Market Agents and Flash Crashes

Mohsen Naderi

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

September 30, 2020

Declaration

I, Mohsen Naderi, 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 thesis.

Over the years, I have worked at several major financial institutions focussing on electronic trading of equities, fx and fixed-income. Although my background and experience play a major role in my research, conclusions and viewpoints presented in this research represents my personal views and not that of my current or any of my previous employers.

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

- Mohsen Naderi and Philip Treleaven. *Challenges of full-scale simulation of an electronic financial market.* in preparation. 2020
- Mohsen Naderi and Philip Treleaven. *Using agent-based modelling to analyse the effects of new regulation on an electronic trading venue.* in preparation. 2020

Mohsen Naderi

Abstract

This thesis studies the use of agent-based modelling to investigate factors that can affect the stability of an electronic trading venue. Automated trading strategies have contributed to tighter bid/ask spread and granular liquidity in the markets. However, over the past few years, there have been several incidents, notably the *flash crash* of 6 May 2010, that raised concerns about the effect of such automated trading systems on the stability of financial markets. There have been different views on why the flash crash happened and what can be done to prevent similar issues in the future. Several changes have been proposed by regulators and market operators to improve the stability of financial markets. It is essential that these suggestions are well understood and scientifically analysed to clarify their ability to improve market stability and understand any negative side-effects they may bring.

The study proposes an agent-based modelling framework for financial market and market participants to allow analysing the effect of such proposed changes. The objective is to use agent-based models to simulate and analyse the behaviour of market participants in the event of a liquidity shock. Firstly, the study develops an agent-based model of a financial market and its participants and corresponding simulation platform. Using this simulation platform and market data from trading venues, the study examines the behaviour of artificial agents versus data from a real environment. Finally, this simulation platform is used to perform extensive experiments to understand the factors that have been claimed to contribute to a flash crash. It further analyses the effectiveness of solutions that are proposed to prevent it.

- The study first investigates how the *diversity of the trader population* can change the market's reaction to order-flow imbalance. In particular, it focuses on two types of market participants: high-frequency traders and fundamental traders. There have been concerns that a high ratio of high-frequency or fundamental traders can make markets more unstable. The experiments performed in the study confirm that having too many high-frequency traders contributes to higher market volatility. It is observed that not having high-frequency market makers also generates problems as long-term investors may not be able to provide short-term liquidity when needed causing price fluctuations.

- The study examines *electronic trading controls* that are being implemented by trading venues to maintain an orderly market. Trading rules are designed to provide a framework in which a market participant is expected to behave. Circuit breakers are extra controls that can stop trading when some safety conditions are broken. Circuit breakers are commonly

triggered by a large price move in a short period and they stop trading in the venue so that a human can review the status and decide whether to resume trading. Some of the trading venues have automated circuit breakers that enforce a short-term suspension of trading and automatically resume trading using an auction. There also have been suggestions for more complex measures to trigger circuit breakers than large price shift. The study examines how effective these measures are on increasing market stability.

- The study finally investigates the *interaction between markets*. It investigates how a liquidity crisis from one security in one market can expand to other securities and other markets. The flash crash started in one contract in the future market but soon expanded to ETF market and equity markets. This experiment analyses the factors that can affect that interaction by either reducing the effects of such event on other markets or amplifying the crisis and make it worse.

The first contribution of this thesis is the introduction of an agent-based framework to study the behaviour of a financial market with different classes of trading agents. An extensive study of this platform is performed and it is used to simulate the flash crash. The second main contribution of the thesis is the development of a simulation platform that is capable of simulating an electronic trading market with high-frequency traders. The study employs the same techniques and tools that are used in building high-frequency trading platforms to build a platform targeted for analysing regulation and market rule changes. The third contribution of this thesis is to analyse some of the factors that are claimed to contribute to flash crash or methods that are designed to prevent such events. The study analyses the effect of fundamental and high-frequency trader population on market stability. It examines studied the effect of circuit-breakers, minimum quote life and order-to-trade ratio limits. The final contribution of this thesis is to extend the base model further to more than one financial market and study how liquidity issues expand from one market to another.

Acknowledgements

I wish to express my greatest gratitude to my principal supervisor, Philip Treleaven, for his encouragement and support throughout the course of this work. Without his patience and guidance, this thesis would never have been completed. I am also grateful to my second supervisor, Piotr Karasinski, for inspiration, enthusiasm and guidance and to Guido Germano for being the assessor and for his constructive feedback. I am grateful to my thesis committee, Professor Dave Cliff and Dr Giuseppe Nuti for accepting to be my external examiners and for their insightful comments and feedback.

It has been a pleasure to be part of the Centre for Financial Computing at the Department of Computer Science of the University College London. I have enjoyed its great research atmosphere. During my career in finance, I enjoyed working with some of the brightest people in this field at Merrill Lynch, Knight Capital, Goldman Sachs, RBS and Deutsche Bank. I am grateful for all their support during my studies and for the fruitful exchanges at work. I am also grateful to CME and BATS for permission to use their data for my research.

It would not have been possible to write this doctoral thesis without the help and support of the kind people around me. I would like most of all to thank my wonderful wife, Elham, my daughter, Roshanak, my mother and father, my family, and my friends for their encouragement and patience. I am deeply indebted to my dear wife Elham for her continuing support without which doing this research would not have been possible. Her high writing standard has influenced my way of thinking fundamentally. This thesis is dedicated to her.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This thesis investigates the use of agent-based modelling to analyse *flash crashes* and the effectiveness of regulatory changes that have been proposed to prevent such events in the future. This chapter first provides a high-level overview of the May 2010 flash crash and concerns about the expanding role of electronic trading systems in today's financial markets. Next, research objectives and an overview of the methodology used to perform this research are explained. Finally, this chapter concludes with an outline of the thesis and a brief description of the chapters.

1.2 Research Motivation

While it is still common to see an open-outcry trading floor with human traders on TV screens, the reality is that most of today's trading in equities, futures, options, and foreign exchange (FX) happens in electronic exchanges. London Stock Exchange moved from open-outcry to electronic screen in 1986 as part of the deregulation of the markets (Lawson et al. 2006). Most of the other exchanges and trading venues have followed that suit and use computers to receive orders from members, match buy and sell orders and disseminate information about trades or un-match buy/sell orders in the orderbook to market participants. This expansion in the use of electronic platforms by exchanges and trading venues has contributed to a rapid increase in the automated trading strategies in the markets. One study suggests that high-frequency trading firms accounted for 60-73% of all US equity trading volume (Aite Group 2009). Other studies estimated 56% of equity trades (in terms of value) in the US and 38% in Europe, handled by high-frequency trading firms (Tabb Group 2010; Nuti et al. 2011). Using computer programs to make the investment decision and execute those decisions without involving a human trader has affected the market microstructure and trading dynamics. Some reports suggest that electronic

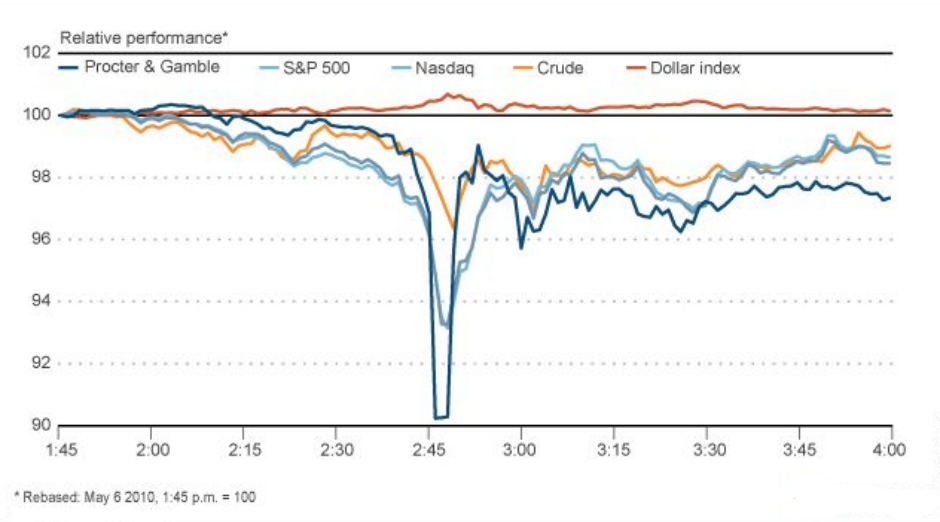


Figure 1.1: Flash Crash of 6 May 2010 (Reuters)

trading and high-frequency traders have improved liquidity and reduced bid/ask spreads and that benefits investors (e.g. Angel et al. 2010). On the other hand, some argue that although some of the indicators of market quality have improved, investors have not gained overall and the total cost of trading has increased (e.g. Zhang 2010; Brogaard et al. 2013).

Use of electronic platforms by trading venues enables market participants to use computer algorithms to manage orders and execute them on behalf of traders. Different terminology used to refer to use of computers for trading. Among those *Program Trading*, *Algorithmic Trading (AT)*, *Electronic Trading* and *High-Frequency Trading (HFT)* are among most commonly used terms to refer to such activity. Unfortunately, there is no commonly agreed definition of these terms and the exact type of activity each of these refers to. Even when CFTC was proposing regulation for high-frequency traders, it formed a working group to come up with a definition for high-frequency trading and classify trading activities that fall into that category (CFTC 2012). Throughout this thesis, we use *Electronic Trading* as a general term to refer to any trading activity that is performed on a computer based trading venue. It could be originated by a human trader or a computer. We use term *Algorithmic Trading* to refer to trading activity that is originated by a human or a fully independent system, but computers can decide about details of its execution. For example, a computer can choose to slice main (parent) order into smaller ones and determine the timing and destination of each slice. We use the term *High-Frequency Trading* to refer to activities that are fully controlled by a computer and can place and cancel orders at high speed (generally in times measured in microseconds up to a couple of milliseconds).

Moreover, there have been incidents over the past few years that have raised concerns about the effects of electronic trading systems on market stability. The most famous incident was the flash crash of 6 May 2010. On this day, U.S. stock collapsed and recovered to almost the same level within minutes (See figure 1.1). This is also referred to as the May 2010 flash crash

or simply *the flash crash*. During that downturn, approximately one trillion dollars were wiped off market value; although most of this was recovered shortly. Securities and Exchange Commission (SEC)[1] and Commodity Futures Trading Commission (CFTC)[2] conducted an official investigation into the event (CFTC and SEC 2010b; CFTC and SEC 2010a). The outcome of this inquiry suggests that a large sell order of E-Mini S&P Index future triggered the event. High-frequency traders and market makers bought part of this order-flow leaving them with significant short positions. As sell pressure continued, high-frequency traders started selling E-Mini to get out of their risk positions and that exacerbated the decline in the price. As prices dropped, retail order flow that is usually internalised by market makers was forwarded to market, and stop-loss orders triggered adding to the selling pressure. This selling pressure quickly expanded from E-Mini to SPY, the exchange-traded fund (ETF) representing that index. It then expanded to underlying stocks in the index and other correlated stocks and indexes like Dow Jones Industrial Average and Nasdaq Composite.

The flash crash of May 2010 was not the only time that trading venues were severely affected because of the issues with automated trading strategies. There have been other incidents with similar properties such as the crash of ETF markets (CNBC 2015), trading issues with Knight Capital (Bloomberg 2012a) and Goldman Sachs (Bloomberg 2013).

Due to the speed of such trading systems, any problem affecting their behaviour can spread across the market much faster than human traders operating and controlling those strategies. A number of regulatory changes have been proposed to prevent such problems happening in the future. It is important that these analyses and suggestions are scientifically analysed, and their effectiveness and side effects are studied before they are implemented. This thesis is concerned with the modelling and simulation of a financial market in which the majority of trading is performed by electronic trading systems. The aim is to provide a framework for studying the factors that contribute to a flash crash and analyse the effects of proposed regulatory changes before they are implemented.

1.3 Research Objectives and Method

This thesis provides an agent-based model of a financial market to study a flash crash. Its first objective is to review the related work in the literature to determine a suitable agent-based model of an electronic market, as presented in Chapter 2. Motivated by the outcomes of this review, the study uses a Zero-Intelligent (ZI) agent-based model of the market participants (Gode and Sunder 1993). ZI agents do not follow a strategy and do not have a memory of their past behaviour. Instead, they use a random probability distribution to choose the direction and the price of their orders. They have been successfully used to generate market models that replicate some of the characteristics of the financial markets. Most of these market models are simplified versions of real markets: i.e. each agent only trades one unit of security at a time, and orderbook

[1] U.S. Regulator for stocks and options market
[2] U.S. Regulator for future market

is cleared after each trade. The study relaxes these constraints and allows continuous trading in the orderbook and different sizes for the orders. This is a closer replication of conditions in a financial market, and such a setup allows us to use agent properties studied on financial market data to be used directly for simulation and analysis.

Its second objective is to build and study a simulation framework that can replicate a real market and flash crash, as discussed in Chapter 4. To achieve this objective, extensive experiments are then performed extensive experiments to show that the market behaviour emerging from these types of agents mimics the properties of a real financial market. To this end, the research first examines if these agents can reproduce the properties shown using previous similar models. It then places such agents in a mixed-mode simulation environment where orders from the simulated agent interact with historical data from a financial market. It studies the behaviour of different classes of agents placing passive and market orders. To further analyse agent behaviour, the study examines if a market composed of such agent produces similar stylised facts to a real electronic trading venue. Finally, it examines if this simulation setup can replicate the flash crash on CME using the same set of trading agents and an aggressive incoming sell order.

According to the official flash crash investigation report, a fundamental seller has been blamed for triggering the flash crash, and high-frequency traders have been claimed to have exacerbated the crash. Therefore, the study's third objective is to use the simulation framework to examine how the population of fundamental traders and high-frequency traders affects the market's response to a liquidity shock. To this end, it simulates a market with the population of participants chosen to be proportional to CME on the day of the flash crash, as discussed in Chapter 5. The study investigates whether the size of the market crash is different when the population of fundamental or high-frequency traders changes compared to other market participants. The focus is on the market's response to liquidity shock and understanding how much of the short-term flow imbalance can be absorbed by market participants.

As described earlier, there have been a number of suggestions about controls and trading rule changes to prevent flash crashes happening in the future. The study's fourth objective is to analyse the effectiveness of the proposed regulatory changes and their side-effects. The flash crash was problematic for investors not only because of its effects on S&P E-Mini futures but because it expanded from the Futures market to the ETF, and from the ETF market to the Equity markets. Its final objective is to study how the liquidity issue expands from one market to another, as discussed in Chapter 7. It investigates the five research objectives by carrying out extensive experiments using the agent-based model, and on the software platform that was specifically implemented for this thesis.

1.4 Thesis Outline

This thesis is organised as follows.

Chapter 2: Background. The purpose of this chapter is to identify the key challenges in analysis and modelling of the flash crash by reviewing a number of previous approaches. This chapter presents the official investigation report by market regulators on 6 May 2010 flash crash highlighting how the flash crash happened and the behaviour of different market participants during the flash crash. This chapter also reviews other flash crashes, technical glitches that cause market-wide issues similar to a flash crash, and some sharp market moves that were driven by real economic factors but present a market impact that is similar to a flash crash. These events can be positively or negatively affected by some of the controls that are designed to prevent flash crashes. This chapter presents the related work in the area of analysis and modelling of the flash crash, highlighting previous research info modelling flash crash or a market that includes high-frequency traders using agent-based modelling that influenced the study's agent-based framework. Finally, this chapter reviews currently available agent-based modelling software platforms. The discussion is mainly concerned with the flexibility of these software platforms to model a trading agent and their performance to allow running a representative model of a real market.

Chapter 3: Experimental Platform and Data. This chapter introduces a number of key concepts in out model and platform and provides the reader with the necessary background for understanding this thesis. This chapter discusses the simulation platform that we built to perform the experiments described in this thesis. Finally, provides technical details on the market data that is used in the study to perform experiments. The investigations to analyse the effects of proposed regulatory changes in the following chapters are carried out using the platform described in this chapter.

Chapter 4: Agent-Based Model of the Flash Crash. This chapter describes the agent-based framework that the study has develops to model a financial market that aims to replicate a real market. To model market participants, this research uses a zero-intelligent agent model with two modifications: agents have a position constraint, and their limit price is non-uniform and market-price dependent. The study conducts extensive experiments on the effect of each of these changes to a zero-intelligent agent model using simulated markets and mixed-mode simulation. In the mixed-mode simulation, the study performs experiments with artificial agents interacting data from real markets such as CME and BATS. Finally, the chapter ends with experiments replicating the market conditions of the flash crash on CME, which used as a baseline.

Chapter 5: Diversity of Trader Population. This chapter examines whether the relative population of high-frequency traders and fundamental traders compared to the rest of the market participants affect the stability of the market and its ability to handle short-term liquidity shock. First, it presents experimental evidence of the market behaviour when the relative population of fundamental traders changes and its effect on the market price. This experiment investigates two scenarios: (i) changing the population of fundamental traders (buyers and sellers together) and

(ii) changing the proportion of fundamental buyers vs sellers while keeping their total population fixed. Next, it experiments with changing high-frequency traders' population while keeping the population of other market participants fixed. This experiment also investigates two scenarios: (i) changing population of high-frequency traders compared to other market participants and (ii) changing the position limits of high-frequency traders.

Chapter 6: Algorithmic Trading Controls. This chapter investigates a number of mechanisms that have been proposed to control algorithmic trading strategies and prevent an event similar to the flash crash or at least limit damages of such event. The study investigates three of the most famous proposed methods: circuit breaker, minimum quote life, and order-to-trade ratio. Circuit breaker stops trading completely either for a predefined period of time or until it is manually resumed by human operators controlling the market. The experiment is conducted with the single market circuit breaker and the cross-market circuit breaker. Minimum quote life forces an incoming order has to stay in the market for a minimum time before it can be cancelled or modified. Order-to-trade ratio puts a higher bound on the number of order updates that can be applied by a participant compared to the number of trades that are done by each participant. In all these experiments the study compares the response of a market to a liquidity crisis when that control is applied to the case and when this control is disabled.

Chapter 7: Interaction between Markets. Finally, this chapter studies the interaction between two markets and how a liquidity crisis propagates from one to the other. It first experiments with the scenario where the same security trade in multiple markets and one market experiences a liquidity issue. Furthermore, it investigates the case where the liquidity crisis happens in highly correlated security. In this case, the study investigates the effect of the liquidity crisis on other securities and how the problem spreads. An example of such a scenario will be an index future and the ETF following the same index.

Chapter 8: Conclusions and Future Work. This chapter concludes this thesis with a summary of the main findings and contributions. Additionally, this chapter outlines possible directions for future research.

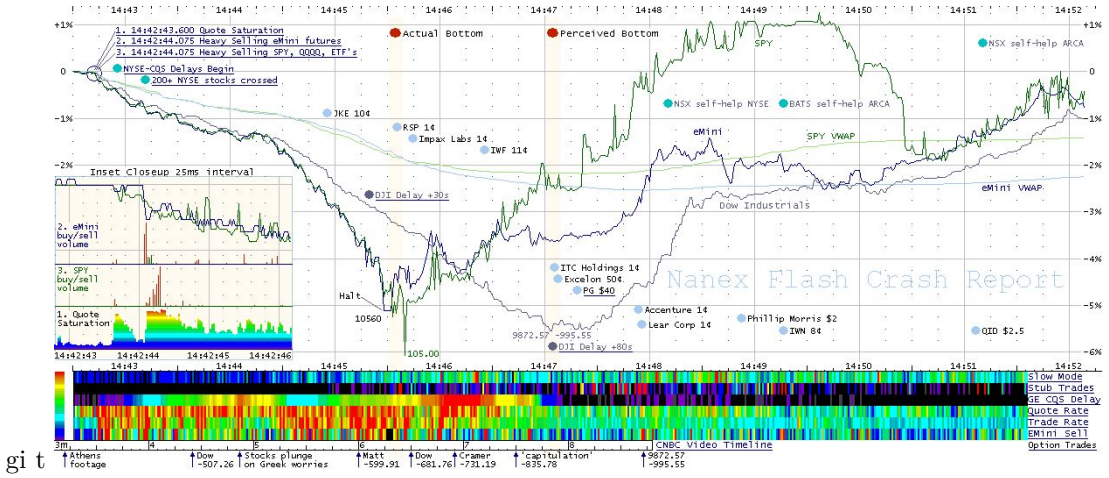
Chapter 2

Background

2.1 Introduction

A flash crash is a sudden fall in financial market prices followed by a quick recovery. The best-known incident of this kind, which made the term *flash crash* popular, happened on 6 May 2010. On that day U.S. equities and futures markets collapsed and recovered to almost the same levels in a short period of time. This incident and a number of similar events that happened recently have raised concerns among market participants, regulators and politicians about the stability of financial markets with extensive presence automated trading systems. There have been different views on why the flash crash happened and what can be done to prevent similar issues in the future. It is essential that these suggestions are well understood and scientifically analysed to clarify their ability to improve market stability and to recognise any negative side-effects they may bring. This research proposes an agent-based modelling framework for financial markets and traders to allow the analysis of the effect of such proposed changes.

This chapter starts by reviewing the flash crash of 6 May 2010 in detail in Section 2.2. It then reviews other market disruptions that have similar characteristics to a flash crash and are likely to be affected by the solutions that are proposed to prevent a flash crash in Section 2.3. There have been many suggestions from market participants, regulators and even politicians on what needs to be done. Section 2.4 looks at some of the solutions that are proposed to detect or prevent events similar to the flash crash in the future. Section 2.5 provides a review of previous research into the analysis of the flash crash and how it differs from this work. The study proposes to use agent-based modelling to model a financial market and its participants. Section 2.7 provides an overview of available agent-based simulation platforms that can potentially be used for this research. Section 2.8 provides a summary of this chapter.



gi t

Figure 2.1: Detailed View of the Flash Crash (Nanex 2010)

2.2 May 2010 Flash Crash

On 6 May 2010, U.S. equities and futures markets collapsed and recovered to almost the same level in a period of about twenty minutes. This quick drop and recovery is called a *flash crash*, and as May 2010 is the most well-known case so far it is sometimes called *"the flash crash"*. During this flash crash, the quick drop and recovery in securities prices occurred shortly after 14:30 EST[1]. It took about five minutes for markets to hit bottom and they recovered to the same level within about twenty minutes (See Figure 2.1). The Dow Jones plunged 1,000 points (almost 6%) before recovering. During the downturn, approximately one trillion dollars in market value were wiped out, but most of this loss was recovered shortly.

The Securities and Exchange Commission[2] (SEC) and the Commodity Futures Trading Commission[3] (CFTC) launched a joint investigation into the event. The preliminary report of the investigation was published on 18 May 2010, just a few days after the event (CFTC and SEC 2010b). The report notes that concerns over the European debt crisis caused a turbulent start of the day for US markets which led to a "significant but not extraordinary down day". This extreme volatility in the markets suggests the occurrence of a temporary breakdown in the supply of liquidity across the markets. The report investigated the top 10 buyers and sellers during the time from 2:00 pm to 3:00 pm EST. The report highlighted a large fundamental seller who traded during that time period, selling the contracts during the market going down and during its recovery. The report mentioned that the order was issued on E-Mini contracts as a hedge to an existing equity position; it did not name the large fundamental trader. The media identified the fundamental trader initiating this order as the mutual fund Waddle, citing internal documents prepared by exchange operator CME Group (Reuters 2010). The final report, published on 30 September 2010, examines the execution of this large sell order (75,000 contracts) in E-Mini S&P Index futures. It also considers all other market participants and

[1] Eastern Standard Time
[2] U.S. Regulator for stocks and options market
[3] U.S. Regulator for future market

their behaviour (CFTC and SEC 2010a).

This section, mainly based on the official report investigating the 6 May 2010 flash crash, reviews the illegal or questionable behaviours that might have caused the flash crash and their harm to the market. To be able to discuss behaviour and interaction among market participants we need to classify them into categories and discuss their actions as a group. This section reviews the classification of traders by using their trading history information on the day of the flash crash and on days prior to the flash crash on the CME. This classification is used in the official report produced by market regulators (CFTC and SEC 2010a) and is also used by many researchers that have investigated the flash crash. Then it looks into the regulators' initial view of the sequence of events that lead to the crash on that day. Finally, it summarises different views on who was responsible for the flash crash and how they had been involved.

Because of the widespread effect of the flash crash on the market, media, market participants and regulators immediately started investigating to identify the root cause. Initial reports that the crash was caused by a mistyped order was proved wrong. The media, researchers and politicians had been many follow-up discussions and investigations on why this event happened and how it could be prevented in the future. This research examines a number of these suggestions focusing on how these can be modelled and how their effects on the market can be studied before implementing such changes.

Many of the studies suggest specific types of traders or interaction between different types of traders as the cause of the crash or a contributor to its expansion. Agents participate in a financial market with different aims and capabilities which affect their behaviour. The study first examines at the CME market and its participants on the day of the flash crash in Section 2.2.1 to provide further detail on the classification used in the official report by regulators. It is a reasonable proxy to the structure of similar liquid markets. It also reviews some of the behaviours that are considered problematic and harmful that have been blamed to cause or contribute to the problems on the day of the flash crash in Section 2.2.2. The sequence of events that happened during the crash is examined in Section 2.2.3. Finally, the current view of who was responsible for the flash crash is discussed in Section 2.2.4.

2.2.1 Classifying Market Participants

The official investigation report uses market participant categories based on the definition provided by (Kirilenko et al. 2011). It divides market participants into six categories:

- High-frequency traders
- Intermediaries
- Fundamental buyers
- Fundamental sellers
- Noise traders
- Opportunistic traders

It uses trading information on the CME on the day of the flash crash and three days before

the flash crash for this classification. Initially, it splits all market participants into one of two general classes: *market makers* and the rest of the market participants. It assumes market makers would normally be active in the market every day, including the days prior to the flash crash. As a result, any market participant that was active in the market during all three days prior to the flash crash is marked as a market maker. Market makers buy and sell a large number of contracts, but hold a relatively low level of inventory. Market making manifests itself in both a low standard deviation of position holdings and a low ratio of overall net holdings to trading volume. This report then splits market makers into two categories, High-frequency Traders and Intermediaries, based on the data for three days before the flash crash. After removing market participants categorised as market makers, trade data on the day of the flash crash was used to designate traders into other trading categories.

- *High-Frequency Traders* are defined as market makers with very large daily trading frequency. For classification purposes, the top 3% of the market makers sorted by the number of trades were designated as high-frequency traders.

- *Intermediaries* are defined as the market makers who did not fall into high-frequency traders category.

- *Fundamental Buyers/Sellers* are defined as those who were either buying or selling in one direction during the trading day and held a significant net position at the end of the day. They are further separated into fundamental *buyers* and *sellers* depending on both the direction of their trade and the accumulation of their net positions.

- *Noise Traders* are those traders who traded fewer than 10 contracts.

- *Opportunistic Traders* are defined as those traders who do not fall into the other five categories. Traders in this category sometimes behave like the intermediaries (both buying and selling around a target position) and at other times behave like fundamental traders (accumulating a directional long or short position).

Table 2.1 provides statistics for each category of traders based on the above classification. Panel A shows the data for the three days before the flash crash. It is used to split participants to market makers and other participants. Panel B contains the data on the day of the flash crash. It is used to classify the rest of the participants. Note that this classification is only possible with privileged access to proprietary and sensitive data available only to exchanges and regulators. One needs to know the identity of each market participant associated with each trade to be able to make the classification above. Market data that is available to market participants is anonymous, meaning it provides the details of the orders and trades such as price, size, and timing but it does not reveal the identity of the market participant placing that order or making the trade.

Panel A: May 3-5

Trader Type	% Volume	% of Trades	# Traders	Avg Trade Size	Limit Orders % Volume	Limit Orders % Trades
High Frequency Trader	34.22%	32.56%	15	5.69	100.000%	100.000%
Intermediary	10.49%	11.63%	189	4.88	99.614%	98.939%
Buyer	11.89%	10.15%	1,013	6.34	91.258%	91.273%
Seller	12.11%	10.10%	1,088	6.50	92.176%	91.360%
Opportunistic Trader	30.79%	33.34%	3,504	4.98	92.137%	90.549%
Noise Trader	0.50%	2.22%	6,065	1.22	70.092%	71.205%
All	2,397,639	446,340	11,875	5.41	95.45%	94.36%

Panel B: May 6

Trader Type	% Volume	% of Trades	# Traders	Avg Trade Size	Limit Orders % Volume	Limit Orders % Trades
High Frequency Trader	28.57%	29.35%	16	4.85	99.997%	99.997%
Intermediary	9.00%	11.48%	179	3.89	99.639%	99.237%
Buyer	12.01%	11.54%	1,263	5.15	88.841%	89.589%
Seller	10.04%	6.95%	1,276	7.19	89.985%	88.966%
Opportunistic Trader	40.13%	39.64%	5,808	5.05	87.385%	85.352%
Noise Trader	0.25%	1.04%	6,880	1.20	63.609%	64.879%
All	5,094,703	1,030,204	15,422	4.99	92.443%	91.750%

Table 2.1: Market participants (CFTC and SEC 2010a)

2.2.2 Problematic Behaviour

There have been some questionable market participant's behaviours. These behaviours can cause problems for other participants during the flash crash, and some even claim these behaviours have caused the flash crash. In this section, we review some of these behaviours.

Spoofing and Layering

Spoofing is defined as a trader placing a bid or offer on a security with the intent to cancel before execution. *Layering* is a more specific form of spoofing whereby a trader places multiple orders that he does not intend to execute. These fake orders trick other market participants by creating the false impression of heavy buying or selling pressure. Layering is an advanced form of spoofing because it implies there are multiple orders and market participants on one side of the market.

For example, if a trader wanted to buy shares in ABC below the current market price, he could layer three big sell orders above the current price. Assume ABC trades in dollar increments and is currently trading at \$100 x \$101. A trader who wanted to buy 100 ABC shares and used layering could put in a sell order of 2000 shares for sale at \$102, 2000 share at \$103, and 2000 shares at \$104. An algorithm or a trader that is executing a sell order might see these orders, as a selling pressure, and jump in front of them to push the shares price lower. Then the trader would put in a buy order to pay \$99 for 100 shares, a much smaller amount. After the buy order gets executed, the trader would immediately cancel all his large layered sell orders. Spoofing was not specifically made illegal until the 2010 Dodd-Frank bill. It has been claimed that in the past sell-side traders would sometimes use some type of spoofing, especially in illiquid markets. In a security that was more illiquid and didn't trade often, a trader would put a big order above

or below the market to see how the market reacted. In fact, doing this puts him at more risk. Layering exposes the trader to an excessive amount of risk if he miscalculates the pressure or speed of one side of the market. If a trader miscalculates, he can lose tens of millions of dollars in a matter of minutes and can have an uncomfortably large position on his account. In the example above where a trader offers 6000 ABC shares between \$102 and \$104, assume another large broker could receive an order to buy 20000 ABC shares at the market. In a matter of seconds, the market could be trading up to \$104 and the spoofing trader could be short of 6000 ABC shares below the spot price even though he wanted to be long and buy 100 ABC shares instead.

