading companies and data providers started accumulating data down to tick-by-tick levels and made it available either commercially or freely to researchers. Initially the most of the researchers concentrated on the characteristics of the high-frequency data series. Some explored the microstructure of the prices (see Lyons et al. [73]), and some, such as Dacorogna et al. [74], became the base for a lot more research on high-frequency finance and trading.

Pricing of financial derivatives in the new pricedata-rich and high-frequency environment attracted recently also a fair amount of interest. The newly developed models, such as Cartea [75] and Cartea and Meyer-Brandis [76] among many others, start from modelling the tick-by-tick data and the times between the trade arrivals. Most of the researchers do not explicitly look into intraday traded short term derivatives such as intraday options. The reason for this is probably the fact that this kind of derivatives are still at the beginning of their development cycle (see section 5.1.3 above). Scalas and Politi [77] is one of the few who focus specifically on intraday options. They suggest a stochastic model for the asset price using a pure jump-diffusion process to describe the tick-by-tick price stream. The authors model the price and the tick-by-tick logarithmic returns using the trade arrival times and the inter-trade durations. They derive the distribution of the trading epochs representing it as an n-fold convolution of the distribution of the durations. The log price $X(t)$ is then represented as the sum of all log returns Y_i that occurred as a result of trade i. The price itself is created by exponentiating the log price or by the product of the exponentiated log returns. Scalas and Politi [77] carry on by deriving the conditions for which the martingale condition is satisfied for the price that has been created in the above described way. Equipped with a price process that is a martingale they proceed onto using it for pricing intraday options.

5.1.5 Our Solution

As described in section 5.1.2 the existing research does not take into account the non-Markovian consequences of intraday technical trading. We suggest a solution for extending the price process aiming to satisfy the Markov condition. We do this on a pattern by pattern basis, since every pattern requires a bespoke approach. We also design an example that utilises the approach suggested in the experiment.

5.1.6 Contributions

The contribution of this experiment is to help create more realistic pricing models for intraday options. This is important for trading and hedging of these options. In addition to that the research can be used for implementing improved methods for hedging and position management of an automated market-making system. The pattern analysis that we carry out in the experiment can be applied directly when those patterns are detected to be traded on the underlying security. The research can also be used as an example how to analyze additional technical patterns in order to create more comprehensive intraday pricing models.

5.1.7 Structure of the Chapter

The structure of this experiment is as follows. In the next section 5.2 we present a brief overview of the probabilistic framework for pricing European options under the risk-neutral measure and an underlying asset whose price satisfies the Markov property. In section 5.3 we introduce the model that deals with the non-Markov behaviour in presence of technical trading. The following section 5.4 presents the technical patterns that this experiment uses. We describe briefly the patterns and identify the set of parameters that the model needs for each one of the patterns. Section 5.5 presents the simulations for three different patterns: a generic randomly generated continuation technical pattern, a range breaks pattern and a moving average one. We demonstrate the effect on the price of a European call option on the security when technical trading on each of these patterns is active. The last section 5.6 concludes and presents suggestions for improvement and future work.

5.2 Option pricing within Markov framework

5.2.1 General considerations

In this section we are reviewing the general framework for pricing European options on financial securities with the intention to fit it in the environment that this experiment focuses on. Since this research does not concentrate on how the overall price process evolves, it is safe to assume that the underlying price X_t is a continuous stochastic process defined for the time period $t \geq 0$. The overall environment in which the price exists is further defined by a sample space Ω with a sigma algebra of events $\mathscr{F}$. The information retrieval is represented with a filtration $\mathscr{F}_t$ such as $\mathscr{F}_t \subset \mathscr{F}$ for $t \geq 0$. This results in a filtered measured space $(\Omega, \mathscr{F}, \mathscr{F}_t)$. A probability measure P is defined on the measured space $(\Omega, \mathscr{F})$ and is used initially for the price process S_t. The resulting filtered probability space is $(\Omega, \mathscr{F}, \mathscr{F}_t, P)$.

Given these settings and also the important assumption that the security price is a market process the standard option pricing approach is to define an alternative probability measure Q that is equivalent to P. The new probability measure is defined in a way that the discounted price process is a martingale. This measure is usually called risk-neutral. The current value (at time t) of a contingent claim on the price X_t that makes a payment $C_T(X_T, T)$ at a maturity time T can be expressed as:

$$c_t = \mathbf{E}_Q(D(t,T)C_T(X_T,T)|\mathscr{F}_t) \tag{5.25}$$

In the above equation (5.25) $\mathbf{E}_Q$ is expectation under the risk-neutral probability measure and $D(t,T)$ is a discount process.

5.2.2 The discount process and the risk-free interest rates

In general interest rates are not accrued during the day and are added overnight. That has always been the approach for longer term options. Once we move to intraday instruments and higher frequency trading the approaches vary. Scalas and Politi [77] provide a justification why risk-free interest rate can be fully ignored. It is based on the assumptions that first the risk-free interest rate levels are orders of magnitude smaller than the underlying security returns, and second the proportionate rates for small intraday periods would be too small to be worth including. If any of the above two conditions is not met it is justifiable to introduce the notion of intraday interest rates. This approach contributes also to a more general formulation of the model. The simplest implementation that comes as a natural extension of the existing convention is to accrue equal proportions of the daily interest rate at a smaller interval. That could be $\frac{1}{24}$ every hour, $\frac{1}{24*60}$ every minute or $\frac{1}{24*60*60}$ every second. Depending on the platform the concept that we are introducing allows to extend the implementation even to subsecond intervals. The continuous version of this idea is to use an instantaneous rate that integrates up to the daily value. This approach allows also using non-constant interest during the day. In real-life implementations of this concept the values for the interest rates can be updated by a high-frequency fixed income trading platform. A simple choice that does not compromise the idea is to assume a constant interest rate r for the lifetime of the option. In this case the equation (5.25) for the value of the contingent claim becomes:

$$c_t = e^{(r(T-t))}\mathbf{E}_Q(C_T(X_T, T)|\mathscr{F}_t) \tag{5.26}$$

5.3 Non-Markovian price in the presence of technical trading

When one or more technical patterns are actively traded the price process cannot be considered as Markov any more In this experiment we propose a method for overcoming this problem by extending the state space of the underlying process in order to fulfil the Markov property. This is achieved by analyzing a set of patterns one by one in order to identify the parameters that need to be included in the extended process.

The intuition behind this approach goes as follows. If a certain technical pattern is actively traded the full disclosure of information at any point in time must also include the current state of that pattern. 'Has it been observed already?' 'Is it currently being created?' 'How many of its parts have been identified?'

Define the parameters of a pattern occurrence and estimate them from the price data. For certain patterns that estimate can be derived using the underlying characteristics of the price process. For example the likelihood of observing a break out pattern can be directly derived by the current view of market volatility. We can either measure this or use the value currently implied by the options market of the underlying asset. For more complicated patterns the likelihood of occurrence can be estimated by processing a historical sample. When the methodology is used to reflect the effects of different types of trading behaviour the likelihood of occurrence is defined by the frequency of appearance of the events that are expected to initiate the analyzed traders' behaviour.

5.4 Technical Patterns

The different kinds of technical patterns and the specifics around their trading define the way we extend the state space to satisfy the Markov property requirement. The directional type of the pattern, i.e. continuation or reversal, defines the way in which the technical traders will execute trades when the pattern is observed. The trading frequency of the pattern defines whether it will be traded next time it occurs. Some patterns, for example the ones based on price mean reversion, are traded every time they are observed. On the other hand, patterns of the 'break out' type are traded once per observational period. In our intraday trading based experiment, the information that a break out pattern has already been observed means that it will not be traded anymore on that day. That kind of feature is coded with a binary variable that accepts the value of one if the pattern has already been observed and zero if it is still to be watched for.

We assume that the security price process P_t is one part of an extended process X_t that is of Markov type:

$$X_t(P_t, \Xi_t) \tag{5.27}$$

In the above formulation Ξ_t is a process that is built of random variables specific for each pattern following the logic described above. These are the variables which values need to be known at time t in order to have the full information regarding the technical pattern development up to this time. In the following subsections we describe the variables that are required by every pattern. The variables that we include here are different from the parameters that define the patterns. The latter are part of the settings and they are assumed to be known all the time, hence they do not need to be included in this.

5.4.1 Break out

The ‘break out’ type of technical patterns trade once per observational period after the price fluctuates beyond some pre-defined range in either positive or negative direction (see chapter 2, section 4.4.2). The information whether the pattern has already occurred is coded by a random variable η that takes values 0 or 1. If $\eta = 0$ the pattern has not been observed yet on the current trading day and this is reflected in the option pricing. If $\eta = 1$ the pattern has already been observed, its trading time has expired and the existence of the pattern is ignored for the rest of the day. The variable η is reset to 0 at the beginning of each trading day.

5.4.2 Mean reversion

The mean reversion type of technical patterns have been reviewed in chapter 2, section 2.5.5. The trading strategy based on the pattern watches the value of two moving averages calculated on the price with different backward looking period. A trading signal is generated when the two average series cross each other. The information that needs to be coded in the process X_t, if such trading strategy is active, has to contain the current values of both long and short period moving averages. This can be done by defining two random variables $\bar{l}_t$ and $\bar{s}_t$ for the longer and shorter averages respectively.

5.4.3 Head-and-shoulders

The ‘Head-and-shoulders’ is a more complicated technical pattern. It has been described in chapter 2, section 2.5.6. The Markov property is achieved by knowing the level of current development of the ‘Head-and-shoulders’ formation. The potential development of a pattern is monitored with the help of linearization process that runs parallel with the price updates.

In chapter 6 we develop an alternative idea of price linearization. We use the notion of price prominence that we introduce in that chapter. We demonstrate that this is a better reflection of the way technical analysts and traders identify certain types of technical patterns. ‘Head-and-shoulders’ is an example of such a pattern. In order to include the possibility of technical trading on the 'Head-and-shoulders' pattern the price process must be extended with the price prominence levels and the development of the pattern elements based on these prominence levels.

5.4.4 Trend based on support and resistance levels

The technical analysts' idea of support and resistance levels on the price series has been introduced in chapter 2, section 2.5.4. In chapter 6 we develop an automated detection of the pattern using price prominence. The mechanics of the pattern formation is the following: two support levels in the direction of the current price trend form a potential trend line; the trend is either confirmed by a third support level in the same direction, or invalidated by the price breaking through the line. When the trend is confirmed by a third support level on the trend line, the pattern is considered complete and the technical trading on the pattern is initiated. The parameters of the currently existing trend line need to be included in the process X_t in order to achieve Markov property. This can be done by adding two additional variables s_t and i_t for the line slope and intercept respectively.

5.5 Simulation

This section demonstrates the effect of technical trading on the intraday prices and returns of a security and following from that the effect on the price of a European intraday call option that has the traded security as an underlying asset. We built a simulation framework that provides price series that fluctuate randomly following a discrete GBM process. The simulation comprises the following steps: first we generate minutely returns for a 12 hours trading day for one hundred thousand days by drawing from a normal distribution with mean zero and standard deviation that is equivalent to 15% annual volatility; we accumulate the return changes to create the intraday cumulative return series and finally we generate the price series as the exponent of the returns. When a technical pattern event is emitted by the system either as a result of automated detection or simulation, a technical trading is simulated that reflects on the price as described in chapter 4, section 4.3.4. After a technical pattern has been observed we generate a technical trading return component by drawing from a Gamma distribution with a shape of $\lambda = 7$ that represents the trading intensity and rate $\beta = 0.0001$. The generated values are added to the minute return values and the price series is regenerated. The option price is calculated as the averaged simulated value of a European call option that expires during the trading day. We simulate options with strike prices that range from 1.0 to 1.03 and time to maturity between 1 and 11 hours. The option price that is the result of the simulation is compared with an option price of the same security without the technical trading effect.

For the price series X_t we use a discretized geometric Brownian motion (GBM) with a zero drift and constant volatility σ:

$$\Delta X_t = \sigma X_t \Delta W_t \tag{5.28}$$

with random standard normal increments ΔW_t. The simulation of the technical patterns on the price process is described in the following sections.

There are a range of factors that define how the technical trading of a price pattern affects an intraday option on that security. These factors include:

- **The type of the pattern.** A pattern that triggers technical trading in the direction of the current price moves creates an opposite effect to a pattern that initiate reversal trading.
- **The frequency of the pattern.** In the case of patterns that can occur once a day, the effect on the price is concentrated only at the time when the pattern appears.
- **The option maturity time.** The likelihood of price pattern occurrence is different through the trading day. An option that expires when patterns are more likely to appear will be affected more by the technical trading caused by those patterns.
- **The expected time of pattern occurrences.** The arguments that were stated in the last point are valid but from the aspect of the likelihood of the occurrence of the price patterns.

5.5.1 A generic technical pattern

Before simulating the occurrences of any of the analyzed patterns we simulate the more general case of observing a technical pattern as a whole. In fact it does not make any difference how a pattern evolves or what pattern it is, once it has been observed. Following this line of logic we simulate a continuation pattern of generic type. Another way to explain this is that we simulate only the times when the pattern is observed. The occurrences of the pattern are simulated by a random process with buffers around the pattern trigger times. This is needed to achieve more realistic simulation. The buffers guarantee a minimum time between two consecutive patterns. The direction of the signals based on the pattern occurrences follows the current price momentum. This effectively creates a continuation technical pattern.

Figure 5.1 depicts the value added to an intraday call option when the underlying security is subject to technical trading based on the above pattern. The positive surface demonstrates that the effect on the option continues through the day and is valid along the analyzed range of strikes.

5.5.2 Break out

This pattern is straightforward to simulate as it is. In order to demonstrate results of higher significance we adjust the parameters of the pattern in order to increase its frequency. This can be achieved by either decreasing the break out threshold of the pattern or by increasing the volatility of the simulated series. In this simulation, similar to the previous one we set the option expiry times at the end of each hour during the active part of a trading day. In figure 5.2 we plot the difference between the value of a generic call option and a call option for which the security is subject to range breaks technical pattern trading. The positive z-axis values demonstrate that ignoring the information about the

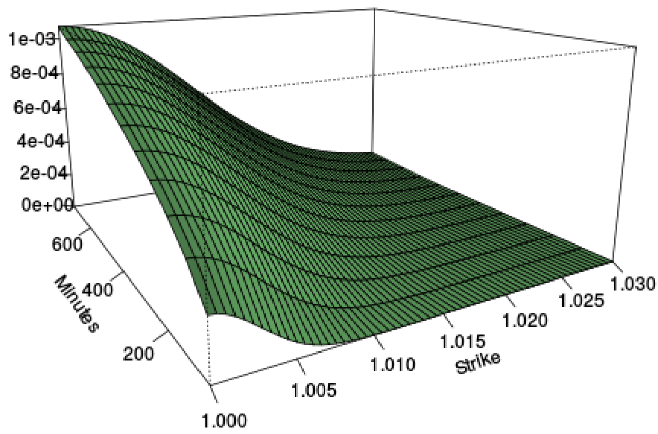


Figure 5.1: *Surface of the value added to an intraday call option on a security that is subject to randomly generated technical trading. The vertical z-axis measures the increase of the option value in currency units when the effect of technical trading is added to the price process simulation.*

technical trading leads to underestimating the option value. The nature of the pattern suggests that it would take some time to the price to reach and break the range of the pattern. That explains the fact that the shorter maturity options (one and two hours) are not affected by the technical trading.

5.5.3 Mean reversion

The mean reversion is easy to simulate as it is. Its simple formulation allows the detection of the pattern to be included in the actual pattern simulation. We use the comparison between one minute as the short period moving average, i.e. the close price of the current minute, and 30 minutes moving average as the long period to generate moving average trading signals. Such a strategy creates signals with relatively high frequency. Given the higher frequency we assume that the strategy is traded only during the minute when the signals trigger. The intraday option expiration times are at the end of every hour, the same as in the previous simulations. Figure 5.3 depicts the surface of the value added to a call option when moving average strategy is active on the security. When considered on the same scale, the effect of the moving averages trading is smaller than the effect of the range breaks. There are two reasons for that. Firstly, the range breaks trading acts in the direction of the current price move and secondly, the moving average signal is of higher frequency and following from that, less capacity, i.e. it initiates smaller amount of trading volume. On the other hand, the simulation demonstrates that the effect of the moving averages trading is observed from the beginning of the day.

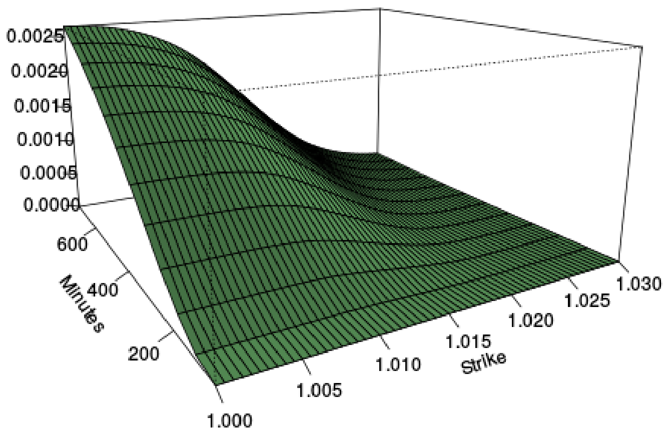


Figure 5.2: *Surface of the value added to an intraday call option on a security that is subject to range breaks trading. The value that is added to the option is of the same magnitude as the option price without technical trading. As a result of that the option can double its value after technical trading is included.*

5.5.4 Volatility Analysis

In this section we present the effect of the technical trading on the intraday return volatility. For the strategies that we simulated above, we measured the realized volatility for each of the option maturities that we simulated. These maturities vary from one to twelve hours. The volatility that has been used for the price process simulation is 0.034% that corresponds to an annual volatility of 15% scaled down to minutely frequency. The numbers are scaled down using the square root of time rule and assuming 260 trading days in a year and 12 hours trading per day.

A summary of the results is depicted in Figure 5.4. When technical trading is not present the realized volatility that any of the options face is equal to the underlying volatility used for the simulation of the price process. These are the black points on the chart. The range break out trading results in an increase of the realized volatility. The trades that this type of technical strategy generates are unevenly distributed within a trading day. Fewer trades are expected at the beginning of the day when the price did not have enough time to reach and cross the borders of the monitored range. As a result of this the effect on the volatility grows during most of the trading day. The increased volatility contributes to the higher option prices when the security is subject to range break out technical trading. For the moving average case, the trading of the technical pattern generates a negative effect on the volatility. The simulation demonstrated that all option maturities experienced lower realized volatility than the base case.