For an exchange or a trading venue, detecting and preventing spoofing and layering is a complicated task. Compared to triggering a circuit breaker which could be automated by a simple comparison between two numbers, detecting spoofing or layering depends on detecting "patterns" in market participant behaviour. Even detecting spoofing off-line using market data is not simple and requires information about the trader who placed the order. For a complex entity like an investment bank, it is plausible that one division of the bank was looking to sell an asset and placed an order to sell and at a later stage strategy decided not to trade and it happened at the same time when another division of the bank placed an order to buy the same asset.

Some of the market participants have built internal systems to prevent such activity from going outside the bank by internalising their order-flow. Internalising the order-flow comes with other benefits such as lower transaction cost and information leakage outside the bank. On the other hand, there have been recent regulatory changes on the internal crossing platforms especially in many of the investment banks which made operating such platforms more regulated and complex which may outweigh the benefit. Either way, it is a complex problem to decide internally within a market participant, *e.g.* a large investment bank on how to deal with opposing requirements inside the system and to not to have a behaviour that when seen from outside gives others the "impression" that it was intended for wrong reasons.

Self-crossing

When a market participant places orders on both sides of the orderbook that trade against each other it is called "self-crossing". Self-crossing on the surface is not logical as that participant makes no gain on the trade. It can also cause confusion for other market participants as it produces the impression of the market trading volume where it may not exist. This behaviour is prohibited in some markets. Although it looks meaningless on the surface, it could be that a complex market participant, *e.g.* a large investment bank has different businesses with different trading systems placing these two opposing orders that traded against each other in the market.

Smoking

Market data is an important source of information about the available liquidity and flow. Many strategies use this information to decide about controlling their order placement. If

this information is ambiguous, delayed or stopped, it can cause harm to their trading efficiency. Not all market participants have the same capabilities to handle market data. Limitations on bandwidth and latency caused by physical distance, equipment, providers connecting them to the market, and processing power or technical capability can put some participants at a disadvantage. This disadvantage is especially important when the market is very volatile, which in turn increases the rate of market change and market data produced. It has been claimed that some participants place and cancel orders at short intervals not with the intention of trading but to produce a high volume of market data that delays or confuses other participants, and then benefit from their wrong order placement. This behaviour is called *Smoking* as it limits the visibility of other participants of what is going on in the market.

Hot-Potato Effect

Hot-Potato refers to a situation when a specific security circulates rapidly among market participants. This effect is especially common when the market is moving in a direction against what market makers have predicted, and there are no other participants to trade with. For example, when the market is moving down during a flash crash and there are no fundamental buyers, market maker A that already holds a short position becomes aggressive and trades with another market maker B that is still trading normally. Shortly after trading, market maker B realises it cannot hold onto this position. It becomes aggressive trying to trade out of the position, and passes this short position to market maker C. Market maker A, now flat, is ready to trade on both sides of the market and C may pass the position back to A. This behaviour has two negative effects on the market. This set of trades provides no gain for market makers as a whole. It also negatively affects other market participants as it produces a high market trading volume which in a normal scenario could be interpreted as a liquid market but is actually quite the opposite.

2.2.3 Sequence of Events

The SEC and CFTC reports suggest the sequence of events causing the flash crash happened as follows:

- Due to unsettling political and economic news about the European debt crisis, *volatility* was unusually high, and *liquidity* was thin during the hours before the flash crash.

- The main trigger for the sudden decline was a large sell order in S&P 500 index E-Mini futures by a mutual-fund group. Because this automated algorithmic trade was programmed to take account of trading volume, not price or time, it was executed unusually rapidly in 20 minutes instead of the several hours that would be typical for such an order.

- High-frequency traders initially helped to absorb the selling pressure, buying E-mini contracts. High-frequency traders usually get in and out of positions quickly, sometimes

holding them for less than a second. Initial sell pressure pushed many of them into holding long positions.

- Ten minutes later, however, they began forcefully selling to reduce their long positions. High-frequency traders rapidly passed the same positions back and forth creating a *Hot-Potato* effect that generated lots of trade volume but little net buying. Traditional buyers were unable or unwilling to step in, and the depth of the buying market for E-minis and S&P 500-tracking ETFs fell to a mere 1% of its level that morning.

- The sell algorithm used by the mutual fund responded to this increased volume by increasing the rate at which it fed orders into the market, even though orders that it had already sent to the market were arguably not yet fully absorbed by fundamental buyers or cross-market arbitrageurs. This response created a positive feedback loop.

- The initial liquidity crises in index futures were expanded to individual stocks in the index by market arbitrageurs.

- Some market makers reacted to this increased risk by widening the spreads between the levels at which they would buy or sell, others withdrew completely, and some resorted to manual trading but could not keep up with the explosion in volume. As price declined, *retail order flow* that is normally internalised by market makers was forwarded to the market. *Stop-loss* orders were triggered, adding to the selling pressure.

- New York Stock Exchange stopped trading briefly while other exchanges and alternative trading venues kept going. As NYSE was the primary market for many of the stocks traded, it caused some of the trading systems to stop trading or it put them on alert on the suspension. These pauses and alerts involved human intervention in most cases and slowed down the speed of decline allowing the market to catch up.

High-frequency traders, who have been blamed by some for the collapse in liquidity, were net sellers at this time, but so were most other participants. However, some high-frequency traders continued to trade throughout the crash, even as others reduced or halted trading.

2.2.4 Who was responsible

The official report suggests that the flash crash was triggered by a large order sent by a market participant that was executed unusually fast with no regard for price. It blames high-frequency traders for exacerbating the crash but not for starting it. It is worth pointing out that the above points were the conclusion of the official investigation report. There are researchers and market participants that do not fully agree with this conclusion. Some recent research argues that flash crashes are not isolated occurrences, but have occurred quite often over the past century.

Nearly five years after the flash crash, on 21 April 2015, the U.S. Department of Justice filed charges against Navinder Singh Sarao, a London-based trader, on 22 criminal counts, including fraud and market manipulation for his alleged role in the flash crash. According to criminal

charges, Sarao and his company, Nav Sarao Futures Limited, used an automated program to generate large sell orders, pushing down prices, which he then cancelled to buy at the lower market prices. This practice is called *spoofing* and is illegal. Just prior to the flash crash, he placed orders for thousands of E-mini S&P 500 stock index futures contracts which he planned on cancelling later. These orders amounting to about "$200 million worth of bets that the market would fall" were "replaced or modified 19,000 times" before they were cancelled. The CFTC investigation concluded that Sarao "was at least significantly responsible for the order imbalances" in the derivatives market that affected stock markets and exacerbated the flash crash. Sarao began his alleged market manipulation in 2009 with commercially available trading software whose code he modified "so he could rapidly place and cancel orders automatically."

Some people argued that blaming a 36-year-old small trader who worked from his parents' house in suburban west London for sparking a trillion-dollar stock market crash is like *"blaming lightning for starting a fire"* and that the investigation was lengthened because regulators used *"bicycles to try and catch Ferraris."* (Bates 2015)

2.3 Other Market Disruptions

The flash crash on 6 May 2010 was not the only time that trading was disrupted in a short period of time because of the issues with automated trading systems and strategies. There have been many events in recent years with similar issues. Section 2.3.1 looks into a few examples of flash crash where the price has moved quickly and recovered to the same level without a clear fundamental driver. It then reviews two other types of market disruptions that share some but not all of the properties of a flash crash. These market disruptions are of interest for two reasons. These types of disruptive events have mostly produced large market moves over a short period of time similar to a flash crash and affected investors in different ways. Regulation changes that are proposed to prevent flash crashes can have adverse effects on the ability of market participants to handle such scenarios efficiently. As a result, it is imperative to understand these types of events and the effect that market regulation changes can have on the market participants. Some of the market crashes and disruptions are caused by technical or control failure by a single market participant or exchange which is reviewed in Section 2.3.2. But, there are market crashes that are caused by real economic factors which are discussed in Section 2.3.3.

2.3.1 Flash Crashes

In recent years, there have been a number of flash crashes reported that have similar properties to the flash crash on 6 May 2010. In this section, we look into some recent examples and their possible causes.

India's NSE

On 5 October 2012, India's National Stock Exchange (NSE) Nifty Index lost 900 points (over 15%) in a matter of seconds. Almost sixty billion dollars were temporarily wiped from the stock

market value of India's biggest companies, listed on the NSE in Mumbai (Reuters 2012). The stock exchange was forced to pause trading for a while. The NSE claimed that a human error caused the crash (NSE 2012).

ETF Crash

On 10 August 2015, Exchange Traded Funds (ETF) markets observed irregular volatility. Minutes after trading started, some ETFs and stocks sank 30% or more from the previous day's closing level. Prices quickly recovered later in the day after many investors had sold at deflated prices. Trading in 327 ETFs was halted (CNBC 2015). ETFs are securities that are designed to represent a specific mixture of underlying assets and issuers are required by regulation to maintain a very close set of underlying assets that they have sold to their customers. Exchanges halt trading when stock and ETF prices move violently. But different exchanges do not necessarily rely on the same rules in deciding when and how to pause trading. As a result, when trading in an ETF is halted by the exchange but trading in the underlying securities continues, the issuers are in a very difficult position to match their underlying portfolio, which in turn contributes to moving the underlying security if there is not enough liquidity, indirectly affecting the price of the ETF. After the event, a group of top ETF issuers, traders and other financial firms issued a public letter to the Securities and Exchange Commission saying the industry had reached a consensus that the markets are susceptible to similar events occurring at any time. The group argued that inconsistent rules between exchanges governing when stocks and ETFs are halted and when trades declared invalid "contributed" to the trading turmoil. They asked the SEC to intervene to bring those rules in sync (Reuters 2016). The SEC approved new rules designed to protect mutual fund investors from the effects of a sudden sell-off. Under the new rules, funds have to classify investments into the categories of highly liquid, moderately liquid, less liquid and illiquid. The rules also exempt *in kind* ETFs, those that honour redemption in securities instead of cash, from requirements on how many highly liquid and illiquid assets they can hold. The new rules still require funds to keep a certain level of highly liquid assets that can be converted into cash in three days.

Bank of America Merill Lynch

On 26 September 2016, the SEC reported that its investigation found that *Merrill Lynch* caused market disruptions on at least fifteen occasions from late 2012 to mid-2014 causing "mini-flash crashes" (SEC 2016). According to the SEC, the erroneous orders that passed through Merrill Lynch's internal controls caused certain stock prices to plummet and then suddenly recover within seconds. Among the mini-flash crashes were 99-percent drops in the stocks of Anadarko Petroleum Corporation on 17 May 2013, and Qualys Inc. on 25 April 2013, and a nearly 3% decline in Google's stock in less than a second on 22 April 2013. Their internal controls in place to prevent erroneous trading orders were set at levels so high that in practice they were ineffective. For example, Merrill Lynch applied a limit of 5 million shares per order to one stock that only traded around 79,000 shares per day. Merrill Lynch had agreed to pay a $12.5 million

penalty for maintaining ineffective trading controls that failed to prevent erroneous orders from being sent to the markets.

E-Mini Stop-Orders

On 7 December 2016, multiple buyers purchased around 16,000 contracts of *E-mini S&P 500* valued at $1.8 billion (Wall Street Journal 2016). It was the biggest E-mini trade by more than a factor of two in 2016. The sequence of trades at new highs caused a sharp market rally for the rest of the day and the two succeeding days. It has been reported that these trades were caused by a series of stop orders triggered by a single contract trading at $2225.00. The contracts traded as stop-loss orders, and traded all at the same nanosecond. Stop-orders are placed by investors to limit their loss when the market moves far away from what they expect. One of the problems with these orders is that they normally activate simultaneously just as a price passes a specific point. If the market gets very close to that limit point but does not cross it, they do not activate. Another issue is that because the limit price is commonly set by a human, it tends to be a round number.

2.3.2 Technical Glitches

There have been recent market crashes that were not caused by trading strategies but purely by technical problems with the trading platform. This section reviews some of the most famous examples of such technical issues.

Facebook IPO on Nasdaq

On 18 May 2012, Facebook had its Initial Public Offering (IPO) on Nasdaq. It was valued at $16 billion and was one of the biggest and most anticipated events in technology companies in years. However, the Facebook IPO turned into a catastrophe. The pricing of the first transaction took a half hour longer than planned. There were delays in trade confirmations. The order-book had crossed quotes, and some orders were not handled as expected. About 30 minutes after the first transaction, Nasdaq reported an issue confirming trades from the opening auction. Although in the minutes after Facebook's IPO issues executives at the Nasdaq stock exchange received an e-mail pleading for a trading pause. The stock exchange decided to proceed with trading (Bloomberg 2012b). The problem had been caused by errors in Nasdaq's software but because exchange decided to continue trading many of the market participants were left with unconfirmed trades or orders. Nasdaq later established an appeals process for investors whose instructions weren't carried out. The SEC also conducted an investigation into the incident. The report pointed out concerns about the design of the system and the response of exchange officials. Nasdaq's parent company had to pay the largest fine ever levied against an exchange for poor systems and decision making (New York Times 2013).

Knight Capital

On 1 August 2012, trading errors happened at Knight Capital, one of the largest market makers and high-frequency traders in the US equities market. The problem was caused by the release

of software to enable a new feature for the Retail Liquidity Program in the New York Stock Exchange. It caused a piece of program to be deployed into production that was intended to be used only in a test environment for controlled testing purposes. The incident lasted only about 45 minutes. It caused the firm to send excessive orders in stocks of 148 companies leaving Knight Capital with a large portfolio before this program stopped (SEC 2013). The trading incident caused the company to take a pre-tax loss of $440 million (Bloomberg 2012a). Knight Capital Group share prices lost almost 70% as a result and sent it to the brink of bankruptcy. This event almost put the market making firm out of business but a group of Wall Street firms led by Jefferies later rescued the company with a cash infusion. Knight Capital Group was acquired by Getco LLC in December 2012.

Goldman Sachs

On 20 August 2013, a programming error at Goldman Sachs caused unintended stock-option orders to flood the U.S. Option markets. An internal system that Goldman Sachs used to help prepare to meet market demand for equity options inadvertently produced orders with inaccurate price limits and sent them to exchanges (Bloomberg 2013). The SEC investigation found that Goldman Sachs did not have adequate safeguards after the firm implemented new electronic trading functionality designed to match internal options orders with client orders. A software configuration error inadvertently converted the firm's "contingent orders" for various options series into live orders and assigned them all a price of $1. During the first 15 minutes of opening the market 405 out of the 500 biggest options trades were triggered for tickers starting with H through L and priced at $1. Almost 130 of those were in 1,000-contract lots. As an example, about 240 September $103 put contracts for the iShares Russell 2000 Exchange-Traded Fund traded at $1 at 9:32, down from as much as $3.32 two minutes earlier. The next trade was executed at $3.27 at 9:33. Many of the executed trades were later cancelled or received price adjustments according to the options exchanges' rules on *clearly erroneous trades* (SEC 2015).

The SEC's report highlighted a number of issues. The firm employed unreasonably wide price checks for its options orders during pre-market hours. An employee lifted several electronic circuit breaker blocks that automatically shut off outgoing options order messages once the rate of messages exceeds a certain level. The policies regarding these circuit breakers were not properly disseminated or fully understood by employees with responsibilities relating to the circuit breakers. The bank did not maintain adequate controls designed to prevent the entry of orders that exceeded the firm's capital threshold. The firm computed its capital usage level only every 30 minutes, and did not have an automated mechanism to shut off orders if the firm exceeded its capital threshold. It failed to include a number of business units in the firm's capital utilisation calculation, thereby underestimating the firm's trading risk. The SEC charged the bank with violating the market access rule in connection with the trading incident. Goldman Sachs agreed to pay a $7 million penalty to settle the charges. For Knight Capital, a programming error cost the firm its own existence. Goldman Sachs, on the other hand, said the

Figure 2.2: EURCHF rate change after SNB decision (The Telegraph 2015)

error "would not be material to the financial condition of the firm."

2.3.3 Real Market Crashes

Previously, we reviewed a number of market disruptions that were caused by a technical problem in software or unexpected behaviour of a trading strategy or unforeseen interaction between different trading strategies. However, there have been a number of situations where there has been a sharp market move that has been caused by a fundamental economic factor. In many cases, the market anticipates the change in the economic future and incorporates that information into the pricing of the asset resulting in the longer term a slower movement in market prices. However, there have been recent events that turned out completely opposite to general market expectations and caused a sudden market move. These cases were based on real and fundamental economic parameters but at the same time resulted in an outcome with many similarities to flash crashes previously discussed in this chapter. This section reviews some of these events.

SNB Rate Decision

During a financial crisis, the Swiss National Bank (SNB) put in place[4] a cap on the value of the Swiss Franc to the Euro. Specifically, it wanted the keep the value of the Franc below 1.2 to the Euro. On 15 January 2015, SNB announced it would stop the Euro currency cap placement. Although this possibility had been discussed before by market participants, it was not expected that the cap would be removed suddenly with no prior indication (Reuters 2015).

As shown in Figure 2.2, the market moved sharply following the announcement of the decision. In this case, the market did not recover to the same level, but at the same time, it triggered some of the controls placed on different banks' internal infrastructure that were

[4] As of 6 September 2011

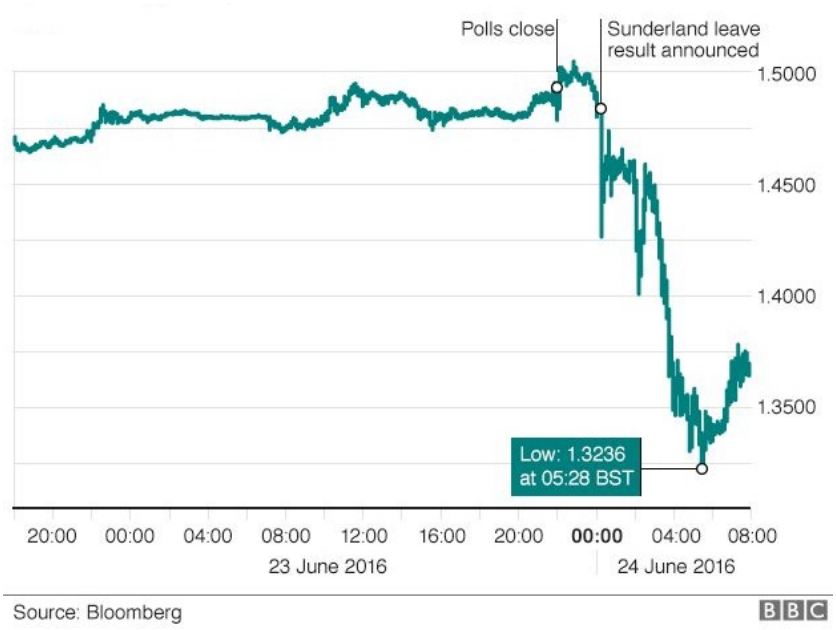


Figure 2.3: GBPUSD rate on Brexit referendum (BBC 2016)

designed to prevent situations like a flash crash. As a result, market participants who could not handle the situation quickly accumulated large losses. The FX market is not as regulated as stocks or futures markets and trading happens quite differently. Most of the trading is Over-The-Counter (OTC), meaning it is traded and settled between two counter-parties without an exchange. As a result, available information is much more limited on the effect of the SNB decision on the market participants.

Brexit

On 23 June 2016, the UK government held a referendum to ask the UK electorate if they wanted to "Remain" in the European Union or wanted to "Leave". The poll during the days before the referendum showed a very close competition. The FTSE 100 index and pound were affected by close opinion polls during the campaign. In the last week of referendum market sentiment showed growing confidence in a vote to remain in the EU. The FTSE 100 rallied every day during this week before the referendum was up 5.3%.

Before the results started to come in, the pound had risen, as traders bet on a Remain victory. Just after polling stations closed it moved as high as $1.50. But following strong Leave votes in north-east England, it tumbled to $1.43 reversing initial gains (The Guardian 2016a). It took another dive to $1.33 after 03:00 BST as Leave maintained its lead (see Figure 2.3). The final result saw, 51.9% of the participating UK electorate voting to leave the EU; the turnout was 72.2% of the electorate. It pushed the pound down 10% against the dollar. The pound was also down more than 7% against the euro. Shares were also hit. The FTSE 100 index began falling by more than 8%, then recovered some losses to close at 2.5% lower. Some traders said these moves were more extreme than those seen during the financial crisis of 2008. The pound collapsed to its lowest level in over 30 years. It also suffered its biggest one-day fall in history

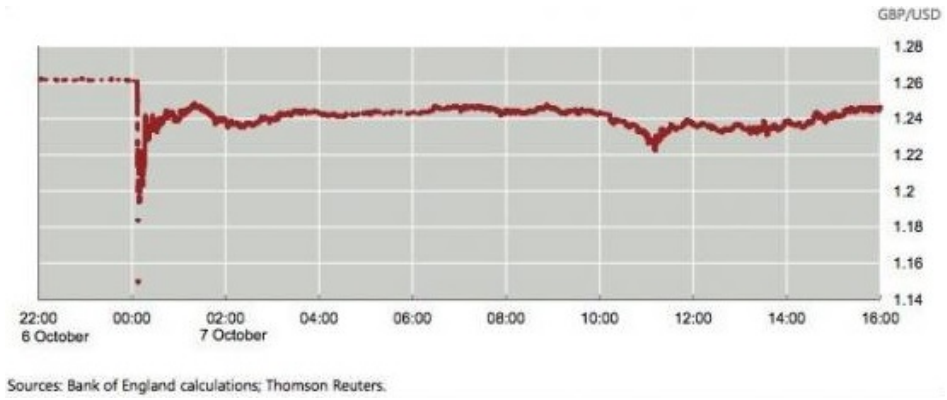


Figure 2.4: GBPUSD Crash on 7th October (BBC 2016)

as panicking investors contemplated the prospect of a vote to leave the European Union.

Uncertainty following Brexit caused more volatility and is no clarity if the follow up incidents were caused by real economic concerns or technical issues. On 7 October 2016, the value of the pound fell from \$1.26 to \$1.14 just after midnight in about 40 seconds before bouncing back to \$1.24 (see Figure 2.4). The pound fell by 6%, having dropped by 10% before one of the more outlying major trades was cancelled. Traders were baffled by the overnight *flash crash* in the pound, with theories emerging that pointed to possible causes including rogue computer trades, an accidental "fat finger" transaction and tough comments from the French president, Francois Hollande, on Brexit negotiations (The Guardian 2016b).

The Bank of England's Prudential Regulation Authority (PRA) launched an investigation into the crash. It announced that the October 7 crash was "set apart by the lack of a clear fundamental trigger". Media reported that PRA was not particularly concerned by the initial trigger, but focused heavily on the second stage of the slide. It coincided with a large number of rapid-fire sell orders placed in Tokyo by Japanese trading operations of Citigroup (Financial Times 2016). Citi's traders are not believed to have started the slide in the currency in thin Asia trading but its Tokyo desk played a key role in sending the pound to its lowest levels in 31 years. Some claim the crash started by an algorithm that triggered the sell after it picked up Twitter-reading traffic on the comments made by the French president, Francois Hollande. People with knowledge of events at Citi that day said one of the US bank's traders placed multiple sell orders when the currency slumped in unusually fragile market conditions.

Citigroup has been forced to defend its trading business after it emerged as the world's largest foreign exchange bank may have contributed to the flash crash. "Sterling fell sharply following a news event just after midnight UK time, when the GBP spot foreign exchange market was extremely illiquid. Citi managed the situation appropriately and our systems and controls functioned throughout the period." a spokesman for Citi said (The Telegraph 2016). The "news event" referred to by Citi was reports that Francois Hollande, France's president, wanted European leaders to take a firm negotiating stance with the UK over Brexit and said "if Theresa May and company want *hard Brexit*, they will get hard Brexit."

However, the Bank of England's Financial Stability Report, which examined the flash crash, concluded that "while the story may have acted to reinforce the negative pressure on sterling, it was not the initial trigger". The Bank for International Settlement (BIS), an umbrella research organisation for central banks, said that a number of factors contributed to the crash but found no evidence of deliberate manipulation or "fat finger" mistakes by traders (The Bank for International Settlement 2017). These factors included: significant sell orders for sterling from traders; automatic stop-loss orders; a report containing negative news for sterling; and inexperienced traders working at that time of night in Asia with lower risk appetite. The BIS concludes that the October crash was not a unique or unprecedented event but a new data point in what appears to be a series of flash crash events occurring in a broader range of fast electronic markets than was previously the case in the post-crisis era (Independent 2017).

2.4 Proposed Solutions to Prevent a Flash Crash

Since the 2010 flash crash, there have been further incidents that have renewed concerns about the safety of markets in a time when a large part of trading is automated. These incidents called into question many of the regulatory and technological changes over the last decade, which facilitated an era of high-speed trading on electronic exchanges and alternative venues. The regulators point to a number of lessons to be learnt. In times of turmoil, automated trading can trigger extreme price swings, especially if the algorithm does not take account of prices. The way in which these automatic orders interact with high-frequency and other computer trading strategies can quickly erode liquidity, even amid very high trading volume. More work also needs to be done to understand how stock-markets and derivative markets interact, especially with respect to index products.

US Lawmakers used the flash crash investigation report to put pressure on regulators to do more to revise market rules. Some suggested that if necessary, Congress must "put in place new rules of the road to ensure the fair, orderly and efficient functioning of the U.S. capital markets." The SEC has brought in uniform policies for cancelling trades struck at clearly irrational prices. It eliminated "stub quotes", which allowed market makers to buy stocks for a penny if there were no other bids. In February 2011 an advisory panel convened to investigate the flash crash recommended rule changes that would oblige high-frequency traders to maintain orderly markets and limit brokerages' ability to execute trades internally. It also recommended the permanent adoption of circuit breakers to halt trading temporarily in an individual security if price or volume movements caused concern. Another area that has been investigated is market data.

UK Government Office for Science sponsored a project to study the "future of computer trading in financial markets" (UK Gov 2012) as part of *Foresight* Programme. The Foresight Programme, over the long term, looks at major cross departmental, multidisciplinary projects in key areas of policy. This project examined the technological advances which have transformed market structures in recent years and explored how computer trading will evolve over the next ten years. It acknowledges that the volume of financial products traded through computer

automated trading taking place at high speed and with little human involvement has increased dramatically in the past few years. Over one-third of UK equity trading volume is generated through high-frequency automated computer trading while in the US this figure is closer to three-quarters.

The studies by the regulators, government agencies, and academic researchers have proposed different solutions to prevent events similar to the flash crash. Mechanics of trading in a market and its trading rules affect its viability and stability as a trading venue. Some of the studies focus on information asymmetry and the effects of fundamental traders and high-frequency traders on market stability. They argue that directional order flow from informed traders causes noise traders and other uninformed traders to leave the market. This departure, in turn, leaves market makers trading with informed traders with positions in which they cannot find any counter-party to take the other side. Some other studies focusing on high-frequency traders argue about the problems these traders cause or benefits they bring to the market especially their effects on market volatility.

A large amount of study by regulators has focused on trading rules and has proposed some measures that need to be considered to prevent events similar to the flash crash. One can view trading rules on the exchange as the equivalent of rules of the road. These rules aim to ensure there is orderly behaviour in the market and one market participant intentionally or by mistake does not damage the experience of other market participants. Some of the frequently proposed measures include the use of circuit breakers, limit minimum-quote-life and order-to-trade ratio. These are somehow similar to driving measures on the road. A *circuit breaker* is an intuitive way to limit the damages of a rapid market crash by stopping the market from trading and preventing further damage. Another important tool is speed limit control that aims to limit the general speed of the market and reduces the chance of a rapid crash in the market and controls its effect and spread. This is similar to speed control on the road that brings down the speed of traffic to reduce the risk of accidents. *Minimum quote life* acts as a speed limit control and places a limit on how fast a market participant can change its order in the market. A softer version is *order-to-trade ratio* and is equivalent to variable speed control in the highways. Instead of measuring and monitoring constantly, it will measure the speed over a longer range to make sure the overall speed stays below a safe limit. It is notable that this measure does not control or punish minor deviation from the limit if it is compensated by slower than limit speed within two measurement points.

In the last few years, there has been a move towards fragmented trading in financial markets. While the costs and benefits of this market structure change are beyond the scope of this research, the move towards having multiple trading venues will likely have a meaningful impact on the future of trading. Some markets already have restrictions on placing and cancelling orders. But when there are competing market venues with different rules, such as whether to have a minimum quote life or a maximum ratio of quotes per trade, investors will be able to

choose for themselves the market they believe provides the best trade execution and service.

2.4.1 Limit High-Frequency Trading

A recent study suggests high-frequency traders participate in more than 60% of US equity trading. This trend is closely followed in European equity markets and other highly liquid trading venues like Futures. This heavy presence has caused concerns that any problem with high-frequency traders can have a dramatic effect on other financial market participants. There have been discussions that high-frequency traders pose a risk to the stability of the market. Their technological advantage is seen by some participants to make markets unfair. The official flash crash investigation report concluded that high-frequency traders did not cause the flash crash, but contributed to it by demanding immediacy ahead of other market participants. Most individual investors do not fully understand how high-frequency trading works but are concerned by their technological advances. Some politicians and market participants see the flash crash as confirmation that high-frequency trading is dangerous. In response, the high-frequency traders point out that the algorithm at the centre of the story was not executed by a high-frequency trader, but by a standard mutual fund.