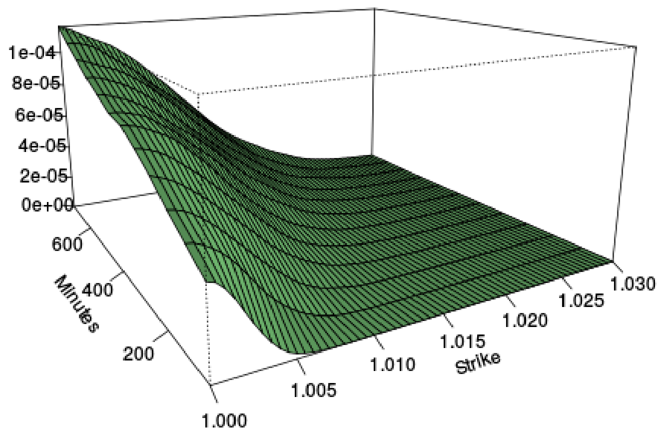


Figure 5.3: *Surface of the value added to an intraday call option on a security that is subject to moving averages trading*

5.6 Conclusions

In this chapter we described how technical trading of specific price patterns affects the value of intraday options on that security. The methodology that we developed in chapter 4 allows to identify technical patterns that are actively traded on a security and measure the intensity of that trading. Once the technical trading intensity information is available it has to be reflected in a model that prices intraday contingent claims on that security. This is achieved by adding the technical trading information to the price data. The combination of the current price and the state of the traded technical patterns results in a Markov process that can be plugged into intraday pricing models.

There are two important considerations about the framework that we introduced in this chapter. The first one is that the information that needs to be added to the current price state is pattern specific. This is a major difference from the existing literature on the subject where technical trading is usually consolidated into a single process. In this experiment we explored the specifics of each of the analyzed patterns and summarized the random variables that are needed to extend the state space in order to make the compound process Markov. The second consideration is that the above observations are more valid and easier to isolate when the technical trading activity occurs at or close to the maturity time of the claim. This is the reason this research focuses on intraday technical trading.

In section 5.5 we demonstrated the effect on an intraday European call option value when the technical trading on the underlying security is ignored. The strength of the effect depends on the type of the pattern - continuation or reversal, on the frequency of the pattern and on the maturity time of the option.

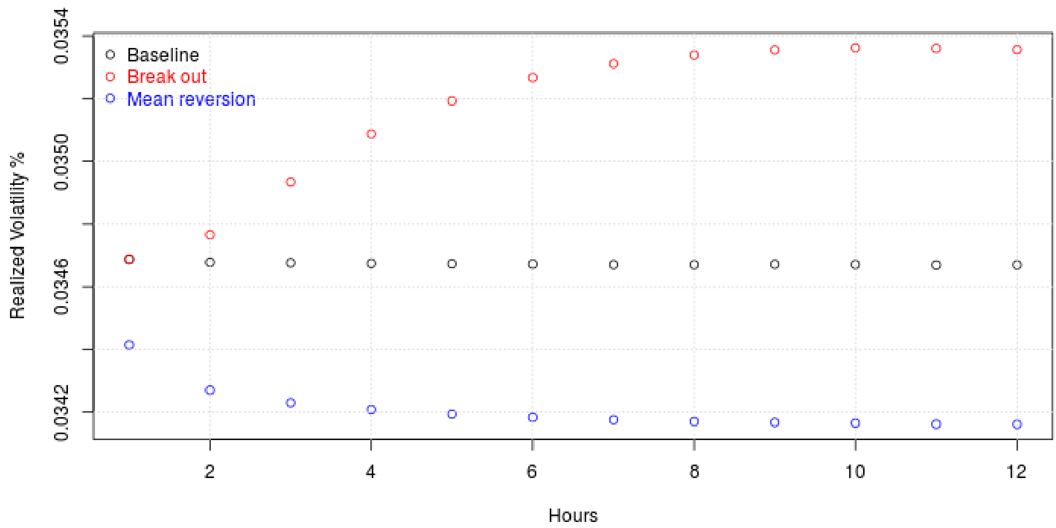


Figure 5.4: *Realized volatility of a European option on a simulated security with time to maturity from 1 to 12 hours. The black points show constant volatility when the price is not affected by technical trading. The red points show the effect on the volatility when range break out strategy is traded and the blue points show the effect of moving average trading.*

There is a range of directions that this research can be extended to. The portfolio of technical strategies parameters can be extended to include more of the commonly traded patterns as well as any that the implementation of chapter 4 demonstrates to be actively traded. A logical next step could also be the extension of existing contingent claims pricing models with the technical patterns trading process and real data test. The concept that we introduced in this experiment can be transferred to higher frequencies to be integrated into market making algorithms.

Chapter 6

Technical Trading, Prominence and Liquidity

This chapter describes the third experiment of the thesis. The experiment introduces a new method for automated detection of technical price patterns. The method uses the topographical concept of prominence to linearize the price path. The method is then used for locating the occurrences of two types of technical patterns: "Head and Shoulder" and trends based on support and resistance levels. In the second part of the experiment we test a broad range of securities for patterns that are actively traded and explore the relationship between the result and the liquidity profile of the analyzed securities.

6.1 Introduction

6.1.1 Existing approaches in automation of technical price patterns

In chapter 2, section 2.4 we presented an overview of the most popular methods for automation of technical patterns. The methods can be divided into two groups. The first one is based on linearizing the price into a sequence of straight lines. The second group comprises the methods that smooth the price line and by doing this create a smaller set of minima and maxima. Based on the utilized approach to smooth the price series the methods can be divided into the following categories: kernel based smoothing, regression based linearization, modal smoothing and polynomial regression based smoothing. Using kernel based smoothing on the price series for automated detection of the most common patterns has been introduced by Lo et al. [17]. Since then the kernel based smoothing has been used by a wide range of implementations both in academic research and in the financial industry. As reviewed in chapter 2, section 2.4 Shaw [18] explains the shortcomings of the kernel based regression. These are mainly the sensitivity to the choice of the bandwidth and the distortion at both ends of the observed time interval. Shaw [18] introduces the modal smoothing using orthogonal polynomials (see Burden and Faires [78]) as a cure for this problem. The intuition behind Shaw's modal based method is that the significant part of the price movements can be described with the help of only a few basic levels and shapes, such as an overall price level, slope and some curvatures. Starting with the most significant one, Shaw derives the next one finding an orthogonal polynomial that corresponds to the desired feature. As stated there, this approach does not suffer the shortcomings of the kernel based regression but introduces too much noise in the resulting series. The method that Lo mentions in Lo et al. [17] and then Shaw delves further into is the local polynomial regression method (see Fan and Gijbels [19]). It produces a smoothed price without suffering the problems of the previous two methods. The choice of the bandwidth is also important but there are ways to choose an optimal bandwidth see (Fan and Gijbels [19], Ruppert [79] and Ruppert et al. [80]).

6.1.2 The problem

The existing research on automating price patterns identification concentrates on smoothing the price series. By smoothing the series two positive effects are achieved at the same time. First, the background noise is removed and second, the price series is approximated by a 'nice' function that it is much easier to be analysed further. The different implementations of this concept usually suffer some specific implementation related problems. We discussed these in the previous section and in chapter 2, section 2.4. One of the most important characteristics of the technical price patterns is that they are creations of the human vision. Our experience from analyzing the technical patterns and discussions with professional technical analysts is that the way the human eye creates patterns from the chart is completely different to the way the currently used algorithms would do it. When a technical analyst looks at a chart, he linearizes the series instead of smoothing it. The lines are defined by the levels that stand out from their environment, i.e. the ones that are more prominent. This is a completely different approach from smoothing and can lead to very different results.

6.1.3 Our Solution

In this experiment we propose a solution to the above problem using a concept from topography called *prominence*. Using this concept we can linearize the price series by automatically ignoring the small changes without affecting the significant levels. We claim that this approach is much more aligned with the way the human eye analyzes a noisy price series and looks for patterns. In this experiment we introduce the concept of topographic prominence and explain how it applies to the price series. We program an implementation of the concept and run the code to automate the detection of the technical pattern 'Head-and-shoulders' as well as technical trends based on support and resistance levels.

6.1.4 Contribution

The first major contribution of this experiment is to extend the existing methods for automation of technical price patterns by a new one that is conceptually different to the existing computerized implementations. It is closer to the way a human technical analyst operates and it produces better results for certain patterns. The next contribution is presenting an algorithm for the method implementation. The third contribution comprises an empirical investigation on identifying the patterns using the method and performing a test for the patterns whether they are actively traded.

6.1.5 Structure of the Chapter

The plan of this chapter is as follows. In the next section 6.2 we introduce the method of topographic prominence for automation of technical patterns. We explain the concept, we discuss the implementation and demonstrate how it works. In section 6.4 we introduce the notion of liquidity and present different ways for quantifying it. In the following section 6.5 we analyze a range of securities with different liquidity to explore how technical trading activity is affected by liquidity. The last section 6.6 concludes and lists suggestions for improvement and future work.

6.2 Automation of Technical Patterns Using Topographic Prominence

6.2.1 The art of chart reading

Technical analysts and technical traders are also known as *chartists*. They claim to be experts in chart reading. What does chart reading mean? Is this an art or a science? According to the technical analysts we discussed this subject with, it is a mixture of both. They claim that the ability to read charts is a skill that similarly to any other type of art requires talent and learning. Our objective in this research is not to discuss the validity of this statement. We want to focus on the way the chart reading is performed. The better we understand it, the better algorithms we will be able to create for automating it and substituting the human chartists with a computer programme. When a technical analyst looks at a price chart he tries to simplify it to the extent that the chart contains only a small number of movements. This process of filtering and aggregation appears to be similar to smoothing but has some significant differences. The price fluctuations are compared by magnitude within their surrounding. If there is a bigger move in the same direction that is first close to the observed one, and second they are not separated by a big move in the opposite direction, the small fluctuation is completely ignored. The human eye simply ignores the small fluctuations and considers a flat line that connects the big ones. This phenomenon is not specific for the process of looking into price charts. It is a feature of the human vision and visual interpretation that humans have developed a long time ago. Following this line of reasoning it is natural to relate the way technical analysts look at charts to the way people look at a mountain range. The small peaks that are on the same ridge as a bigger one are easily ignored and even completely unnoticed.

6.2.2 The topographic notion of prominence

In topography the notion that is used for measuring the elevation of a mountain summit relative to their surrounding is called *prominence*. The topographical meaning and the mathematical theory behind it is explained in Maizlish [81]. There are different flavours of the notion defined by some specifics around the way it is measured but for our purposes we need the most general definition. Following from Maizlish [81] and other sources, we define prominence of a summit as the elevation from the highest saddle that connects this summit to a higher one. The latter is called a parent peak. The prominence is measured by identifying first that parent peak. This is done by comparing the vertical distance that need to be descended from the initial peak before starting to climb each of the higher peaks. The higher peak that requires the least descend from the initial peak becomes the parent peak. The intuition behind prominence is that it is a measure of the significance of a summit. A very high summit can be easily ignored if there is higher summit just next to it on the same ridge. On the other hand a summit would attract attention and be considered significant if it is surrounded only by lower summits, to a great extent independent of its own height.

6.2.3 The concept of prominence applied to price charts

Justification

The way the technical analysts and traders look at the price charts when they are trying to define 'highs' and 'lows', and 'support' and 'resistance' levels is very similar, if not identical to the above described concept of prominence. The art of chart reading that has been described in section 6.2.1 can be considered as a simple filtering that is based on the prominence of the moves. The price moves are disregarded unless they have a prominence above certain level that is considered significant.

Positive and negative prominence

The obvious difference between a price chart and a topographic terrain is that the price chart can be looked from either positive or negative side. On a mountain range the peaks are always the objects with the higher elevation and the saddles are the locations with the lower elevations connecting the peaks. On a price chart when the security price is in an uptrend the up moves are the 'peaks' and the down moves are the 'saddles'. Once the overall trend reverses the peaks and saddles swap places. Our solution deals with this confusion by introducing the notions of positive and negative prominence. Positive prominence is a simplified two-dimensional version of the topographical notion that has been introduced in section 6.2.2. Negative prominence is defined as a positive prominence on the reflection of the price chart on the horizontal axis. Using the combination of the negative and positive prominence we can associate every price move on the chart with a prominence value that measures the relative significance of the move.

Filtration of information

Another important requirement for the validity of a technique that processes the price stream for the purposes of automation is to preserve the way information is filtered through the system. In other words any stopping times (see Grimmett and Stirzaker [82]) with respect to the original information retrieval should remain stopping times after the price data is processed. The intuition behind this requirement is that the pattern detection algorithm should not identify patterns based on price data that would not have been available to a technical trader at that time. The smoothing algorithms that we reviewed in section 6.1.1 do not take special care for fulfilling this requirement. It is usually intrinsic for a smoothing algorithm to use data from around the current point and as a result of that to end up using information that would not have been available. The prominence approach can be very efficiently managed to deal with this requirement. At any time that is considered as current, the algorithm looks for the parent peak only in the past. Once again this imitates closely the way a technical analyst would operate.

Outliers

Very big movements in a data series are usually considered as outliers and the most of the smoothing algorithms are designed to deal with them by smoothing the points and removing their effect on the surrounding ones. When identifying technical patterns this is not the desired solution. Very often a technical pattern is defined by the big movements that cross certain previously defined levels. Our approach allows these, so-called outliers, to serve the same function as they would do by a technical analyst. This and the above points are demonstrated in figure 6.1.

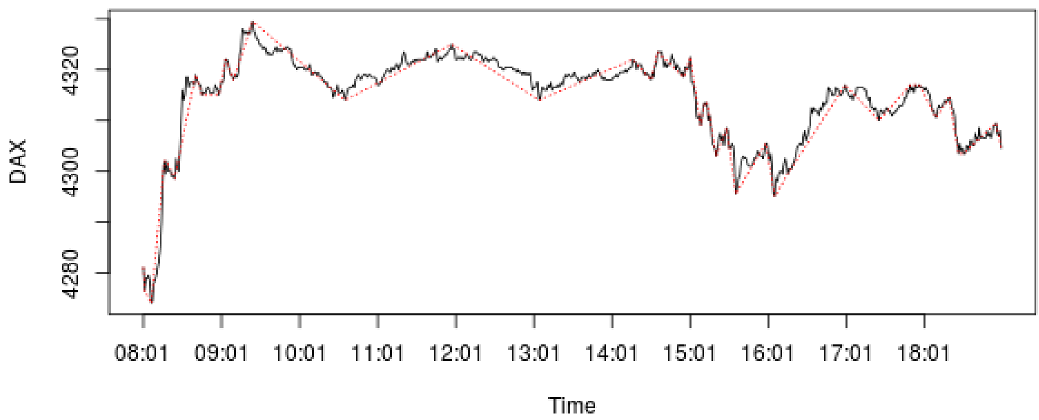


Figure 6.1: *Prominence based linearization of minutely data of the DAX futures on 03/01/2005.*

Comparison with other linearization methods

In chapter 2, section 2.4, along with the methods based on smoothing, we reviewed the approach of Osler and Chang [21] for automating patterns by identifying the significant peaks and troughs and then connecting them with straight lines. The method that they use relies only on the preceding trough or peak to measure the height of the current peak or respectively trough. As we explained in the literature review chapter 2, section 2.4 this method differs from the real technical analysis approach because the latter creates the measure by taking much broader surrounding of the peak or trough and decides their height or respectively depth based on the whole observed period. Figure 6.2

displays an example of this. The red circled point on the figure is preceded by a deep trough and an automating algorithm such as the one Osler and Chang [21] use would consider it more significant than the one in the blue circle. On the other hand, once the blue circled point is registered a technical analysts would ignore the red one and from that point on consider only the blue one, since it creates a better measure of the local maximum that the price achieved at that moment. The prominence based measure works in a very similar way to the human trader, in contrast to the above mentioned automated algorithm. The red point prominence is very small since it is 'parented' by the blue peak.

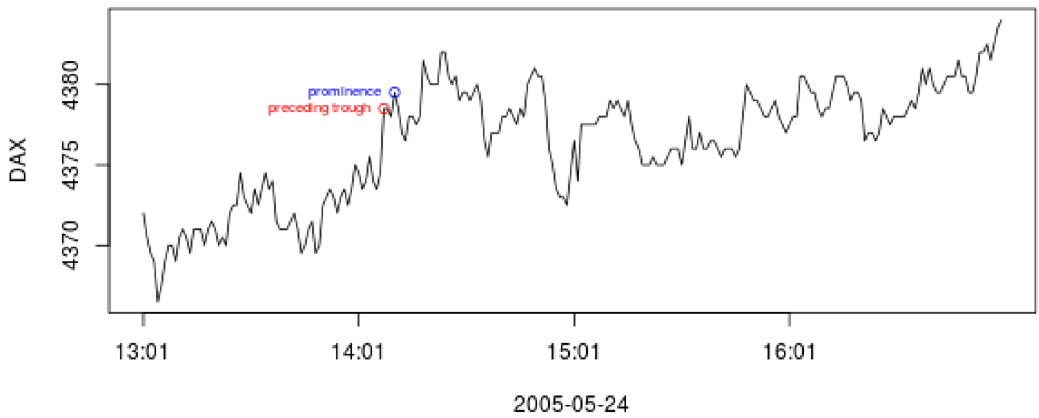


Figure 6.2: *The price of the DAX futures in the afternoon of 24/05/2005 creates the two small peaks that are circled red and blue. The first one is preceded by a deeper trough and an algorithm using only the preceding trough value would assign a higher significance to it. The blue circled peak has a higher prominence and is a better reflection of the local maximum achieved during that period. The blue circled peak would be the one selected by technical analysis and by the prominence based algorithm.*

Technical patterns

The automation criterion divides the commonly used technical patterns into two major categories, 'easy' and 'challenging'. The former are easy to implement, since they are based on pure numerical parameters. These include moving averages and range break-outs. The automation of these patterns is straight forward. The automation algorithm implements the pattern logic with specific values for the parameters. The latter are patterns that rely on being spotted by the technical analysts. These are all the patterns that use 'tops', 'bottoms', 'support', 'resistance' and similar types of technical notions. An example for these patterns are: double tops/bottoms, head-and-shoulder, trends and trend reversals. The prominence based linearization method is intended for the 'challenging' group of patterns. Using the positive and the negative prominence separately the method can detect the price levels that would be considered as support and resistance values by a technical analyst. Using the combined linearization the prominence based method can feed an algorithm for spotting more complicated patterns such as ‘head-and-shoulders’ and multiple top/bottoms.