One of the things that brought high-frequency traders into the centre of attention for flash-crash was the "Flash Boys" book by Lewis. The New York Times Best Seller (New York Times 2015) book focuses on the rise of high-frequency trading in the US equity market and states that "The market is rigged" by high-frequency traders who *front-run* orders placed by investors. Bradley Katsuyama, a trader, is the main character of Flash Boys. The speed of data is a major theme in the book; the faster the data travels, the better the price of the trade. Flash Boys mentions the construction of Spread Networks', a $300 million project of fibre optic cable running through mountains and under rivers, to connect the financial markets of Chicago and New York. This link reduced the latency of data from 17 to 13 milliseconds. Lewis claims that access to this fibre optic cable, as well as other technologies, presents an opportunity for the market to be controlled by the big Wall Street banks. The book concludes by observing that there is now a new microwave link between Chicago and New Jersey, which follows an even straighter route than the Spread Networks' 827-mile cable. Microwaves always follow a direct path, whereas cables must, at least occasionally, detour around physical barriers. The new route also takes advantage of the faster speed of signal travel that is possible through the air compared to signal travel speed through glass fibres. With these two advantages, this new link is 4.5 milliseconds faster than the Spread Networks.

A number of solutions have been proposed to control the speed of high-frequency traders and to ensure that this does not give them an unfair advantage over other market participants.

Katsuyama co-founded a new exchange, IEX (the Investors Exchange) that opened for trading on 25 October 2013. To counter this speed disadvantage to investors, it aims to make the playing field for trading fairer. To overcome the latency advantage, it has its matching engine located in Weehawken, New Jersey. While the initial point of presence is located in a data centre

in Secaucus, New Jersey, it places a 38-mile coil of the optical fibre in front of its trading engine. This link adds a 350 microseconds delay, a round-trip delay of 700 microseconds (0.0007 seconds), and is designed to negate certain speed advantages utilized by some high-frequency traders.

2.4.2 Circuit Breakers

Assuming electronic trading venues are equivalent to the highways in a transport network. The first measure, traffic control, is applied when something seriously bad happens. The traffic movement is shut-down until the police arrive and once the situation is understood and resolved then the traffic is allowed to move. This solution is called a *circuit breaker* and will stop trading in the market. A circuit breaker is an intuitive way to limit the damages of a rapid market crash by stopping the market from trading and preventing further damage. This trading halt allows slow market participants to catch up with the current status of the markets and gives human traders time to react and adjust or stop such strategies.

Circuit breakers are one of the main tools proposed to prevent a repeat of the flash crash. Triggering a circuit breaker is an extreme measure that imposes a heavy cost on ongoing traffic but at the same time could be considered most effective as one can think nothing bad is going to happen after this point. Unfortunately, triggering circuit breaker in a connected world that has high-speed connectivity may not always be the measure. Similarly, if one route going in the direction of the problem is closed that blocked the road, but the rest are still working, actually makes the problem worse by redirecting extra traffic to already overloaded paths. It has been claimed that the circuit breaker trigger at NYSE was a positive tool to cool down the pressure during the 2010 Flash Crash. Others argue that although official trading pauses can be a good way to provide time for sanity to return to markets, uncoordinated breaks can do more harm than good. On 6 May 2010, the New York Stock Exchange stopped trading briefly while other exchanges and alternative trading venues were kept going. This led to a diversion of order flows that added to the pressure on those markets. The SEC has since introduced "circuit-breakers" for individual shares that halt trading across all markets.

2.4.3 Minimum Quote Life

Minimum Quote Life (MQL) can be forced by the exchange to set a lower limit on the time an order has to stay in the orderbook before the participant can cancel or modify the order. For example, if an exchange set MQL on an instrument to be 50 milliseconds, the participant sending an order at time $T0$ cannot cancel this order or modify it before $T0 + 50ms$. This is to make sure other market participants have at least 50 milliseconds to receive this order via their market data platform, process the market data update in their trading strategy and react to this order if necessary by placing new orders or modifying their existing orders before this order is modified or cancelled by the participant. This still does not protect market participants against other faster participants that can receive and process this market data about the order and are likely to take action by hitting that order or changing their own orders before slower participants have a chance to react.

2.4.4 Order-to-Trade Ratio

Another proposed solution is to impose a maximum order-to-trade ratio. Benefits of this solution are to increase the likelihood of a viewed quote being available to trade, and to reduce hyper-active order book participation. Moreover, this will align quoting activity with actual trading in the market. Many exchanges do not charge market participants for sending messages but charge a transaction fee on trades. Each order message processed by a trading venue incurs a cost as it requires a small amount of energy and computing power to process. Trading venues usually carry large amounts of excess capacity as message activity can peak at several times the rate of normal traffic flow. As a result, they need to heavily invest in high-performance computing systems in order to process messages quickly. In addition to the cost incurred by trading venues, other market participants that follow intraday market activities must invest more in their computing systems and communication channels as the level of traffic increases. A quote-to-trade maximum requires market participants to pay for some of their message activity cost in trading fees.

But these benefits do not come for free. The potential costs associated with this approach are: reducing depth, and increasing bid-ask spreads, and exacerbating liquidity withdrawal in volatile times. Whereas the minimum time constraint would apply to every quote, the order-to-trade ratio allows for greater flexibility. Given that much of the time, the order-to-trade ratio will be a non-binding constraint, the depth and bid-ask competitiveness will not be affected. However, when market participants near the maximum quote-to-trade ratio they will likely be more cautious about placing quotes as they will be penalized if they withdraw too many quotes. This hesitance is most likely to occur in volatile times. UK Government sponsored research concludes that not enough is currently known to determine whether the benefits will outweigh the costs of any of the proposed solutions above and, more empirical data needs to be collected and analysed (Brogaard 2011).

2.5 Studies of the Flash Crash

This section reviews previous research on the analysis of the flash crash and the application of agent-based modelling into the flash crash. It examines studies of the flash crash from three aspects. Some of the previous work has focussed on detecting or predicting the flash crash before it happens. Another aspect that is frequently investigated is the role of high-frequency traders in the market and their involvement in the flash crash. And the last aspect it reviews is the modelling of the flash crash.

2.5.1 Predicting a flash crash

There have been studies on the indicators that could have forecasted the flash crash. One well-known study introduces *Volume Synchronised Probability of Informed Trading (VPIN)* (Easley et al. 2010) that is designed to indicate flow toxicity. Generally, if a very high portion of incoming order-flow to the market is from informed traders, and there are no other market participants,

market makers will be forced to trade at a loss as they will not be able to find any other participant with whom to trade out of their position. There have been arguments about this measure. Studying it over a longer time, Andersen and Bondarenko claim that it has reached high levels in the past without causing a crash (Andersen and Bondarenko 2014). Authors of VPIN have responded to the study defending their work (Easley et al. 2014).

Aldridge also claims that flash crashes have been frequent and their causes predictable in market microstructure analysis (Aldridge 2014).

2.5.2 The role of high-frequency trading

Research by Jacob Leal et al. uses a model that does not borrow zero-intelligence based agents from other works. It uses an agent-based model for low and high-frequency traders (Jacob Leal et al. 2015). Low-frequency traders work in chronological time, *i.e.* their activity is exogenous and constant over time. For each agent in simulation, the trading frequency is decided using a uniform distribution between θ_{min} and θ_{max} minutes. On the other hand, high-frequency traders work in event time. Working in event time means high-frequency traders become more active when the market is heavily traded. Agents submit buy or sell limit orders with equal probability $p = 0.5$. High-frequency traders adopt directional strategies that try to profit from the anticipation of price movements. To do this, high-frequency agents exploit the price and order information released by low-frequency agents. At the beginning of each trading session t, active low-frequency and high-frequency agents know the past closing price as well as the past and current fundamental values. In each step the trading proceeds as follows:

- Active low-frequency traders submit their buy/sell orders to the limit order-book market, specifying their size and limit price.
- Knowing the orders of low-frequency traders, active high-frequency agents start trading sequentially and submit their buy/sell orders. The size and the price of their orders are also listed in the order-book.
- Low-frequency and high-frequency agents' orders are matched and executed according to their price and then arrival time. Unexecuted orders rest in the order-book for the next trading session.
- At the end of the trading session, the closing price is determined. The closing price is the maximum price of all executed transactions in the session.
- Given the closing price, all agents compute their profits and low-frequency agents update their strategy for the next trading session.

The research concludes that high-frequency traders front-run other market participants and they add to market volatility.

Another research by Hagstrmer and LarsNordn aims at distinguishing different high-frequency trading strategies and categorising them to *market-making* high-frequency traders and *opportunistic* high-frequency traders (*e.g.* arbitrage and directional). It uses a proprietary dataset that allows them to observe all limit order submissions, cancellations, and

executions, complete with the identities of the traders from NASDAQ-OMX Stockholm. It carries out the analysis of one highly volatile month (August 2011) and one relatively calm month (February 2012). The study finds that market makers constitute the lion's share of High-Frequency Traders trading volume (63-72%) and limit order traffic (81-86%) (Hagstrmer and LarsNordn 2013). It also reports that market-making high-frequency traders have higher order-to-trade ratios and lower latency than opportunistic High-Frequency Traders. It uses tick size changes as exogenous events to study high-frequency traders' influence on short-term volatility. On European stock exchanges, the tick size (minimum price increment) depends on the stock price level. For example, when the price of a stock increases from SEK[5] 49 to SEK 51, the tick size increases five fold from SEK 0.01 to SEK 0.05. The study hypothesizes that an increased tick size makes market making more profitable, and other strategies, such as arbitrage trading, more costly. This is because market makers typically earn the spread, whereas opportunistic traders tend to pay the spread. Nordic-OMX Stokholm introduced several changes that facilitate the activity of high-frequency traders by cutting both the cost and the latency of trading. The changes included central counter-party clearing (October 2009), capped trading fees (January 2010), and the INET trading platform (February 2010). A natural experiment based on tick size changes found that the activity of market-making High-Frequency Traders mitigated short-term intraday price volatility.

Hanson utilized an agent-based market simulation using a paradigm of zero-intelligence traders to examine the impact of high frequency trading on various aspects of the stock market. He adapts the model of (Gode and Sunder 1993), a continuous double auction setting, to include algorithmic high-frequency traders who retrade by marking up their shares by a fixed percentage and examines the effects the number of HF traders and their markup percentage on traded volume, market efficiency, trader surplus and volatility. He concludes that all observed properties vary directly with the number of high-frequency traders. Results also reveal that market volatility increases with the number of high-frequency traders (Hanson 2016).

2.5.3 Modelling of the flash crash

To be able to analyse effects of proposed changes to market structure and trading rules and also reason market behaviour using different new conditions, one needs to be able to either simulate a financial trading market or to model it fully analytically.

Many important economic processes are complex in the sense that it is difficult to decompose them into separate parts that can be studied in isolation and aggregated to give the whole picture. Furthermore, economic agents do not seem to possess in reality the perfect rationality and computation abilities that the classical economic theory attributes to them. Issues of beliefs and prediction about the future become important and the individual beliefs and choices when aggregated shape the economic indicators, the market prices and ultimately the world that the agents must deal with.

[5]SEK is the abbreviation for the Swedish currency Krona

Table 2.2: S&P 500 market participation description (Kirilenko et al. 2011)

Trader Type	# of Traders	Trade Speed	Position Limits	Market Volume
Small	6880	2 hr	-30 - 30	1%
Fundamental Buyers	1268	1 min	$-\infty$ - ∞	9%
Fundamental Sellers	1276	1 min	$-\infty$ - ∞	9%
Market Makers	176	20 sec	-120 - 120	10%
Opportunistic	5808	2 min	-120 - 120	33%
High-Frequency	16	0.35 sec	-3000-3000	38%

There has been a study to analyse the flash crash using agent-based modelling (M. Paddrik et al. 2012). The study uses trader agent types defined by SEC/CFTC report with the same order placements (Kirilenko et al. 2011). They also use a zero-intelligent agent model with two enhancements. Firstly, instead of using a uniform distribution to determine the price of an order, they use a normal distribution around the last traded price because they argue that high-frequency traders, which are a large part of order-flow, place 60% of their orders within one price tick from the last traded price (CFTC and SEC 2010a). Secondly, they add order cancellation mechanics to the model as they argue it was a large contributor to the flash crash.

Using zero-intelligent agents with limits imposed on them based on the findings of (Kirilenko et al. 2011) with summaries in table 2.2 and using the same proportion of trading agents in a simulated environment, another paper also uses a different distribution for order size and price distribution. The details of the price distribution and order size are not provided in the paper. Interaction between traders happens through a limit order-book market using price-time priority to match incoming orders in combination with a Poisson arrival and cancellation process with a mean that is relative to the speed that a trader class is allowed to place and cancel orders.

Verification of the model is done through two factors:

- Traded volume and cancellation rate for each of the agent classes matches data provided from the real market.
- The Shape of the order-book resembles a “V” shape.

The second step is to compare stylised facts between the real market price and the simulated market. It is done on the following basis:

- Price returns follow a normal distribution with fat tails.
- Volatility clustering of “absolute” price returns is present.
- The Absence of auto-correlation of returns shows there is no predictability of price movements.
- Aggregation of returns increases as the time scale over which the returns are calculated increases and the distribution approaches the Gaussian form.

Another group uses a zero-intelligent agent model similar to one introduced in (Maslov 2000) with two enhancements. Firstly, instead of using a uniform distribution to determine the price of an order, they use a normal distribution around last traded price because they argue,

based on information from the CFTC report that high-frequency traders, which are a large part of order-flow, place 60% of their orders within one price tick of the last traded price. Secondly, they add order cancellation mechanics to the model as they argue it is a large contributor to the flash crash.

2.6 Market Simulation Platforms

Although, many of research we have referred above (*e.g.* (M. Paddrik et al. 2012)) use general multi-agent simulation platform like NetLogo, there are platforms specifically designed to simulate a financial market. Many of major financial institutions have in-house platform that allows them to replay and model trading of their own strategies to analyse their performance before they are being deployed to production environments but they are not available to research community. There are market simulators developed and made public in the research community with different targets. For example (Cliff 2012) provides a limit orderbook exchange model for educational purpose. Another example is (Booth 2013) which is designed for performing research into predicting the orderflow.

2.7 Agent-Based Modelling and Simulation

Agent-Based Modelling is a useful method to investigate behaviour and phenomena emerging from the interaction of autonomous agents (Macal and North 2010). Agent models are relatively simple and their behaviour is driven by some local utility function of their own. On the other hand, the system as a whole could show some behaviour that cannot be described using the sum of the behaviour of its components. One may think for instance of ant or bee colonies, or flocks of birds as well-known natural examples of the emergence of ordered, coherent collective behaviour without any central control or authority.

The idea of agent-based modelling was developed as a relatively simple concept in the late 1940s. Because it requires computation-intensive procedures, it did not become widespread until the 1990s. Current use of agent-based modelling is not limited to computer science. Niazi and Hussain (2011) used scientometric analysis to study journal articles indexed in the ISI web of knowledge, published within a twenty year period, between 1990-2010. Their study shows agent-based modelling is widely used in a number of non-computing related scientific domains including life sciences, ecological sciences and social sciences. Some examples of applications include supply chain optimisation and logistics, modelling of consumer behaviour, social network effects, distributed computing, workforce management, traffic management, and portfolio management.

In these and other applications, the system of interest is simulated by capturing the behaviour and interactions of individual agents. Agent-based modelling tools can be used to test how changes in individual behaviours will affect the overall, emergent system behaviour. Agent-based modelling is used to help understand the system as a whole or predict and assess the effect of simple behavioural or model changes in the agent on the system as a whole.

(Todd et al. 2014) proposes the use of data visualisation in the processes used by market surveillance, enforcement and academic research in the area of financial market regulatory. (M. E. Paddrik et al. 2016) present different ways of visualising the data in an electronic trading venue to support regulators better analyse the order flow in the market. (Mark Paddrik et al. 2017) investigate the use of agent based model of the limit order book to study data available from electronic markets to help reulators, exchanges and participants to better understand the stability and resiliency of a market. They have confirmed these findings by comparing their results to user identifiable order flow data from CME and NYMEX. They claim that the data can reliably signal a high likelihood for an immediate flash crash event about one minute before it occurs.

2.7.1 Trading Agent Models

Agent-based model has been used to study trading strategies and the performance of such strategies. For example, Schoreels, Logan, et al. (2004) used agent-based system building strategies using simple genetic algorithms and compared their performance against human traders competing on building the portfolios and observed that they provided comparable performance. They compared the behaviour of dynamic agents against static agents in trading DAX-30 and observed that the dynamic approach has superior performance (Schoreels, Logan, et al. 2004). They also investigated the effect of using varying amounts of training data on the specificity and robustness of a multi-agent based solution for use in trading simulations using historical equity market data. The results indicated that larger training data sets lead to more general solutions and overall better performance when tested in environments with varying conditions (Schoreels and Garibaldi 2005).

Agent-based modelling is different from designing an intelligent agent or multi-agent systems whose goal is to implement a model for the agent with the ability to learn, perform a task, or solve a practical or engineering problem.

(McGroarty et al. 2018) present a multi-agent model framework in python to simulate a limit orderbook and different trading agents. This platform uses discrete time simulation with fixed intervals to simulate the behaviour of the market. In each cycle agents are chosen to perform an action based a probability distribution of that agent class. All the agents have access to the limit order book information.

2.7.2 Agent-Based Simulation Platforms

The Foundation for Intelligent Physical Agents (FIPA) was formed as a Swiss-based organisation in 1996 to produce software standards specifications for heterogeneous and interacting agents and agent-based systems. In March 2005, the FIPA joined the IEEE Computer Society and now is an IEEE Computer Society Standards Organisation that promotes agent-based technology and the interoperability of its standards with other technologies.

FIPA specifications represent a collection of standards intended to promote the inter-operation of heterogeneous agents and the services that they can represent. See (Poslad

2007) for an overview of the FIPA approach. The complete set of specifications including the ones that did not or have not yet made it to standardisation can be viewed in terms of different categories: agent communication, agent transport, agent management, abstract architecture and applications. Of these categories, agent communication is the core category at the heart of the FIPA multi-agent system model. The most widely adopted of the FIPA standards are the Agent Management and Agent Communication Language (FIPA-ACL) specifications (FIPA 2002).

Swarm is a multi-agent software platform for the simulation of complex adaptive systems. It is an open source GPL (Free Software Foundation 2007a) licensed simulation framework that was initially developed at Santa Fe Institute (Minar et al. 1996). It is now maintained by Swarm Development Group[6], a non-profit organisation. In this system, the basic unit of simulation is the *swarm*, a collection of agents executing a schedule of actions. Swarm supports hierarchical modelling approaches whereby agents can be composed of swarms of other agents in nested structures. Swarm provides object-oriented libraries of reusable components for building models and analysing, displaying, and controlling experiments on those models. Computation in a Swarm application takes place by objects sending messages to one another. It is written mainly in Objective-C with some Java code for the interface. It requires the GNU C compiler, Unix, and the X Window system.

MASON has been developed at George Mason University to meet the needs of swarm style multi-agent systems research. MASON is a single-process discrete event simulation core and visualisation library written in Java (Luke et al. 2005).

There are many agent-based simulation frameworks available but their execution model does not permit efficient and distributed simulation which is becoming more important with large-scale economic and molecular biology agent models, where either the execution time or memory requirements outstrip single machine capabilities (Coakley et al. 2012).

JADE

Java Agent Development Environment (JADE) is a software framework that simplifies the development of agent applications in compliance with FIPA specifications (Bellifemine et al. 2000). It has a middle-ware layer that complies with the FIPA specifications and a set of graphical tools that support the debugging and deployment phases. A JADE-based system can be distributed across machines (which need not share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be changed at run-time by moving agents from one machine to another, as and when required.

JADE is completely implemented in Java. It is free software and is distributed by Telecom Italia in open source under the terms and conditions of the LGPL (Free Software Foundation 2007b). It is still being maintained at the time of this writing. It aims to provide a platform to describe agents, a communication mechanism compatible with FIPA-ACL, a *Directory Facility*

[6]http://www.swarm.org

to announce which agents are available and an *Agent Management System* that controls the platform and creates and destroys the agents (JADE 2015). It includes:

- A runtime environment where JADE agents can "live" and that must be active on a given host before one or more agents can be executed on that host.
- A library of classes that programmers optionally use (directly or by specialising them) to develop their agents.
- A suite of graphical tools that allow administering and monitoring the activity of running agents.

Each running instance of the JADE runtime environment is called a *Container* as it can contain several agents. The set of active containers is called a *Platform*. A single special *Main container* must always be active in a platform and all other containers register with it when they start. The first container to start in a platform must be a *main* container while all other containers must be normal containers and must be told where to find their main container to register with. JADE agents are identified by a unique name and, provided they know the names of others, they can communicate transparently regardless of their actual location. Besides the ability to accept registrations from other containers, the main container differs from normal containers as it holds two special agents: the Agent Management System that provides the naming service and represents the authority in the platform, and the Directory Facilitator that provides a Yellow Pages service by which an agent can find other agents providing the services it requires to achieve its goals. (Bellifemine et al. 2000)

FLAME

Flexible Large-scale Agent Modelling Environment (FLAME) is an agent-based modelling framework designed to utilise parallel architectures (Greenough et al. 2008). It is based on the concept of a communicating *X-machine* as a formal basis for the development of an agent-based simulation framework (Kefalas et al. 2003). An X-machine is a form of a state machine with internal memory. The framework has been used for simulating biological systems (Kiran et al. 2008), economic modelling (Deissenberg et al. 2008) and other application areas.

Agents are specified in an XML based language called XMML. FLAME is a template driven system and parses a description of a model from XMML and applies the data to a set of standard templates that automatically generate simulation code in C. The program that does the parsing is called *Xparser* which also statically calculates the optimal execution order of agent functions for efficient parallel processing. The modeller supplies agent functions in C which are compiled together with the resulting simulation code generated from applying templates to XMML definition. The final binary is then linked to the FLAME communication library, called Message Board, to produce the simulation program. The resulting code can be compiled and executed on both serial and parallel systems. For the parallel version, FLAME produces code that uses the *Message Passing Interface (MPI)* for execution on parallel computing systems. During the simulation, agents are able to pass only through defined state changes and communicate with

other agents through a collection of message boards. Thus Xparser generates these message boards to manage the community.

To illustrate FLAME we review an example from (Greenough et al. 2008). This model as defined in XMML is a collection of point agents in two-dimensional space that interact with neighbouring agents, within their radius of influence, by repulsion. The XMML definition of this system consists of its environment, the X-Machine definitions and their definition of messages that agents use to communicate. The *environment* block specifies essential global data for the model such as the names of function files. This example only specifies "functions.c", the name of the file that contains the C code functions associated with the model.

The next step is to define the agents within the model. In this example, there is only one very simple agent called Circle. The agent has two main elements: its memory and its functions. All the information that an agent can maintain between execution cycles is defined in the memory section. The functions section defines the states the agent can enter and how they are related. The ordering of the functions in this section is important and determines the execution sequence. Depends tags for each function specify any dependency on information from other agents provided through the message boards. In this case, the *inputdata* function is dependent on *location* messages from other agents.

Each state transition function has access to the internal memory of the agent, as well as input and output streams of information. Agent transition functions take in this memory structure and update the values, effectively transitioning the agent instance to the next state ready to be consumed by the next function. The execution of a transition function is repeated for all agent instances of the associated type in the relevant state. Once all the functions have been called (in the correct order to meet dependencies) an iteration (time step) of the simulation is complete. Each message board handles the messages of a single message type. Because message boards are the only means by which agents communicate with other agents, the agent model is inherently parallel. Each agent can execute independently as long as the input message board contains the expected messages

FLAME has been extended to map a subset of its templates to Graphical Processing Units (GPU) (Richmond et al. 2010). For computational power, GPGPUs have provided exceptional speed-ups. To execute the same functions at the same time, ABM GPGPU frameworks use asynchronous agent updates because they are either cellular automaton based or are an extension of the FLAME model for GPGPUs. However, this condition means that they are better suited for homogeneous agent models, where there is only one type of agent with the same functionality. GPGPUs are also restricted by their amount of on-board memory they have direct access to. This restriction can be mitigated by using multiple GPGPUs but then the same memory independent node problem as with HPCs results. All have shown exceptional speed increases but are very limited in the scope and complexity of the models they can run due to the smaller amount of memory available and the homogeneous nature of the agents used

in models.

2.8 Summary

This chapter first reviewed the flash crash of 6 May 2010 as well as a number of other market disruptions that have many properties in common with the flash crash. We then reviewed a number of changes that have been proposed to improve market stability and to prevent such events happening in the future. Especially we reviewed solutions proposed to limit high-frequency trading, introduce circuit breakers and also minimum-quote-life and maximum order-to-trade ratio limits. We finally looked at the previous studies of the flash crash and research that applied agent-based modelling to study the mechanics of trading in the market and analysis of the flash crash. The next chapter describes the agent-based model of the market that this research uses to investigate the flash crash and to study the proposed solutions.

Chapter 3

Experimental Platform and Data

3.1 Introduction

This research uses agent-based modelling to analyse market behaviour and simulate a flash crash. To simulate trading venues at the same scale of a real financial market with different trading agents a high-performance platform is required. This chapter discusses the details of the trading agent simulation platform that is implemented for this research. This platform is used in the next chapters to study the flash crash and analyse the effect of the changes to the market structure or trading rules.

Trading is a process by which the buyer and the seller interact to agree on the details of the securities they exchange. There are three main forms for the trading process used by electronic trading platforms: Quote-Driven, Quote-Streaming and Order-Driven.

In a Quote-Driven setup, trading is considered a service provided to a counter-party acting as a client (*e.g.* an individual investor or asset manager) by a dealer (*e.g.* an investment bank or a broker-dealer). When the client wants to buy (sell) x shares of a stock, it sends a *Request for Quote (RFQ)* to a dealer and asks for a *quote* for that quantity. If the dealer is willing to trade, it responds with a quote specifying the price at which it is willing to sell (buy). After receiving the quote, if the client accepts the provided quote, it places an order and a trade happens. To obtain a better price, a client may want to ask more than one dealer for a quote and choose the best one. In a quote-driven market, detecting the current fair trading price of an security (price discovery) and availability of counter-parties with whom to trade (liquidity discovery) is not straightforward. Placing an order comes with the risk of information leakage about pending intention to trade and dealers can provide different price for the same request from different clients. A modified version of the quote-driven system is *Quote Streaming* where a dealer sends a continuous stream of quotes to its clients. This model of streaming quotes is popular for FX

and fixed-income trading.

Commonly used trading mechanism that can address the problems stated above with price and liquidity discovery. In an *Order-Driven* market, buyers and sellers have similar roles and all market participants send their request to buy or sell a stock to the market as an *order*. An order specifies the stock to be traded, the buy or sell direction, and the minimum or maximum price that participant would accept for the trade. If an existing order matches these criteria, a trade happens. Otherwise, the new order is placed in the *orderbook* to match against future incoming orders. If more than one order matches the incoming order, the market chooses the order(s) to trade from orders in the orderbook using its order *matching algorithm*. Most stock and future exchanges (*e.g.* London Stock Exchange, New York Stock Exchange, and CME) use this form of trading. In the rest of this thesis we focus on order-driven markets and in the next section we describe how an order-book driven market works.

To simulate an agent-based model of a financial market, a multi-agent simulation framework is needed. This framework should be able to simulate an electronic trading market and its participants' behaviour. This chapter starts by describing the different parts of an electronic financial market in Section 3.2. Section 3.3 reviews existing agent-based modelling frameworks in further details. Next, we introduce our proposed model of an electronic financial market in Section 3.4. This section provides a high-level description of how this research models a financial market and presents the main components of the system and how they interact with each other. Section 3.5 provides details about the implementation of this simulation platform. This research uses very detailed orderbook level data from financial markets to evaluate the behaviour of different trading agents when they interact with a real market. Section 3.6 describes the market data used in this research. Finally, Section 3.7 provides a summary of this chapter.

3.2 Electronic Financial Markets

In the rest of this thesis we focus on order-driven markets and in this section we describe how an order-book driven market works. An order-driven market works as a central point to match buy and sell orders. The common form of this market is to have a continuous double-auction of remaining buy and sell orders in the book. A continuous double-auction means that as soon as an incoming order arrives and the orderbook has a matching order, a trade happens. The alternative is to have scheduled auctions. In the latter form, orders are received and kept in the orderbook for the duration of an auction, even when matching orders exist. At the end of the auction, all orders are matched on a single price. For the purpose of this research, we consider continuous double-auction markets.

Most order-driven markets publish information about unmatched orders in the orderbook and trades that happened to market participants. This information is called *market-data*, and we describe it in further detail in Section 3.2.7. Market-data makes liquidity and price discovery simpler. There is a delay from the time that market publishes the state of the orderbook to the

time a market participant receives that data and processes it. Then again, there is a delay for a potential order from that participant to reach the trading venue, and as a result by that time the orderbook state may have changed in between.

As described previously, the participant sending an order can specify the price, and size. We now look into these characteristics in more detail as well other common characteristics we may mention in this thesis.

3.2.1 Trading Platform

An electronic trading *platform* refers to a wide range of software and hardware systems, network infrastructure and communication protocols that enable computer-based systems to communicate with one another and execute orders. An electronic trading market has its participants with their own internal structure connecting to the market to place orders and receive market data. Systems inside the participants could be structured in different layers. Such a platform, especially in a large financial institution, could be quite complex because many internal systems with different requirements need to connect to a number of external markets.

It consists of market gateways that communicate with each market using their dedicated protocols. Because some trading systems need to communicate and trade in more than one market, these internal gateways can form a market access layer to abstract differences between markets and provide a uniform interface to internal trading systems. On the downside, such abstraction may not be able to provide all the functionality in detail that direct access can make available. Another issue with such a layer is the extra latency it adds that may not be acceptable for some of the trading systems, such as high-frequency market making systems. An example architecture of such a system inside an investment bank is shown in figure 3.1.

Similar arguments hold for market data published by any of the financial markets. Because the communication protocol used by each of these markets could be different, a feedhandler is required to decode and normalise it to provide a unified view of all external markets in a consistent and similar form to the internal systems. There is also a possibility that this normalisation is done by an external market data vendor. Similarly to market access, trading systems that are latency sensitive may need to connect to each market directly.

To understand these layers, let's look at an example. Let's assume the portfolio trading system decides to buy a large number of shares in company ABC. Sending such an order directly to the market would result in a huge market impact. Instead, it sends the order to an execution strategy that trades based on a benchmark, *e.g.* VWAP. The execution strategy splits the order into slices and then sends the slices in a time window to the market. For each slice, especially if it is an aggressive order, the strategy looks for different sources of liquidity such as MTFs, dark pools, and the exchange, and may decide to split the slice into further chunks and send them to different venues.