6.2.4 Implementation considerations

The implementation that we propose follows closely the logical steps of the method. In order to compute all prominences we run the function ones on the price series for the positive ones and then on the negated prices to compute the negative prominences. The implementation contains the following steps:

1. Identify all peaks and troughs along the series. The first point of a flat part is considered peak or trough, depending on its location, and the rest are ignored;
2. For each peak find the peaks that are higher, these are the potential parents;
3. Find the two higher peaks that are in the closest proximity
4. Compare the deepest troughs on the way to each one of the two higher peaks to identify the parent peak and the prominence.
5. Apply a high pass filter to the prominence values.

The last point above is not part of the prominence algorithm. We use the prominence value to filter the moves in order to leave only the significant ones based on their prominence. The threshold values for the high pass filter can be set following different approaches. Two obvious choices are to select only certain top percentile of the moves, for example the top 10%, or to select only the prominences that are bigger than a predefined fraction of the total move on that day. We are in favour of the second approach since we consider it to be closer to the way a technical analyst operates. By taking the total price move on the day we define a scaling factor. When technical analysts investigate a price series for patterns they only consider formations that stand out on the scale of the chart. In an intraday case the analysts would be looking at a daily chart and the scale would be defined by the distance between the lowest and the highest values observed on that day. In algorithm 1 we present a pseudo-code that follows the above steps (see Appendix B for an actual implementation of the algorithm in R):

6.2.5 Technical patterns automation

In this research we use the above developed methodology to automatically identify two of the most famous price patterns in technical analysis: Head-and-shoulders and support and resistance based trends. We run the detection algorithm on a broad range of securities over a long period of time. The first step is to extend the price time series with the prominence values and then use these values to identify the patterns.

Data

The datasets that we use for the analysis consist of foreign exchange pairs and exchanged traded futures. Table 6.1 displays the securities and the time period over which our price history extends.

Algorithm 1 Price prominence

$M \leftarrow max(price)$
$m \leftarrow min(price)$
$maxProminence \leftarrow M - m$
$peaks \leftarrow findPeaks(price)$
$troughs \leftarrow findTroughs(price)$
$prominence[maxPosPeak] \leftarrow maxProminence$
$prominence[maxNegPeak] \leftarrow -maxProminence$
for each $peak \in peaks$ **do**
  $parents \leftarrow findParents(peak)$
  $saddle \leftarrow max(minTrough(peak, parents[1]), minTrough(peak, parents[2]))$
  $prominence[peak] \leftarrow price[peak] - price[saddle]$
end for
for each $trough \in troughs$ **do**
  $parents \leftarrow findParents(trough)$
  $saddle \leftarrow min(maxPeak(trough, parents[1]), maxPeak(trough, parents[2]))$
  $prominence[trough] \leftarrow price[trough] - price[saddle]$
end for

The foreign exchange data have been downloaded from the online provider Dukascopy[1] and the futures from the data collection and distribution company TICK DATA[2]. The frequency of the futures data is minutely. Each record consists of open, high, low and close price values. In addition to that the records hold the traded volume and the number of up and downward price moves. The foreign exchange data consists of open, high, low and close values of the current best bid and offer prices. The price series for every instrument has been converted into R timeSeries object. Every time series is then split into lists of daily data for which we calculate the prominence and add the prominence values to the time series. These lists of time series are then used as inputs for the algorithms that identify the intraday technical patterns.

Table 6.1: Analyzed datasets

Type	Symbol	Security	Period
FX	EURUSD	EURUSD	Jan 2010 - Aug 2015
FX	GBPUSD	GBPUSD	Jan 2010 - Aug 2015
FX	USDJPY	USDJPY	Jan 2010 - Aug 2015
Fut	AD	Australian Dollar Futures	Jan 2005 - Oct 2013
Fut	BL	Euro-Bobl 5-Year Futures	Jan 2005 - Oct 2013
Fut	BN	Euro-Bund 10-Year Futures	Jan 2005 - Oct 2013
Fut	BP	British Pound	Jan 2005 - Oct 2013

[1] see www.dukascopy.com
[2] see www.tickdata.com

Table 6.1 cont.			
Type	**Symbol**	**Security**	**Period**
Fut	CD	Canadian Dollar	Jan 2005 - Oct 2013
Fut	DA	DAX Index	Jan 2005 - Oct 2013
Fut	DJ	DJIA ($10) Futures	Jan 2005 - Oct 2013
Fut	EC	Euro FX	Jan 2005 - Oct 2013
Fut	ES	S&P 500 E-Mini Futures	Jan 2005 - Oct 2013
Fut	FT	FTSE 100	Jan 2005 - Oct 2013
Fut	FV	US 5-Year T-Note Futures	Jan 2005 - Oct 2013
Fut	GL	Long Gilt Futures	Jan 2005 - Oct 2013
Fut	GO	Low Sulphur Gasoil Futures ICE	Jan 2010 - Oct 2013
Fut	JB	Japanese 10-Year Bond	Jan 2005 - Oct 2013
Fut	JE	Japanese Yen E-Mini Futures	Jan 2006 - Oct 2013
Fut	JY	Japanese Yen Futures	Jan 2005 - Oct 2013
Fut	ND	NASDAQ 100 Index Futures	Jan 2005 - Oct 2013
Fut	NE	Nikkei 225 Futures OSE	Feb 2008 - Oct 2013
Fut	NG	Natural Gas Futures NYMEX	Jan 2005 - Oct 2013
Fut	NK	Nikkei 225 Futures CME	Jan 2005 - Oct 2013
Fut	NQ	NASDAQ 100 E-Mini Futures	Jan 2005 - Oct 2013
Fut	NZ	New Zealand Dollar Futures	Jan 2010 - Oct 2013
Fut	PT	S&P Canada 60 Futures	Jan 2005 - Oct 2013
Fut	QG	Natural Gas E-Mini Futures	Jan 2005 - Oct 2013
Fut	QM	Crude Oil E-Mini Futures NYMEX	Jan 2005 - Oct 2013
Fut	SF	Swiss Franc Futures	Jan 2005 - Oct 2013
Fut	SP	S&P 500 Futures	Jan 2005 - Oct 2013
Fut	TP	TOPIX Futures OSE	Jan 2005 - Oct 2013
Fut	TU	US 2-Year T-Note Futures	Jan 2005 - Oct 2013
Fut	TY	US 10-Year T-Note Futures	Jan 2005 - Oct 2013
Fut	US	US 30-Year T-Bond Futures	Jan 2005 - Oct 2013
Fut	VH	STOXX Europe 50	Jan 2005 - Oct 2013
Fut	WT	WTI Light Sweet Crude Oil Futures ICE	Jan 2010 - Oct 2013
Fut	XB	RBOB Gasoline Futures NYMEX	Oct 2006 - Oct 2013
Fut	XX	EURO STOXX 50	Jan 2005 - Oct 2013

Table 6.1 cont.			
Type	**Symbol**	**Security**	**Period**
Fut	YM	Dow Jones ($5) E-mini Futures	Jan 2005 - Oct 2013

In the following sections we describe the algorithms for automated detection of the analyzed technical patterns using the price prominence.

Head-and-shoulders

The implementation steps of the intraday Head-and-shoulders pattern on a price series with computed prominence are as follows:

1. Select the peaks with prominence greater than a predefined minimum level.
2. From the filtered set of peaks identify all sequences of three consecutive peaks that could form a pattern, i.e. the middle peak is higher than the two other peaks.
3. Exclude the triplets that are not followed by a lower peak.
4. Exclude the horizontally asymmetric groups (see Osler and Chang [21]), these are the ones where the times between the two consecutive peaks differ by a factor of 2.5 or more.
5. Identify the neckpoints. These are the troughs between the shoulders and the head. Exclude the sets where a neckpoint is higher than a shoulder.
6. Identify the neckline and compute its slope.
7. Filter for patterns where both shoulders are above the neckline. This requirement combines vertical and horizontal symmetry of the pattern. An example of the neckline above the right shoulder is when the neck line is steeper than the slope of the right shoulder. This can occur when the right neckpoint is much higher than the left one creating a steep neckline and at the same time the right shoulder takes longer time to form ending up with a relatively flat slope.
8. Locate the peak following the pattern, independent of its prominence, and filter for the cases where the neckline is above that peak.

The above algorithm is applied independently for the positive and the negative (reverse) Head-and-shoulders pattern using the price positive and negative prominences respectively. The numbers of detected patterns per security are summarized in table 6.2. The Head-and-Shoulders is a relatively low frequency pattern. It very rarely appears more than once a day and for some of the securities it featured only small number of times during the analyzed price history.

In figure 6.3 we plot four examples of days when our automated algorithm detected the Head-and-Shoulders pattern. Each plot displays the minutely prices during that day with a black line. On the price line we plot the pattern by drawing a solid red line that traverses from the beginning of the left shoulder to the end of the right shoulder. The first and third examples demonstrate reverse patterns and the other two demonstrate normal patterns. The neckline of the pattern is plotted with a dotted red line. It is extended to the time of the peak following the right shoulder to demonstrate that the last step of the above requirements is satisfied.

Figure 6.4 presents an interesting case where the right shoulder of the pattern is succeeded by a more prominent peak. It appears that the algorithm missed the actual pattern triggering too early. That impression is created when we have the opportunity to look at the price series ex-post. In reality a technical trader would have been able to watch the prices as they have been arriving. As soon as the right shoulder of the pattern is formed, the trader would have acted, without knowing or being able to anticipate what appears to be a better alternative of the pattern.

Table 6.2 summarizes the number of occurrences of the pattern for each security during the analyzed periods. We count the upward and downward pointing patterns separately. The former triggers a sell signal and the latter a buy signal.

Table 6.2: Number of occurrences of Head-and-Shoulder patterns per security

Head-and-Shoulder Patterns		
Security	**Buy Signals**	**Sell Signals**
AD	37	31
BL	16	12
BN	23	15
BP	14	14
CD	10	10
DA	162	159
DJ	38	39
EC	6	10
ES	71	83
EURUSD	0	1
FT	159	149
FV	8	9
GL	12	11
GO	4	1
JB	3	6
JY	12	21
ND	54	35
NG	6	3
NK	27	28

Table 6.2 cont.		
Security	**Buy Trades**	**Sell Trades**
NQ	45	51
NZ	1	0
PT	51	45
QG	3	1
QM	15	14
SF	9	10
SP	91	60
TP	12	13
TU	11	7
TY	11	6
USDJPY	2	1
US	14	9
VH	26	14
WT	1	2
XB	10	0
XX	67	50
YM	59	56

The target of our analysis is not to assess the profitability of the technical patterns but to identify how actively the patterns are traded. The profit and loss profile of the trades that are triggered by the pattern provides a mean of justification of the pattern usage. A technical pattern that conceptually or by implementation is guaranteed to generate a series of losing trades does not need to be tested any further, since it is obvious that nobody would base his strategy on such a pattern. In figure 6.5 we illustrate the average post-trade profile aggregated across time and analyzed securities. The total cumulative average return for the 90 minutes following the signal trigger is represented on the chart by a thick solid red line. The return profiles of all individual securities are drawn with grey lines in the background. The return profile, depicted by the red line, appears to be appealingly positive. The return increases steadily during the immediate post-trade period. This picture provides a satisfactory justification for our analysis but it is not a proof that this is a profitable technical trading strategy. The profitability of a trading strategy based on the intraday Head-and-Shoulders pattern depends on many other factors in addition to the entry trigger. These include among others execution costs and slippage, holding period and risk management. By looking at the post-trade return we demonstrate that the idea and implementation of the pattern do not guarantee immediate losses.

Support and resistance levels

The intraday price trend based on technical support and resistance levels is relatively easy to identify using the price prominence. The detection algorithm mimics the process of how a human trader identifies the pattern. After defining a prominence threshold the pattern automation algorithm tests all possible triplets of peaks with significant prominence for creating a trend line. The potential candidates must meet the following conditions:

- the slopes of the lines connecting the points must not differ more than 10%;
- the trend line must enclose the levels that are between the points defining the line.

A summary of the number of intraday trend lines detected for each security is presented in table 6.3. As expected the signals are evenly distributed between buy and sell, i.e. positive and negative trends respectively. The post-trade analysis depicted in figure 6.6 shows that the pattern algorithm is 'well behaved'. On average the post-trade return is slightly upward during the first few minutes and then flat for the rest of the period. The return profiles vary less across different securities compared to the Head-and-Shoulders results.

Table 6.3: Number of trading signals based on trend defined by technical support and resistance levels per security

Trend Signals		
Security	**Buy Signals**	**Sell Signals**
AD	869	940
BL	238	284
BN	447	566
BP	1008	1122
CD	681	788
DA	887	917
DJ	84	106
EC	1039	1200
ES	719	842
EURUSD	1113	1144
FT	656	744
FV	268	434
GBPUSD	1065	1119
GL	295	357
GO	205	244
JB	67	79
JE	33	39

Table 6.3 cont.		
Security	**Buy Trades**	**Sell Trades**
JY	933	1099
ND	62	103
NE	37	56
NG	590	691
NK	164	216
NQ	842	929
NZ	256	281
PT	251	273
QG	82	131
QM	549	653
SF	747	860
SP	576	632
TP	29	69
TU	55	107
TY	310	517
USDJPY	1126	1131
US	307	498
VH	14	16
WT	389	415
XB	498	537
XX	398	569
YM	934	987

The next two patterns are not based on the price prominence since they belong to types that are fully quantifiable. They have been included in this experiment in order to extend the results of chapter 4 with the securities added to this research.

Range Breaks

The Range Breaks patterns can appear only once per trading day. After the occurrence of the signal the algorithm is deactivated until the beginning of the following day. Thus table 6.4 shows the number of the observed patterns per security that correspond to the number of days the pattern has been observed. The Range Breaks is a family of technical patterns that trigger trading signals during the day but the positions that result from trading these signals are usually held until the end of the current or the beginning of the following trading day. In figure 6.7 we plot the post-trade return of the signals based on the pattern but following from the above stated observations the immediate post-trade returns are not of such a great importance as they are for patterns that are actively traded multiple times within a trading day. A better view of the strategy is received by looking at the overall performance during the analyzed period. We assume that all instruments are distributed equal amount of capital, the trades are open on the signal trigger and closed at a pre-specified time at the end of the day. The cumulative return of such a portfolio is depicted in figure 6.8. The demonstrated positive slope of the cumulative return justifies the existence of the trading strategy. The total return numbers are deceiving since the strategy is presented before transaction cost, assuming perfect execution and without any risk management. In other words, the presented strategy is not a recipe for printing money but it is appealing enough to attract the attention of technical analysts and traders. The flat part on the right hand side of the chart is due to the fact that we do not have data for the most of the analyzed securities for that period.

Table 6.4: Number of occurrences of Range breaks patterns per security

Range Breaks Patterns		
Security	**Buy Signals**	**Sell Signals**
AD	135	169
BL	3	1
BN	12	17
BP	86	114
CD	134	187
DA	522	637
DJ	315	412
EC	122	166
ES	360	492
EURUSD	70	92
FT	273	381
FV	1	3
GBPUSD	35	40
GL	0	1
GO	301	316
JB	5	1
JE	135	130
JY	163	166
ND	554	658
NE	41	64
NG	814	1153
NK	385	482

Table 6.4 cont.		
Security	**Buy Trades**	**Sell Trades**
NQ	523	676
NZ	64	83
PT	324	503
QG	850	1137
QM	1008	1191
SF	180	218
SP	367	482
TP	15	16
TY	11	12
USDJPY	34	62
US	109	147
VH	329	415
WT	391	458
XB	710	794
XX	441	585
YM	299	442

Moving Averages

The moving average technical signals are the highest frequency signals that we analyze in this research. The pattern is defined by two parameters These are the long and the short period over which the moving averages are calculated. As in chapter 4 we investigate two flavours of this strategy: long period of 30 minutes and long period of 120 minutes. In table 6.5 we summarize the strategy signals appearing on average per trading day. The 30 minutes strategy produces on average twice as many signals. The nature of the basic moving average strategies, a type of which we investigate here, is that the buy and sell signals alternate. This is caused by the fact that the strategy triggers a signal when the short term moving average crosses the long term one. This always happens in a direction opposite to the previous one. Thus there is no need to create separate statistics for the buy and sell signals.

Table 6.5: Number of occurrences of moving average signals per day for each security

Moving Average Signals		
Security	**30 Minutes MA**	**120 Minutes MA**
AD	72	33

Table 6.5 cont.		
Security	**30 Minutes MA**	**120 Minutes MA**
BL	89	44
BN	89	43
BP	78	35
CD	74	33
DA	87	42
DJ	23	13
EC	83	38
ES	88	40
EURUSD	69	31
FT	72	34
FV	80	35
GBPUSD	69	31
GL	60	28
GO	72	32
JB	18	13
JE	10	7
JY	83	38
ND	23	11
NE	46	22
NG	60	25
NK	45	21
NQ	76	35
NZ	57	25
PT	39	18
QG	30	11
QM	62	26
SF	73	33
SP	55	25
TP	27	15
TU	79	39
TY	93	42
USDJPY	71	33
US	85	38

Table 6.5 cont.		
Security	**30 Minutes MA**	**120 Minutes MA**
VH	17	9
WT	70	30
XB	51	21
XX	96	47
YM	80	36

The high frequency of the intraday moving average signals defines the relatively shorter holding period of a trading strategy based on these signals. In figures 6.9 and 6.10 we plot the 30 minutes post-trade returns of the 30 minutes and 120 minutes moving average signals. Most of the signals would be overridden by the following opposite signal long before the plotted 30 minutes expires. On average the post-trade returns are slightly positive. We use a logarithmic scale for the y axis to present a better view on the different securities.