These layers in the system provide an abstraction for the higher levels and allow them to

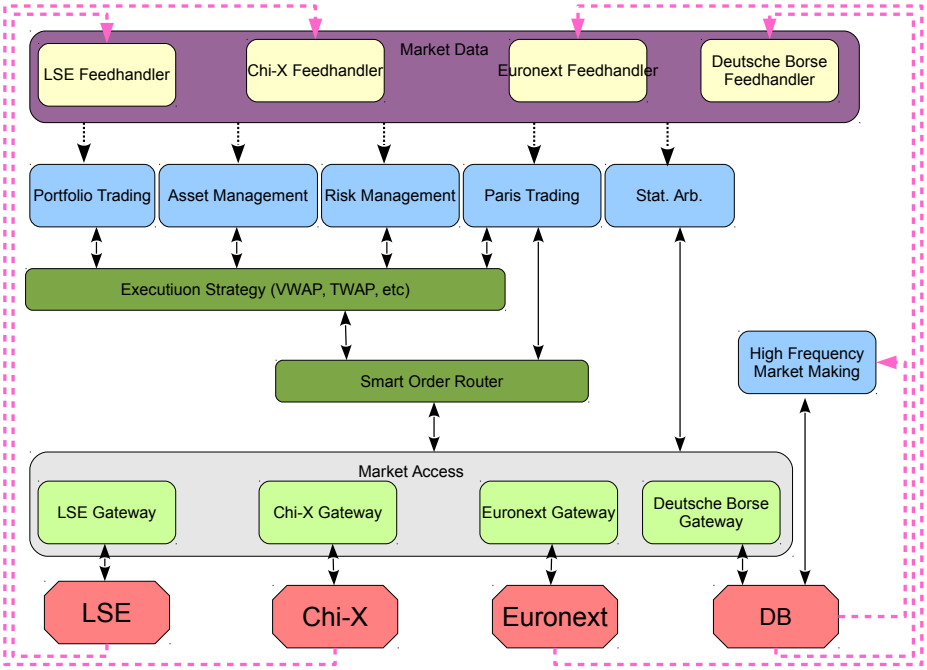


Figure 3.1: An example of a electronic trading platform

focus on their target rather than details of trading. On the other hand, for a high-frequency trading strategy, going through these different layers of abstraction may not be beneficial because each layer adds latency for its processing and communication. A latency sensitive strategy like statistical arbitrage may lose an opportunity because of the latency. A market making strategy may rely on detailed information about its market that is not easy to abstract. Such strategies may directly communicate with the trading venue instead of using this stack.

3.2.2 Market Connectivity

A high-frequency trading system requires fast market data and execution access. Market-data feeds are one of the main inputs to a high-frequency trading system. Having accurate and fast market data allows the trading strategy to react to any market movements faster than other participants and grab opportunities while they exist. There are a number of ways to get access to market data feed from a trading venue.

The traditional option would be to receive the data via a market data vendor such as Thomson-Reuters or Bloomberg. These vendors connect to all trading venues, translate the trading venues' information to their normalised form and then dpublish this data to their clients. The benefit of such a solution is that only one physical connection to the market data vendor is required. Also, the data are normalised and provided via the same messaging protocol or API, which means faster time to market and easier maintenance because any changes to the trading venue protocols are handled by market data vendor and are hidden from the clients. On the negative side, there is a higher latency involved in the connectivity from the trading venue to market data vendors and back to the clients compared to a direct connection from the trading

venue to the client. The normalisation and publication layer by the vendor also adds latency.

The second option is to have a direct connection to all trading venues. Market data feed handlers can be implemented in-house or could be provided by third-party vendors. This option reduces the connectivity latency and can improve on the layers of software involved with normalising and publishing the data too. Another benefit is that whereas normalisation could lose some valuable information that can be useful for a trading strategy, implementing the trading venue protocol could keep this information.

Finally, one could save on the latency by locating the trading application in the trading venue data centre (*co-location*) or somewhere physically very close (*proximity*). The choice of each of the options above depends on many factors. The sensitivity of the trading strategy to latency, the speed of other players in the market, and the speed of the trading venue's matching engine are the most important ones.

3.2.3 Market Gateways

Participants need to use a protocol to communicate with the trading venue to send orders, modify them, or cancel them. The trading venue also notifies the participants about order status such as the current status and its execution. For example, a sequence of events could be as follows:

- A participant sends an order to buy 200 shares of ABC at the price of 100.20.
- At the time of submission, there is no sell order that can be matched at that price. As a result, it is placed in the orderbook and the market sends an acknowledgement back confirming that the order has been placed in the orderbook.
- An order is received to sell ABC at a price that can be matched against. This order (or the portion of the order matching against this one) is 50 shares. An execution report is sent back to the agent confirming that 50 shares have been matched and there are still 50 shares open in the market.
- The trading venue informs the participant that the remaining 1000 shares of its order are also executed, and the order is fully filled and removed from the market.
- Finally, the participant decides to cancel its remaining open shares in the market.

Trading venues use different protocols to communicate with their participants. *Financial Information eXchange (FIX)* is a standard protocol that has been traditionally used between clients and brokers. Trading venues are adopting this standard as their communication protocol either as their default protocol or as a protocol along with their default protocol. Versions FIX 4.2 (FIX Protocol Limited 2000) and FIX 5.0 (FIX Protocol Limited 2006) of the standard are the most common versions at the time of this writing.

Figure 3.2 shows an example of interaction between a participant and the market. In this example, the participant places a limit order to buy 200 shares of ABC at $10.70. At the time

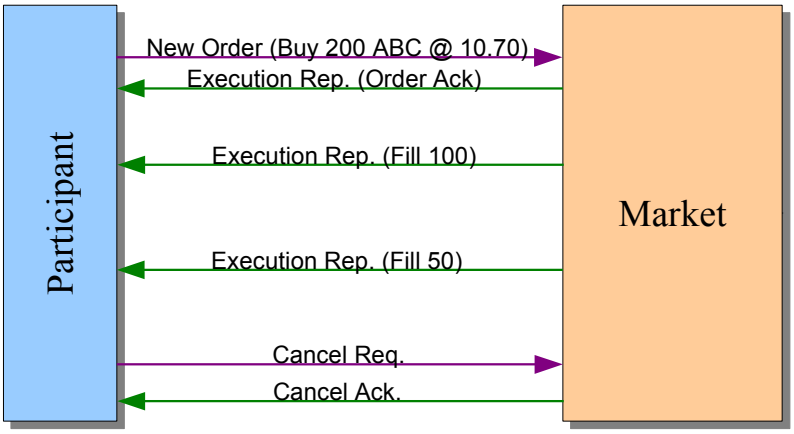


Figure 3.2: Interaction between a participant and market

of submission, there is no sell order that can be matched at that price. As a result, it is placed in the orderbook and the market sends an acknowledgement back confirming the order is placed in the orderbook. Sometime later, an order is received to sell 100 ABC at a price that can be matched against (to sell at $10.70 or lower). It could have also been a bigger order that the rest of its shares are matched with orders with higher priority in the orderbook, e.g. have higher limit price or have the same limit price but arrived before this order. An execution report is sent back to the agent confirming 100 shares are executed, and there are still 100 shares open in the market. Exchange receives another order to sell 50 shares at $10.70 or lower. Note in this case this order is on top of the priority queue and will be matched against any incoming order now. Finally, the participant decides to cancel its remaining open shares in the market and trading venue acknowledges that order has been cancelled and removed from orderbook. It is also possible that this order has been executed just before trading venue has received the cancel request. It may have sent another execution report notifying of the filled amount which is not received by the market participant yet. In this case trading venue sends a cancel reject instead of acknowledging the cancel request.

3.2.4 Trading Venues

An order-driven market works as a central point to match buy and sell orders. The common form of this market is to have a continuous double-auction of remaining buy and sell orders in the book. A continuous double-auction means that as soon as an incoming order arrives and the orderbook has a matching order, a trade happens. The alternative is to have scheduled auctions. In the latter form, orders are received and kept in the orderbook for the duration of an auction, even when matching orders exist. At the end of the auction, all orders are matched on a single price. For the purpose of this research, we consider continuous double-auction markets.

Most order-driven markets publish information about unmatched orders in the orderbook and trades that happened to market participants. This information is called *market-data*, and we describe it in further detail in Section 3.2.7. Market-data makes liquidity and price discovery

simpler. There is a delay from the time that market publishes the state of the orderbook to the time a market participant receives that data and processes it. Then again, there is a delay for a potential order from that participant to reach the trading venue, and as a result by that time the orderbook state may have changed in between.

As described previously, the participant sending an order can specify the price, and size. We now look into these characteristics in more detail as well other common characteristics we may mention in this thesis.

3.2.5 Order Specifications

The first characteristic is the *order price*. If the participant is willing to sell or buy at the best price available on the market without specifying a limit, it sends a *market order*. If the participant knows the maximum price it is willing to pay to buy or the minimum price at which it is willing to sell at, it sends a *limit order* and specifies a *limit price*. A limit order is called a *passive order* if the limit indicated on the order to buy (sell) is below (above) the current best price available in the market. A passive order is not executed immediately and is placed in the order book. A limit order is called *aggressive* or *marketable* if the limit price of the order is equal or greater (less) than best price available on the orderbook and therefore results in a trade immediately. Some markets also support an order price to be specified relative to a *reference price*. Such an order is called a *pegged order*. The reference could be the best buy (sell) price in the order book or the mean of the best buy and sell order (mid) price.

The second characteristic of an order is the *order size*. For a stock, the order size is the number of shares the participant is willing to buy or sell. If the order is not executed, the trading venue notifies other participants of the order and its size, and the price is published on the market data feed. Trading venues allow participants to make a choice about the visibility of their orders. An order could be fully *hidden*. If a hidden order does not match against another order, it will reside in the orderbook but no information about it will be published on the market data feed. To compensate for the fact that these participants are hiding their intentions, hidden orders normally have lower priority than visible orders at the same price. An order could be partially hidden, called *iceberg* order. An iceberg order has just some portion of it visible to other participants. If the visible part is executed, the invisible part becomes visible up to the visible size limit. It is a common practice in trading venues to give the hidden part of an iceberg order lower priority than visible orders at the same price. Some trading venues have limitations about which orders can be hidden orders or icebergs. For example, LSE does not support fully hidden orders and accepts iceberg orders only if the order size is above a certain limit and its visible size also has a minimum.

The last characteristic we describe here is the order lifetime. It specifies how long the participant is willing to leave the order in the orderbook if it is not matched. A participant can decide to cancel its order or modify it before the lifetime of the order expires. If an order is not matched or cancelled by participants, the market automatically cancels the order at the

end of the specified lifetime and notifies the participant that its order has been cancelled and is no longer in the orderbook. The lifetime of an order is also called its Time-in-Force. If no expiration time for an order is specified and the order can remain in the book until it is matched or cancelled, it is called *Good-Till-Cancel (GTC)*. If a specific expiration date/time is specified, it is known as a *Good-Till-Date (GTD)* order. Good-Till-Date orders are used by slow traders to send orders that stay on the book for multiple days, which could be useful to maintain order priority for low liquidity instruments that do not trade often. On the other hand, Good-Till-Cancel orders are also used by high-frequency traders to specify a short window (sometimes only a few seconds) to make sure the order is cancelled even if the trading engine has technical issues or loses connectivity to the market and cannot cancel it. A *Day* order is a type of order that stays in the book until the trading day ends. Finally, *Immediate-or-Cancel (IOC)* is a type of order that has zero lifetime. It should be either matched with an order currently in the orderbook or cancelled otherwise. Such an order never stays in the orderbook.

There are other characteristics that could be specified on an order, *e.g.* minimum size of the trade *etc.*, but the properties described above are the most important ones and the ones needed for understanding the rest of this thesis.

3.2.6 Matching Algorithm

The orderbook at each point in time contains all the unmatched orders. The bid side of the orderbook includes all buy orders, and the ask side includes all sell orders. Each side lists orders with their identifier, the time they were sent to the market, their size, and their price. Electronic order books normally maintain *price and time priority*. If a buy order comes to the book, it matches all orders at the lowest ask price before matching them against any order at a higher price. For orders at the same price, it matches against orders that were sent to the market earlier than others. Figure 3.3 shows a sample orderbook with four orders.

Bid				Ask			
OrderId	Time	Size	Price	Price	Size	Time	OrderId
x002	08:00:00.003	500	160.90	161.00	1000	08:00:00.001	x001
				161.00	1500	08:00:00.006	x004
				161.10	500	08:00:00.004	x003

Figure 3.3: Sample orderbook

In our sample orderbook, assume a new order x005 to buy 2000 shares at 161.20 is received at 08:00:00.009. The best ask price is 161.00, and there are two orders x001 and x004 at that price. Because x001 is sent five milliseconds before x004, it has higher priority. Order x001 has just 1000 shares, while the incoming order is for 2000 shares. A trade will happen against this order and x001 will be removed from the book. The remaining 1000 shares of x005 are now matched against x004. Because the size of x004 is 1500 shares, 1000 shares will be executed and 500 shares will remain in the book. Note that price is the highest priority and although x003 was in the book before x004, x004 has price priority. Figure 3.4 shows the state of the

orderbook after two executions. Note that the trade happens at the price of the order in the orderbook and not the price of the incoming order. In our example, the incoming order had a limit of 161.20 but because the current orders were at 161.00, the resulting trade was priced at 161.00.

Bid				Ask			
OrderId	Time	Size	Price	Price	Size	Time	OrderId
x002	08:00:00.003	500	160.90	161.00	500	08:00:00.004	x004
				161.10	500	08:00:00.004	x003

Figure 3.4: Status of orderbook after matching of incoming order

Some trading venues maintain *price/size priority* and not *price/time* priority. When processing a new order, such a venue matches all orders at the same price regardless of their time. The orders at the same price on the book get executed against a slice of the incoming order in proportion to their size. If the state of the book were as described in Figure 3.3 and a new order x005 were received to buy 2000 shares at 161.20, it would be matched against both x001 and x004. The total number of shares available at 161.00 would be 2500 shares, and the order would be requesting 2000 shares, which is 80% of that total size. As a result, 80% of x001 (800 shares) and 80% of x004 (1200 shares) would be executed. The remaining shares from both orders would stay in the orderbook. The state of the orderbook after processing order x005 is shown in Figure 3.5.

Bid				Ask			
OrderId	Time	Size	Price	Price	Size	Time	OrderId
x002	08:00:00.003	500	160.90	161.00	200	08:00:00.001	x001
				161.00	300	08:00:00.006	x004
				161.10	500	08:00:00.004	x003

Figure 3.5: Status of sample orderbook with price/size priority after the execution

Most trading venues and securities cannot trade in fractional size. For example, one cannot own half a share of a company on most of the trading venues. Trading venues using price/size priority have detailed rules about the allocation of fractional sizes to orders residing in the orderbook to deal with such cases. For our research and experiments, we use price time priority matching rules.

3.2.7 Market Data

To maintain fairness, trading venues publish information about the current unmatched orders in the orderbook to their participants. A trading venue can provide this information at different levels of detail. Below we look into common forms of market data provided by trading venues.

Market by order

Market by Order is the highest level of information available to participants. Such a market data feed notifies the participants of every single unmatched order entering the orderbook and all order modifications and executions. Let's consider the sample orderbook described in Table 3.3.

As soon as order x001 enters the book, the venue sends a message informing other participants that there is a new order x001 to sell 1000 shares at a limit price of 161.00. It does the same for all other orders x002, x003 and x004 when they arrive. When an aggressive order x005 comes into the venue and matches x001 and part of x004, the venue sends two new messages specifying that 1000 shares of x001 and 1000 shares of x004 have been executed. The market by order information published on the market data feed for our sample scenario is shown in Figure 3.6.

Time	OrderId	Action	Side	Size	Price
08:00:00.001	x001	Add	Sell	1000	161.00
08:00:00.003	x002	Add	Buy	1000	160.90
08:00:00.004	x003	Add	Sell	500	161.10
08:00:00.006	x004	Add	Sell	1500	161.00
08:00:00.009	x001	Execute		1000	
08:00:00.009	x004	Execute		1000	

Figure 3.6: Market by order

Some venues provide an extra execution identifier field on the execution messages, which is unique for each matching. In our example, such an execution identifier would be the same, *e.g.* t001 for both x001 and x004 executions. This identifier informs the participants that both orders have been executed against one incoming order rather than two separate orders. On the other hand, some venues send the same message for the case that x001 is matched against incoming order x005 and order x001 is cancelled by the participant itself. They just send a modify notification of the size of the order to zero or a delete message. Many such venues send information about the trades separately too, but matching the orderbook feed against the trade feed is imprecise, and if one of them (normally the trade feed) is slower, it cannot be used in real-time because it adds latency.

Market by level

The different prices on each side of the orderbook are called price-levels. Instead of publishing the details of individual orders, a trading venue may just provide information about price levels of the book. Such a market data feed is called *Market by Level*. Some venues also publish the number of orders forming that price level. Figure 3.7 below shows the view of the book obtained from such a market data feed. Understanding what is actually happening in the orderbook is more difficult from this feed than from a Market-by-order feed. The accuracy of any logic used to analyse the orderbook behaviour highly depends on the availability of the number of orders and also the frequency at which the price level update is published.

Bid				Ask			
# of orders	Time	Size	Price	Price	Size	Time	# of orders
1	08:00:00.003	500	160.90	161.00	2500	08:00:00.006	2
				161.10	500	08:00:00.004	1

Figure 3.7: Market-by-level view of orderbook

Top of the book

If the trading venue just provides information about the best bid and ask price and the total size available at each best price, it is called a top of the book or a Level 1 feed. This type of data feed also provides information about the trades. Figure 3.8 shows a level 1 feed corresponding to the orderbook in Figure 3.3 with corresponding changes after order x005 is processed.

	Quote				Trade	
Time	Bid Size	Bid Price	Ask Price	Ask Size	Trade Price	Trade Size
08:00:00.001	500	160.90	x	0		
08:00:00.002	500	160.90	161.00	1000		
08:00:00.005	500	160.90	161.00	2500		
08:00:00.007	500	160.90	161.00	500	161.00	2000

Figure 3.8: Top of book market data

Dark pools

Dark pools are trading venues that run an electronic orderbook but do not provide any visibility into the current state of the orderbook. They just inform the market of the trades after they happen. No information about the orders in the book is published before they are executed. The market data feed from a dark pool is a level 1 feed that does not provide information about quotes and just publishes trades.

3.3 Evaluation of Existing Platforms

Most of the financial trading systems are event-driven and react to incoming orders, market data feeds and other sources of information like economic announcements, news and social media. CEP systems provide a generic platform to implement such trading systems and allow the designers to focus on designing the strategies and trading engine. The volume of data being processed by a trading system is very high and latency in processing such data is critical. This means the cost of communication in such systems is considered to be very high as it introduces latency. Financial firms use co-location to place their trading strategies in the same data centre as the trading venues's matching engine to avoid communication delays.

In next section we looks at some of the available agent-based modelling and complex event processing platforms and evaluates their suitability for this research.

3.3.1 Agent-Based Modelling Frameworks

To study a financial market using agent-based modelling, a software framework is required to simulate trading agent models and trading venue model. Initially, a number of agent-based modelling frameworks were investigated to be used as the core platform for this research. These frameworks were introduced in Section 2.7. This section reviews the use of these frameworks for this research and reports on the results of this step.

Swarm was the first platform to be investigated. Swam has an advantage point that it allows swarm style multi-level agent models, *i.e.* an agent can be composed as a swarm of other

agents. On the other hand, its source code is not updated recently and its practical capabilities were not enough to simulate a market with a large number of trading agents. *Mason* was investigated next. Mason is more flexible in terms of capabilities of individual agents compared to Swarm although it does not provide the same multi-level swarm view of the system. Its source code is not also actively maintained.

Because of the limitations of Mason and Swarm, the next option investigated was JADE. *JADE* does not provide the multi-level swarm view but provides enough capabilities at the agent level to implement trading agent models. A prototype model was implemented using this JADE. Unfortunately, it failed to provide sufficient performance to allow simulating different scenarios with large number of trading agents. Specially implementing high-frequency trading agents that trade at high speed and produce large number of orders in short period of time presented a challenge in the communication performance of JADE.

FLAME is a agent-based modelling platform which was designed for performance and supports running on GPU. Its modelling view is a bit different as instead of modelling a trading agent in a program its modelling framework is XML file defining the agent and environment as X-machines. It allows providing user-defined functions to the model in C and C++. After some trial the efforts to convert a trading agent model to X-Machine and its limitation out-weighted the possible gain on the platform availability and performance.

3.3.2 Complex Event Processing Engines

Each strategy needs to listen to set of events received from different sources and then if needs to react (e.g. place a new or order or cancel/amend an existing order) it will generate another event to send to trading venue. There are platforms available to make building such systems much easier and flexible.

Complex Event Processing (CEP) provides methods and tools to describe and detect patterns of events occurring on a number of event streams. These patterns could be specified by data content, correlation between events, timing and frequency of events. CEP systems are shown to be useful in many areas including finance, health-care, sensor networks, business process management, and network monitoring. The amount of data in many of these areas is very high, it needs to be processed in real-time and the rate of the data is variant. This study reviewed a number of research CEP systems to evaluate if such a system could be used as a core for for simulating agent-based model of a financial market.

Aurora (Carney et al. 2002) at Brandeis University, Brown University, and MIT models queries using a network of building blocks and supports eight primitive operations in these building blocks. It starts with unoptimised network and gathers statistics about the cost of the nodes and their activities and implements a number of heuristics to optimise the heavily loaded sub-networks. *Medusa* (Cherniack et al. 2003) and its next generation *Borealis* (Abadi et al. 2005) are distributed versions of Aurora. They monitor the load of machines running Aurora and balance the load of queries on the machines. *STREAM* (Arasu et al. 2003) at Stanford, defines

queries using an extended version of SQL called CQL (Continues Query Language) (Motwani et al. 2003). It generates a query plan that consists of operators and queues connecting them. It optimises memory requirement by considering constrains on data streams and uses an scheduling algorithm to reduce inter-operator queue sizes. *TelegraphCQ* (Chandrasekaran et al. 2003) at Berkeley, considers the query plan as a set of modules communicating through an API. The system routes data touples through query modules and constructs a query plan that consists of adaptive routing modules. These adaptive modules can re-optimise the plan as it is running. *Cayuga* (Demers et al. 2007) at Cornell uses a non-deterministic finite automata to define the model and uses algebraic optimisation on the automata. *NextCEP* (Schultz-Moeller et al. 2009) also uses the same model to define queries and introduces optimisation techniques for next and union operators.

In recent years, CEP on is not an active area of research and as a result most of the platforms listed above are not maintained or supported anymore. A number of commercial products were also studied that are designed specially for CEP. *Coral8*[1] was founded by Dr. Rajeev Motwani, who was from STREAM project at Standford. *StreamBase*, was founded by Dr. Mike Stonebraker, from Aurora project at MIT. and *Apama Progress*[2] are the most notable products. Although some of these platforms were available to the author has worked, licensing of them for the purpose of this research would not have been straightforward.

Esper (EsperTech 2017) is an open source CEP with commercial support from *EsperTech Inc.* Its initial version was written in Java but later release added support for .Net platform as well. Esper and Java was used to build the first generation of the platform for this research. As better integration with other platforms especially Python was needed to be able to use the tools for running different setups in parallel and interacting with data analysis tools the implementation was moved to C++ as it is a native language producing binary code which can be easily wrapped in Python and R. As Java compile to byte code instead of machine binary codes and runs on JVM, interaction between JVM and other languages specially python is not that simple and efficient compared to C++. In the second iteration we have implemented our event processing engine in C++ similar to many high-frequency trading core platforms used in real-markets.

3.4 Simulation Model

An agent-based model of a system consists of autonomous agents interacting with each other within an environment. Performance of an economy is a joining result of its market structure (the rules that governs exchange), its environment (agent's taste and endowment of information) and agent behaviour (trading strategy). The model presented here considers both market participants and the trading venue itself as autonomous agents. This model of a financial market constitutes of a number of trading agents (market participants) and a single or multiple trading

[1] taken over by Aleri and now both part of Sybase
[2] taken over by Software AG

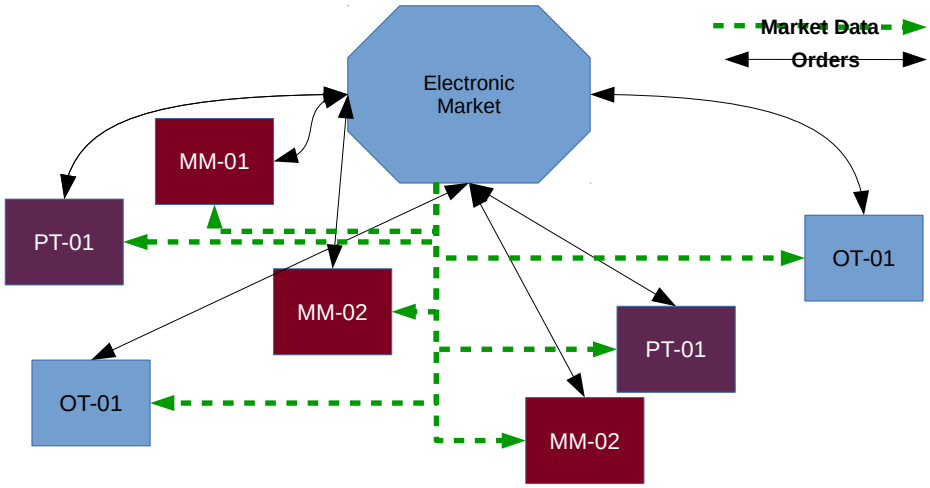


Figure 3.9: An example of a financial market

venues. The topology of this model is like a *star*, meaning trading agents only communicate to the trading venue and there is no direct interaction between trading agents. Trading agents can only trade a single asset. In our initial experiments agents only receive information from the same market they are trading on. The last experiment that looks at the interaction between markets, it is assumed that the agents receive information from more than one market but still trade on a single market. For a real-world trading agent, the concept of environment could be quite complex and the agent interacts with many sources like news, human psychology, etc. But for the purpose of our experiment, it is assumed all the information is translated into the market price and it is the only source of information. For the trading venue, the model is an order-driven market which uses a price-time based priority for matching.

The simulation platform consists of two main components, trading agents and trading venues. Each of these components its built using smaller modules. These modules communicate internally using a generic event processing core. This allows each module to be designed and tested independent of other modules. On top of these core components, some extra glueing system is needed to define and configure simulation scenarios and record required output data for further analysis. The next sections provide further detail on each of these components of the system.

3.4.1 Topology

Figure 3.9 is an example of a financial market with seven trading agents and a single trading venue. Trading agents are divided into *classes* (highlighted by its colour and name prefix) based on their behaviour. We assume agents in the same class share a similar behavioural, *i.e.* their response to market events follows a similar path. Trading agents communicate to the trading venue using an *order channel* (shown in black) which allows them to place a new order, modify or cancel an existing order. Trading venue also notifies the trading agent via the same channel if/when agent's order gets executed or forcibly rejected or cancelled by the trading venue. The

trading venue publishes the current state of orderbook to all of its participants via *market data channel* (shown in green). This model assumes trading venue is publishing order-by-order market data (For further details on order-by-order market data see Section 3.2.7).

Order channels are point-to-point and can transport *NewOrder*, *ReplaceOrder*, and *CancelOrder* events from a market participant to trading venue and responses like order acknowledgement or execution reports back. Market data is supposed to be a broadcast[3] channel that supposed to provide timely information to all market participants. In reality, there is always a delay and no matter what technology is used there could always be a difference between the time that each market participants receives the data compared to others. In our model, these delays are indirectly incorporated to speed of trading of each class of market participants.

3.4.2 Communication

A trading agent is a component that is going to publish *NewOrder* and *CancelOrder* events and is interested in *OrderFill* messages that are addressed to itself and also different type of *MarketData* updates. On the other hand, the trading venue is interested in receiving *NewOrder* and *CancelOrder* messages and registers to publish *OrderFill* and *MarketData*. The framework handles all the bindings and passing of messages between components.

Also, in a real world all these communications happen via networking mechanism which is subject to delay and fault. Depending on the technology used by each market participant they may actually experience a different delay. This delay may not also be fixed or linear as it was shown for the case of the flash crash. For example if the network pipe used by market participants has capacity of transferring only 1,000 messages per second, when market is very volatile resulting in exchange to publish 1,500 messages per second for a few minutes, depending on the setup market participant is going to either miss some of the intermediate updates or is going to experience an increasing delay on its market data messages on this peak traffic (burst). Either way, that participant is not going to have an up to date view of available liquidity in the market.

Even if the capacity of the connection is enough, but one participant is physically located in the same data centre or city as the trading venue and another one is in a different country (e.g. a market participant based in New York and trading in London Stock Exchange) the second participant is always going to have a delayed view of the market. So, when it sends an order hoping to get matched against liquidity seen on market data published by exchange, another participant with faster access to the market may have observed this earlier and taken that liquidity.

3.4.3 Trading Agent

A trading agent is a system that follows a trading strategy based on some control parameters or incoming orders from other systems or human traders. Its actions are based on its inputs plus current and historical market data and its previous actions and current state. To implement

[3] in reality it is a multicast channel

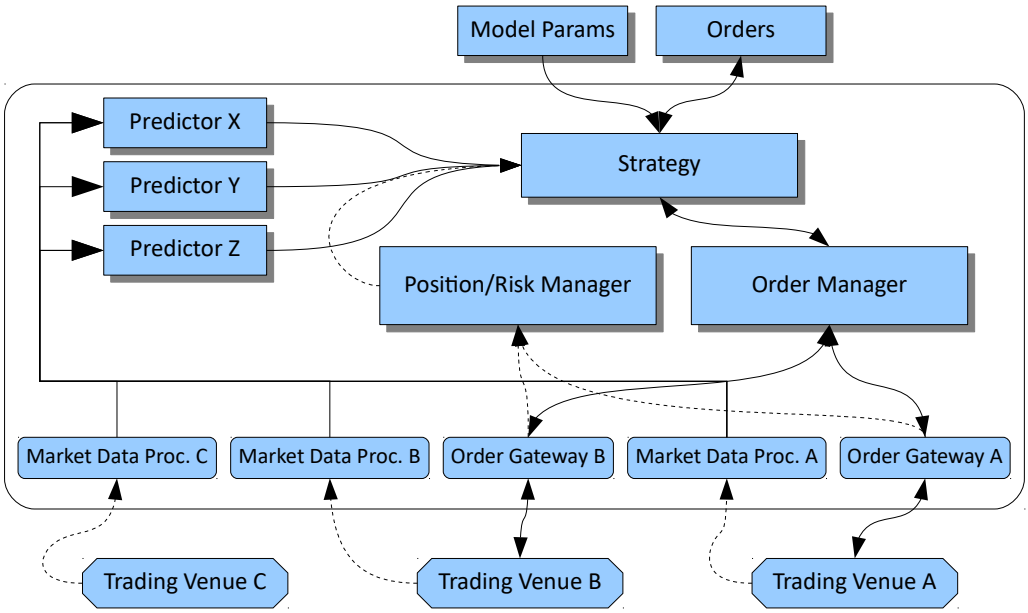


Figure 3.10: Trading Agent Architecture

such a system, a few building blocks are required: strategy, predictor(s), order manager, position manager, order gateway(s) and market data processor(s). A diagram of a trading agent architecture used in this platform is shown in Figure 3.10 that has three predictors and consumes market data from three trading venues but only trades on two of those trading venues.