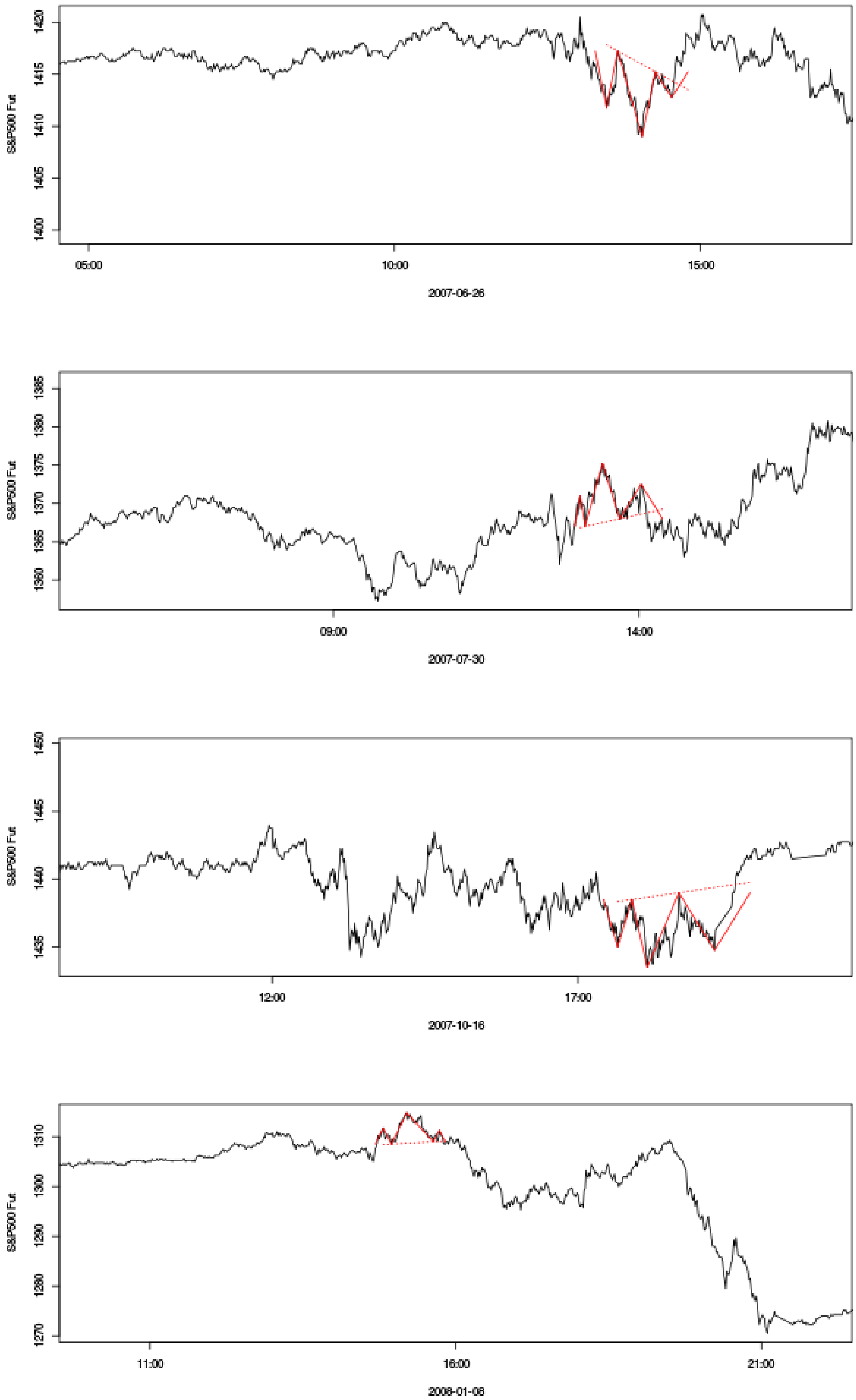


Figure 6.3: *Examples of Head-and-Shoulders intraday patterns on the S&P500 futures. The solid red line traverses from the beginning of the left shoulder to the end of the right shoulder. The neckline is plotted with a dotted red line.*

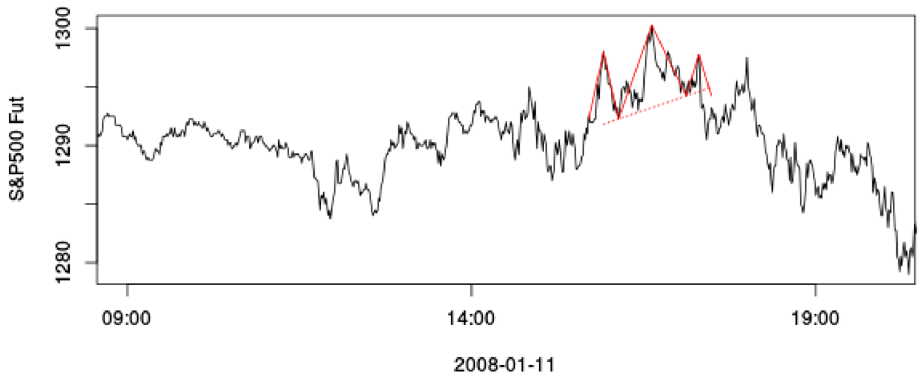


Figure 6.4: *An example of a Head-and-Shoulders pattern where the right shoulder of the pattern is succeeded by a more prominent one.*

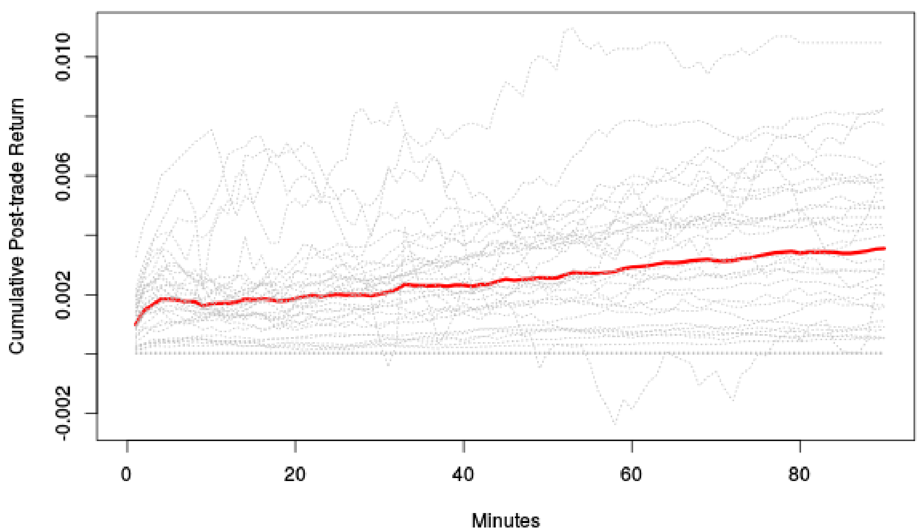


Figure 6.5: *Average cumulative return from 1 to 90 minutes after the completion of a Head-and-Shoulders pattern. The grey lines in the background are the average post-trade returns of each of the analyzed securities. The solid red line is averaged over pattern occurrences and across instruments.*

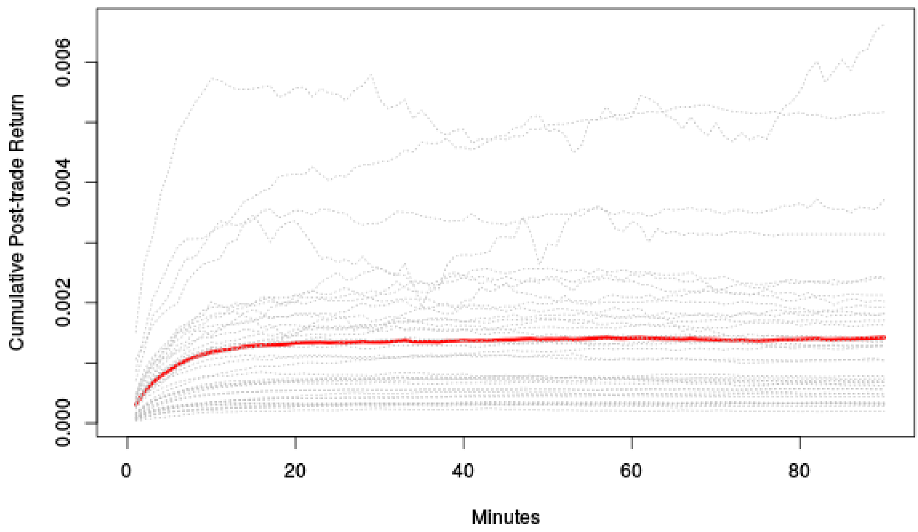


Figure 6.6: *Average cumulative return from 1 to 90 minutes after the formation of a technical trend pattern based on support and resistance levels. The post-trade return of each security is represented by a grey line in the background. The solid red line is averaged over pattern occurrences and across instruments.*

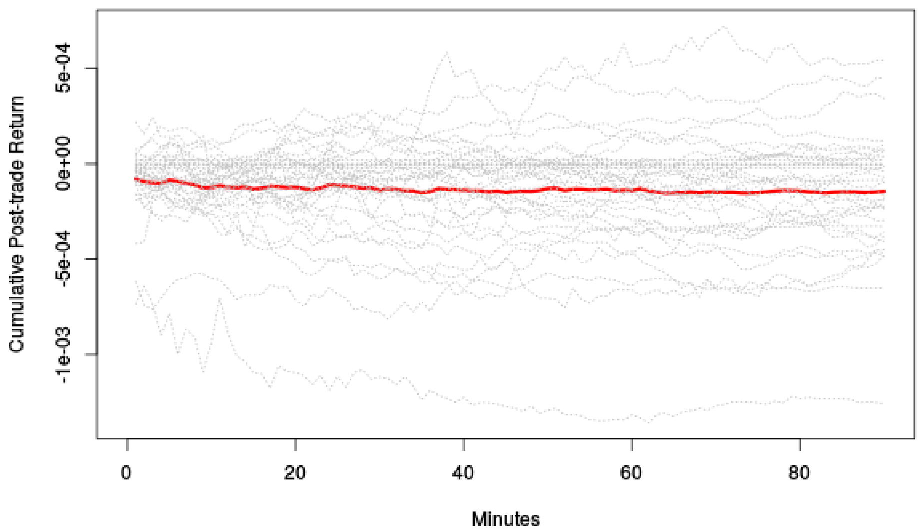


Figure 6.7: *Average cumulative return from 1 to 90 minutes following the occurrence of a Range Breaks pattern. The grey lines in the background are the average post-trade returns of each of the analyzed securities. The solid red line is averaged over pattern occurrences and across instruments.*

Figure 6.8: *Total cumulative return across all analyzed securities of the Range Breaks pattern along the analyzed time period.*

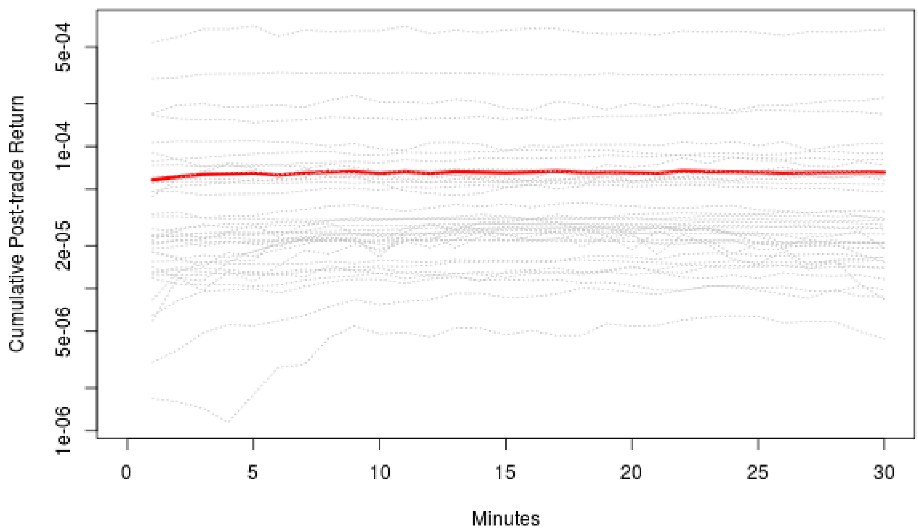


Figure 6.9: *Average cumulative return from 1 to 30 minutes following the occurrence of a 30 minutes moving average signal. The grey lines in the background are the average post-trade returns of each of the analyzed securities. The solid red line is averaged over pattern occurrences and across instruments. The y axis is logarithmic.*

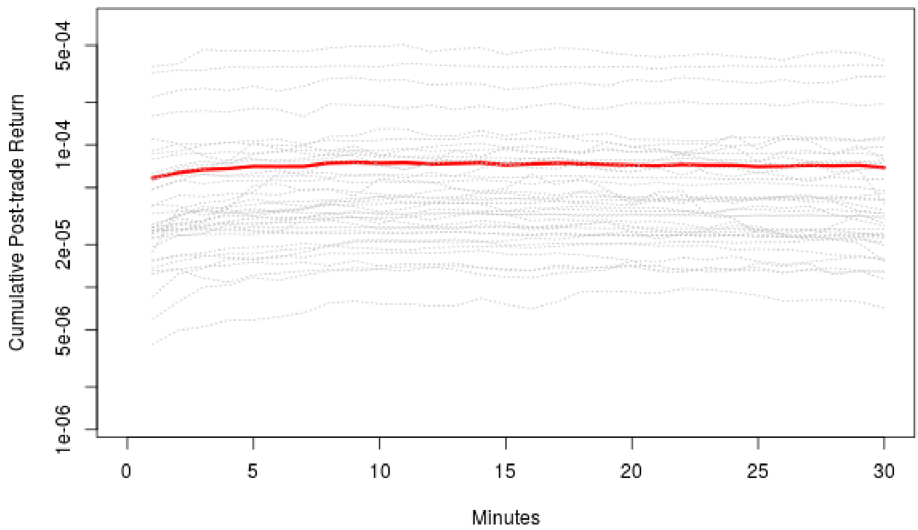


Figure 6.10: *Average cumulative return from 1 to 30 minutes following the occurrence of a 120 minutes moving average signal. The grey lines in the background are the average post-trade returns of each of the analyzed securities. The solid red line is averaged over pattern occurrences and across instruments. The y axis is logarithmic.*

6.3 Technical Trading Detection Results

In this section we present the results of the methodology that we developed and described in chapter 4 for detection and tracking of technical trading. We tested all the securities for which we implemented the automated pattern recognition algorithms described in the previous section 6.2.5. We present the result for each of the four technical patterns that we tested. As previously explained (see chapter 4) the test measures the trading intensity λ of potential technical traders assumed to be trading actively at or around the occurrence of the pattern. The outcomes of the *active* periods (the periods immediately after the pattern has been observed) are compared with periods with similar price and market environment without the presence of the technical pattern. Each of these periods is a randomly selected one hour interval on the day when the pattern occurred. The end of the interval is no later than five minutes before the occurrence of the pattern. We called these periods *passive*. Following from that the results of each of the tests consist of maximum likelihood estimations for the trading intensities λ. These are derived from the return impact series X produced by the particle filtering implementation (see chapter 4, section 4.3.4). For each of the patterns we present the results in the following format:

- A chart of active and passive trading intensity estimates for each security. We plot the maximum likelihood estimates for the trading intensities λ for the passive and active return sets.
- A chart of the distributions of the return impacts. This is a summary of the technical trading contribution to the total security return. The idea is to compare visually the distributions and identify the ones where the active periods are significantly different compared to the passive periods.
- A measure of the distance between the active and passive estimates. We compute the Kullback-Leibler divergence between each pair of gamma distributions implied by the technical trading detection results. The KL divergence of a gamma distribution with a shape λ_a and a scale parameter β from another gamma distribution with the same scale and a shape λ_p is given by:

$$\log(\Gamma(\lambda_p)) - \log(\Gamma(\lambda_a)) - (\lambda_p - \lambda_a)\psi_{\lambda_p} \tag{6.1}$$

The derivation of the above result is described in appendix C

6.3.1 Head-and-shoulders

The technical trading detection results for the intraday Head-and-shoulders pattern based on price prominence are presented in figures 6.11 and 6.12. As introduced in the previous section 6.3, the first chart compares the estimated trading intensity levels. The estimates marked *active* are based on the periods immediately after the pattern occurred, and the ones marked *passive* are from the same days but when the pattern was not present. The second chart compares the components of the minutely return that were estimated by the detection algorithm to be a result of the pattern trading activity. The first observation that can be made from the two charts is that the analyzed securities can be divided into three groups. The first one are the securities for which both the difference between the active and passive period intensities stands out and also the centres of the return distributions are not close to each other in standardized terms. Typical examples for this group are the US and European equities indices futures, the natural gas futures, the crude oil futures and the EURUSD spot. The second group are securities that appear to have a big difference between the trading intensities in absolute terms but the centres of the distributions are much closer in units of standard deviations. An example for that is the Canadian equity index futures (PT). The third group are the securities for which on both charts the difference between the active and the passive periods appears to be insignificant. These are the Japanese bond futures and the US 5-year treasury note futures. The first group are the securities for which we can make the conclusion that the Head-and-shoulder technical pattern is actively traded.

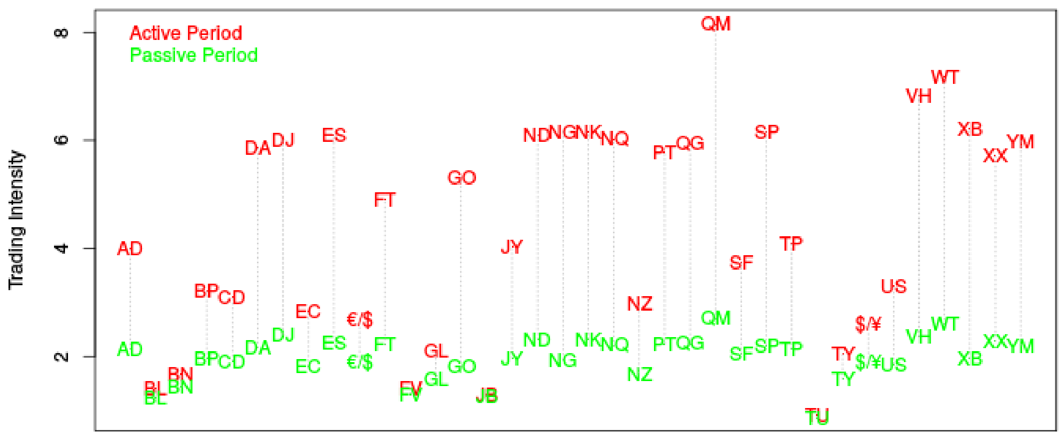


Figure 6.11: *Trading intensities estimates of the Head-and-Shoulders technical pattern*

6.3.2 Support and resistance levels

The results for detection of intraday technical trend trading are plotted in pictures 6.13 and 6.14. The only securities that appear to be actively traded are the NASDAQ 100 index futures, the natural gas and the crude oil futures. These securities demonstrate significant differences of the trading intensities between the active and passive periods as well as significant distance between the centres of the return distributions. Neither the fixed income futures nor the currencies appear to attract the technical trend traders. The European equity index futures results present an uncertain result.

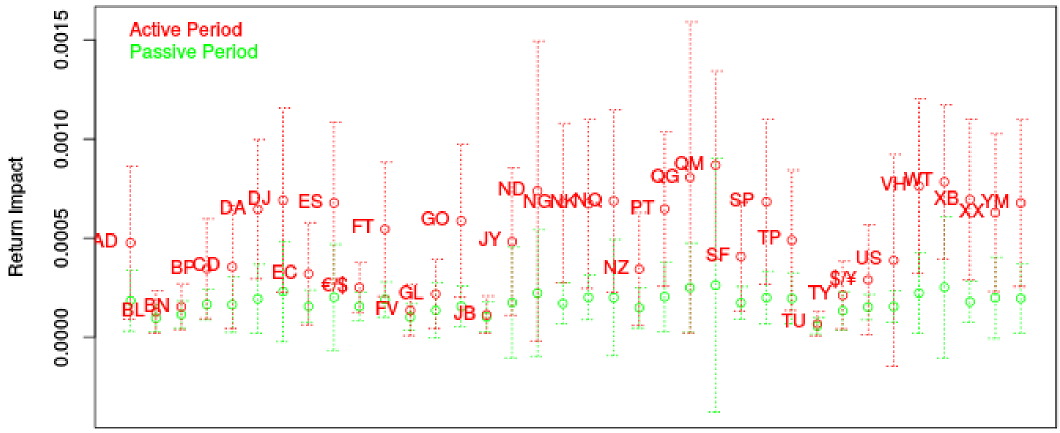


Figure 6.12: *Average return impact of Head-and-Shoulders technical trading. The bands are at plus/minus one standard deviation*

6.3.3 Range Breaks

The range breaks technical price pattern results are summarized in pictures 6.15 and 6.16. The first observation that can be made is that the scale of the trading intensities is smaller than that of the last two patterns. The results for the active trading intensities show that there are securities for which the active intensity values are relatively high and outside the range defined by the false positive analysis in chapter 4, section 4.3.5. This is valid for most of the analysed equity index futures and the energy futures. One possible explanation for these results could be the presence of active technical trading on the analysed pattern. In the case of the currency pairs, the analysis shows very low values for the trading intensities for this price pattern.

6.3.4 Moving Averages

The results for the two parametrizations of moving average strategies that we explored, are very similar (see pictures 6.17, 6.18, 6.19 and 6.20). The crude oil futures and the Japanese equity index futures, NIKKEI 225, demonstrate activity based on moving average technical strategy. The NIKKEI 225 futures appears to be more affected by the 120 minutes moving average than the 30 minutes one.

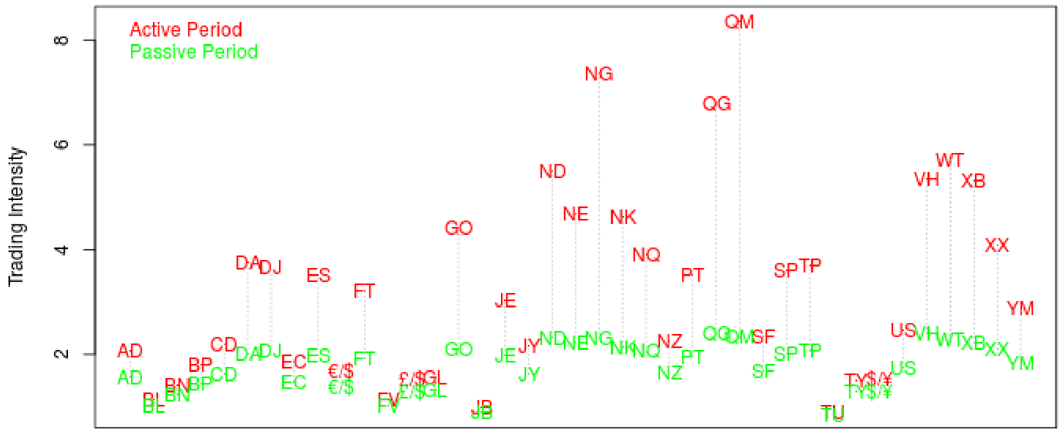


Figure 6.13: *Trading intensities estimates of signals based on trend defined by support and resistance levels*

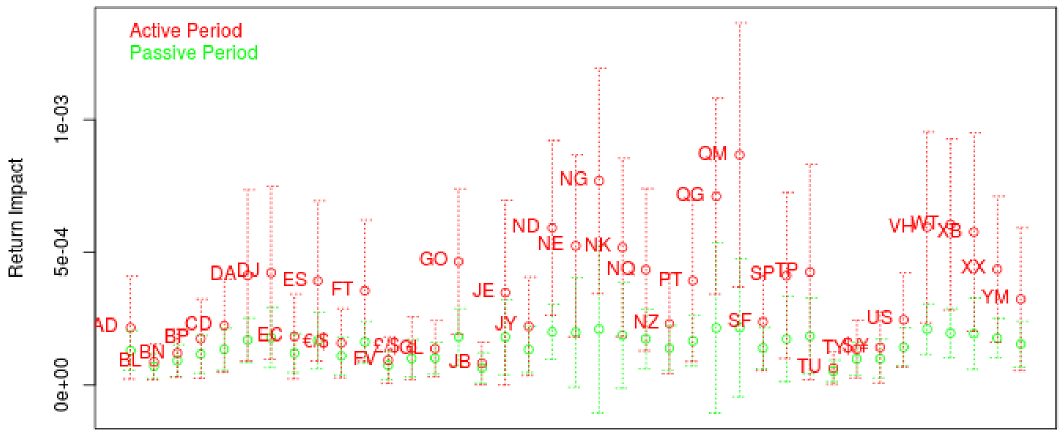


Figure 6.14: *Average return impact of technical trend trading. The bands are at plus/minus one standard deviation*

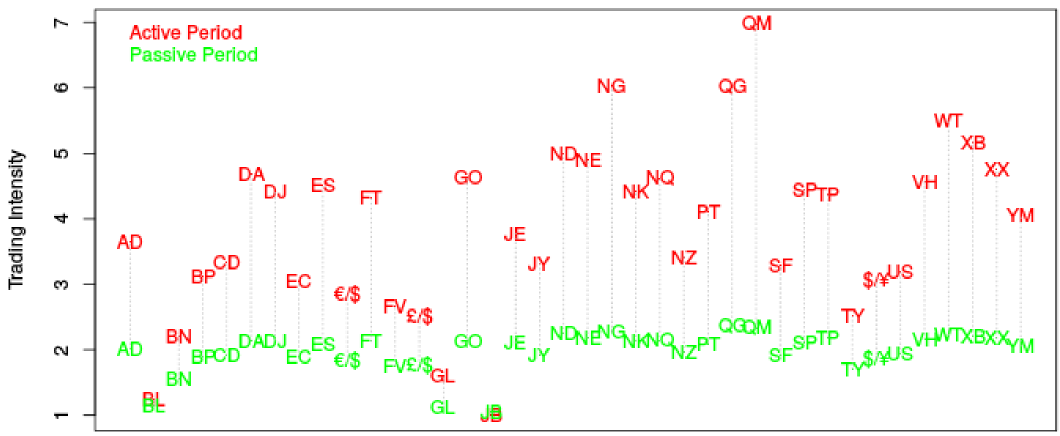


Figure 6.15: *Trading intensities estimates of the Range Breaks technical pattern*

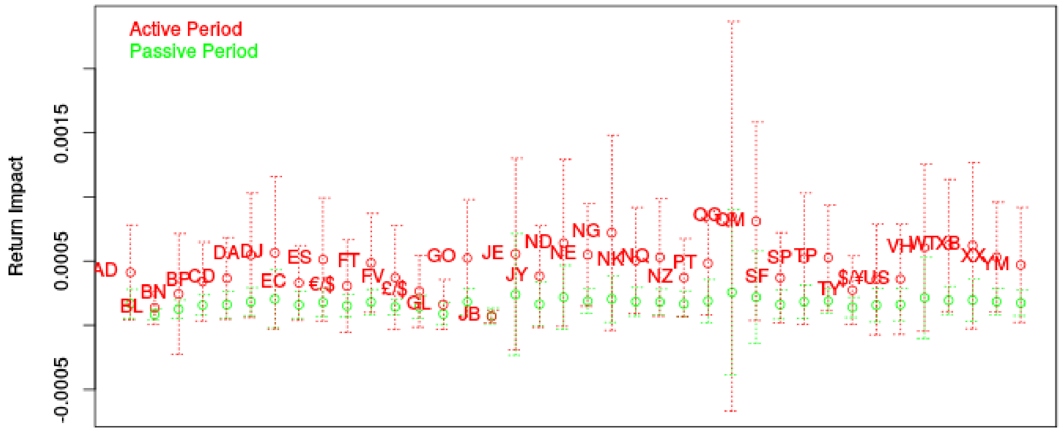


Figure 6.16: *Average return impact of Range Breaks technical trading. The bands are at plus/minus one standard deviation*

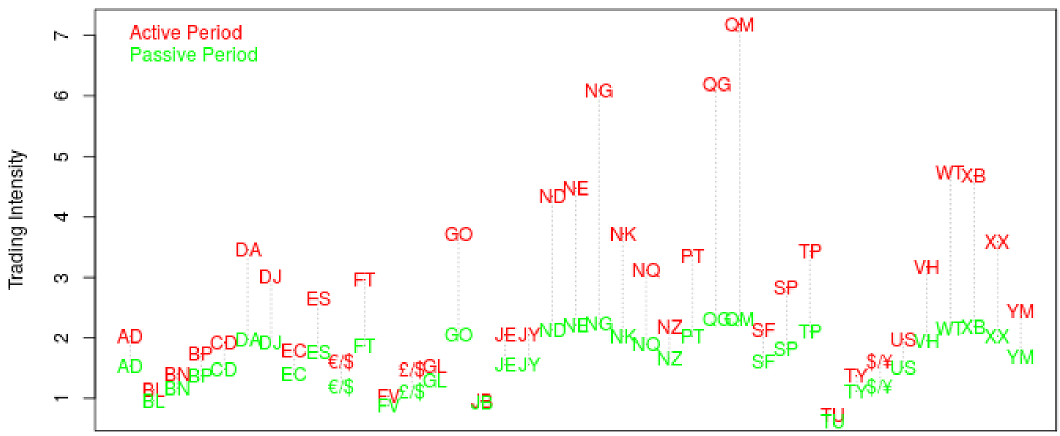


Figure 6.17: *Trading intensities estimates of 30 minutes moving average signals*

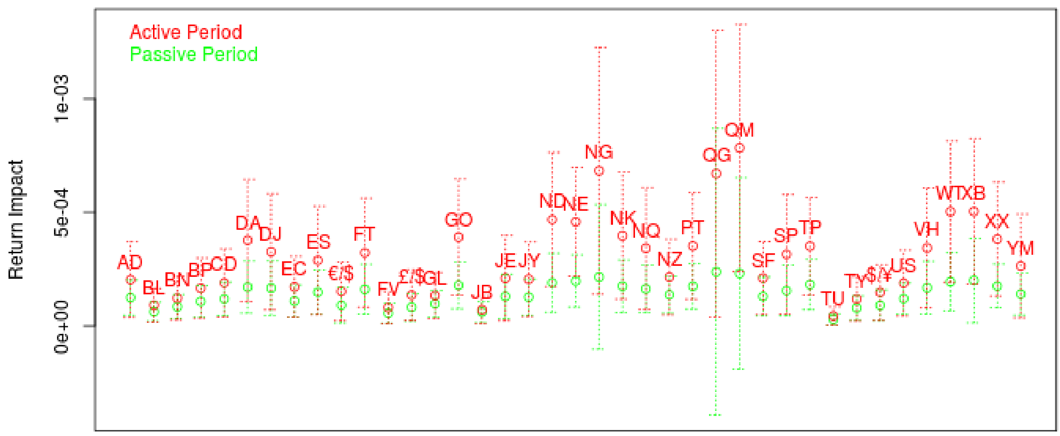


Figure 6.18: *Average return impact of 30 minutes moving average trading. The bands are at plus/minus one standard deviation*

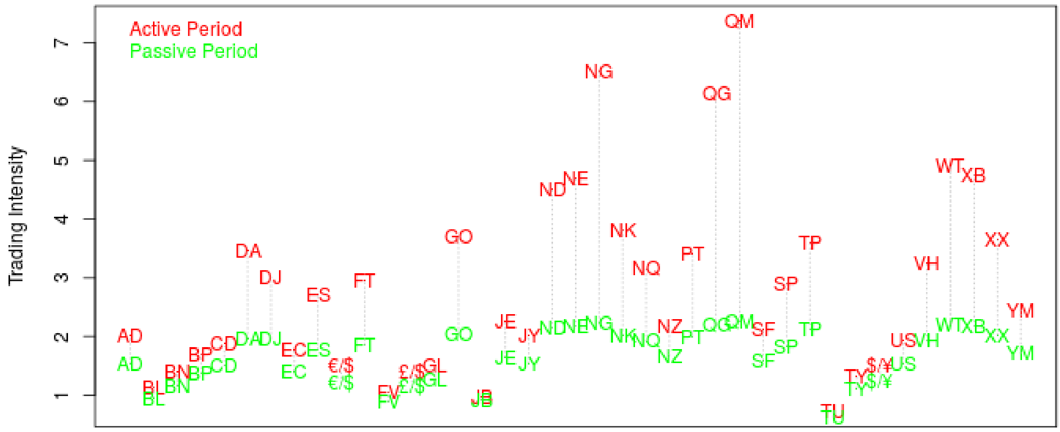


Figure 6.19: *Trading intensities estimates of 120 minutes moving average signals*

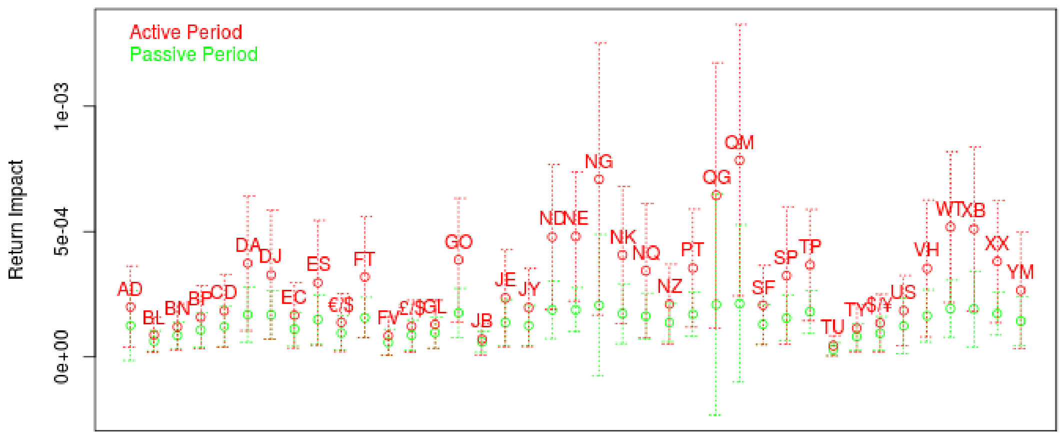


Figure 6.20: *Average return impact of 120 minutes moving average trading. The bands are at plus/minus one standard deviation*

6.3.5 Measuring the difference between Active and Passive Intensity

In this section we summarize the trading detection results using an objective measure for comparing the intensities distributions. Table 6.6 contains the Kullback-Leibler divergence results for the pairs of active and passive trading intensities for all analyzed securities. It is well known that the KL-divergence is asymmetric and the way we employ it for this analysis is to measure the divergence of the active intensity relative to the passive one that is considered the base case. As already pointed out in chapter 4, section 4.3.5, the KL-divergence is a measure of the difference between two distributions. It does not directly compare the means of the distributions. That means that a high value of KL-divergence does not guarantee active trading intensity estimate higher than the passive one. We use the KL-divergence results as an additional way to identify different security price behaviour at active periods compared to passive ones. Figure 6.21 visualizes the data from table 6.6 plotting the securities on the x axis and using differently shaped and coloured symbols for each technical strategy. Following from the false positive analysis in chapter 4, section 4.3.5, we define a KL-divergence significance threshold for each technical pattern. The thresholds are set at the highest values observed for each of the patterns in the false positive simulations, these can therefore be viewed as one percent significance thresholds. The values for the thresholds are 0.72 for the Head-and-shoulders, 0.54 for the Range breaks, 0.94 for the Trend and 1.32 for the moving average patterns.

Table 6.6: Kullback-Leibler divergence of active lambdas

	Technical Strategies				
Security	**H & S**	**RB**	**Trend**	**MA 30**	**MA 120**
AD	0.788	0.665	0.095	0.094	0.091
BL	0.015	0.007	0.008	0.029	0.022
BN	0.024	0.161	0.018	0.038	0.032
BP	0.417	0.408	0.055	0.066	0.05
CD	0.389	0.534	0.116	0.087	0.067
DA	2.555	1.368	0.745	0.582	0.568
DJ	2.207	1.113	0.614	0.325	0.301
EC	0.312	0.367	0.065	0.077	0.057
ES	2.608	1.271	0.59	0.254	0.267
EURUSD	0.182	0.307	0.042	0.102	0.046
FT	1.434	1.043	0.449	0.338	0.352
FV	0.005	0.264	0.01	0.026	0.024
GBPUSD	-	0.189	0.029	0.087	0.034
GL	0.098	0.137	0.026	0.029	0.029
GO	2.72	1.299	1.141	0.654	0.654

Table 6.6 cont.					
Security	**H & S**	**RB**	**Trend**	**MA 30**	**MA 120**
JB	0	0.001	0.008	0.002	0.003
JE	-	0.656	0.296	0.099	0.136
JY	1.021	0.513	0.108	0.097	0.089
ND	2.476	1.475	1.846	1.074	1.185
NE	-	1.497	1.241	1.057	1.323
NG	3.455	2.491	4.063	2.648	3.142
NK	2.548	1.126	1.292	0.696	0.775
NQ	2.666	1.31	0.817	0.408	0.413
NZ	0.528	0.549	0.126	0.094	0.089
PT	2.299	0.917	0.63	0.448	0.503
QG	2.465	2.318	3.1	2.601	2.782
QM	4.029	3.489	5.396	3.854	4.199
SF	0.68	0.501	0.147	0.109	0.108
SP	2.788	1.162	0.627	0.309	0.334
TP	0.85	1.016	0.613	0.435	0.523
TU	0.003	-	0.004	0.016	0.016
TY	0.085	0.213	0.028	0.043	0.029
USDJPY	0.157	0.401	0.031	0.093	0.043
US	0.58	0.428	0.167	0.081	0.063
VH	3.229	1.217	1.577	0.407	0.48
WT	3.127	2.017	2.149	1.347	1.468
XB	3.484	1.707	1.861	1.267	1.368
XX	2.15	1.374	0.897	0.605	0.637
YM	2.604	0.934	0.329	0.198	0.186

6.3.6 Summary of the Results

Using the thresholds defined in the previous section, one can easily identify that the Head-and-shoulders pattern produced the most significant results across all analyzed securities. The natural gas and the oil futures stand out as the securities that attract the most interest with the technical traders. The range breaks pattern values are significant for most of the equity indexes and the oil futures while the trend pattern is restricted within the oil futures. The intraday moving average strategies demonstrate the lowest levels of activity.