Strategy

Strategy is the core component of a trading agent. Strategy listens to input from predictor(s), aggregate this information and also looks at its current position and risk and open orders in the market and makes a decision to place a new order, amend its existing order or cancel it. Depending on the complexity and type of predictors, the strategy sometimes could be simpler and only decide about how much exposure it wants to have to the market at one or more price level and leave the decision about the details of the orders to order manager.

Signals

Signals[4] are components that provide processed information to strategy. They listen to different sources including market data from the trading venue that strategy is trading on and related trading venues, trading history of the strategy, economic announcements, and other sources like twitter, etc. and produce an output value that indicate a prediction. For example, a signal can predict the market move in the next 5 minutes or next 30 second and provide an output value between -1 and 1 to show the market is expected to go down or up in the predicted time horizon. This is used by the strategy to decide to buy or sell the asset. The output values of the signals may not be a single number. For our example above, signal can join the output (-1,1) value with another number (0,1) to indicate signal calculator's confidence in its prediction. Signal

[4]also referred to as predictors or alphas

prediction can be generic about the instrument or market or very detailed and specific. For example can produce an output to indicate if market in general is in the Bullish or Bearish mode. On the other hand, signal predict that placing a buy (or sell) order at specific price for a specific security is expected to return N dollars in the next 5 minutes.

Order Manager

Order manager is a component that handles the details of placing and managing orders to the trading venue. It makes sure that the number of open orders and their behaviour is in-line with the requirement from the trading venue.

Risk Manager

Risk manager controls the current position of the trading agent plus limits and control on its risk. In some cases, risk manager is also more complex and behaves as risk control component and can protect trading agent not only from breach of its configured limits but also its open exposure in order manager and potential position it may take in the future.

Order Gateway

Order gateway provides an abstraction layer for sending orders to and processing responses from different trading venues. There are different protocols used by trading venues to communicate with their participants. One of the most common protocols is Financial Information eXchange (FIX) protocol. But even this protocol has different versions and allows optional tags on the messages. As a result, a trading strategy, it would be simpler to use an abstraction layer.

Market Data Processor

Market data processor performs normalisation on market data and provides a uniform view for the strategy regardless of the market that it is receiving data from.

3.4.4 Trading Venue

Each trading venue has three main modules: a matching engine, a number of client gateways and one or more market data publishers. Additionally, it includes a market control module that watches agents and market behaviour. It can impose high-level market controls like circuit-breaker or participant-specific controls like order-to-trade ratio limits. Different components of a trading venue is shown in Figure 3.11.

Matching Engine

A matching engine is the core component of a trading venue and handles trading for a single security. It maintains an orderbook and processes new orders, replace and cancel request and based on its matching rules and available orders in the orderbook can either match orders existing orders and report the fills back to participants or add or modify the orders in the orderbook.

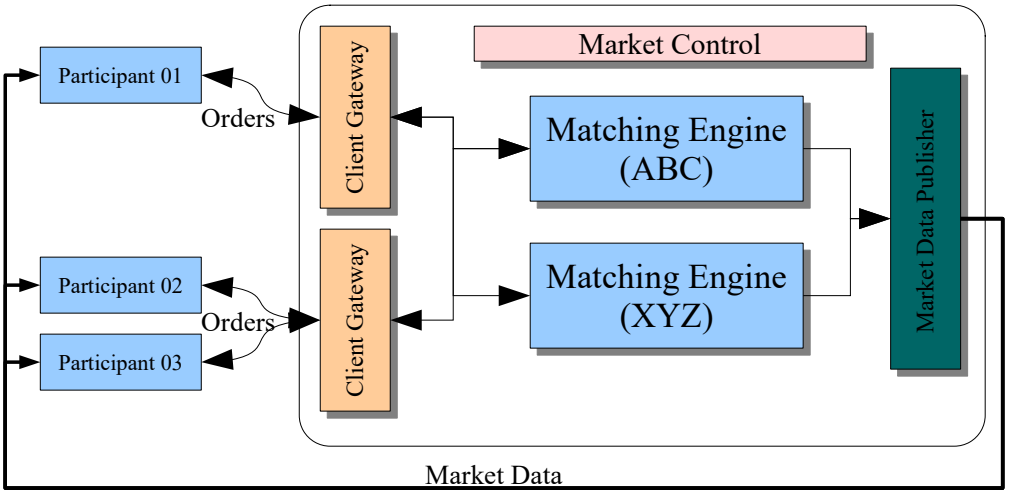


Figure 3.11: Trading Venue Architecture

Client Gateway

Client gateway handles the communication between market participants and matching engine. It enables the trading venue to support multiple communication protocols with the participants. For example, it can provide support for FIX protocol for general participants as well as OUCH protocol support for high-frequency and algorithmic trading participants. Gateway also handles low-level communication details like sequencing of messages, heartbeats, and replay of lost messages. Additionally, it can provide extra features like cancel on disconnect, *i.e.* cancel participant's existing orders in the orderbook if its communication link to the trading venue is disconnected.

Market Data Publisher

Market data publisher is responsible to publish the changes in the orderbook after any incoming order or update to the book back to the market participants and other listeners. While everyone communicating with matching engine is a market participant, there are agents listening to market data that to do not trade on that market and use this information to either calculate their PnL and risk or trade on other trading venues or markets.

3.5 Platform Implementation

This section discusses the implementation detail of the simulation platform that is developed for this research. Our research investigates how we can have an agent-based model that is as close as possible to a real market. As a result, it is important that the agent-based modelling framework is flexible enough to model agents that mimic the behaviour of market participants as close as possible. As we will be modelling a mixture of trading agents including high-frequency traders, it is also important that the platform has performance and scalability to simulate such models efficiently.

Listing 3.1: Strategy Initialisation

```
ZeroIntelligenceTrader::ZeroIntelligenceTrader()
{
    registerSubscriber(mg::OrderFill, &Strategy::onOrderFill, this);
    registerSubscriber(mg::OrderReject, &Strategy::onOrderReject, this);
    registerSubscriber(md::OrderbookUpdate, &Strategy::onOrderbookUpdate, this);
    registerSubscriber(md::Trade, &Strategy::onMarketdataTrade, this);
    registerPublisher(mg::NewOrder);
    registerPublisher(mg::ReplaceOrder);
    registerPublisher(mg::CancelOrder);
}

void ZeroIntelligenceTrader::onStartup()
{
    Strategy::onStartup();
    registerTimer(getNextPlacementTime(), this);
}
```

3.5.1 System Architecture

Because of high performance requirements of analysing tick by tick market data and the time it took for currently available platform to run the simulation, an specific platform has been build to run these simulation.

A trading engine designed for a high-frequency and low latency platform needs to meet a few requirement. Firstly, it needs to have high-throughput and be able to process high amount of information. Secondly, it needs to do its operation in very short amount of time, ideally in single digit micro-seconds. Thirdly, its code-base needs to be clear and testable. Recent events as shown in previous chapters have shown any mistake in a trading platform operating at high-speed could be very costly. Regulators have also increased their due-diligence and requirements on such platforms.

There is a trade-off between first two requirement; to increase throughput one can use multiple CPUs/cores on the same machine and distribute the operation on different cores. On the other hand, this can introduce latency. Distributing the operation on more than one core requires. At very high-speed each of the operation at very low-level matters. Two CPU cores running to same address page of the memory can cause performance hit because of cache miss on different memory cache levels. A simpler but high-performance version of such system has been implemented for this research.

3.5.2 Event Processing Core

Listing 3.1 shows a code snippet of such a setup showing how and strategy could be defined. All the strategies subscribers for order filled messages from exchange as well as market data updates published by trading venue. Exchanges also subscribe to reverse set of events and topics. Communications on the order channel (e.g. a OrderFill message) are one to one and messages are only received and processed by its target strategy who generated the original order. Market data messages (e.g. OrderbookUpdate) is a multicast message from exchange to all participants trading on that venue.

3.5.3 Simulation Platform

The agent-based simulation framework implements an event driven simulation core. Events are the only mechanism that trading agents and trading venue can interact with each other. Events are messages passed between two component in the system. To make market data more efficient, platform also support broadcast messages that are published to all components that are interested in that message type. Initially, each component registers type of events it is going to publish and type and source of events it is interesting in receiving. Each component in this system is going to be activated at start of the system to do an *init* step to initialise its internal state or send an message. System core also support special *timer* event which component can register to receive. System supports basic timer which means it only happens once. If a components needs to repetitive timer, it needs to register for another timer on the processing of each event. Simulation core itself is the source of these timer events. To allow a component to have different timer, each timer registration can include a *closure*[5] so that component can differentiate between different timers that it has registered.

This is similar to real-world algorithmic and automated trading platforms used by market participants. Trading strategy may have more inputs to subscribe to. It may subscribe to market data from other trading venues or other instruments trading on the same venue as well as news and other macro indicators.

3.5.4 Simulation toolkit

In our platform, each scenario is defined in a JSON file. To make our experiments repeatable, all the random variables use the same generator which initialised from a fixed seed. To assist managing these simulation configurations, they are all automatically generated from a scenario definition. The simulation setup used for an example case is shown in listing 3.2.

3.6 Exchange Data

To test our strategies we have used data from CME, NYSE and BATS Europe. This test data allows us to compare our results from a simulated market with similar measures for a real market. It is also used to test how good our zero intelligence trading agents behave in a more realistic environment against other intelligent traders.

To test our strategies, some market data from real markets is needed. This test data allows us to compare our results from a simulated market with similar measures for a real market. It is also used to study how good our zero intelligence trading agents may behave in a more realistic environment against real market trading agents. To test these strategies this research has used market data from Chicago Mercantile Exchange (CME), New York Stock Exchange (NYSE) and BATS Europe. This data is used to test trading agent strategies to compare the results from a simulated market with similar measures for a real market. This section provides only an overview of how this data is used in this research.

[5] https://en.wikipedia.org/wiki/Closure_(computer_programming)

Listing 3.2: Simulation ZI config

```
{
    "simulation" : {
        "start_time" : "2010-05-10 14:00:00",
        "end_time" : "2010-05-10 14:30:00",
        "runs" : 10
    },
    "symbols": [ { "ticker": "ABC", "tick_size": 1 } ],
    "exchanges": [ {
            "type": "price_time_priority_exchange",
            "name": "EX",
            "symbols" : [ "ABC" ]
    } ],
    "agents": [
        {
            "type" : "zero_intelligence_trader",
            "name" : "zit",
            "count" : 12,
            "symbol" : "ABC",
            "side" : { "type" : "fixed", "value" : {
                "type" : "bernoulli", "p": 0.5, "values", [-1,1] }},
            "size" : { "type" : "fixed", "value" : 1 },
            "price" : { "type": "uniform", "min" : 0, "max" : 200 }
            "order_lifetime" : { "type": "fixed", "value" : 1 },
            "position_manager": { "mode" : "risk_reducing" },
        },
        ...
    ],
    "event_logger": { ... },
    ...
}
```

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.50	9428.00	40	2
19	500	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21
14	300	9425.50	9430.00	150	12

Figure 3.12: 5-Level deep orderbook

3.6.1 CME

Chicago Mercantile Exchange (CME) is the largest future exchange in the world (Statistica 2016). It is also of particular interest to our research because the flash carash of May 2010 started at CME.

CME Market Data Platform supports three market data formats. This research uses MDP market data feed which is designed for algorithmic trading and high-frequency traders. MDP is sent to market participants over UDP[6] Multicast which means market data update will be sent to all market participants almost at the same time subject to network and connectivity delays. MDP messages are published with Simple Binary Encoding (SBE). SBE is based on simple primitive encoding, and is optimised for low bandwidth, low latency, and direct data access.

MDP is an event-driven messaging platform. Events can result from activities such as: Order Entry/Acceptance, Market State Changes, Start of Week Book Population. A single event will be represented by a series of messages sent per market data entry type, and the end of each event will be indicated in the last message for that event.

CME provides a multiple-depth book for most products. Figure 3.12 illustrate the mechanics of a 5-deep book. The book is represented by an equal number of rows in a table for each of the bid and ask sides. The rows indicate the quantity and order counts available at each price level.

CME maintains the Aggregate Depth view with the following data blocks:

- Add - create/insert a new price at a specified price level
- Change - change quantity for a price at a specified price level
- Delete - remove a price at a specified price level
  - An instruction of “Delete Through” - Deletes all book levels on one side of the book.
  - An instruction of “Delete From” - Deletes top ’n’ levels on one side of the book.

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). The Bid and Ask sides are updated independently with separate data blocks. The Trade Summary data is the first type of message sent on the market data feed for a trade. A Trade Summary message represents a distinct match comprised of all orders that traded together as the result of a single aggressing

[6]User Datagram Protocol

order. In general, if a trade occurs, CME will send a delete or change data block to update the book. The trade data block itself is not used to update the order book.

Beginning in January 2017, CME Globex enabled Market by Order functionality along with the Market by Price (MBP) functionality. Market by Order disseminates individual orders and quotes at every price level for the given instrument. This research does not use Market by Order data from CME.

Market Data Platform (MDP)

MDP 3.0 includes the introduction of Simple Binary Encoding (SBE) and Event Driven Messaging to the CME Market Data Platform. Simple Binary Encoding (SBE)(FIX Protocol Limited 2016) is based on simple primitive encoding, and is optimised for low bandwidth, low latency, and direct data access. SBE and event-driven messaging provide: Independence between number of events, messages, and packets, including: - Multiple messages per packet - A single event over multiple packets. Each packet contains a complete message as defined by the FIX specification, which allows client systems to start processing the message once the first packet is received. Fixed-length fields, which allow direct data access to fields in the message based on offsets and eliminate the need to parse entire messages.

It is an Event Driven Messaging platform. Events can result from activities such as: Order Entry/Acceptance, Market State Changes, Start of Week Book Population, Channel Resets and Recovery, and Statistics Generation. As market events occur, messages are sent in the real-time market data feed in packets containing FIX messages. Event-based market data is sequential per event (i.e., all messages for Event 1 will be processed and sent before any messages for Event 2 are processed and sent).

A single event will be represented by a series of FIX messages sent per market data entry type, and the end of each event will be indicated in the last message for that event. Messages within an event will be disseminated in a specific order by market data entry type. Each message will include an indicator which identifies whether there is more information for that type of message in the following packet(s).

Trade Summary

The Trade Summary data is the first type of message sent on the market data feed for a trade. A Trade Summary message represents a distinct match comprised of all orders that traded together as the result of a single aggressing order, elected stop order, mass quote, or a market state event. Under certain circumstances, there exists the possibility of multiple rounds of order matches in a given event, and as a result, separate Trade Summary messages sent for each round as described below. A single Trade Summary message can be split across multiple packets if the total number of related entries cannot be fit in a single UDP packet.

Multiple Depth Book

Figure 3.12 illustrate the mechanics of a 5-deep book. All books that are not top-of-book will use the same mechanics. CME provides a multiple-depth book for most products. Client systems must determine the book depth for an instrument from Security Definition message. The aggregate book reports summarised order quantities and order counts at a given price level. The depth represents the number of price levels supported via the market data feed. The book is represented by an equal number of rows in a table for each of the bid and ask sides. The rows indicate the quantity available at each price level. An aggregate depth book is sequenced by price, descending for bid and ascending for ask.

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.50	9428.00	40	2
19	500	9427.00	9428.50	600	35
34	750	9426.50	9429.00	850	55
25	400	9426.00	9429.50	350	21
14	300	9425.50	9430.00	150	12

Figure 3.13: Example of CME 5-level deep orderbook

CME maintains the Aggregate Depth view with the following data blocks:

- Add - create/insert a new price at a specified price level
- Change - change quantity for a price at a specified price level
- Delete - remove a price at a specified price level
  - An instruction of “Delete Through” - Deletes all book levels on one side of the book.
  - An instruction of “Delete From” - Deletes top ’n’ levels on one side of the book.

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). All data blocks are issued for a specified entry type, price, and price level. The incremental instruction approach assumes the use of the Market Data Incremental Refresh message. The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

CME sends an add data block if there is a new price level. Client systems should then shift price levels down, and delete any price levels past the defined depth of the book. CME sends a delete data block to remove a price level in the book. Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If available, an add data block will be sent to fill in the last price level. The change data block is sent to update characteristics of a price level without changing the price itself, or impacting any other prices on the book. The change data block is sent to update the order count and / or quantity for a price level. The change data block is not sent when the price changes at a given price level.

In general, if a trade occurs, CME will send a delete or change data block to update the

book. The trade data block itself is not used to update the order book.

Beginning in January 2017, CME Globex will enable Market by Order (MBO) functionality along with the current Market by Price (MBP) functionality. MBO disseminates individual orders and quotes at every price level for the given instrument. MBO will improve transparency of markets and allow customers to view position(s) while preserving market participant anonymity.

3.6.2 NYSE

New York Stock Exchange (NYSE) is the major US equity exchange. NYSE is currently owned by Inter-Continental Exchange (ICE). NYSE provide a number of different market data feed, for historical reasons or feeds targetted at different class of users (NYSE 2016). Two of the major feeds used for algorithmic trading are Arca Integrated Feed and ArcaBook. Other than live data feed for real-time trading, it provides extra information overnight known as Trade and Quote or TAQ which could be used for research and optimization of the algorithms.

NYSE ArcaBook shows the full limit order book for NYSE Arca traded securities on a real time basis as well as information to NYSE Arca opening, closing, halt auctions, and indicative match price/volume, auction imbalance, and market imbalance data. NYSE ArcaBook is disseminated through a direct data feed originating from NYSE SFTI network utilising a multicast feed.

NYSE Arca Integrated Feed is a real time data feed that provides a unified view of events, in sequence as they appear on the NYSE Arca matching engine. The data feed includes depth of book (with add, modify, delete orders), trades (with corrections and cancel/errors), order imbalance data, and security status messages. This product was introduced in 2011 and was not available at the time of May 2010 flash crash.

New York Stock Exchange (NYSE) is the major US equity exchange and is currently owned by Inter-Continental Exchange (ICE). It provide a number of different market data feed, for different historical reasons or feeds targetted at different class of users (NYSE 2016). Two of the major feeds used for algorithmic trading are Arca Integrated Feed and ArcaBook. Other than live data feed for real-time trading, it provides extra information overnight known as Trade and Quote or TAQ which could be used for research and optimization of the algorithms.

NYSE Arca Integrated Feed

NYSE Arca Integrated Feed is a real time data feed that provides a unified view of events, in sequence as they appear on the NYSE Arca matching engine. The data feed includes depth of book (with add, modify, delete orders), trades (with corrections and cancel/errors), order imbalance data, and security status messages. The product is not shaped nor throttled, so this value added data feed requires customers to establish connectivity that is sufficient to support a substantial increase in data content and bandwidth. NYSE Arca Integrated Feed helps customers overcome the challenges of determining event sequences from disparate NYSE Arca data products, increasing transparency of the specific quotes that drive trades.

NYSE ArcaBook

NYSE ArcaBook shows the full limit order book for NYSE Arca traded securities on a real time basis. Also included in this product are data elements providing information to NYSE Arca opening, closing, and halt auctions, as well as indicative match price, match volume, auction imbalance, and market imbalance data.

All of the data is disseminated through data vendors or through a direct data feed originating from NYSE SFTI network utilising a multicast feed. This product enhances market transparency and provides consumers with a complete liquidity picture from one of the leading US marketplaces.

3.6.3 BATS

BATS is one of the recent exchanges registered in US and also owns BATS Europe which is one of the largest MTFs in Europe. It is famous for its state of the art technology, and fast matching engine which makes it an interesting trading venue for high-frequency trading participants. BATS publishes detailed information about their trading infra-structure performance and its latency.

BATS publishes market data via two interfaces, PITCH and Multicast PITCH. They are similar in terms of the content and differ mainly on the technology used for publication. BATS also publishes detailed information about their trading infra-structure performance.

PITCH vs Multicast PITCH Feed

PITCH feed is distributed via TCP protocol, which is point-to-point and each customer need to make a connection to one of the assigned market data publishers at BATS. On the other hand, multicast PITCH is published on UDP multicast which is one to many protocol.

First difference is on the encoding of the fields in the messages. PITCH feed encodes fields in fixed-length string format while Multicast PITCH uses binary encoding. For example price in a PITCH message is a string of ASCII numbers with four decimal points; a price of 100.21 is encoded as string “1002100” on the PITCH feed while it is encoded as binary form on Multicast PITCH. This makes Multicast PITCH faster to decode as participant does not have to parse strings to convert it to numbers and it is also smaller. Table 3.14 shows an “Add Order” messaage sent on the PITCH feed. This message is 45 bytes long while the equivalent message on the Multicast feed is only 25 bytes long and needs less parsing of strings (See Table 3.16).

The second difference is on timestamp of the messages. PITCH uses millisecond resolution timestamps from midnight on the messages and each message carries full timestamp. On the other hand, multicast PITCH uses a nano-second resolution timestamp but instead of sending the full timestamp in each message, it sends “Time” (See Table 3.15 messages periodically and each message only has the offset from last time message in nano-seconds.

Field Name	Offset	Length	Value	Description
Timestamp	0	8	Timestamp	Timestamp
Message Type	8	1	"A"	Add Order Message
OrderID	9	12	Base 36 Numeric	Day-specific identifier assigned to this order.
Side Indicator	21	1	Alpha	B = Buy S = Sell
Shares	22	6	Numeric	Number of shares being added to the book (may be less than the number of shares entered).
Symbol	28	6	Alphanumeric	Symbol, right added with spaces. Common Symbology Format.
Price	34	10	Price	Order price.
Display	44	1	"Y"	Always "Y". Maintained for compatibility reasons.

Figure 3.14: Multicast PITCH Add Order Message

Time				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x20	Time Message
Time	2	4	Binary	Number of whole seconds from midnight London time
Total Length = 6 bytes				

Figure 3.15: Multicast PITCH Time Message

Orderbook Update Messages

In this section we introduce important messages sent by BATS. The aim of this section is to provide enough details to understand the semantics of the feed and as a result this is not a comprehensive list of messages. The full details of the messages and fields can be found in the PITCH feed specification (BATS Europe 2017b) or Multicast PITCH specification (BATS Europe 2017a).

Add Order Message Add order message informs participants that an order by a participant has been added to orderbook. This is sent when an incoming order from participant cannot be matched against existing orders in the orderbook. It has a "long" used on European market that has a longer price (8 bytes) and quantity (4 bytes) fields.

Add Order — Short				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x22	Add Order Message — Short
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Day-specific identifier assigned to this order
Side Indicator	14	1	Alphanumeric	B = Buy Order S = Sell Order
Quantity	15	2	Binary	Number of shares being added to the book (may be less than the number entered)
Symbol	17	6	Alphanumeric	Symbol right padded with spaces
Price	23	2	Binary Short Price	The limit order price
Total Length = 25 bytes				

Figure 3.16: Multicast PITCH Add Order Message

Order Executed Message Order executed message informs participants that an order that was in the orderbook has been matched against an incoming order. This message does not specify the price as it is assumed that participant are aware of the price of this order from previous

add order or modify messages. This message also specifies only quantity that is matched in this specific occasion. Market can send different order executed messages at different time when an order is partically matched against incoming smaller orders. It is up to market participants to maintain the remaining size of the order in the orderbook from the sequence of messages received from market.

Order Executed				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x23	Order Executed Message
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Order Id of a previously send Add Order Message
Executed Shares	14	4	Binary	Number of shares executed
Execution Id	18	8	Binary	Bats generated day-unique execution identifier of this execution. Execution Id is also referenced in the Trade Break Message.
Execution Flags	26	4	Alphanumeric	Type flags based on MMT v3.04 standard
Total Length = 30 bytes				

Figure 3.17: Multicast PITCH Execute Order Message

Reduce Size Message Reduce size message informs participants that an order size has reduced by the participant. Reducing quantity maintains order priority in the queue. It has a "long" used on European market that has a longer shares (4 bytes) field.

Reduce Size — Short				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x26	Reduce Size Message — Short
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Order Id of a previously send Add Order Message
Cancelled Shares	14	2	Binary	Number of shares cancelled
Total Length = 16 bytes				

Figure 3.18: Multicast PITCH Reduce Size Message

Modify Order Modify order message informs participants that an order size or price has changed. This message affects the priority of orders in the orderbook. It has a "long" used on European market that has a longer price (8 bytes) and shares (4 bytes) fields.

Modify Order — Short				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x28	Modify Order Message — Short
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Order Id of a previously send Add Order Message
Shares	14	2	Binary	Number of shares associated with this order after this modify (may be less than the number of shares entered)
Price	16	2	Binary Short Price	The limit order price after this modify
Total Length = 18 bytes				

Figure 3.19: Multicast PITCH Modify Order Message

Delete Order Message Delete order cancel message informs participants that an order has been cancelled in full by the participant.

Delete Order				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x29	Delete Order Message
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Order Id of a previously send Add Order Message
Total Length = 14 bytes				

Figure 3.20: Multicast PITCH Delete Order Message

Trade Message Trade message is sent to inform market participants that an incoming order has been match against an order that has not been visible on market data feed before. For example when an incoming order is matched against a hidden order. It has a "long" used on European market that has a longer price (8 bytes) and shares (4 bytes) fields.

Trade — Short				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x2B	Trade — Short
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Order Id	6	8	Binary	Obfuscated Order ID or Order ID of the non-displayed executed order
Side Indicator	14	1	Alphanumeric	Always B for hidden trades.
Shares	15	2	Binary	Incremental number of shares executed
Symbol	17	6	Alphanumeric	Symbol right padded with spaces
Price	23	2	Binary Short Price	The execution price
Execution Id	25	8	Binary	Bats generated day-unique execution identifier of this trade. Execution Id is also references in the Trade Break Message.
Trade Flags	33	5	Alphanumeric	Type flags based on MMT v3.04 standard
Total Length = 38 bytes				

Figure 3.21: Multicast PITCH Trade Message

Trading Status Message Trading status message is sent to inform market participants that security's trading status has changed; for example when it starts "trading" or when it is "suspended".

Trading Status				
Field	**Offset**	**Length**	**Data Type**	**Description**
Length	0	1	Binary	Length of this message including this field
Message Type	1	1	0x31	Trading Status Message
Time Offset	2	4	Binary	Nanosecond offset from last unit timestamp
Symbol	6	8	Alphanumeric	Symbol right padded with spaces
Status	14	1	Alpha	T = Trading R = Off-Book Reporting C = Closed S = Suspension N = No Reference Price V = Volatility Interruption O = Opening Auction E = Closing Auction H = Halt[1] M = Market Order Imbalance Extension P = Price Monitoring Extension
Reserved1	15	3	Alpha	Reserved
Total Length = 18 bytes				

Figure 3.22: Multicast PITCH Trading Status Message

3.7 Summary

This chapter discussed the details of the trading agent simulation platform used in this research to perform different experiments. First, it described the general system architecture providing details on the internal design of trading agents and trading venues. Then, is reviewed available agent-based modelling and complex event processing platforms. The aim was to evaluate their suitability for this research. Those frameworks proved to be either lacking performance needed to simulate trading of an exchange with many market participants and high rate of orders. In case of FLAME, we lacked on the flexibility of matching to our use-case and the cost benefit balance pushed us in the direction of developing a new platform. This platform is based on the same design patterns that a high-frequency trading platform or a high-speed exchange matching engine use. The review of available platform resulted in a decision to build a new system and this chapter provided information on the design of that platform. Finally, it discussed the market data that is used in this research.

Chapter 4

Agent-Based Model of the Flash Crash

4.1 Introduction

This study proposes to use agent-based modelling as a tool to analyse the flash crash. Section 2.7 provides an overview of agent-based modelling and previous research into its applications into financial markets. We briefly touch upon using intelligent agents for trading as it provides a basis for our models of market participants. The main focus of this section is research that focuses on modelling a whole market and specifically modelling the flash crash.

This chapter describes how this research uses agent-based modelling to analyse the collective behaviour of trading agents in a financial market setting. The focus is to understand how a flash crash happens and what properties in the agents and market affect it. It aims to provide a model that not only represents the properties of the financial market itself but also trading agents provide a reasonable proxy to market participants and their behaviour in a real financial market. The target of this modelling is to provide a base for the next chapters where this research investigates the changes in properties of agent and market model and studies how these changes affect the financial market.

Section 4.2 describes the agent model that is used as a base and studies its characteristics. Market participants are classified into a number of categories that represent their different behaviours. To show our models are representative of a real trading agent, Section 4.3 verifies artificial agent-based models using data from the real market. Section 4.4 builds a market using only simulated agents and show such a model presents similar properties that can be observed on a real trading venue. Section 4.5 presents how this model can be used to simulate the flash crash. Section 4.6 provides a summary of the discussions presented in this chapter.

4.2 Agent Model

We want to come up with a model for a trader agent would be as simple as possible but at the same time replicate market properties as closely as possible. We start by the simplest model of an agent, a zero intelligent trading agent with no limitation. One of the simplest agent-based models of trading market is presented by (Gode and Sunder 1993). It was designed to analyse how much of efficiency of the market can be attributed to human traders and how much comes from rules of the market. It is based an agent-based model to test if hypothesis presented by (Becker 1962) that "*Households may be irrational but yet markets quite rational*" and that we should not impute all observed irrationalities of individuals to markets or to impute all rationality of markets to their participants.

It compares three scenarios: In the first group it has human traders trading rationally driven by profit and having strategy; a second group consists of artificial "*zero intelligence*" (ZI) machine traders with no strategy that are trading randomly, and the last group of zero intelligence traders with "*budget constraints*" (ZI-C). This experiment was performed on a double auction market which is not continuous, *i.e.* as soon as a trade happens, all outstanding orders in the orderbook will be cleared and trading resumes from an empty orderbook.

4.2.1 Budget-Constrained Zero-Intelligent Traders

To have a baseline we have implemented similar agents both with and without budget constraints. For simplicity, we assume one agent can place an order of size one unit. Agents trade in fixed time intervals and at each step, a random trader is chosen. The chosen trader will place a buy or sell order (with equal probability). The price of the order is uniformly distributed in a range of possible prices for the asset (0-200 in our experiment). All orders are considered to be limit orders. Figure 4.1 shows a sample data from a run of the model. As can be seen from the graph, price movement seems to be quite random in a very wide range, it does not seem to stabilise to an equilibrium price.

In a budget-constrained zero-intelligence trader, the trading agent already has a long or short position and can only trade toward reducing its position. For example, if it has a long position, it also has a cost associated with each unit of asset acquired and cannot trade at a loss. As a result, if it is long and selling the unit u_i with associated cost p_i, it can place a sell order asking at a price that has to be greater than p_i. Gode and Sunder experiments, put an order into units of asset that each agent holds and agents are forced to trade (Gode and Sunder 1993). Agents are limited to trade units in that specific order. That experiments allocates positions from a liquidity curve that represents the whole of the market. In our experimental setup, agents can place their order at random times but can only trade one unit of asset at a time.

Figure 4.2 shows a sample data from a run of the constrained model. The price constraint limits the higher-bound for offer price for buyers or lower-bound for the ask price for sellers as trading agents are prevented by the constraints from trading at a loss. As can be seen from

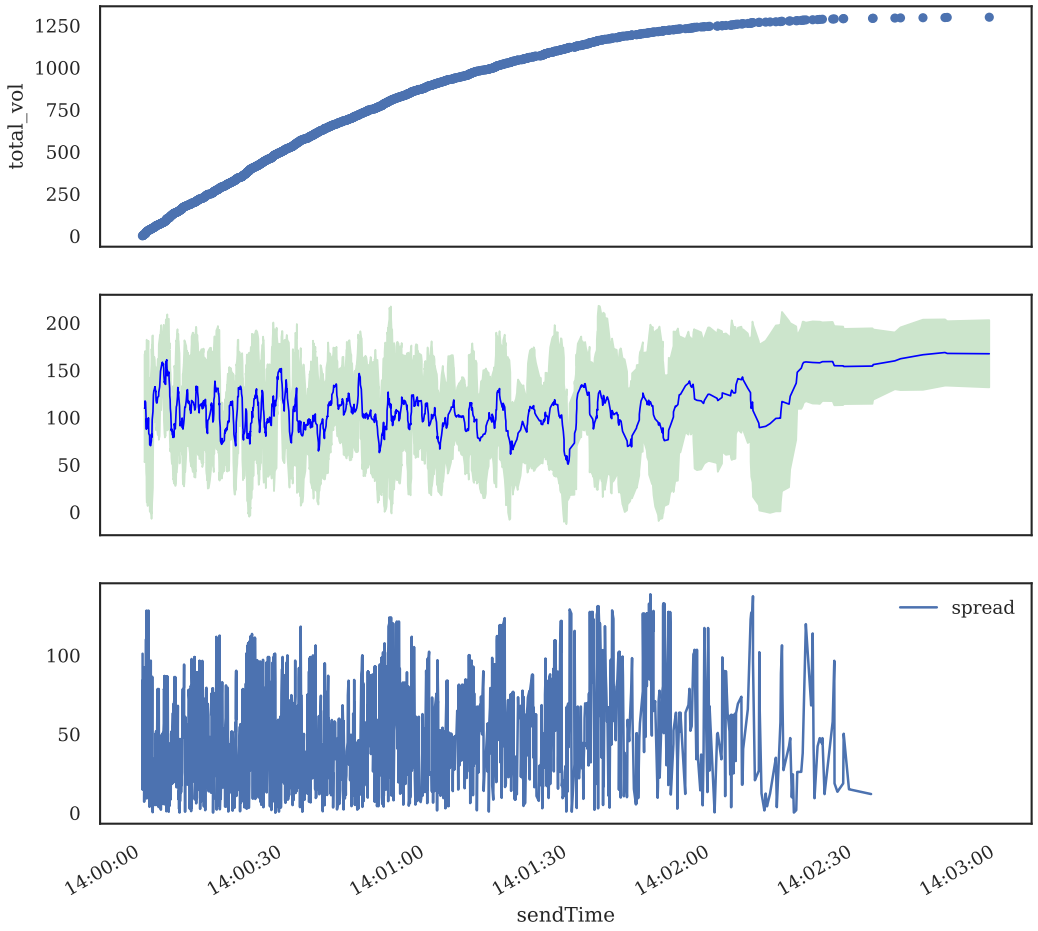


Figure 4.1: An example of a market with ZI traders with no constrains

the graph, the range of price movement is significantly smaller compared to the non-constrained graph. Also, trading price converges to an area around the equilibrium.

Figure 4.3 shows total liquidity available by limit price of each agent. This gives an indication that if we were to trade all the orders in a single auction, what would be the volume of this crossing auction.

Looking at a few more examples of the liquidity constraint graph, we can see this behaviour is almost repetitive even when we have imbalanced liquidity curves. In this experiment, each trading agent is either a buyer or seller and not both at the same time. Their decisions about the direction of the trade is taken regardless of current trading price, and even when there is liquidity imbalance, *e.g.* there is more liquidity to sell than there are buyers, trading cannot happen with this simple model. Also, as the agents do not take into account last trading price (as they do not have memory) price movement does not put a feedback look into the future decision of trading agents.

4.2.2 Price-aware Zero-Intelligent Traders

Models we have tested so-far can produce a market that has some characteristics that are close to real-markets but still lack some of the requirements of modelling agents that previously

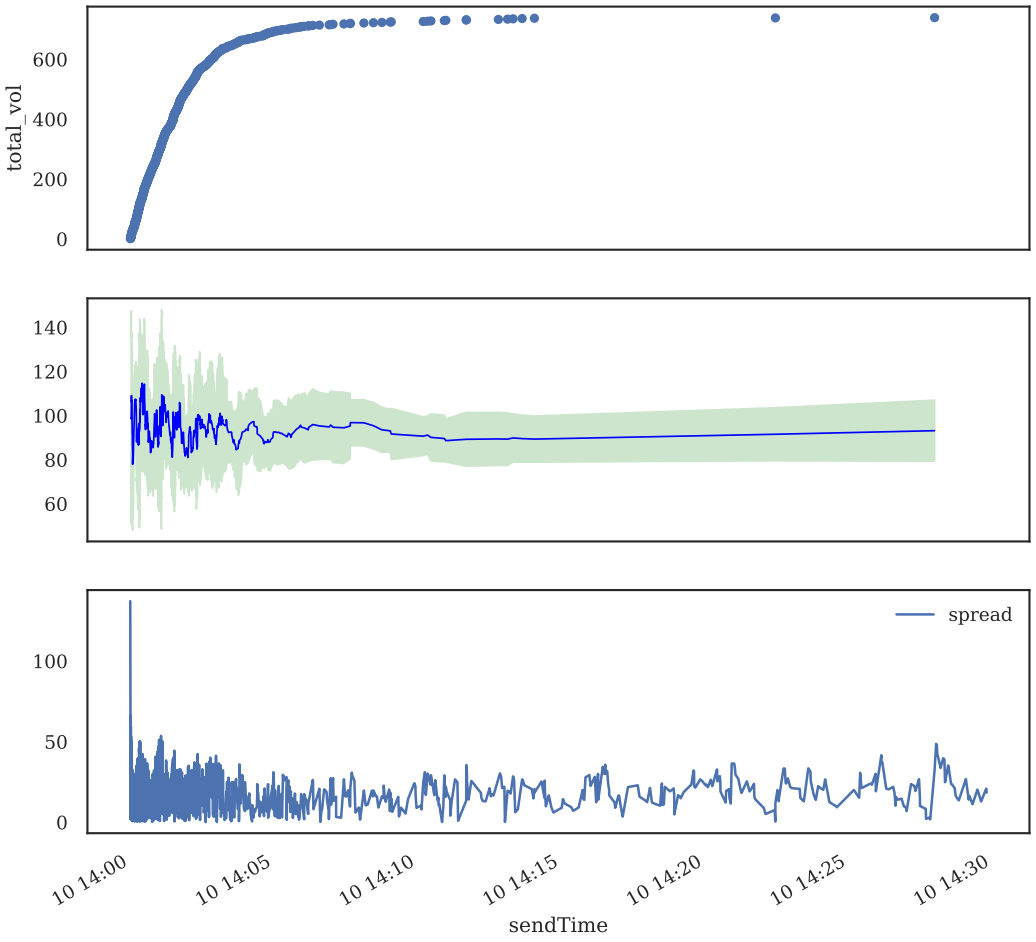


Figure 4.2: An example of market with ZI traders and price constrains

described as contributing to flash crash. One of the missing characteristics is the lack of view from the current trading price of the market. In this section, we look at the ways that current trading price can be incorporated into a zero-intelligence model. There are two ways that market price can be included in a simple trading agent. First option is to trigger its decision to trade and possibly its direction. The second option is to choose price of new order relative to current market trading price.

For the first case, one well-suited example from real market is *stop-loss* or *take-profit* orders. Stop-loss orders have been claimed to be one of the contributors of the flash crash. An stop-loss order is issued by a trading agent who has already bough an asset and has a long position. This trade has been done with the view that the price of that asset is more likely to go up in the future bringing in profit for the trader that holds that long position. This prediction may not always happen and and the price may as well go down against the prediction. On the other hand, this agent would normally had bough with a long term view and it is acceptable to have comparably small price fluctuations but need a way to hedge its risk against market moving completely in a different direction. Such trader can issue an stop loss order which is (possibly multi-day, long term) limit order far below current trading price of the market. This would put

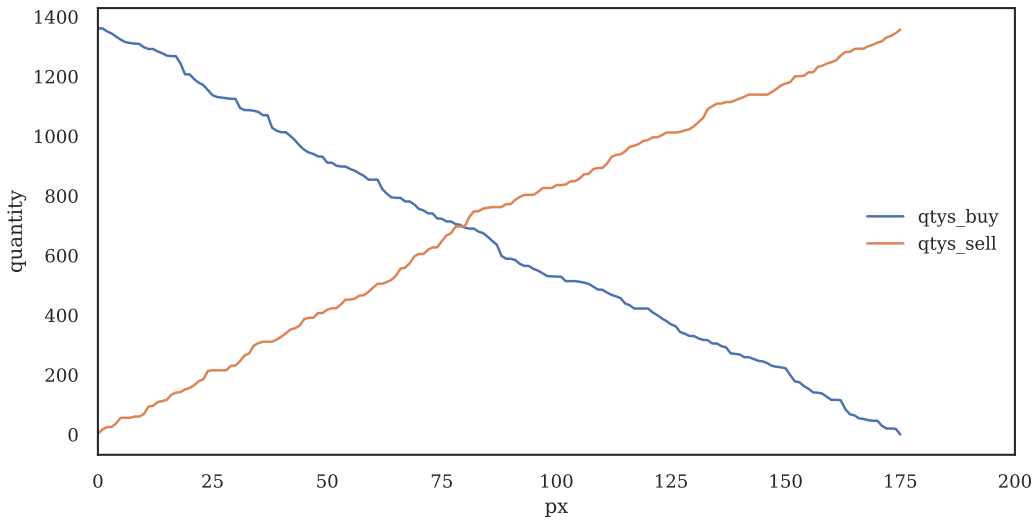


Figure 4.3: Cummulititative liquidity at each price level for price constraint ZI

a lower bound on the price this asset would be sold at and limits the agent's loss. For example, if trader is buying ABC at $100 assuming this will go to about $110. To limit its risk, it will put an stop-loss order at $85. This means if the prediction about price going up turns out to be in-correct, trader will sell its holding at $85 and will limit its loss at $15 per asset unit. There are opposite, take profit orders that trader can also put a take profit order at $115 for example, assuming that the price will not go above this and this is very good profit to make as if this order is not there automatically and trader misses the time to sell its holding it may loose the chance and price may go down again.

In our investigations, stop loss orders are more important than take profit orders. Stop loss orders are in the same direction as the market price move. If price is going down, it normally means there are more agents willing to sell than agents willing to buy. When the price passes below the limit on stop loss orders, it triggers more sell orders adding to the market pressure. Take profit acts on the opposite direction as when the price is going up, this will issue a sell order and can provide more liquidity to the market. When the market goes above/below stop-loss price the trading agent holding that position will issue a market order to trade that position and limit its risk. The stop-loss limit is assumed to be the same for all the orders with has similar cost associated with them, as a result in out example it is a line shifted below or above the liquidity constrain line.

Figure 4.4 shows data from a sample run of the constrained model with stop-loss limits. As it can be seen from the graph, range of price movement is wider with more spikes for this model of trading compared to the model were there were no stop-loss constraints.

Second way that a trading agent can incorporate current market trading price into its decision making process. For example, many of market participants place their orders relative to current best available orders in the market. It has been reported for example that, high-frequency market makers place most of their orders at the best bid/ask price or within

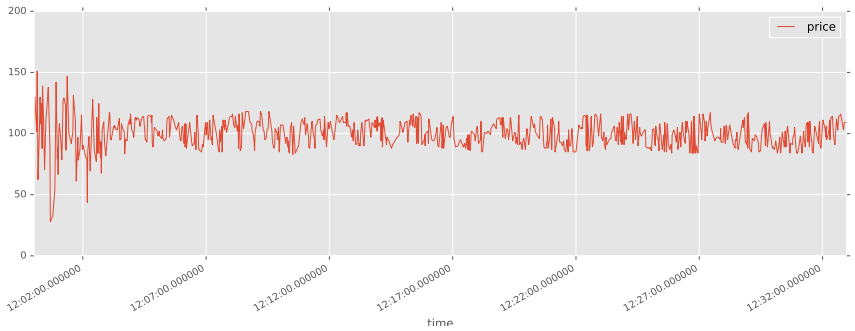


Figure 4.4: An example ZI-C with stop-loss trader

one price-tick away from last traded price. Current trading price and its history can also be used by a market participant as a source of information to decide no only on the limit price of the new order it is going to place but also on the decision to trade and its direction as well. We have previously reported about a research into modelling of flash crash that uses momentum and mean-reversion based signals for non-high-frequency trading agents based on the history of traded price. These are methods with no memory from their past behaviour.

Before we go into more complex models, we want to investigate whether the market model arising from such trading agents we discussed so-far have representation of real-market agents. For this purpose, we will look into the behaviour of our simple trading agents in a real market.

4.3 Verification of Agent Model

In this section, we perform experiments to investigate how agent models we have for each type of traders behave when they are placed in a real market environment. Regulation does not allow an agent with no strategy or purpose to be put into a real trading environment and we need to come up with a way of co-simulating our agent models with data from real-markets. To this end, we need to make some assumptions about the data and behaviour of other participants in a real market. Making assumptions for execution of an agents order is not specific to our case. When a financial institution develops a new strategy or algorithm for trading they need to have a reasonable idea about its performance in real-market before they put that into trading. As part of this, it needs to be back-tested against previous market data. As the first step, this implies an assumption that an strategy that could have worked in the past is going to work in the future as well. It also needs to make assumptions about the feedback and change in the behaviour of other market participants which is not going to be easy to simulate. Normally, these initial assumption would be compensated by confidence interval around projected profit and loss (PnL) or adjustment of the expected fill ratio. In the next two sections, we show a number of methods that can be used for this purpose.

4.3.1 Execution of Market and Aggressive Limit Orders

As described before, market order is a type of order when participant does not specify a price and is willing to trade at the price available on the other side of the orderbook. A limit price is

considered aggressive, when its prices is higher/lower that the orders available on the opposite side of the orderbook. Simplest case to simulate an agent interacting with market is to limit the artificial agent to place only market orders. When a participant places a market order, we can fill that order with available liquidity on the opposite side of the book. This indirectly assumes that we have zero latency receiving market data from the market and also on the outgoing market orders into the market. These assumptions are not realistic. There is latency on receiving and processing market data and also one way back when placing an order into the market. There is also latency in calculating of the decision in the trading strategy. If this agent and its communication channels to receive market data and send orders to exchange is not zero or at-least the fastest compared to other market participants, there is a chance that another market participant has placed an order in the same direction and has traded against available orders in the orderbook before our order arrives at trading venue. The result could be that the order will be rejected or could possibly trade at worse price than initially expected.

Another complexity with this method of mixing real market data with artificial agent is that the data needs to be kept consistent in the view of agent placing the order and other agents that are watching and processing the orderbook. We need to make sure market data received by the trading agent placing that market order changes in-line with the model deciding about the execution of the order. If an trading agent has an strategy to place an order in a certain scenario in the market and places an order and gets filled; it should also see the corresponding market data change, *i.e.* that top of the price or the quantity related to its executed in the market should be deducted from the orderbook that is visible to strategy. Otherwise, it could end-up with an infinite loop of repetitive orders. This is also a problem for high-frequency trading systems as they receive and process their fill message on the order channel before trading venue itself has a chance to update its market data. Such trading systems needs to incorporate their order placement into their market data feed to have a consistent view of orders available in the orderbook.

For our simulation, we assume that all of the orders of this agent is going to be fully executed. To keep agent view of orderbook consistent, We modify the orderbook visible to the agent and reduce the corresponding quantity and price levels from the orderbook visible to the strategy. We are also making assumption that this execution (which could be different to the execution that happened in the market for the corresponding order) does not change the behaviour of other market participants. This means we are not going to modify any other market data other than executed quantity. Figure 4.5 shows a sample orderbook with 4 orders. If our artificial trading agent places a market order x010 to buy 400 shares, its order will be executed straight-away and order x001 will be reduced to 600 shares remaining in the orderbook (as it is visible to our strategy).

Bid				Ask			
OrderId	Time	Size	Price	Price	Size	Time	OrderId
x002	08:00:00.003	500	160.90	161.00	1000	08:00:00.001	x001
				161.00	1500	08:00:00.006	x004
				161.10	500	08:00:00.004	x003

Figure 4.5: Sample orderbook

4.3.2 Execution of Passive Orders

The approach discussed above works for market orders or aggressive limit orders that can be traded straight-away. For limit orders that are passive, interacting with data from a real market is more complex.

One solution that we are going to use is to consider orders from our artificial agents like hidden liquidity in the orderbook. When we add our order to the book, it will be added to the orderbook at its proper position based on the matching algorithm of the orderbook. In case of a price-time priority matching algorithm, the order will be added to the end of orders with the same price. The order will be then executed inline with the orders after this order. This model is also making same assumptions as above about the effect of our order in the behaviour of other market participants.

Assuming the same orderbook as in Figure 4.5, if the strategy places a passive sell order x010 with limit price of 161.00 to sell 500 shares, it will be placed in the orderbook after order x004. Then if we receive orders on the other side, for example two market orders for 1500 and 2000, assuming nothing else has changed in the meantime, the first order would be executed against orders x001 and 500 shares from x004 and the second order of 2000 shares will be matched 1000 shares with x004 and also executes our order x010 of 500 shares before going to match against x003.

4.3.3 Market Data as Super Trader

One final option for mixed simulation of data from real-market with artificial agents is to run a normal orderbook and process all the orders from market data similar to orders from real-market. In this example, if we place a passive order into the book similar to the case we discussed in the previous section. When there is an aggressive order x021 to match against the orderbook, it will execute against order x004 which was before our order and as it will be fully filled matching against our order, order x005 sitting behind our order will not be executed.

4.4 Verification of the Market Model

Second step is compare stylised facts between real market price and simulated market (Cont 2001). The first set of properties are on the trading price of the asset in the market. We have verified that Absence of autocorrelations: *(linear) autocorrelations of returns* are insignificant, except for time scales less than 10 minutes for which microstructure effects come into play. The unconditional distribution of returns is normal and seems to display a *fat tail*. As we increases

the time scale δt over which returns are calculated, their distribution looks more and more like a *Gaussian distribution*. In particular, the shape of the distribution is not the same at different time scales.

On the other hand when we look at the available liquidity in the orderbook, the shape of order-book resembles "V" shape

4.5 Simulation of Flash Crash

Simulation of flash crash is done as follows: An agent is introduced to the model, which tries to sell a large number of contracts. The agent examines the previous minute of trading and executes an aggressive sell order for 9% of the trading volume. Market makers and high-frequency traders are constrained by a rule, which forces them to lower their position level if they reach their position limit. Additionally, market makers were calibrated to withdraw from the market if the price falls 24 ticks below the moving average. Furthermore, fundamental traders withdraw from the market and stop loss orders are triggered if the price drops 70 ticks below the starting price. It is shown that this event provides similar volume and price profile to actual flash crash.

4.6 Conclusion

Later we looked the modelling of the agents and how a zero-intelligence agent model can be used to simulate different classes of market participants. Finally we looked at how a flash-crash can be replicated in our simulated environment.

Chapter 5

Diversity of Trader Population

5.1 Introduction

A financial market is a place where possible buyers and sellers of an asset (*e.g.* stocks or bonds) are brought together to allow them to trade with each other. A financial market is designed to facilitate the allocation of resources in society by matching the needs of different market participants with each other. In a primary market, a company that needs to spend the capital to develop a product and bring revenues in the future is matched with an investor that has the capital and is looking for income in the future. A secondary market allows participants to exchange the assets they have invested in. Investors participate in the secondary mark because either their needs have changed or they are looking to switch to a new asset because that asset better meets their investment targets. Such investors that participate in the market with a long-term view are fundamental traders.

Any trade involves a buyer and a seller that need to be available at the same time and can agree on the size and price for the trade. For a financial market to perform efficiently, it needs to bring together fundamental buyers and sellers that are getting in or out of a position. The needs and investment targets of such fundamental traders do not change very fast. Thus, when a fundamental trader decides to trade may not match the exact time another fundamental trader is available on the opposite side of the trade. Market makers are supposed to fill this timing gap by providing liquidity to market when needed which in turns can help to improve both price discovery and lower transaction costs for fundamental traders. There are also market participants that see such short-term supply and demand imbalance as an opportunity that they can benefit from and trade with the aim of profiting from this situation. If there is no other participant that is willing to participate on the opposite side of the transaction, no trading can happen. Therefore, it is important there are a *diverse* set of market participants to maximise

the possibility of trades happening.

During the flash crash, this supply and demand between buyers and sellers broke down. The official report by (CFTC and SEC 2010a) argues that a *fundamental seller* that was trading at unreasonably high speed was responsible for starting the flash crash. It pushed a large number of sell orders in a short period of time when there were not enough buyers. It also names *high-frequency traders* as a type of participants that contributed to the flash crash by becoming aggressive sellers when their portfolio reached their risk limits, and their sell pressure pushed the market further down. In this chapter, we are going to investigate if the diversity of traders' population can affect a financial market's ability to deal with a liquidity crisis similar to the flash crash.

A fundamental trader has been claimed to be responsible for starting the flash crash. Section 5.2 investigates how an increase in the population of fundamental traders compared to rest of market participants can affect the behaviour of a financial market. We also study the effect of the change in the population of fundamental buyers vs. fundamental sellers on the market's response to a liquidity crisis. High-frequency traders have been reported to have exacerbated the flash crash by becoming aggressive sellers when they hit their risk limits. A recent study by TABBS group suggests more than 60% of US equity trading happens with high-frequency traders. That trend is closely followed in European equity markets and other highly liquid trading venues like Futures. This heavy presence has caused concerns that any problem with high-frequency traders can have a dramatic effect on the financial market and its participants. We study the effects of an increase in the population of high-frequency traders compared to rest of market participants on the market's response to a liquidity shock in Section 5.3. During the flash crash, high-frequency traders became aggressive sellers when their risk-limit have breached. To this end, we further investigate the effect of high-frequency trader risk-limit on the market's response to a liquidity crisis. Section 5.4 summarises the findings of this chapter.

5.2 Fundamental Trader Population

We have previously categorised traders based on their trading pattern during the flash crash into six categories: high-frequency traders, intermediaries, fundamental buyers, fundamental sellers, noise traders and opportunistic traders (See Section 2.2.1 for details). Fundamental traders (buyers/sellers) in that classification were traders that have been trading in the same direction all day and have accumulated a significant position in that direction by the end of the day. This is based on the assumption that fundamental traders have an investment horizon of more than a day and large capital to invest.

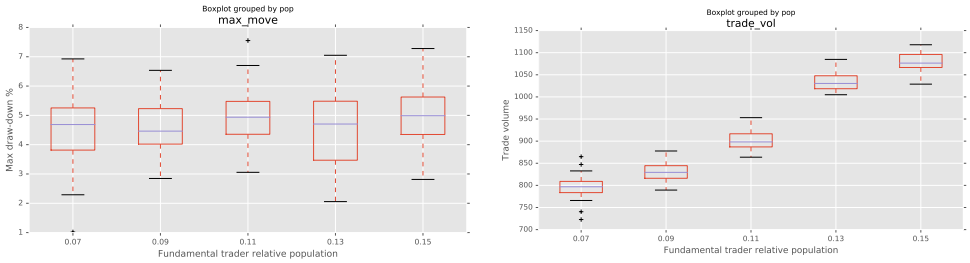
In general, *fundamental trading* refers to investors that evaluate securities by attempting to measure the intrinsic value of a security by looking at economic factors, known as fundamentals. Such a trader decides on buying or selling shares of a company by studying financial reports of the company to understand its earnings, expenses, tangible/intangible assets and liabilities. Tangible assets include land, equipment or buildings that a company owns. Intangible assets, on

the other hand, include non-physical assets such as trademarks, patents, branding or intellectual property. A fundamental trader also looks into the overall global economy, country and industry condition, as well as the company's management and product line to estimate a true value for the shares. This analysis results in a value assigned to the security that is compared to the security's current price. Fundamental traders use the comparison to determine is underpriced or overpriced and based on that they decide to buy or sell it.

Technical trading is used as opposite of the fundamental trading. Technical traders, on the other hand, believe there is no reason to analyse a company's fundamentals because they assume this information is incorporated in the company's share price. Technical traders evaluate securities using statistics generated by market activity, such as past prices and trading volume. Technical traders do not attempt to measure a security's intrinsic value. Instead, they use historical price movements of the security and overlay charts to identify patterns and trends to predict its future price movements.

Fundamental analysis and technical analysis are normally used for different timeframes. Technical analysis can be utilised on a timeframe of weeks, days, and even minutes. On the other hand, fundamental analysts often look at data over a number of years as it takes time for a company's value to be reflected in is share price. Thus, when a fundamental analyst estimates intrinsic value, it needs to wait until the stock's market price changes to its "correct" value to gain from its investment. It assumes that the short-term market is wrong, but the market will correct itself over the long run which in some cases can be as long as several years. Also, fundamentals are the actual characteristics of a business. New management cannot implement changes overnight, and it takes time to create new products, run a marketing campaign, build supply chains, etc. Furthermore, the data that a fundamental analyst use is generated much more slowly than the price and trading volume data used by technical analysts. Financial statements are filed quarterly, and changes in earnings do not emerge on a daily basis like price and trading volume information. Not only the means of deciding about buying a security could be different; the goal could be different too. One can buy an asset because they believe it can increase in value, while on the other hand, one can buy the asset because they think they can sell it to somebody else at a greater price.

Some consider a trading venue as a place that is originally designed for fundamental traders. Any other type of traders is seen as either facilitating the work for fundamental traders or parasite trying either gain on the side of such traders or eat into fundamental trader's profits. From such point of view, a market should only consist of fundamental traders that have long-term and real interest in the underlying asset rather participants that are looking to benefit from trading or market fluctuations. In Section 5.2.1 we investigate characteristics of a financial market that has more fundamental traders compared to rest of market participants and analyse its response to a liquidity crisis. In Section 5.2.2 study the case where only the population of fundamental buyers or fundamental sellers increases.



(a) Market draw-down

(b) Trading Volume

Figure 5.1: fundamental trader Market draw-down by population

5.2.1 Changes in Population of Fundamental Traders

If a fundamental trader, *e.g.* an asset manager is buying stocks of a technology company, it is expecting this company to grow and produce profit and asset manager is expecting to profit from either its share price increase or its dividend payments. As the number of shares that such a trader is looking to buy is not likely to be available at the orderbook at a price they are willing to pay, they normally execute this order via an algorithmic execution system. These algorithmic execution systems will break that parent order into a number of child orders and send those orders to market or markets that they can trade in them. Algorithmic execution strategies share some of the characteristics of high-frequency trading algorithms, but the main difference is that the initial decision to buy or sell, its price limits and quantity is already decided either by a human trader or another high-level asset management or portfolio management system. Based on the official report by (CFTC and SEC 2010a) an execution algorithm used by a fundamental seller was a major factor in starting the flash crash. The algorithm in question has traded a large volume at a fast speed with no limit on the price. Similar quantities of the same asset have been traded before the flash crash with no error but on previous occasions, the time limit for the execution of the order has been much longer allowing it to execute in much smaller quantities and slower speed.

To understand how fundamental traders affect the market, we will run the experiment with the same setup we used to re-generate flash crash and keep all the parameters the same but will change the number of fundamental traders. At this step, we do not alter the proportion of fundamental buyers and fundamental sellers but only increase their relative population to the rest of the market participants.

Figure 5.1a shows the maximum market draw-down for each of the scenarios. As can be seen from the graph, the relative population of fundamental traders did not significantly change the response of the market the short-term liquidity shocks.

Figure 5.1b shows the trade volume for each of the scenarios in our experiments. It can be seen that the trading volume will increase with the increase in the number of fundamental traders as they are looking to get in or out of their target positions and can tolerate minor price costs.