One interesting observation is that the spot currencies appear on the lowest side of the trading activity. They could not break above the threshold for none of the analysed patterns. Given that

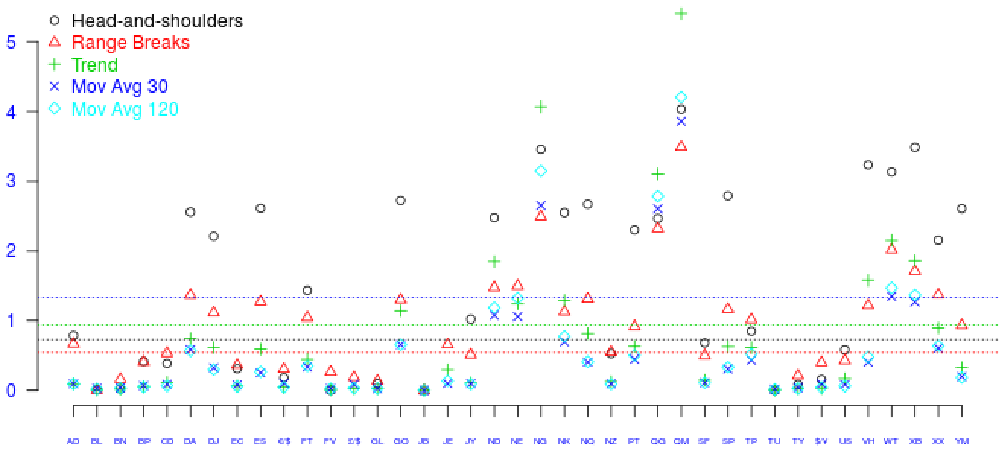


Figure 6.21: *Kullback-Leibler divergence of active trading intensities for each security and technical strategy. The horizontal lines plot the thresholds derived from the false positive analysis. The line colours correspond to the technical patterns in the chart legend. The blue line is used for both moving average patterns. Divergence values above the thresholds are considered significant.*

we selected three of the most liquid currency pairs this result does not reflect our expectations nor it confirms our experience. It is worth noting that a different data source has been used for the currency data. On the other hand, the range breaks pattern is the one that dominates the currency technical trading although by a marginal difference. This result confirms our experience and direct observations of technical trading.

6.4 Overview of Market Liquidity

In this section we introduce the notion of market liquidity. Liquidity is a feature of financial markets that has been studied and discussed by a number of researchers and financial markets professionals. It attracted even more attention during and after the 2008 recession. Here we present the widely accepted definitions and approaches taken by other researchers and also our experience on the subject.

6.4.1 Definition

Liquidity is fundamental category of any markets and trading. The very primary meaning of liquidity is that there are market participants prepared to buy and sell at price levels that will allow some trades to be executed. The meaning that is implied when the term liquidity is used in the financial markets is much broader. Liquidity has become a measure of how frictionless the market of a security is. Liquidity is closely related to other market characteristics. During the 2008 global financial crisis period the financial industry had the opportunity to observe these relationships in a painful way. Initially the crisis started as a lack of credit. That quickly converted into lack of liquidity. This caused levels of price volatility hardly ever experienced before.

There are different aspects of liquidity. The three most widely used (see Von Wyss [83] and Gabrielsen et al. [84] among others) are depth, breadth and resilience. Lybek and Sarr [85] define also tightness and immediacy. Each of these categories describes a certain aspect of liquidity:

- **tightness** is a measure of the direct transaction costs associated with a trade. An example of that is the spread between the bid and the offer that are paid when trading at an exchange.
- **immediacy** measures the time needed for executing a transaction. This depends on how technologically advanced the trading platform on which the trades are executed is.
- **breadth** is an orderbook related factor that measures the volume that can be traded without moving the current price. In other words, it measures the size of the top level of the orderbook.
- **depth** is closely related to breadth. It is the volume that can be traded without impacting the price too much. Instead of focusing on the top level of the orderbook it looks at the whole 'depth' of it, i.e. how big the limit orders at every level of the book are.
- **resilience** is the measure that creates a longer term picture of the market liquidity. It measures the volume that a security market is capable of trading for a certain period of time without causing big impact on the price.

Lyons et al. [73] define the breadth and the depth as the liquidity measures that are the most important for a market maker, since they directly reflect the price impact of a market order. Lyons et al. [73] also introduce the interesting relationship between liquidity and market efficiency. They demonstrate that efficient markets support stable levels of liquidity that are not predictable. If liquidity becomes predictable that leads to market inefficiencies and hence trading opportunities.

6.4.2 Measuring liquidity

By associating numerical values to the above descriptions we transfer the notion of liquidity from philosophical category into a quantifiable factor. The different types of liquidity require different approaches for measuring their levels. The types that are based on the orderbook are relatively easy to define a method for. In this research we investigate how liquidity is related to technical trading on a security. We need a measure for liquidity that reflects the overall liquidity during the whole trading period that is analyzed. It is the resilience from the types above that reflects this requirement. The different ways of measuring this type of liquidity rely on different market variables. The most popular of these methods are based on one or more of the following: price, volume, volatility and/or variance.

The trading volume has been traditionally considered to be positively linked with liquidity. During the recent years with the development of ultra high-frequency trading this view has been reconsidered. In a few cases the automated trading algorithms created a self accelerating loop of sending orders to the market. The result of this was a very high volume combined with extreme volatility. The culmination of this effect was observed on May 10, 2010 and became known as the "flash crash". The conclusion of the investigation report by the United States Securities and Exchange Commission was: "in times of significant volatility, high trading volume is not necessarily a reliable indicator of market liquidity"[3]. Trading volume continues to be an important factor for measuring volatility but as Gabrielsen et al. [84] point out, it is only the first step of a much more involved analysis.

A very common liquidity measure that uses not only volume but also price change is the conventional liquidity ratio. It measures the traded volume per unit of absolute price change (see Gabrielsen et al. [84]):

$$LR_t = \frac{\sum_{t=1}^{T} P_t V_t}{\sum_{t=1}^{T} |P_t - P_{t-1}|} \tag{6.2}$$

where t is a trading period, P is the closing price and V is the traded volume. This ratio can be calculated for a group of securities as well as for a single security. The meaning of the liquidity ratio is very close to the idea of resilience that has been described above. A high value for this ratio means high liquidity during the analysed period.

A similar idea has been followed by Hui and Heubel [86]. The liquidity measure that they introduce calculates the maximum price change per unit of traded volume:

$$LR_{HH} = \frac{\frac{P_{max} - P_{min}}{P_{min}}}{\frac{V}{S\bar{P}}} \tag{6.3}$$

where P_{max} and P_{min} are respectively the maximum and minimum traded price during the analyzed period, V is the total traded volume, S is the total outstanding volume of the traded asset and $\bar{P}$ is the average price during the period. The Hui and Heubel ratio is in fact a measure of how illiquid the current market for that asset is. The higher the ratio, the worse the liquidity is assumed to be. The normalisation of the numerator is needed to convert the price change in return terms and make results comparable across different times. The normalisation of the denominator has been introduced by Hui and Heubel to make the ratio comparable across different securities. It is needed to neutralise the differences in percentage of outstanding shares for different equities. That adjustment is not necessary for other types of securities such as currency pairs and futures on stock indexes, government bonds or commodities. Gabrielsen et al. [84] criticize the Hui and Heubel ratio for being too slow for the automatic liquidity adjustment when applied to daily price data. However the ratio is still valid when used on intraday data.

Amihud [87] introduces an illiquidity measure that has a very similar logic to the Hui and

[3] Findings Regarding the Market Events of May 6, 2010 Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues, September 30, 2010, `https://www.sec.gov/news/studies/2010/marketevents-report.pdf`

Heubel ratio. He defines the illiquidity of a stock as the average ratio of the daily absolute return and the total traded volume in dollar terms on that day:

$$ILLIQ_y = \frac{1}{D_y} \sum_{d=1}^{D_y} \frac{|R_{yd}|}{VOLD_{yd}} \tag{6.4}$$

The above equation is averaged over a trading year. D_y are the number of trading days in the year y for that security. We deliberately omit the subscript i from the original that identifies the i^{th} equity, since this research does not focus only on equities. The ratio preserves its meaning when the time periods are scaled down for analysis of intraday data. In this case the ratio can be averaged over a day or a week and the absolute returns can be calculated for a minute or an hour.

The above liquidity measures are the most widely accepted ones that use the trading volume. There are measures for liquidity that use only the observed security price or quantities derived from it. Gabrielsen et al. [84] describe a ratio that has been introduced by Marsh and Rock [88]. The idea of this liquidity ratio is to use the number of transactions instead of the traded volume. The ratio computes the price change per transaction and is calculated as follows:

$$LR_{MR} = \frac{1}{M} \sum_{m=1}^{M} |\frac{P_m - P_{m-1}}{P_{m-1}}| \tag{6.5}$$

where M is the total number of transactions during the period and P is the asset price. As in the above instances we omit the stock index i since our intention is to use the ratio in its generic meaning as a liquidity measure of any traded security rather than just stocks. The meaning of the ratio, as explained by Gabrielsen et al. [84], is that an asset that is traded rarely in very big chunks is apparently less liquid than an asset that is traded close to continuously in very small amounts. The two assets can end up with comparable amount of traded volumes, and therefore the number of transactions create better insight on the way the assets have been traded and hence on their liquidity.

Another liquidity measure that is based only on the price changes is the variance ratio. Gabrielsen et al. [84] introduce it as:

$$VR = \frac{var(R_T)}{T var(Z_T)} \tag{6.6}$$

In the above equation 6.6 R_T is the long-term variance and Z_T is the short-term variance. The multiplying factor T is equal to the number of short periods in a long period. Values smaller than 1 suggest that the market of the analyzed security is not fully efficient, which affects its liquidity. As the authors of the above quoted research point out, this ratio can be computed with different choices for the long and the short periods. Since our research is focused on intraday trading we use relatively shorter periods to identify the current liquidity conditions. The variance of minutely returns is compared to the variance of one- or multi-hour returns. On the other hand, in order to distinguish between more and less liquid securities we compute the measure with using longer periods. In this case hourly returns variance is compared to the variance over a number of days.

6.5 Effects on Technical Trading of Markets with Higher and Lower Liquidity

In this section we explore the liquidity profiles of the securities that were identified to be actively traded by technical traders. For each of these securities we compare the technical trading intensities across periods of high, medium and low liquidity. Figure 6.22 visualizes the daily liquidity profile of the DAX Index futures for the analyzed period. The lack of liquidity during the 2008 crisis is clearly visible.

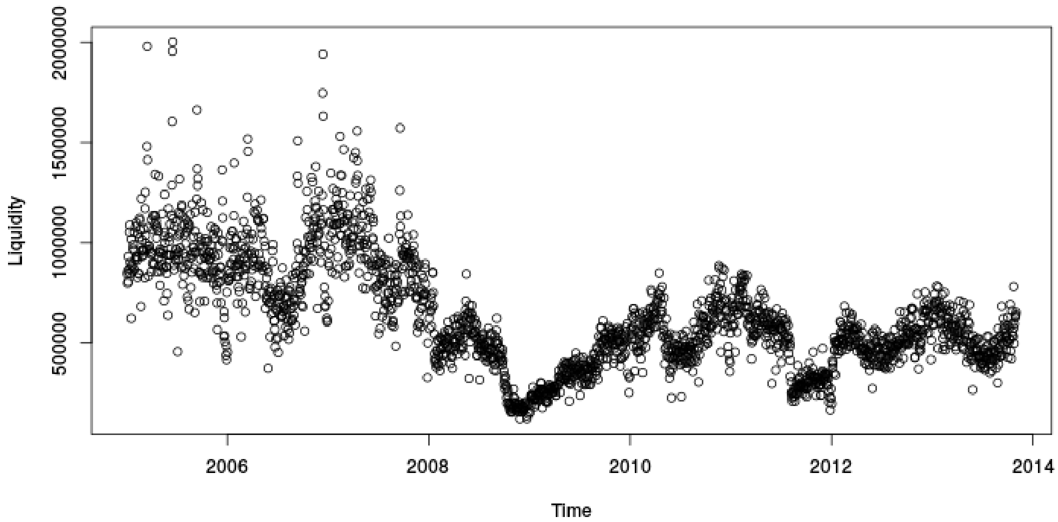


Figure 6.22: *Liquidity profile of the DAX Index futures.*

In section 6.3 we identified the Head-and-shoulders price pattern to be actively traded on a range of securities. The results also showed that the technical trend has been traded on the energy futures: oil, natural gas and gasoline. These results become the basis for the analysis in this section. For each of the securities that we identified as actively traded, we compute the daily liquidity ratio and then define three levels of liquidity: low, medium and high. These levels are based on the 15th and 80th percentile of the observed liquidity values for each of the securities. The technical trading detection results are split into three groups defined by the level of liquidity on the day on which the price pattern occurred. As a result of this we get three different results for the passive and active intensity for each of the traded securities. The Kullback-Leibler divergence is then calculated for the distributions implied by these results. Figure 6.23 depicts the KL divergences. The first thirteen results are for the Head-and-shoulders pattern and the last three are for the technical trend. Some of the securities did not register patterns during all of the liquidity conditions. For example, there were no Head-and-shoulders patterns detected on the price of the gasoil futures during low liquidity days. Another example is the light crude oil futures for which all the Head-and-shoulders patterns have been observed on days with normal levels of liquidity.

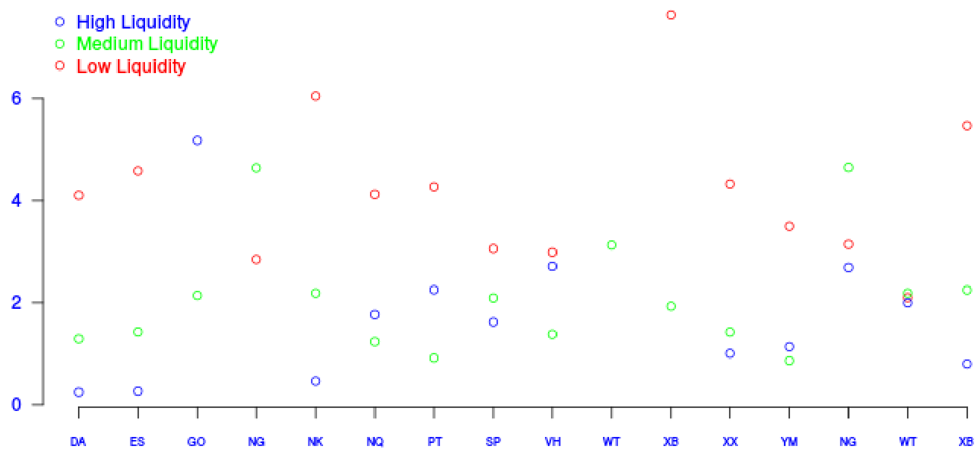


Figure 6.23: *Kullback-Leibler divergence for different liquidity bands. The liquidity bands are based on the 15th and the 80th percentile of the liquidity values calculated for each security. The liquidity values have been computed using the conventional liquidity ratio (Eq. 6.2)*

Our analysis demonstrates that during periods of liquidity shortages the technical trading is more active. We observe this for most of the securities on which the Head-and-shoulders is traded on a range of securities as well as the technical trend on the gasoil futures. This result may at first seem counter-intuitive, however it is confirmed by the ideas of technical trading that we revised in chapter 2, section 2.1. The analysts' and traders' reasoning behind technical analysis and trading is the ability to identify the current levels of supply and demand on a traded security only by looking at the security price fluctuations. When the liquidity is scarce the price can be even more sensitive to short term changes in supply and demand. This line of argument can explain the technical trading activity during these periods.

6.6 Conclusions

This chapter presented the third experiment of the thesis. It continued the idea that was developed in chapter 4 in three different directions. Firstly we proposed a new way for automation of technical pattern detection. It is based on price linearization using the concept of topographic prominence and the way the human eye analyzes price series. This new method is especially adequate for the types of technical patterns that are not completely quantifiable and rely on the 'chart reading' skills of the technical analysts. We developed an algorithm for computing the price prominence and used it in the automated detection of the Head-and-shoulders and the technical trend price patterns. For each of the trading strategies that we analyzed we presented post-trade return analysis.

We used the newly developed pattern detection and the methodology that we developed in chapter 4 to identify technical trading activity to analyze a broad range of securities. In addition to the newly automated Head-and-shoulders and technical trend patterns we carried out the analysis also on the Range breaks and moving averages strategies with the newly added securities. We used the Kullback-Leibler divergence for better interpretation of the technical trading intensity results. We demonstrated that the Head-and-shoulders is the most actively traded of the four different technical patterns that we explored. The other pattern that clearly showed active technical trading was the technical trend. The equity index futures and the energy futures are the securities that are most actively targeted by technical traders.

The third contribution of the chapter is the investigation of the effect of market liquidity on technical trading. We reviewed the concept of liquidity and the measures used to quantify it. We computed and analyzed the liquidity profiles of the securities that we have identified to be exposed to technical trading. The trading intensities presented in section 6.3 were then analyzed with respect to the level of liquidity on the days the patterns have been observed. The analysis showed that the liquidity shortage does not decrease the levels of technical trading.