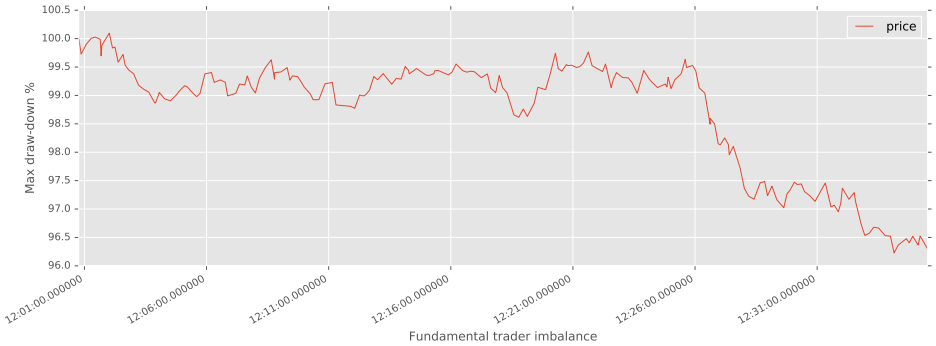


Figure 5.2: Market price move with fundamental-trader imbalance

5.2.2 Fundamental Trader Population Imbalance

As we discussed in the previous section, an increase in the number of fundamental traders as long as they exist on both sides of the orderbook and willing to buy and sell, will not particularly increase the chance of a market crash. Although, it will increase the total trading volume of the market for the same period. We now want to look into what happens if we change the balance of the fundamental traders and have more buyers or more sellers and investigate if that will affect market stability.

First, we run the same set of the simulation without the aggressive fundamental trader. Figure 5.2 shows the market price movements from a sample run from each set of population parameters we used. It can be seen that the fundamental trader imbalance have a small longer-term trend to price movements and can push the price up or down depending on the type of imbalance in the flow. This is a well-known phenomenon in market microstructure.

Now, we run the same set of experiments, but with the existence of an aggressive fundamental seller as we used in our previous sections. Figure 5.3 maximum market draw-down in each set of parameters. It shows when a market has already imbalance toward a sell, adding another aggressive fundamental seller will increase the chance of a market crash. Similarly, when there is an imbalance toward a buy, the market is more resistant toward short-term liquidity imbalance, and some of the shock generated by the aggressive seller is absorbed by the fundamental traders willing to buy and thus limiting market price swing.

5.3 High-Frequency Trader Population

High-frequency trading can contribute to price discovery and lower bid/ask spread. On the other hand, the amount of volume being traded by high-frequency traders is a cause of concern to some of the market participants and regulators. (CFTC and SEC 2010a) highlights the fact that most of the volume in CME is generated by a small number of trading firms. Some of the investigations on the event of May 2011 flash-crash have blamed high-frequency traders for both initiating and speeding up of the market crash (*e.g.* Nanex 2010). In Section 5.3.1 we investigate how the change in the population of high-frequency traders compared to other market participants affects the response of the market to a liquidity crisis.

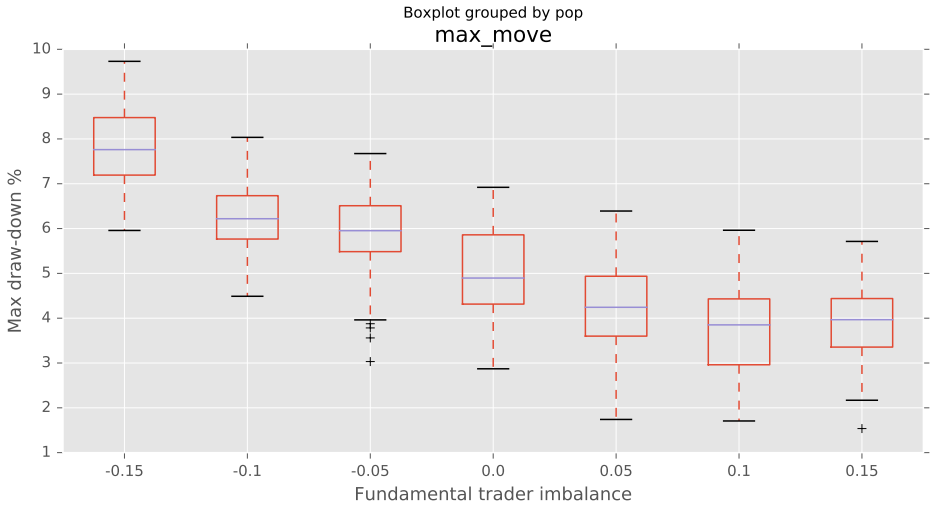


Figure 5.3: Market draw-down by the imbalance of fundamental traders

High-Frequency Traders have limited capital, and their gain comes from turning that capital around in short period of time. As a result, these type of traders get in and out of positions very quickly and often close their position at the end of the day to prevent carrying overnight risk. This capital limit implies, if the market is moving in one direction locking such traders in a position, they may become aggressive in trying to balance their portfolio and contribute to the moving market. In Section 5.3.2, we experiment with different capital constraints on high-frequency trading agents to investigate if such a limit is a contributing factor to market instability.

5.3.1 Changes in Population of High-Frequency Traders

We start the experiment with the same setup as we used in the previous section but going to change the population of market makers between 20% of the market and 70% of the market. In each scenario, we are going to run the simulation for the same period, *i.e.* one hour of trading; have a fundamental market maker starting a one-directional sell order at a fast pace and observe how the market reacts to this incoming flow. For each size of the population, we repeat the same experiment 20 times and look at the max market draw-down. We use maximum market draw-down as a measure of the market crash and use this to compare the behaviour of the market between each of these scenarios. Figure 5.4 shows box plots of max market draw-down for each of the points that we have experimented with. As we can see, as the graph moves to the right meaning population of high-frequency traders is growing, it improves the market reaction for a bit and then it goes down, and after about 65% it stabilises.

We can interpret this behaviour as follows. If there are very few high-frequency market makers who provide liquidity around current market price level, a big one-directional order flow can move the market price very quickly as there is less liquidity at the current level to trade. As a result, when market makers are updating their price, they will use the currently moved market price, and the market goes down quicker. This behaviour is reversed when the number of

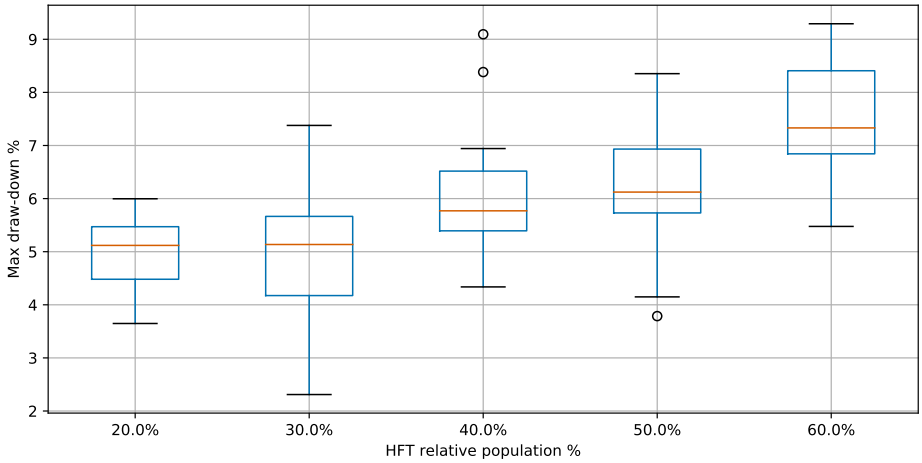


Figure 5.4: Market draw-down by percentage of high-frequency traders

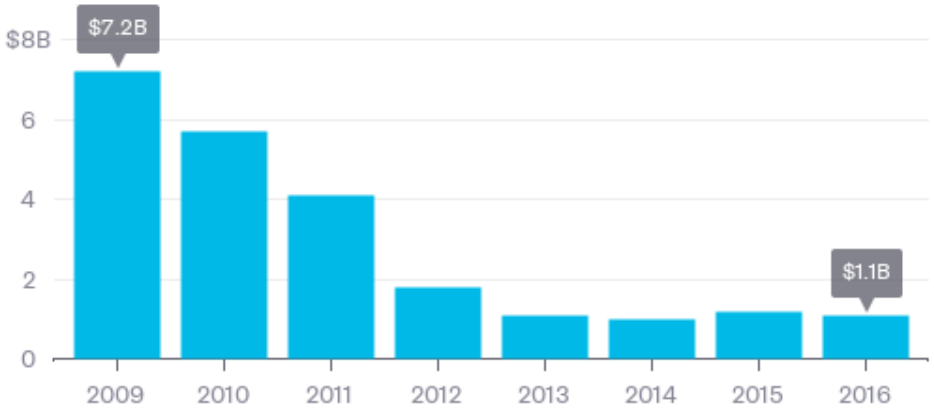


Figure 5.5: Earnings of US equities high-frequency traders (Bloomberg 2017)

high-frequency market makers grow as they start to provide a positive market feedback look. In this scenario, market even small market moves are exacerbated by many market makers fighting for the same order-flow. When many of these market makers themselves become aggressive, they push the price down by their competition for the liquidity. However, as can be seen, there is a limit to this fight and market reaches a kind of equilibrium meaning it cannot get any worse.

In real market condition, there is a natural limit on the higher bound of high-frequency market makers that can trade profitably in the market. Figure 5.5 shows the change in the earning of high-frequency market makers in U.S. equities markets. As can be seen from the graph, their revenue has been shrinking over the past few years. This is due to the fact that they can only benefit from either small market moves or market spread, they can benefit if there is a fundamental trader on the other side of their trade too. Trading in a market that is only filled with high-frequency traders fighting to capture the spread is a zero-sum game for all of them. In reality, it is a loss to those market makers as one need to consider the high cost of ultra-fast connectivity, powerful infrastructure, and best of breed software developers and quantitative analysts they need to make that system work.

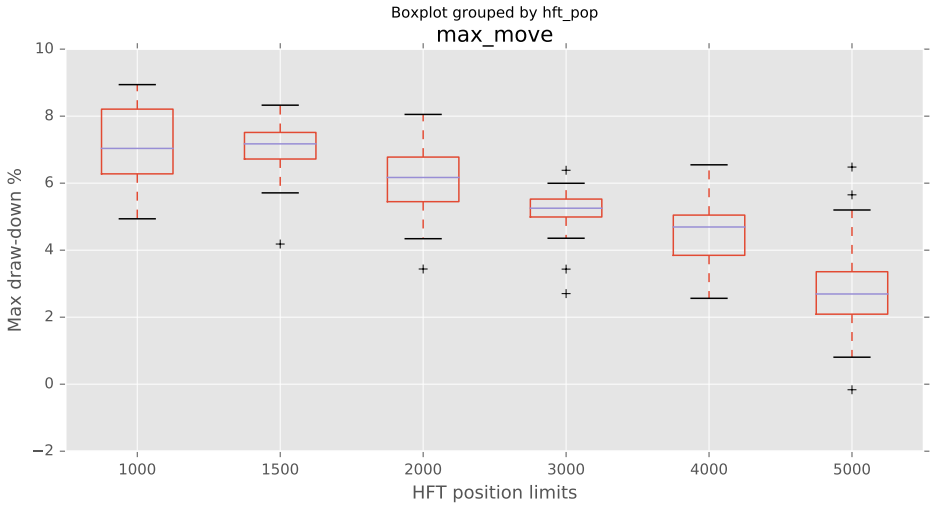


Figure 5.6: Market draw-down by high-frequency traders capital limits

5.3.2 Capital and Risk Limits

Most of the high-frequency traders have limited capital and make gain comes from turning that capital around in short period of time and making small profits on each trade. Such investment style limits amount of risk high-frequency traders are happy to hold onto for a longer time. Their target is to get in and out of positions quickly and close their book flat or at least market neutral at the end of the day.

This capital limit implies, if the market is moving in one direction and such traders have an opposite position, they may become aggressive in trying to balance their portfolio and contribute to moving the market even further in that direction. Imagine a high-frequency trader that has bought ABC stock at $100.05, expecting to sell it at $100.07 but the market moves and currently trades at $100.02 and it is staying at that price for a period of time, or even worse looks like it is moving down further. At some point, such trader may decide to close the position at $100.02 to limit its loss and free-up its capital to be able to get involved in other trades which it expects to make a profit on. Whatever the reason, the move to get out of position, can further contribute to moving the price down.

To investigate the effect of the capital and risk limits of high-frequency market makers on market's ability to handle liquidity crisis, we have repeated our experiment with the same number of high-frequency traders and other market participants and provided the same level of liquidity shock but changed the limits we have exposed on the positions that high-frequency market makers can hold onto. Figure 5.6 shows the market draw-down per each experiment. For each of these scenarios, we have run the simulation for 20 runs.

The default level we used in previous experiments has been 3,000 contracts both on the long and short positions. We have reduced that to 1,500 and increased that to 6,000. As it can be seen on the simulation results increasing the capital limit helps to improve the ability of the market to handle short-term liquidity issues.

Although this is good news, the main issue with this approach is that in reality, it is not practical to directly force market participants to increase their risk limits. In general, it would be much simpler to set an upper bound on the risk a market participant can take onto its book than limiting the lower bound. Exchanges have introduced programmes that give market makers incentives like lower fees or extra information about the flow. In return, they are required to provide some guarantees about the level of service they provide. Some of those requirements include being present at the best bid/offer price for a minimum amount of time (e.g. %95), minimum trading volume (*e.g.* 4% of daily volume) or minimum life-time for their quotes on the book. This will indirectly force market makers to hold larger positions when the market has liquidity issues to be able to pass the metrics mentioned above.

5.4 Conclusion

In this chapter, we experimented with the different ration of population of high-frequency traders and fundamental traders compared to the population of the rest of market participants to see how they affect the stability of the market and its ability to handle short-term liquidity shock.

We saw that high-frequency trader population increase could be helpful up to a point as they provide more liquidity but at some point this advantage will be reversed by their aggressive behaviour when they cannot hold onto their position and become liquidity takers rather than liquidity providers. We also investigated their risk limit exposure increase and observed that increasing their appetite to hold onto larger position could improve the market's ability to handle short-term liquidity shock. This is not something that can be enforced, but exchanges and other trading venues can provide incentives in return ask for statistical behaviour that would have similar implications.

We also looked at fundamental traders and how their population can affect market price. We observed that an increase in the population of fundamental traders does not significantly change the market's ability to handle short-term liquidity shock. Later, we looked into changing the proportion of fundamental traders. This change will produce a buy or sell bias into order-flow and can push the price of the asset up or down. We also saw that such a market could have a worse response when there is a liquidity shock that is in the same direction as the order-flow imbalance. Respectively, it can handle the shock better if there is already an imbalance in the opposite direction as some of the shock would be absorbed by the fundamental traders.

Chapter 6

Algorithmic Trading Controls

6.1 Introduction

Lack of sufficient control mechanism for high-speed trading by exchanges has been blamed as a source of problems with events such as the flash crash. Such controls aim to make sure there is orderly behaviour in the market, and one market participant intentionally or by mistake does not damage the experience of other market participants. There have been claims that high-frequency traders are making markets unfair by putting human traders and slower market participants with less technical abilities at a disadvantage. A number of mechanisms have been proposed to control algorithmic trading strategies and prevent an event similar to the flash crash or at least limit the damages of such event. It is essential that these suggestions are well understood and scientifically analysed to clarify their ability to contribute to their goal and to recognise any negative side-effects they may bring.

This chapter investigates three of the most famous proposed methods (Brogaard 2011): circuit breaker, minimum quote life, and order-to-trade ratio. We examine the effectiveness of these mechanisms using an agent-based model of a financial market and report on positive and negative effects of introducing such controls that we have observed in our simulations. *Circuit-breakers* are a simple way to limit the damages of a rapid market crash by stopping the market from trading and preventing further damage. This trading halt allows slow market participants to catch up with the current status of the markets and gives human traders monitoring and managing automated trading strategies time to react and adjust or stop such strategies. On the other hand, *minimum quote life* and *order-to-trade ratio* aim to limit the general speed of the market and reduce the chance of a rapid crash in the market.

Triggering a *circuit breaker* will stop trading in the market completely either for a predefined period of time or until it manually resumed by human administrators controlling the market.

This control is similar to the mechanisms that are used in many systems that people use in daily life. Electric supply into a building is controlled by a similar system, and if the current used goes above a safe limit the supply of electricity to the building is cut until it is manually triggered back. When there is heavy congestion, accident, or similarly disturbing event in a transport network it is common to block the roads going in the direction of the problematic area. The area will remain blocked until police arrive and the situation is understood, and then traffic is allowed to move again. It could be considered that such a measure is the most effective way of dealing with this problem as one can think nothing bad is going to happen after this point. Unfortunately, triggering circuit breaker in a connected world that has high-speed connectivity may not always be the right solution and can even make the situation worse. Similarly, if one route going in the direction of the problem is closed, but the rest are still working, that blocked road is making the problem worse by redirecting extra traffic to already overloaded paths. It has been claimed that the circuit breaker trigger at NYSE was a positive tool to cool down the pressure during the 2010 Flash Crash. Circuit breakers are one of the main tools have been proposed to prevent the repeat of the flash crash. In Section 6.2 we will investigate how forcing circuit breakers in different scenarios can affect market stability and its response to a short-term liquidity shock. To this end, we investigate two scenarios. We first examine how activating circuit breaker in one market while other markets are trading that asset can affect its price fluctuations. Then we will look into circuit breakers applied on all the markets at the same time and effects of a trading halt on the market response to a liquidity crisis.

There have been discussions that high-frequency traders pose a risk to the stability of the market. Their technological advantage is seen by some participants to make markets unfair. “Flash Boys” a book by (Lewis 2014), a New York Times best-seller (New York Times 2015), brought wide public attention to high-frequency traders. The book focuses on the rise of high-frequency trading in the US equity market. Lewis states that “The market is rigged” by high-frequency traders who front-run orders placed by investors. The speed of data is a major theme of the book; focusing on Spread Networks fibre optic cable connecting the financial markets of Chicago and New York. This link reduced the latency of data from 17 to 13 milliseconds but soon was overtaken by microwave link that reduced the latency by another 4.5 milliseconds. Lewis claims access to this fibre optic cable, as well as other technologies, presents an opportunity for the market to be controlled by the big Wall Street banks.

A number of solutions have been proposed to control the speed of high-frequency traders and ensure that the technological advantages of high-frequency market makers do not give them an unfair advantage over other market participants. The most common solutions proposed to deal with speed advantages that we will investigate are minimum quote life and order-to-trade ratio. Minimum Quote Life puts a lower bound on the time that an incoming order has to stay in the market before it can be cancelled or modified. This minimum time gives higher latency market participants a chance to receive and process market data for this incoming order and

take action if needed. Section 6.3 investigate if enforcing Minimum Quote Life can be helpful in dealing with a liquidity crisis similar to the flash crash. A softer version of this control is order-to-trade ratio which puts a higher bound on the number of order updates that can be applied by a participant compared to the number of trades that is done by that participants. As it does not have a limit per specific order, it allows some order to be cancelled or modified very quickly but enforces a higher level of speed control as these ratios are commonly measured on longer periods, *e.g.* per day. Section 6.4 experiment with different levels of such controls and their effect on market stability during a flash crash. Section 6.5 concludes this chapter.

6.2 Circuit Breakers

A circuit breaker in a trading venue will similarly limit the movement of the asset price within pre-defined boundaries. These boundaries are commonly set around the previous closing price of that asset. This mechanism is useful in very liquid securities as their respective price volatility is low. For less liquid stocks such limits are much more complicated, e.g. mid-cap or AIM-listed stocks in London or penny stocks in the US. Because the prices of such securities are more volatile, setting the price boundaries too tight triggers circuit breaker frequently, and produce a lot of false negatives. On the other hand, setting it too wide will also defeat their purpose and it will allow almost free price swings.

A Circuit-breaker is a mechanism that will stop trading when it detects something in the market has gone wrong. The simplest form of circuit breakers which are also the ones currently used by many exchanges is the one that triggered by market price movements. It will trigger the circuit breaker if the trading price moves outside a pre-set boundary around known good *reference price*. This reference price is often the previous closing price of the asset. For example, NYSE may set its rules to trigger the circuit breaker and stop trading of stock ABC if its trading price moves more than %5 above or below the previous day's closing price.

It has been claimed that the circuit breaker triggers at NYSE helped to cool down the pressure during the 2010 Flash Crash. As the trading only stopped at NYSE and same stocks could be traded on other trading venues, the order-flow that would have gone to NYSE were also routed to other markets by many of the market participants' smart order routeing engines adding to the liquidity issues on the alternative trading venues that were still trading the stock. After the flash crash, the SEC has put new rules in place to synchronise the circuit breaker triggers among all the US equity trading venues.

6.2.1 Experimental Setup

In this section, we will investigate how circuit breakers interact with market participants during a flash crash and how they affect market participants and order flow. We experiment with two scenarios: one where only one of the markets triggers the circuit breaker, and others continue trading. Next, we look into circuit breakers triggered on multiple venues.

For this experiment, we assume that the same security is trading in four markets *MA*, *MB*,

MC, and *MD*. An example of such a market is shown in Figure 6.1.

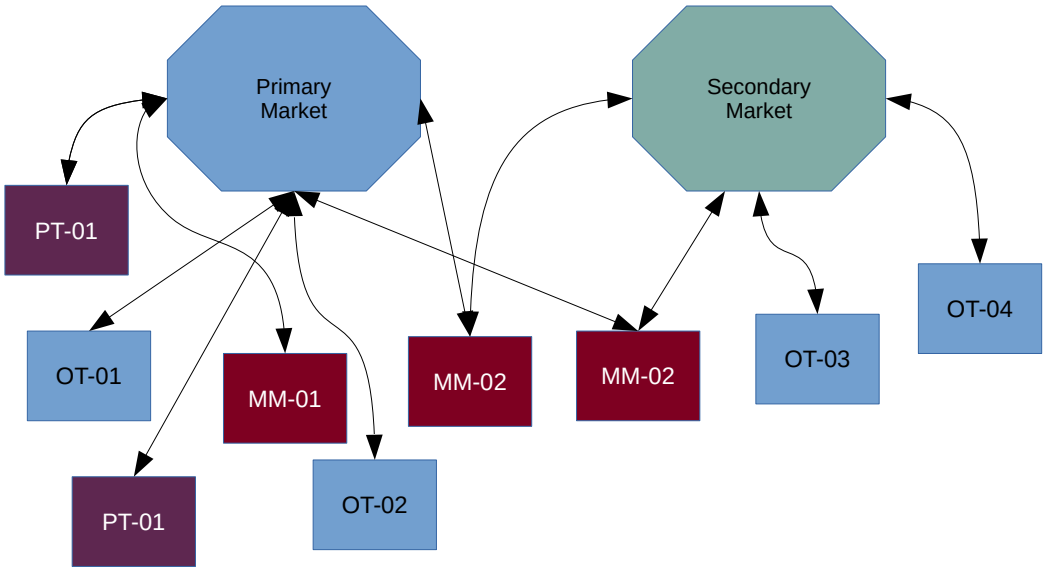


Figure 6.1: Cross-market circuit-breaker setup

One complexity we have in this experiment is that after circuit breakers are triggered, there could be manual intervention on market participants trading parameters and targets. Because this is ad-hoc and not a systematic change, it would be difficult to model. If only a single market were shut-down by the circuit breaker, one could expect that market can start trading straight away after the end of trading halt using the price from other venues that trade the same asset. Unfortunately, this is not a wise idea as the reason that the circuit breaker was triggered in the first place was that the price of the asset seemed unreasonable. The most common form used by many exchanges, *e.g.* LSE and NYSE, it to reopen the market using an auction. This is similar to the mechanism that is used at the start of the trading day to determine the opening price of the market. For our experiment, we also use the same mechanism that we use at the start of trading to set the last known price. Any order that existed in the orderbook before circuit breaker was triggered will be removed from orderbook. Note that the target and the current position of trading strategies will not change as it is the case in a real market.

6.2.2 Single Market Circuit Breakers

To analyse the effect of triggering circuit breakers in a single market on other trading venues trading the same asset we have used three exchanges with the same setup as the previous experiment. Traders and their limits are kept the same with the exception of high-frequency market makers. This is due to the fact that the high-frequency market makers monitor different trading venues closely and mainly trade in multiple markets with a shared inventory. Fundamental traders, on the other hand, are trading to reach a specific position use algorithmic trading platform to slice their orders for them and send those slices to best markets. Thus, when one market stops trading, fundamental traders would need to trade more on the available venues while high-frequency market makers are going to keep their order flow at the same or

similar levels as they have already accounted for this and show close to most of the liquidity they are willing to trade on all the available venues to increase their chance of trading.

We set a %3 trade price limit move to trigger circuit breaker and stop trading on one of the venues out of three available exchanges and continue trading with the other two remaining exchanges. Figure 6.2 shows the difference between running our simulation with and without triggering circuit breakers.

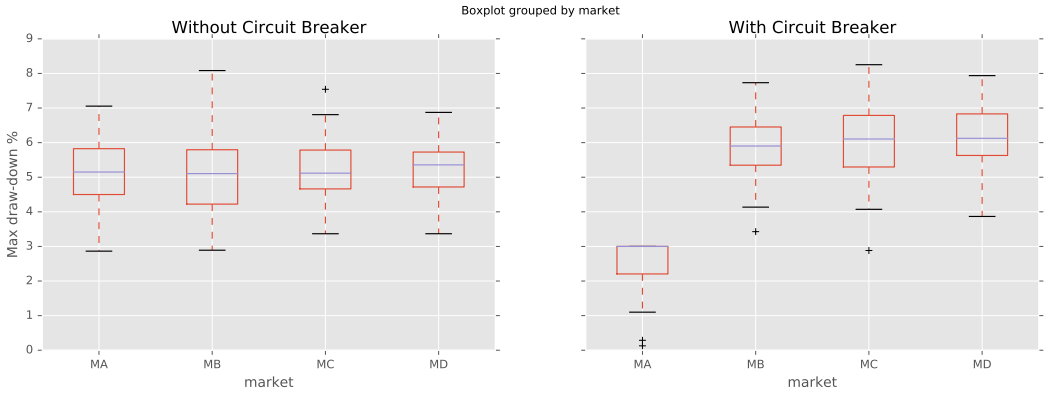


Figure 6.2: Max draw-down of the market with and without circuit breaker

As can be seen from the graph, triggering of circuit breaker on a single market only has an adverse effect on the price move in the other markets trading the same asset.

6.2.3 Cross-Market Circuit Breakers

When a single market triggers the circuit breaker, it is based on its own price moves and is a local decision. To be able to trigger cross-market circuit breakers we need to make sure they are all looking at the same source to decide when to trigger circuit breaker and stop trading.

US stock market regulated by the SEC is the market that has rules to trigger cross-market circuit breakers now. It also has a cross-market source of market data called *National Best Bid and Offer (NBBO)*. All of the US stock exchanges send their best available price and the quantity available at that price level to a central repository and this central repository will calculate a national best bid and best offer and publishes back to exchanges and market participants. Prices on this NBBO feed can be used to trigger national-level circuit breakers across markets.

Figure 6.3 shows the same simulation with multiple markets as we used in the previous section but this time with a national level circuit breaker being triggered if the price moves beyond our %3 limit. For the basic scenario, we have assumed that the orderbook in all the venues will clear and ever participant will start from that.

In reality, though what happens in the market after such market-wide trading halt also depends on the reaction of the humans controlling or watching these automated trading agents. We are assuming here that there has not been any fundamental news change about the asset we are trading and all the pressure to move the market price has come from a liquidity crisis. If this the same interpretation that humans make during such events, then we might expect to get

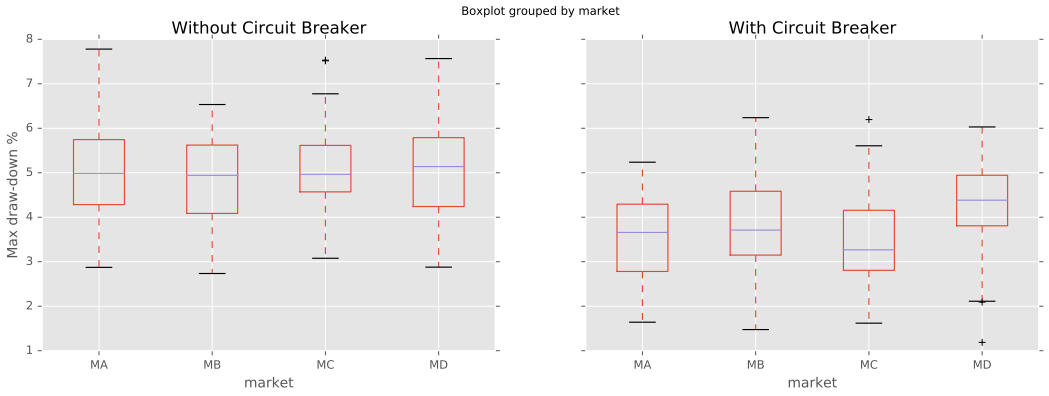


Figure 6.3: Cross-market circuit-breaker results

a similar reaction in the market. With the complex nature of human being, they may not all see this as we expect them too and may interpret this differently, and markets may still continue falling.

6.3 Minimum Quote Life

Minimum Quote Life (MQL) can be set by the exchange to set a lower limit on the time an order is going to stay on the orderbook before the participant can cancel or modify this order. For example, if an exchange set MQL on an instrument to be 50 milliseconds, the participant sending an order x005 at time $T0$ cannot cancel this order or modify it before $T0 + 50ms$. This is to make sure other market participants have at least 50 milliseconds to receive this order via their market data platform, process this market data update in their trading strategy and react to this order if necessary by placing new orders or modifying their existing orders.

To investigate the effect of MQL on the market behaviour during a liquidity shock, we have set up our experiment with exchange enforcing MQL on its matching engine side. We have not modified any of the strategies but enhanced our platform to be able to handle order reject and order replace rejects. MQL is, in particular, relevant to high-frequency market participants because the average time between order updates of other classes of traders in our system is normally larger than MQL. Figure 6.4 shows the result from our experiment running with different values of MQL. When MQL is zero, we get the same behaviour as our baseline experiments. As MQL increases, we can see that there is a slight change in the magnitude of the market move as this limits the ability of high-frequency traders to aggressively trade faster than other participants to get out of their positions.