The research in this experiment can be extended in various directions. One could use the introduced by this research price prominence idea to automate other technical price patterns from more common ones such as double tops and bottoms to less popular and exotic ones such as ‘diamond’ formation. Another obvious avenue to explore is to collect the price data and to test more securities, especially foreign exchange pairs.

This research has been carried out on price data aggregated minutely and intraday technical trading. The same type of analysis could be translated to higher levels of aggregation such as five minutes, hourly or daily intervals. Following from that, the technical trading strategies based on these patterns can have holding periods that extend to more days. The trading detection can be employed for higher frequency price data for the purposes of isolating specific market makers’ or high frequency traders’ behaviour.

The effect of liquidity on technical trading may be explored on different types of securities. It would be interesting to explore securities that are intrinsically less liquid. These include emerging markets currency pairs and less frequently traded futures.

Chapter 7

Conclusion and Future Work

This chapter brings this research to a conclusion. We review the motivation behind the research. The major contributions of the thesis are summarized. Suggestions for future work are presented.

7.1 Concluding Remarks

As stated in the abstract, this thesis investigates methods based on particle filtering for detecting the presence of technical trading on foreign exchange and futures markets. The research combines analytical methods, specific knowledge on technical price patterns and empirical observations to analyze the level of activity of this specific type of trading.

The idea for this thesis came as a result of identifying the potential for contributing to market transparency and efficiency by quantifying the behaviour of technical traders, a large group of traders that traditionally have been to a great extend ignored by the academic community. On one side, the numerous years of experience in the financial industry created the opportunity to view from inside the world of technical analysis. Reading internally published technical analysis views, formal and casual conversations with technical analysts and traders and testing and live-trading technical strategies have been instrumental for understanding the ideas and the behaviour of this unique set of traders on the financial market. Getting an insight that spans from the generic concepts down to the specific price patterns was crucial for investigating this research. On the other hand, the research knowledge and the mathematical and computational expertise provided the necessary basis for approaching the topic with academic rigour.

The existing literature and academic research on technical analysis focuses on different sets of topics that do not naturally link to each other. Most of them also lack or ignore the information that comes from the actual technical analysts and traders. Traditionally the academic research has been concentrated on market efficiency and the profitability of technical trading in general from the perspective of efficient markets. Classic example for this school of thought is the research by Fama [6] followed later by Fama [13] and Fama [14]. An alternative approach to technical analysis is to consider it as an aspect of behavioural finance (see Kahneman and Tversky [16]). The subject has been researched by Kahn [11] and Saettele [4] among others. Bodie et al. [89] present an overview of technical analysis as an example of behavioural finance. The third group of research concentrates onto automating the detection of price patterns occurrences. Osler and Chang [21], Lo et al. [17] and Shaw [18] among others, suggest methods for price smoothing and linearization in order to isolate the set of local minima and maxima used for building certain types of technical patterns. A very important aspect that these approaches fail to reflect is the human perception of a price chart. The way the human eye identifies the significant peaks and troughs on a price line is different from the algorithms that these researches propose. The last group of academic research on technical trading attempts to integrate the effect of that trading in the asset price formation. Shaw and Schofield [50] and Feng et al. [48] develop models for pricing contingent claims in presence of technical trading. These researchers approach technical trading as a single aggregated process and do not distinguish between technical patterns and trading strategies.

The existing research leaves a blank space between the clusters described above and also between each of them and the real technical trading. The idea for this thesis came as a result of identifying this space. Our research creates links between the above described topics and mainly between the existing research and the real world of technical analysis by employing specific knowledge and information on technical analysis and trading from the financial industry.

Firstly, we focused on individual technical price patterns. This approach creates the obvious limitation that only a certain number of technical strategies can be explored. Our response to this limitation was to select patterns from different types. The first two are 'Range breaks' and 'Moving averages'. They are both relatively straightforward to automate using objective criteria and an obvious set of parameters. The former is a continuation pattern, the latter is a reversal pattern. The third pattern that we selected for our analysis is the 'Head-and-shoulder'. It belongs to the group of patterns that are spotted by the technical analysts when looking at the sequence of price peaks and troughs. There are 'chart reading skills' involved in identifying the pattern unlike the previous two patterns. The fourth choice went to a pattern that belongs to a large class of technical analysis focusing on the price trend. This type of analysis works with support and resistance levels and the lines that connect them.

Secondly, we targeted the question: 'How actively is this pattern traded?'. In our research we talk about technical analysis and technical trading strategies. The latter put the former in real terms. A technical pattern or a group of patterns are worth further research and analysis only if there are market participants who regularly invest significant amount of capital or respectively, take significant amount of risk following these patterns.

Lastly, we demonstrated the connection between the information about the individual patterns' trading and the pricing of intraday contingent claims on the traded security. This is the missing link between intraday pricing models that include technical trading in the price process and the specific knowledge about the parameters that define a traded technical strategy.

This line of thought naturally leads to the contributions that this thesis makes. Before presenting these contributions we would like to review briefly the resources and inputs that were used to achieve this. In order to get a realistic view on the technical price patterns and technical analysis as a whole, we interacted directly with technical analysts and followed a wide range of technical analysis reports. A major challenge for any research of this type is to collect the needed amount of data. For this thesis we collected and processed price data for a range of securities. Since the analysis focused on intraday data we collected minutely and higher frequencies data for at least six years. Converting the ideas into academic research required the appropriate research methods and approaches. Finally, for the implementation and testing of the research we needed a powerful and at the same time flexible programming language and environment. R came as a natural choice.

7.2 Major Contributions

In this section we review the main contributions of this thesis. These contributions extend from a new approach for automating technical patterns identification, via presenting a methodology for detection and measuring of technical trading activities to a way of integrating technical trading into pricing of intraday options. In addition to that we developed an experimental environment for testing technical strategies and provided post-trade performance results for the strategies we tested on a range of securities.

Below we present a summary of the contributions this thesis makes.

1. Experimental software environment

   The first contribution of this research is developing an environment for testing trading strategies. The environment is developed within R and PostgreSQL. It uses the scalability and the flexibility of these platforms. Any new strategy can be easily added to it. The way the test system has been built allows for 'look-ahead bias' proof testing of trading strategies.

2. Detection and tracking of technical trading

   The second main contribution of the thesis is using particle filtering methods for identifying the presence of technical trading on the market of the analyzed securities. We model the process of technical trading triggered by a specific pattern and demonstrate how it can be identified within the background noise of all the existing trading on that security. We achieve this by combining the advanced statistical methods with our knowledge and experience in financial markets and technical trading. We applied the methodology that we developed for detecting technical trading but the same methodology can be used for detecting any other type of trading behaviour. The knowledge that a group of traders are expected to trade in a given direction at the occurrence of a price patterns is essential for many different groups of market participants. This type of information contributes for increased market efficiency. It improves the market makers' ability to forecast the current and future liquidity and hence offer better prices. The ability to forecast traders' activities helps also brokers' manual and automated execution services to achieve more optimal execution for client orders. For market participants trading on the market of that security for hedging or speculating purposes technical trading detection can help to improve trade timing and hence lower transaction costs. This knowledge is also essential for any position management or risk management strategy.

3. Option pricing in the presence of technical trading

   In the second experiment we demonstrated how the information on specific technical patterns and their trading intensities can be used for pricing intraday options on the securities exposed to active intraday technical trading. We showed that the existence of technical trading undermines the assumption that the price is a Markov process. For each of the analyzed patterns we described the set of parameters that had to be taken into account in order to extend a pricing model information set with the information for the technical trading.

4. Technical trading, prominence and liquidity

   The first contribution of this experiment is developing a new approach for technical pattern automation based on price prominence. This is a notion that we derived from the topographic prominence and the way the human eye analyzes charts. The existing approaches that have dominated the field start with a method to process the price series, derive patterns based on this method and then compare or check with real technical analysis. Contrary to that, our approach analyzes first how the technical analysts work, processes the price using an algorithm similar to the way they operate and then identifies the patterns.

   The second contribution of this experiment is the application of the above described method to identify technical price patterns on a range of securities. We ran an automated identification for four different price patterns: 'Head-and-shoulders', technical trend, moving averages and range breaks. We analyzed minutely price data for thirty-nine widely traded securities. The data for each security extends for at least six years. The research presents full post-trade performance for trading strategies based on each of the price patterns.

   Having developed a methodology for detection and measuring of technical trading activity we used it for carrying out and extended survey on a range of securities. For each of the technical strategies for which we identified the times of technical patterns occurrences and recorded the performance of the strategies, we tested for presence of actual technical trading. We analyzed the results and concluded that technical trading is, in fact, significant for two of the tested patterns: 'Head-and-shoulders' and technical trend. The results also showed that the energy futures: natural gas, crude oil and gasoline, as well as the most liquid equity index futures attract a significant amount of technical trading.

   There are a number of characteristics that can describe a current state of the financial markets as a whole or the market of a specific security. Market liquidity is one of these features and it is considered one of the most important ones. It was one of the major categories used to measure the extent of the 2008 financial crisis and has continued to concentrate market participants' and economic policy makers' attention. Given the importance of the liquidity factor we explored how it affects the levels of technical trading activity for the patterns that have been identified to be actively traded. Our results showed that during periods of decreased liquidity technical trading is a type of trading that continues at intensities that are the same or higher than during periods of normal liquidity.

7.3 Suggestions for Future Work

As described in section 7.1, one of the ideas of this thesis is to provide links between knowledge from the financial industry and the academia on the broad subject of technical analysis and trading. Future projects that extend this research would follow the same concept. That can be achieved by bringing in, processing and analyzing more information from the industry. Another direction for further development is to extend and transfer the current research ideas to other dimensions of the current research space. Below we present more details on how this could be done together with examples for future projects.

In this thesis we explored a limited number of technical patterns specified via sets of static parameters. One obvious way to extend this research would be to further explore the real world of technical analysis aiming to include more price patterns. The currently included patterns can be analyzed with different parameter sets and enhanced flavours.

Another avenue for 'extensive growth' is to extend the research by adding more securities. Our analysis focused on futures and foreign exchange. Traditionally, equities have been attracting wide range of investment activities including technical trading. Equities have also been the type of securities that academic researchers usually select for their analysis. This would be another way to link the ideas developed in this thesis with the existing literature.

The next interesting extension of this research would be to generalize the idea of trading detection to other types of trading not based on price patterns. Actually, the concept allows a generic version of the research that measures the intensity of any kind of traders' behaviour. Provided the necessary data are available, a model can be developed that monitors the market of a security for fraudulent trading activity.

In this research we used static parameter sets to define the price patterns. If a pattern that is actively traded is misspecified when fed into the trading detection algorithm it will produce negative results. This limitation can be overcome by introducing dynamically derived price pattern parameters. The methodology will continue to operate on a pattern-by-pattern basis but by including some of the pattern parameters into the pattern detection process, it would produce more optimal results.

In this thesis we focused our analysis on intraday trading. For our purposes we defined this as the set of trading strategies based on signals arriving during a trading day. The holding periods of these strategies vary between a few minutes and the rest of the trading or the calendar day. If we keep decreasing the holding period, we get strategies that are driven by very fast, price based, signals and their holding periods start approaching the milliseconds range. This is the realm of the high frequency trading (HFT). Extending this research for detection of high frequency trading strategies would be of an immense importance for the market making business.

Appendix A

R Code Template for Trading Strategies

The R code that is presented in this section demonstrates how a live trading strategy can be implemented in R using the utilities that the language and the platform offer. The generic design follows three major principles:

- The strategy properties and methods should be enclosed within their own space that is not mixed up with the global environment;
- The strategy must provide functionality to act on price updates;
- The strategy must be able to preserve its state between consecutive updates.

The R code template below achieves the above requirements in a relatively simple way:

Listing A.1: Trading Strategy Template

```
tradingStrategy = function(
        ## strategy initialization parameters
        ...
){
        ## the constructor returns an environment
        ## that holds the methods and the variables
        ## of the strategy
        local({
                ## strategy variables
                .var1 = NULL
                .var2 = 0
                .active = FALSE
                .initialized = FALSE
                .trades = list()
```

```
                ## strategy methods / functions
                ## with examples showing
                ## how to get and set object
                ## variables
                activate = function(){
                        if(.initialized)
                                .active <<- TRUE
                }
                deactivate = function(){
                        .active <<- FALSE
                }

                #... More methods ...

                run = function(quote){
                        ## strategy execution code
                }
                ## this is the function that is called on
                ## every price update, the local() structure
                ## used allows it to be called as a () operator
                ## of the strategy object
                watch = function(quote){
                        ############################
                        ## price processing code
                        ############################

                        ## the method does not have a return value
                        invisible()
                }

        })
}
```

A new strategy object is created by calling the above strategy constructor providing values for the strategy parameters:

Listing A.2: Creating a trading strategy object

```
strategy1 = tradingStrategy(p1 = 1, p2 = 2)
```

Once the strategy object is created at every price update *quote* we call the strategy watch method by simply executing:

Listing A.3: Strategy price update

```
strategy1(quote)
```

At any time the current state of the strategy can be monitored by checking the object variables using the *get()* function:

Listing A.4: Get strategy object variables

```
get('.trades', environment(strategy1))
```

Appendix B

R Code for Price Prominence

Below we present an R function that implements the prominence method for an object of class timeSeries, package timeSeries. The function *getTsProminence* is part of the R package created for this research.

Listing B.1: Prominence function implemented in R

```
getTsProminence = function(tSeries, label='Close') {

    len = nrow(tSeries)
    M = max(tSeries[,label])
    m = min(tSeries[,label])
    ## initialize prominence to 0,
    ## it will be the prominence for all flat parts
    tSeries = cbind(tSeries, prominence = 0)
    promL = paste0(label, "_prominence")
    names(tSeries)[length(names(tSeries))] = promL

    ## set min and max prominence to
    ## +/- difference max - min
    maxProm = M - m
    maxPromInd = which(tSeries[,label] == M)
    minPromInd = which(tSeries[,label] == m)
    tSeries[maxPromInd, promL] = maxProm
    tSeries[minPromInd, promL] = -maxProm

    ## get all peaks and troughs
    diffs = diff(as.numeric(tSeries[,label]))
    diffs = c(-sign(diffs[1]), diffs)
```

```
peaks = which(diffs > 0 & c(diffs[-1],-1) <= 0)
troughs = which(diffs < 0 & c(diffs[-1],1) >= 0)

## for every peak / trough find the two
## potential parents (preceding and following)
promFun = function(
                                p, peakInd, troughInd,
                                peakSeries, sgn = 'p'
                                ){
        minFun = list(p = min, n = max)[[sgn]]
        maxFun = list(p = max, n = min)[[sgn]]
        gtFun = get(list(p = '>', n = '<')[[sgn]])
        prnts = peakInd[
                gtFun(peakSeries[peakInd],peakSeries[p])
        ]
        intvl = findInterval(p, prnts)
        trghs = numeric(0)
        ## left trough
        if(p > 1) {
                trghs = minFun(
                        peakSeries[max(1,prnts[intvl]): p]
                )
        }
        ## right trough
        if(p < length(peakSeries)){
          if(intvl < length(prnts)){
                trghs = maxFun(
                        trghs,
                        minFun(
                                peakSeries[
                                  p : min(
                                        prnts[intvl+1],
                                        length(peakSeries)
                                  )
                                ]
                        )
```

```
                )
            } else {
                trghs = maxFun(
                        trghs,
                        minFun(
                          peakSeries[p : length(peakSeries)]
                        )
                )
            }
        }
        peakSeries[p] - trghs
}
tSeries[
        peaks[!is.element(peaks, maxPromInd)],
        promL
        ] = sapply(
                peaks[!is.element(peaks, maxPromInd)],
                promFun,
                peakInd = peaks,
                troughInd = troughs,
                peakSeries = tSeries[,label],
                sgn = 'p'
)

tSeries[
        troughs[!is.element(troughs, minPromInd)],
        promL
        ] = sapply(
                troughs[!is.element(troughs, minPromInd)],
                promFun,
                peakInd = troughs,
                troughInd = peaks,
                peakSeries = tSeries[,label],
                sgn = 'n'
)
```

```
        tSeries
}
```

Appendix C

Kullback-Leibler Divergence for Gamma Distributions

The Kullback-Leibler divergence (see Barber [90] among others) measures the distance between two distributions p and q is defined by

$$KL(q|p) = E_q[\log(q(x)) - \log(p(x))] \geq 0 \tag{C.1}$$

We want to compute the above quantity for two gamma distributions with the same scale parameter β and different shape parameters λ_p and λ_a. We can follow two different approaches to arrive to the same result.

C.1 Approach 1

In this approach we follow Nielsen and Nock [91]. The gamma distribution belongs to the exponential family of distributions. For all distributions that belong to this family the following canonical representation is valid:

$$p(x,\theta) = \exp(t(x)\cdot\theta - F(\theta) + k(x)) \tag{C.2}$$

In the above representation $t(x)$ is the sufficient statistics, $F(\theta)$ is the log-normalizer and $k(x)$ is the carrier measure. For a gamma distribution with shape λ and scale β the above functions correspond to $t(x) = \begin{pmatrix} \log(x) \\ x \end{pmatrix}$, $\theta = \begin{pmatrix} \lambda - 1 \\ -\frac{1}{\beta} \end{pmatrix}$ and $F(\theta) = \log(\Gamma(\lambda)) + \lambda\log(\beta)$

Following the definition of KL divergence one can derive (see Nielsen and Nock [91]) the KL divergence for two distributions q and p of the same exponential family as:

$$KL(p|q) = F(\theta_q) - F(\theta_p) - (\theta_q - \theta_p)\cdot E_p(t(x)) \tag{C.3}$$

When we substitute the above functions in equation C.3 we receive:

$$KL(p_{\lambda_p}|p_{\lambda_a}) = \log(\Gamma(\lambda_a)) - \log(\Gamma(\lambda_p)) - (\lambda_a - \lambda_p)\psi(\lambda_p) \tag{C.4}$$

where $\psi(x)$ is the digamma function $\psi(x) = \frac{d}{dx}\log(\Gamma(x))$

C.2 Approach 2

The same result can be derived directly if we use equation C.1 and the probability density function of the gamma distribution with shape λ and scale parameter β:

$$f(x|\lambda,\beta) = \frac{\beta^{-\lambda}x^{\lambda-1}e^{-\frac{x}{\beta}}}{\Gamma(\lambda)} \tag{C.5}$$

The Kullback-Leibler divergence of two gamma distributions with shape parameters λ_0 and λ_1 and a common scale parameter β, with probability density functions f_0 and f_1 respectively, becomes:

$$KL(Gamma(x|\lambda_0,\beta)|Gamma(x|\lambda_1,\beta)) = E_0[\log(f_0) - \log(f_1)] \tag{C.6}$$

$$\begin{aligned}
&= \int_0^\infty (-\lambda_0\log(\beta) + (\lambda_0 - 1)\log(x) - \frac{x}{\beta} - \log(\Gamma(\lambda_0)) \\
&\quad +\lambda_1\log(\beta) - (\lambda_1 - 1)\log(x) + \frac{x}{\beta} + \log(\Gamma(\lambda_1))) \\
&\quad \times \frac{\beta^{-\lambda_0}x^{\lambda_0-1}\exp(-\frac{x}{\beta})}{\Gamma(\lambda_0)}dx
\end{aligned} \tag{C.7}$$

$$= \log(\Gamma(\lambda_1)) - \log(\Gamma(\lambda_0)) - (\lambda_1 - \lambda_0)\psi(\lambda_0) \tag{C.8}$$

Bibliography

[1] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL http://www.R-project.org/.

[2] John J. Murphy. *Technical analysis of the financial markets*. New York Institute of Finance, Fishkill, N.Y., 1999. ISBN 0735200661 9780735200661. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=0735200661.

[3] Robert W. Colby. The encyclopedia of technical market indicators, 2003. URL http://site.ebrary.com/id/10409653.

[4] Jamie Saettele. Sentiment in the forex market indicators and strategies to profit from crowd behavior and market extremes, 2008. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9780470208236.

[5] Tim Knight. Chart your way to profits the online trader's guide to technical analysis with prophetcharts, 2010. URL http://site.ebrary.com/id/10399027.

[6] E. F. Fama. The behavior of stock-market prices. *Journal of Business*, 38(1):34–105, 1965.

[7] Brian Hurst, Yao Hua Ooi, and Lasse H Pedersen. A century of evidence on trend-following investing. *AQR white paper*, 2012.

[8] Nicolae B. Gârleanu and Lasse H. Pedersen. Efficiently inefficient markets for assets and asset management. Working Paper 21563, National Bureau of Economic Research, September 2015. URL http://www.nber.org/papers/w21563.

[9] L.H. Pedersen. *Efficiently Inefficient: How Smart Money Invests and Market Prices are Determined*. Princeton University Press, 2015. ISBN 9780691166193. URL https://books.google.co.uk/books?id=ZUS_oQEACAAJ.

[10] David Tuckett. *Minding the markets : an emotional finance view of financial instability*. Palgrave Macmillan, Houndmills, Basingstoke, Hampshire; New York, 2011. ISBN 9780230299856 0230299857. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=0230299857.

[11] Michael N. Kahn. A beginner's guide to charting financial markets a practical introduction to technical analysis for investors, 2007. URL `http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=214143`.

[12] B. B. Mandelbrot. The variation of certain speculative prices. *Journal of Business*, 36(4): 394–419, 1963.

[13] Eugene F. Fama. Efficient capital markets: Ii. *The Journal of Finance*, 46(5):pp. 1575–1617, 1991. ISSN 00221082. URL `http://www.jstor.org/stable/2328565`.

[14] Eugene F. Fama. Market efficiency, long-term returns, and behavioral finance1. *Journal of Financial Economics*, 49(3):283 – 306, 1998. ISSN 0304-405X. doi: http://dx.doi.org/10.1016/S0304-405X(98)00026-9. URL `http://www.sciencedirect.com/science/article/pii/S0304405X98000269`.

[15] Charles Mackay. *Memoirs of Extraordinary Popular Delusions and the Madness of Crowds*. R. Bentley, 1852.

[16] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decisions under risk. *Econometrica*, pages 263–291, 1979.

[17] Andrew W. Lo, Harry Mamaysky, and Jiang Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, 40:1705–1765, 2000.

[18] William Shaw. Automatic technical analysis. February 2001.

[19] J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall/CRC, Boca Raton, Fla., 1996. ISBN 0412983214 9780412983214. URL `http://www.worldcat.org/search?qt=worldcat_org_all&q=0412983214`.

[20] Salih N Neftci. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of" technical analysis". *Journal of Business*, pages 549–571, 1991.

[21] Carol L Osler and PH Chang. Head and shoulders: Not just a flaky pattern. *FRB of New York staff report*, (4), 1995.

[22] Federico Garzarelli, Matthieu Cristelli, Andrea Zaccaria, and Luciano Pietronero. Memory effects in stock price dynamics: evidences of technical trading, 2011. URL `http://arxiv.org/abs/1110.5197`. cite arxiv:1110.5197Comment: 21 pages, 8 figures, 2 tables.

[23] Hendrik Bessembinder and Kalok Chan. The profitability of technical trading rules in the asian stock markets. *Pacific-Basin Finance Journal*, 3(23):257 – 284, 1995. ISSN 0927-538X. doi:

http://dx.doi.org/10.1016/0927-538X(95)00002-3. URL `http://www.sciencedirect.com/science/article/pii/0927538X95000023`.

[24] WILLIAM BROCK, JOSEF LAKONISHOK, and BLAKE LeBARON. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, 47 (5):1731–1764, 1992. ISSN 1540-6261. doi: 10.1111/j.1540-6261.1992.tb04681.x. URL `http://dx.doi.org/10.1111/j.1540-6261.1992.tb04681.x`.

[25] D. Avitzour. Stochastic simulation bayesian approach to multitarget tracking. *IEE Proceedings - Radar, Sonar and Navigation*, 142(2):41, 1995. doi: 10.1049/ip-rsn:19951757. URL `http://dx.doi.org/10.1049/ip-rsn:19951757`.

[26] Donka Angelova and Lyudmila Mihaylova. Sequential Monte Carlo algorithms for joint target tracking and classification using kinematic radar information. In Per Svensson and Johan Schubert, editors, *Proceedings of the Seventh International Conference on Information Fusion*, volume II, pages 709–716, Mountain View, CA, Jun 2004. International Society of Information Fusion. ISBN 91-7056-116-8. URL `http://www.fusion2004.foi.se/papers/IF04-0709.pdf`.

[27] Donka Angelova and Lyudmila Mihaylova. Joint target tracking and classification via sequential monte carlo filtering. *Advances and Challenges in Multisensor Data and Information Processing*, 8:33, 2007.

[28] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 2106–2111 Vol.2, December 2004. doi: 10.1109/CDC.2004.1430359. URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1430359`.

[29] Ajay Jasra and Pierre Del Moral. Sequential monte carlo methods for option pricing, 2010. URL `http://arxiv.org/abs/1005.4797`. cite arxiv:1005.4797Comment: 37 Pages, 2 Figures.

[30] H.L. Christensen, J. Murphy, and S.J. Godsill. Forecasting high-frequency futures returns using online langevin dynamics. *Selected Topics in Signal Processing, IEEE Journal of*, 6(4): 366–380, August 2012. ISSN 1932-4553. doi: 10.1109/JSTSP.2012.2191532. URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6172210`.

[31] Rudolph Emil Kalman and Others. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[32] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95, 1961. doi: 10.1115/1.3658902. URL `http://dx.doi.org/10.1115/1.3658902`.

[33] J. Durbin and S.J. Koopman. *Time series analysis by state space methods*. Oxford University Press, Oxford; New York, 2001. ISBN 0198523548 9780198523543. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=0198523548.

[34] Andrew C. Harvey. *Forecasting, structural time series models, and the Kalman filter*. Cambridge Univ. Press, Cambridge u.a., 1990. ISBN 9781107720039 1107720036 9781107049994 1107049997. URL http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=676270.

[35] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Mathematics in Science and Engineering. Elsevier Science, 1970. ISBN 9780080960906. URL https://books.google.co.uk/books?id=nGlSNvKyY2MC.

[36] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proc. F Radar Signal Process. UK*, 140(2):107, 1993. doi: 10.1049/ip-f-2.1993.0015. URL http://dx.doi.org/10.1049/ip-f-2.1993.0015.

[37] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

[38] Markus Hürzeler and Hans R. Künsch. Monte carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193, 1998. doi: 10.1080/10618600.1998.10474769. URL /brokenurl#http://www.tandfonline.com/doi/abs/10.1080/10618600.1998.10474769.

[39] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. doi: 10.1080/01621459.1998.10473765. URL /brokenurl#http://www.tandfonline.com/doi/abs/10.1080/01621459.1998.10473765.

[40] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[41] Rong Chen. Sequential monte carlo methods and their applications. *IMS Lecture Notes Series, Markov Chain Monte Carlo*, 7:147–182, 2005.

[42] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[43] Ming T Lin, Junni L Zhang, Qiansheng Cheng, and Rong Chen. Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421, 2005.

[44] Iman Zabbah and Ehsan Partovi. Enhancing an automated trading strategy using artificial neural networks. In *International Conference on Computer Engineering and Network Security*, 2012.

[45] M. Cristelli, L. Pietronero, and A. Zaccaria. Critical overview of agent-based models for economics, 2011. URL `http://arxiv.org/abs/1101.1847`. cite arxiv:1101.1847Comment: 51 pages, 9 figures, Proceedings of the School of Physics "E. Fermi", course CLXXVI, 2010, Varenna.

[46] W. Brian Arthur, John H. Holland, Blake LeBaron, Richard Palmer, and Paul Tayler. Asset pricing under endogenous expectations in an artificial stock market, 1996. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.128.2902`.

[47] G. Caldarelli, M. Marsili, and Y. C. Zhang. A prototype model of stock exchange. Papers, arXiv.org, 1997. URL `http://EconPapers.repec.org/RePEc:arx:papers:cond-mat/9709118`.

[48] Ling Feng, Baowen Li, Boris Podobnik, Tobias Preis, and H. Eugene Stanley. Linking agent-based models and stochastic models of financial markets. *Proceedings of the National Academy of Sciences*, 109(22):8388–8393, 2012. doi: 10.1073/pnas.1205013109. URL `http://www.pnas.org/content/109/22/8388.abstract`.

[49] John Duffy and Nick Feltovich. Does observation of others affect learning in strategic environments? an experimental study. *Int. J. Game Theory*, 28(1):131–152, 1999. URL `http://dblp.uni-trier.de/db/journals/ijgt/ijgt28.html#DuffyF99`.

[50] William T. Shaw and Marcus Schofield. A model of returns for the post-credit-crunch reality: hybrid brownian motion with price feedback. *Quantitative Finance*, 0(0):1–24, 2011. doi: 10.1080/14697688.2011.642810. URL `/brokenurl#http://dx.doi.org/10.1080/14697688.2011.642810`.

[51] A. S. Kyle. Continuous auctions and insider trading. *Econometrica*, 53(6):1315–1336, 1985.

[52] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, pages 5–39, 2000.

[53] Robert F. Almgren. Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied Mathematical Finance*, 10(1):1–18, 2003. doi: 10.1080/135048602100056. URL `/brokenurl#http://dx.doi.org/10.1080/135048602100056`.

[54] G. Iori, M.G. Daniels, J.D. Farmer, L. Gillemot, S. Krishnamurthy, and E. Smith. An analysis of price impact function in order-driven markets. *Physica A: Statistical Mechanics and its Applications*, 324(12):146 – 151, 2003. ISSN 0378-4371. doi: http://dx.doi.org/10.

1016/S0378-4371(02)01888-5. URL `http://www.sciencedirect.com/science/article/pii/S0378437102018885`. Proceedings of the International Econophysics Conference.

[55] Eric Smith, J. Doyne Farmer, Laszlo Gillemot, and Supriya Krishnamurthy. Statistical theory of the continuous double auction, 2002. URL `http://arxiv.org/abs/cond-mat/0210475`. cite arxiv:cond-mat/0210475Comment: 36 pages, 40 figures, RevTex4, submitted to Quantitative Finance.

[56] Dong Guo, Xiaodong Wang, and Rong Chen. New sequential monte carlo methods for nonlinear dynamic systems. *Statistics and Computing*, 15(2):135–147, 2005. URL `http://dblp.uni-trier.de/db/journals/sac/sac15.html#GuoWC05`.

[57] W S Kendall, F Liang, and J-S Wang. Markov chain monte carlo innovations and applications, 2005. URL `http://www.books24x7.com/marc.asp?bookid=18665`.

[58] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Proc., Radar Sonar Navig.*, 146(1):2, 1999. doi: 10.1049/ip-rsn:19990255. URL `http://dx.doi.org/10.1049/ip-rsn:19990255`.

[59] Sergio Hernandez and Paul Teal. Multi-target tracking with poisson processes observations. In *Advances in Image and Video Technology*, pages 474–483. Springer, 2007.

[60] Kevin Gilholm, Simon Godsill, Simon Maskell, and David Salmond. Poisson models for extended target and group tracking. In *Optics & Photonics 2005*, pages 59130R–59130R. International Society for Optics and Photonics, 2005.

[61] Daniel Fink. A compendium of conjugate priors, 1997.

[62] J. A. Rice. *Mathematical statistics and data analysis*. Duxbury, Belmont, CA, 2nd edition, 1995.

[63] Biswabrata Pradhan and Debasis Kundu. Bayes estimation and prediction of the two-parameter gamma distribution. *Journal of Statistical Computation and Simulation*, 81(9):1187–1198, 2011.

[64] Rama Cont, Arseniy Kukanov, and Sasha Stoikov. The price impact of order book events, 2010. URL `http://arxiv.org/abs/1011.6402`. cite arxiv:1011.6402.

[65] Fishcer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637, 1973.

[66] Robert C Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973.

[67] J. Michael Harrison and David Kreps. Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, 20(3):381–408, 1979. URL `http://EconPapers.repec.org/RePEc:eee:jetheo:v:20:y:1979:i:3:p:381-408`.

[68] J. Michael Harrison and Stanley R. Pliska. Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and their Applications*, 11(3):215–260, 1981. URL `http://EconPapers.repec.org/RePEc:eee:spapps:v:11:y:1981:i:3:p:215-260`.

[69] Darrell Duffie. *Dynamic asset pricing theory*. Princeton Univ. Press, Princeton, NJ, 3. ed edition, 2001. ISBN 069109022X. URL `http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+186932650&sourceid=fbw_bibsonomy`.

[70] Steven E Shreve. *Stochastic calculus for finance 2, Continuous-time models*. Springer, New York, NY; Heidelberg, 2004. ISBN 0387401016 9780387401010. URL `http://www.worldcat.org/search?qt=worldcat_org_all&q=0387401016`.

[71] Oanda. *OANDA - Forex Trading and Exchange Rates Services*. OANDA - Forex Trading and Exchange Rates Services, Toronto, Canada, 2015. URL `http://www.oanda.com/`.

[72] ValeryA. Kholodnyi. Valuation and hedging of european contingent claims on power with spikes: a non-markovian approach. *Journal of Engineering Mathematics*, 49(3):233–252, 2004. ISSN 0022-0833. doi: 10.1023/B:ENGI.0000031203.43548.b6. URL `http://dx.doi.org/10.1023/B%3AENGI.0000031203.43548.b6`.

[73] Richard K Lyons et al. *The microstructure approach to exchange rates*, volume 333. Citeseer, 2001.

[74] M. M. Dacorogna, R. Gencay, U. Muller, R. B. Olsen, and O. V. Pictet. *An introduction to high frequency finance*. Academic Press, New York, 2001.

[75] lvaro Cartea. Derivatives pricing with marked point processes using tick-by-tick data. *Quantitative Finance*, 13(1):111–123, 2013. doi: 10.1080/14697688.2012.661447. URL `/brokenurl#http://dx.doi.org/10.1080/14697688.2012.661447`.

[76] Alvaro Cartea and Thilo Meyer-Brandis. How duration between trades of underlying securities affects option prices*. *Review of Finance*, page rfp013, 2009.

[77] Enrico Scalas and Mauro Politi. A parsimonious model for intraday european option pricing. Economics Discussion Paper 2012-14, Kiel Institute for the World Economy, 2012. URL `http://www.economics-ejournal.org/economics/discussionpapers/2012-14`.

[78] Richard L. Burden and J. Douglas Faires. *Numerical analysis*. Thomson Brooks/Cole, Belmont, CA, 2005. ISBN 0534392008 9780534392000 0534404995 9780534404994. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9780534404994.

[79] David Ruppert. Local polynomial regression and its applications in environmental statistics. Technical report, Cornell University Operations Research and Industrial Engineering, 1996.

[80] David Ruppert, Simon J Sheather, and Matthew P Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432): 1257–1270, 1995.

[81] A Maizlish. Prominence and orometrics: A study of the measurement of mountains. *WWW document, http://www.peaklist.org/theory/theory.html*, 2003.

[82] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford University Press, Oxford; New York, 2001. ISBN 0198572239 9780198572237 0198572220 9780198572220. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9780198572220.

[83] Rico Von Wyss. *Measuring and predicting liquidity in the stock market*. na, 2004.

[84] A Gabrielsen, M Marzo, and P Zagaglia. Measuring market liquidity: an introductory survey, mpra paper 35829, 2011.

[85] Mr Tonny Lybek and Mr Abdourahmane Sarr. *Measuring liquidity in financial markets*. Number 2-232. International Monetary Fund, 2002.

[86] B. Hui and B. Heubel. *Comparative liquidity advantages among major U.S. stock markets*. DRI Financial Information Group study series. Data Resources inc., 1984. URL https://books.google.co.uk/books?id=WRwcAQAAMAAJ.

[87] Yakov Amihud. Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets*, 5(1):31–56, 2002.

[88] Terry A Marsh and Kevin Rock. *Exchange listing and liquidity: A comparison of the American Stock Exchange with the NASDAQ National Market System*. American Stock Exchange, 1986.

[89] Zvi Bodie, Alex Kane, and Alan J. Marcus. *Investments*. McGraw-Hill/Irwin, Boston, 2009. ISBN 9780073382371 007338237X 0077261453 9780077261450 9780071278287 0071278281 9780070151574 0070151571 9780077128425 0077128427 9780071263252 007126325X 9780390668349 0390668346. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9780071278287.

[90] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[91] F. Nielsen and R. Nock. Entropies and cross-entropies of exponential families. In *2010 IEEE International Conference on Image Processing*, pages 3621–3624, Sept 2010. doi: 10.1109/ICIP.2010.5652054. URL `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5652054`.