In reality, we need to remember that market participants will adjust their strategy behaviours by considering market rules and limitations. So, although in our model high-frequency traders get a large number of their cancel and replace requests rejected, such traders have considered this into their trading model and market connectivity software platform and unlikely to try sending replace or cancel-request at the time they know for sure this is going to be rejected because of MQL. On the other hand, our simple model reflects this change in

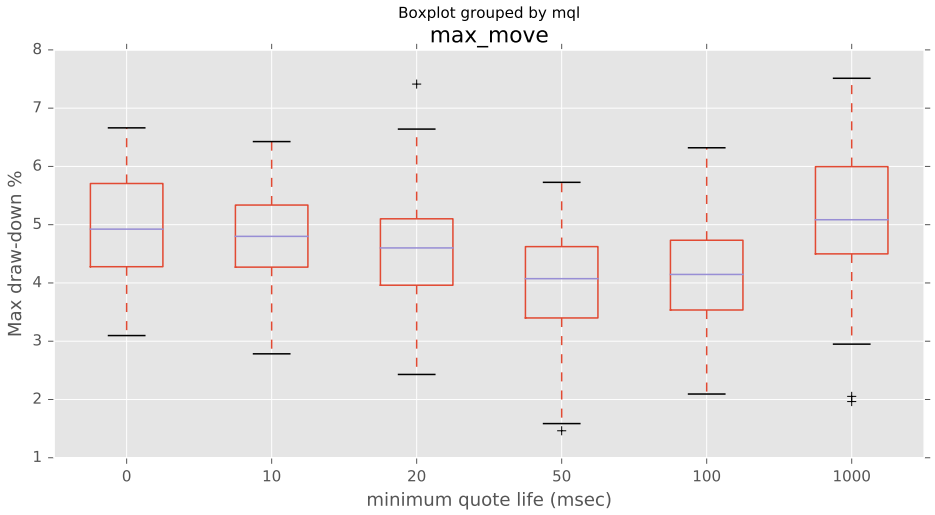


Figure 6.4: Market draw-down with different values of MQL

behaviour in terms of the successful orders, cancel and replace request applied to the exchange matching engine should reflect their real-world counterparts.

6.4 Order to trade ratio

In addition to the cost that exchanges incur from processing high quote traffic, other market participants that follow intra-day market activities must invest more in their computing systems as the level of traffic increases. This negative externality produced by those generating the highest traffic will be reduced with the order-to-trade limits. Enforcing minimum quote life on an exchange will limit the ability of market participants to react to market events. There are events in other markets that affect a participants desire to trade, its direction of trade or the quantity it is willing to trade. If a participant will not be able to react because of the market rules like MQL it will affect its initial desire to commit to a large amount of liquidity in the market. Some markets enforce their limits in a different style to give participants more flexibility but still ensure that there is a good amount of liquidity to slower market participants. One of those mechanisms is limiting *order-to-trade ratio*.

Order-to-trade ration means a market participant has an upper limit on the number of orders it is submitting to the market or number of times it is updating its existing order in the market compared to the number of trades it has done in the market. For example, if the order to trade ratio is limited to ten, it means that a market participant cannot submit and cancel or replace more than ten orders before at least of them is executed.

These limits could be applied in different ways. Exchange can enforce this limit linearly, meaning after each trade the participant is allowed to submit or modify ten times and after that its requests will be rejected. Another option is to apply these limits at a higher level, *e.g.* on a daily basis, of considering all the order requests sent by market participants compared to its trade for the day. If this ratio goes above that limit, it could end up either paying a higher

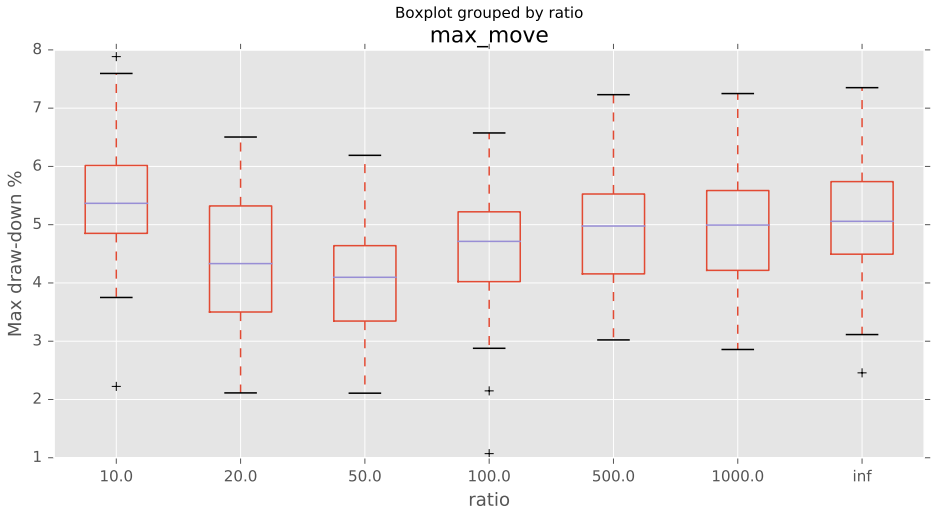


Figure 6.5: Market draw-down by order-to-trade ratio

transaction fee as a penalty or another form of fine. For example, participants failing to comply with order-to-trade ratio could be barred from the market for some time. Exchange can apply a mixture of these measures applied as well. It can have a soft limit (e.g. 10) that would result in the participant paying higher fees and a second level hard limit (e.g. 20) that ignoring that will result in the participant being barred from the market.

We have experimented with the simple model of rejecting the request from the market participant, if it fails to comply with the maximum order-to-trade ratio. Figure 6.5 shows the result of our experiment with different values of order-to-trade ratio. Similar to the previous example, this affects our high-frequency trading agent more than it affects any other type of trading agent model.

As can be seen in the figure, this can also slow down the market crash but its effect is less visible. This is expected as this measure is designed to provide market participants with more flexibility. As a result, market participants can still react quickly to market changes.

6.5 Conclusion

Exchanges and market regulators can enforce a trading emergency mechanism to halt trading of instruments when they think the market volatility is caused only by demand imbalance or a technical error to prevent this short term issue costing market participants. We investigated the single market and multi-market circuit breakers. We observed that a single market circuit breaker does not improve market response to a liquidity shock when that instrument continues to trade on alternative venues. Cross-market circuit-breakers are better tools for this purpose. We have also seen that the mechanism used to start trading after such pause is important to set the market price at a meaningful level.

We then looked at two other limitations that can be enforced by exchanges on their participants. Minimum Quote Life, can reduce the speed of market crash in the time of a

liquidity shock but on the other hand might be a limiting factor for the market makers, especially high-frequency market makers in the amount of liquidity they can provide into the market in the first place. Order to trade ratio limits can provide an alternative mechanism which targets the same behaviour but provides the market participant with more flexibility. This will also slow down market going down on liquidity shock but will its effect would be visible less than MQL.

Chapter 7

Interaction between Markets

7.1 Introduction

This chapter investigates how a liquidity crisis from one security in one market can expand to other securities and markets. The flash crash is a major concern to many market participants and regulators not only because there was a problem with the liquidity in the S&P E-Mini future contract on CME but also the problem on E-Mini expanded to ETF representing S&P 500 and followed to constituent stocks of the index. A similar situation happened during the ETF crash in 2014. There was a liquidity issue on some of the ETFs. The traders of those ETFs were forced to sell the underlying stocks to resolve the issues. This, in turn, created liquidity issues on the underlying stock. Some of constituents were not as liquid as the rest of them and as a result, experienced worse price swings than the ETF itself or other constituents of that ETF.

Section 7.2 studies the case where same security trades on different electronic market, *e.g.* US and European equity market. During the flash crash inconsistencies between trading rules in these exchanges and ECNs was one of the sources of problems. Then, Section 7.3 investigates securities that are not the same but are highly related and analyse how liquidity issues can expand from one to another. Section 7.4 concludes this chapter.

7.2 Multi-Venue Trading

Some the securities can be traded in more than one trading venue. This can be done in different ways. The simplest case is where the same stock with the same financial characteristics can be traded on more than one trading venue. Most of US Stocks are primarily listed in one exchange but can also be traded on other exchanges and Electronic Crossing Networks (ECN) as well. For example, IBM is primarily listed on New York Stock Exchange (NYSE) but the same stock can be traded on NASDAQ, BATS, and DirectEdge. In this case, there is no economic reason for the price of IBM on BATS or NASDAQ to diverge from its price on NYSE. All these venues trade

Figure 7.1: IBM New York and Frankfurt listing

IBM shares in the same currency (US Dollar), they have the same minimum price increment (one cent), follow the same settlement calendar, and trading in all of these venues are cleared in via the same central counterparty. These are fungible; *i.e.* one can buy a share in any of these venues and sell in another venue on the same day and end the day with a flat position. As a result, it is expected the price of IBM shares to be the same on all of these venues and any price changes in one venue is followed by other venues in real-time. It is also required by RegNMS (SEC 2005) that all the trading venues in the US have to forward the order to another trading venue if the other venue has a better price available at the time of the trade. European stocks used to trade only on their primary listing exchange. On 1 November 2007, European Union introduced MiFID[1] to increase competition and consumer protection in investment services. One of the concepts introduced by MiFID was Multilateral Trading Facility (MTF) which is a self-regulated financial trading venue. It means European stocks can be traded on more than one venue which resulted in most of the European large-cap stocks that are part of major national indexes to become available for trading on MTFs and other exchanges. For example, British Telecom (BT) can be traded on its primary listing on the London Stock Exchange but can also trade on other MTFs like BATS Chi-X Europe or Turquoise. Although they still trade on the same currency and follow the same settlement agreement, they may use a different clearing house as a central counterparty. EuroCCP[2], LCH.Clearnet, SIX x-clear and although are fungible it is slightly more complicated than the US.

Another method where a stock can be made available on multiple trading venues is cross-listing. Cross-listing of a company stock is when a firm lists its equity shares on one or more foreign stock exchange in addition to its domestic exchange. Cross-listing is especially common for companies that started out in a small market but grew into a larger market. Royal Dutch Shell, IBM, and Siemens are all examples where the same issue is traded in multiple markets. However, in Frankfurt and Paris, they are traded in EUR, London in GBP and on

[1] Markets in Financial Instruments Directive
[2] Formed by merger of EuroCCP Ltd and European Multilateral Clearing Facility (EMCF)

Figure 7.2: Deutsche Bank Frankfurt and New York ADR

NYSE in USD. Prices are subject to local market conditions, as well as FX fluctuations and are not kept in perfect parity between markets (see Figure 7.1 as an example). Cross-listing is expected to benefit the company by providing a lower cost of capital because the allows a company's shares to gain access to more investors whose access would otherwise be restricted because of international investment barriers. Cross-listing on markets with stringent disclosure requirements also signal the company's quality to outside investors, potential customers and suppliers. The main disadvantages are additional listing fees, increased reporting and disclosure requirements, and increased pressure on executives due to closer public scrutiny (Miller 1999; Roosenboom and Dijk 2009).

Another method that allows a stock to become available in more than one market for trading without the company itself making additional listing is through mechanisms such as American Depositary Receipt (ADR) or variants for other markets like European Depositary Receipt (EDR) and International Depository Receipt (IDR). ADR is a mechanism to repackage a security primarily listed on an Exchange (*e.g.* on Frankfurt in Germany) to enable it to be purchased by an investor outside of that market (*e.g.* within the US on the NYSE). ADR denominated and pay dividends in U.S. dollars and can be traded like regular shares of stock during U.S. trading hours, through U.S. broker-dealers. They simplify investing in foreign securities by having the depositary bank manage all custody, currency and local taxes issues. This is a distinct instrument, as not all the rights may come with the ADR, and the ADR is subject to the fluctuations of the underlying currency. The original issue (on Frankfurt) would be priced in EUR, while the ADR is priced in USD. In most cases, the ADR is convertible back into the original instrument (but needs to go through a process). Figure 7.2 shows an example of Deutsche Bank ADR trading on NYSE compared to its original listing on Frankfurt.

During the flash crash, multiple trading venues trading the same stock have experienced the liquidity problem. Similar issues can happen for cross-listed stocks and ADR. Next section provides details on the experiment that has been set up to replicate and analyse the problem. This experiment is designed to replicate multi-venue trading of the same instrument. This setup can be generalised to cover those scenarios by including currency conversion into this model.

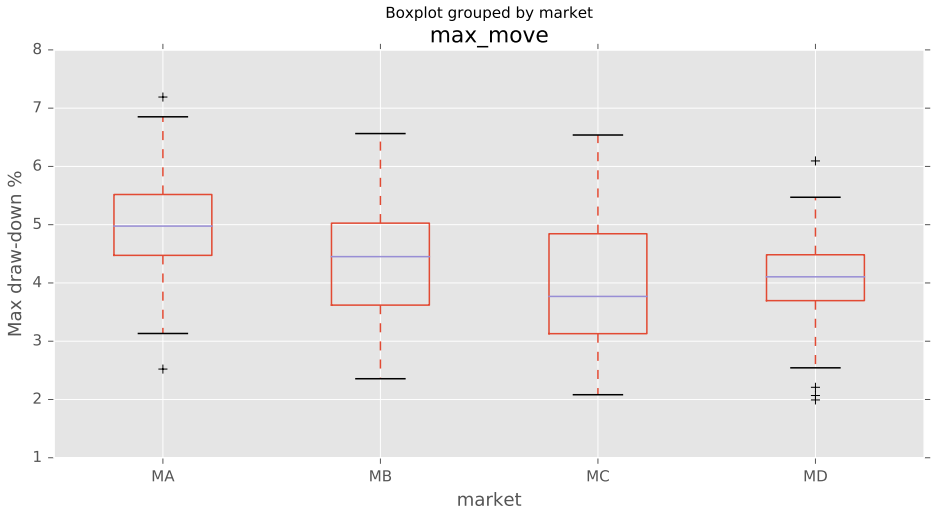


Figure 7.3: Multi-venue trading

7.2.1 Experimental Setup

To investigate the effect of a liquidity shock in one market on other markets that trade the same instrument trading agents are enhanced so that some of them can trade on multiple markets. In this setup, the aggressive fundamental trader that pushes the sell orders were allowed to trade on one of the four markets (*i.e.* MA). They share the same trading parameters as before, but their order placement is slightly different as they decide about their orders based an aggregated view of the orderbook among multiple trading venues. Fundamental market makers trade aggressively on the venue that offers them the best price. Market makers, including high-frequency market makers, still trade on both sides of the orderbook but they will use the most conservative price on each side of the aggregated orderbook to trade.

7.2.2 Results

Figure 7.3 shows the result of running the simulation on the primary venue that has the liquidity issue and other markets which did not initially have the problem. As can be seen the extent of the crash is a bit less compared to a case when everyone was trading on the same venue. This is due to the fact that when there is a price discrepancy market participants are not sure which side is the correct one and it takes time for them to realise the market that is wrong.

7.3 Highly Related Securities

During the flash crash, the initial problem started with S&P E-mini future contract on CME. However, it soon expanded and affected many other securities in other markets. This section reports on the experiments that are performed to analyse how liquidity problem expands from one market to a different security in another market.

Indexes are synthetic instruments that would give an investor synthetic exposure to a set of other instruments. Exchange Traded Fund (ETF) is a weighted basket of securities, commonly stocks or bonds. In the case of ETF, the provider often required by regulation to hold a portfolio

of underlying instruments with the quantities proportional to the weightings.

Both Index and ETF prices depend on the other underlying securities. When a participant is trading, it can consider either constituents are driving the price of the index or ETF meaning there is a liquidity issue in one of the constituents and that asset price moves, it is expected that the price of index or ETF that contains that instrument to move as well with the corresponding weight.

If there is a liquidity issue on the Index Future or ETF itself, it can cause price fluctuations. Price movements could be interpreted in different ways by market participants. One common view would be that this price move is related to a macro environment that this index or ETF is representing. For example, if S&P E-Mini future contract price goes down, it could be interpreted that there is some information about US economy as this index is representing large US companies. In this scenario, participants may also trade underlying instruments down using this information with the expectation that those instruments will be moving down as well with some delay.

7.3.1 S&P 500 E-Mini

S&P 500 Index by Standard & Poor is the weighted price of 500 largest US companies listed on New York Stock Exchange or NASDAQ. The weights are calculated by Standard & Poor based on their weighting strategy for this index. CME trades a future contract on that index called S&P 500 E-Mini or sometimes referred to as just E-Mini. Indexes are commonly traded via their future contracts, and both parties of the trade normally settle by the cash equivalent of that market move.

Before continuing to the issue a brief background on S&P 500 E-Mini contract is presented and why its liquidity problem has expanded that far. The Standard & Poor's 500, often abbreviated as the S&P 500, or just "the S&P", is an American stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ. It is one of the most commonly followed equity indices, and many consider it one of the best representations of the U.S. stock market, and a bellwether for the U.S. economy. The S&P 500 is widely used as a measure of the general level of stock prices, as it includes both growth stocks and value stocks. The "Composite Index", as the S&P 500 was first called when it introduced its first stock index in 1923, began tracking a small number of stocks. Three years later in 1926, the Composite Index expanded to 90 stocks and then in 1957 it expanded to its current 500. The S&P 500 index components and their weightings are determined by S&P Dow Jones Indices. It differs from other U.S. stock market indices, such as the Dow Jones Industrial Average or the Nasdaq Composite index, because of its diverse constituency and weighting methodology. The components of the S&P 500 are selected by a committee. This is similar to the Dow Jones Industrial Average, but different from others such as the Russell 1000, which are strictly rule-based. When considering the eligibility of a new addition, the committee assesses the company's merit using eight primary criteria: market capitalization, liquidity, domicile,

public float, sector classification, financial viability, the length of time publicly traded and stock exchange. The committee selects the companies in the S&P 500 so they are representative of the industries in the United States economy. In order to be added to the index, a company must satisfy these liquidity-based size requirements: (i) market capitalization is greater than or equal to US$6.1 billion (ii) annual dollar value traded to float-adjusted market capitalization is greater than 1.0 (iii) the minimum monthly trading volume of 250,000 shares in each of the six months leading up to the evaluation date, and (iv) the securities must be publicly listed on either the NYSE or NASDAQ. It is a free-float capitalization-weighted index. The index value is updated every 15 seconds during trading sessions and is disseminated by Reuters America.

Chicago Mercantile Exchange (CME) offers futures contracts that track the index and trade on the exchange floor in an open outcry auction, or on CME's Globex platform, and are the exchange's most popular product. E-Mini S&P, often abbreviated to "E-mini" (despite the existence of many other E-mini contracts) and designated by symbol ES, is a stock market index futures contract traded on the Chicago Mercantile Exchange's Globex electronic trading platform. The notional value of one contract is 50 times the value of the S&P 500 stock index The contract was introduced by the CME on September 9, 1997, after the value of the existing S&P contract (then valued at 500 times the index, or over $500,000 at the time) became too large for many small traders. The E-Mini quickly became the most popular equity index futures contract in the world. Hedge funds often prefer trading the E-Mini over the big S&P since the older (big) contract still uses the open outcry pit trading method, with its inherent delays, versus the all-electronic Globex system for the E-mini.

Investors may also invest in all the stocks of the S&P 500 directly, which is usually called index replication. Many index funds and ETFs attempt to replicate the performance of the S&P 500 by holding the same stocks as the index, in the same proportions. Many other mutual funds are benchmarked to the S&P 500. Consequently, a company whose stock is added to the list of S&P 500 stocks may see its stock price rise, as index funds must purchase that company's stock in order to continue tracking the S&P 500 index. Mutual fund managers provide index funds that track the S&P 500, the first of which was The Vanguard Group's Vanguard 500 in 1976. In addition to investing in a mutual fund indexed to the S&P 500, investors may also purchase shares of an ETF which represents ownership in a portfolio of the equity securities that comprise the Standard & Poor's 500 Index. These exchange-traded funds track the S&P 500 index and may be used to trade the index. SPDR[3] funds are a family of ETFs traded in the United States, Europe, and Asia-Pacific and managed by State Street Global Advisors. The name is an acronym for the first member of the family, the Standard & Poor's Depositary Receipts, now the SPDR S&P 500 (NYSE Arca: SPY), which is designed to track the S&P 500 stock market index. For a long time, this fund was the largest ETF in the world

[3]SPDR is a trademark of Standard and Poor's Financial Services LLC, a subsidiary of McGraw Hill Financial

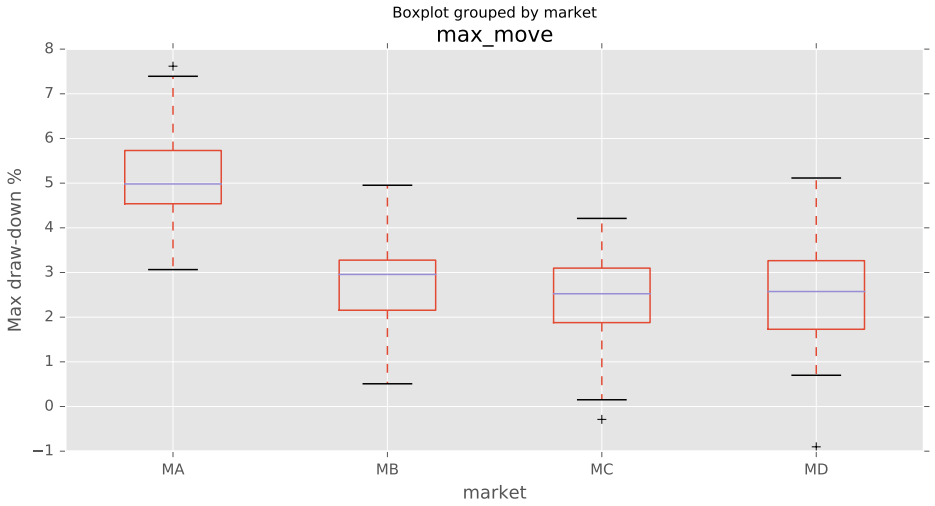


Figure 7.4: Crash expansion from index to a stock

7.3.2 Experiment Setup

This experiment is interested in the second scenario as this is what happened both during the flash crash and the ETF market crash. It models this by allowing traders to trade in two separate markets independently and listen to market data from both. It uses the information from the future market to place orders into the stock market. So, the order placement of the agents in the stock market would depend on the last trading price of the stock market plus a delta representing short-term price moves of the future market.

7.3.3 Results

Figure 7.4 shows the result of the experiment with this setup. As it can be seen, even with this simple setup, liquidity issue in one instrument expands to another instrument that is not pricing directly from the problematic instrument but is only using its short-term price movement as an indicator that affecting the direction of its price movement.

7.4 Conclusion

This chapter investigated the interaction between different instruments during a liquidity crisis. The first experiment studied how the same instrument trading on multiple venues can be affected when one of the trading venues impacted by liquidity issues. It was observed that liquidity issue in one venue expands very rapidly into other markets that are trading the same instrument but it is still slightly better than the case where everything is trading on the same trading venue. This is due to the fact that not all market participants are connected to the same venue and interpret everything the same, so this would provide a smaller window for prices to fluctuate. The second experiment examined how correlation in the future price movement between two instruments can contribute to price fluctuations. The level of fluctuation was considerably smaller than the previous experiment.

Chapter 8

Conclusions and Future Work

This research has investigated the use of agent-based modelling to analyse flash crashes. This chapter presents the contributions and conclusions of this thesis and outlines possible directions for future research.

8.1 Contributions and Conclusions

In recent years there have been a number of incidents in financial markets caused by electronic trading systems. One of the most famous incidents happened on 6 May 2010, when U.S. stock and future markets experienced a flash crash. During the flash crash, markets collapsed nearly 6% and recovered to almost the same level in a short period of time. This flash crash temporarily wiped one trillion dollars in market value during the downturn. The flash crash of May 2010 and similar incidents have raised concerns among market participants, regulators and politicians about the stability of financial markets in the presence of electronic trading systems. A number of regulatory changes have been proposed to prevent such problems happening in the future. It is important that the effectiveness and side-effects of these proposed changes are studied before they are implemented.

This research proposed an agent-based modelling framework and corresponding simulation platform to study market behaviour and effects of proposed regulatory changes. This framework is used to analyse and argue about emergent market behaviour resulting from (i) the behaviour of market participants, (ii) trading rules, and (iii) controls imposed by the market. The first step was to devise a system model that is as close as possible to a real market. Such model enabled us to compare the results from simulation with that of real market and use the effects of regulations and control changes shown in the simulation to contend what is expected to happen in a real market as a result of a similar change. To model market participants, a zero-intelligent agent model with two improvements is used. Firstly, agents have a position constraint instead of

a budget constraint. Secondly, the price of orders placed by the agent comes from a non-uniform distribution that depends on last traded market price instead of a uniform distribution limited by agent's budget constraint.

The characteristics of such probabilistic agents were studied by using a market model with a limited number of agents. Furthermore, agent's behaviour was validated by placing each type of agents into a real market and study its performance when interacting with historical data from a real market. At the next step, a market was built composed only of such artificial agents and shown that this model produces the comparable stylised facts to a real market. This framework was used in the rest of the thesis as a baseline to simulate and analyse some of the proposed regulatory changes. To decide about the population of agents and their type that compose the simulated market, the same class of trading agents and population as the one reported in official investigations by CFTC and SEC into May 2010 flash crash was used. The characteristics of the baseline model are presented in Section 8.1.1.

Using this baseline framework, the population of the different class of traders was investigated, especially focussing on high-frequency traders and fundamental traders. Main findings of that investigation are reported in Section 8.1.2. This framework was then used to experiment with regulation changes and controls that have been proposed to improve the fairness and stability of electronic trading venues. Results of these experiments are presented in Section 8.1.3. Sections 8.1.4 reports findings regarding the interaction between markets. Finally, Section 8.2 discusses future work.

8.1.1 Agent-Based Model of the Flash Crash

Regarding the agent-based model of the flash crash, as presented in Chapter 4, first a model of a financial market was devised using agent-based models of market participants. Each market participant modelled as a zero-intelligent agent. The trading venue itself is also implemented as a type of agent rather than being part of the environment. Trading agents can only communicate with trading venues, and there is no communication channel between trading agents. The initial baseline includes only one trading venue, but in later experiments, that model was expanded to have a system with multiple trading venues. Because trading venues are a type of agent in the system and not part of system environment, modelling a multi-venue trading environment can be done by instantiating more trading venues in the system model and configuring trading agents to connect to more than one trading venue.

A system model was devised using a zero-intelligent model of market participants that are modified to have position constraint and place their orders using a probability distribution that depends on the last market traded price. The model was further expanded for market-making agents so that these agents can trade on both sides of the market, *i.e.* place both buy and sell order, at the same time. The main findings are stated below.

First, this implementation of the framework was verified. To this end, a market was simulated consisting of fundamental, *i.e.* agents that only trade in one direction and place

either buy or sell order, traders with position constraint and used a wide set of liquidity profiles. The results confirmed that the trading price converged to a price that was close to equilibrium price for that liquidity profile. Then, zero-intelligence agents that were aware of the current market price were introduced, and their order placement picked a price from a distribution that is more likely to produce a price that is close to the previous trading price. Results indicated that this extra change to the agent model increases market volatility.

We then examined the behaviour of a zero-intelligent trading agent interacting with historical order-flow from a real market. To this end, this study proposed methodologies to handle passive limit orders and aggressive orders placed by the artificial agent into an orderbook that has the mixture of orders from an artificial agent as well as historical data from a real market. We used this setup to verify the behaviour of the two most important classes of trading agents: fundamental traders and market makers. For fundamental traders that always trade in one direction, this study investigated agent's execution performance compared to a benchmark, and for market makers that trade in both direction, this study analysed agent's profit and loss. The results indicated that these agents are a reasonable representation of their counterparts considering that these agents do not follow any trading strategy and use simple probabilistic models for order placement.

Furthermore, a market was constructed with agents that were proportional to the agents on the day of May 2010 flash crash. For this purpose, this study used the same classification of trading agents that is used by CFTC and SEC report and composed a market model with six classes of traders, *i.e.* fundamental buyers, fundamental sellers, market makers, high-frequency traders, opportunistic traders, and noise traders. For each of these trading agents, the position limits and order placement speed from that official investigation report was used. We studied the stylised facts that this model produced compared to the same stylised fact reported for different financial markets. We finally introduced a fundamental trader trading at a high-speed and placing orders that its size is proportional to the trading volume of the market for the previous period and observed that the artificial model also produces a flash crash.

Overall, the main finding in Chapter 4 was that the agent-based models can be used to model and study the flash crash. Therefore, this model was used as a baseline in Chapter 5 to analyse the behaviour of a market when the relative population of market participant changes. We used this model to discuss some of the controls proposed for electronic systems Chapter 6 and to reason about the interaction between markets in Chapter 7.

8.1.2 Diversity of Trader Population

CFTC and SEC report blames a fundamental trader to be responsible for starting the flash crash. A recent study suggests more than 60% of US equity trading happens with high-frequency traders. That trend is closely followed in other highly liquid trading venues like Futures and European Equity markets. High-frequency traders have been blamed for exacerbating the flash crash and even in some views for causing the flash crash. Chapter 5 used an agent-based model

of a financial market to examine the different sets of the population of high-frequency traders and fundamental traders compared to the rest of market participants to see how they affect the stability of the market and its ability to handle short-term liquidity shock.

First, fundamental traders were examined to study how their population can affect market price. It was observed that the increase in the population of fundamental traders does not significantly change the market's ability to handle short-term liquidity shock. Then, the change in the proportion of fundamental buyers and sellers were investigated. This change produces a buy or sell bias into order-flow and can push the price of the asset up or down. We also observed that such a market could have a worse response when there is a liquidity shock that is in the same direction as the order-flow imbalance. Respectively, it can handle the shock better if there is already an imbalance in the opposite direction as some of the shocks would be absorbed by the fundamental traders.

The study then experimented with high-frequency trader population and shown that an increase in their population can be helpful in dealing with a liquidity crisis if its scale is limited up to a point because they provide more liquidity. At some point, this advantage was reversed by their aggressive behaviour when they cannot hold onto their position and become liquidity takers rather than liquidity providers. The study also investigated their risk limit exposure increase and observed that increasing their appetite to hold onto larger position could improve the market's ability to handle short-term liquidity shock. This is not something that can be enforced, but exchanges and other trading venues can provide incentives in return ask for statistical behaviour that would have similar implications.

8.1.3 Algorithmic Trading Controls

High-speed of trading and lack of sufficient control mechanism by exchanges has been blamed as a source of problems with events such as the flash crash. Such controls aim to make sure there is orderly behaviour in the market and intentionally bad behaviour or mistake by a market participant does not damage the experience of other market participants. There are claims that high-frequency traders are making markets unfair by putting human traders and slower market participants with less technical abilities at a disadvantage. A number of mechanisms have been proposed to control algorithmic trading strategies and prevent an event similar to the flash crash or at least limit damages of such event. It is essential that these suggestions are well understood and scientifically analysed to clarify their ability to contribute to their goal and to recognise any negative side-effects they may bring. Chapter 6 investigated three of the most famous proposed methods: circuit breaker, minimum quote life, and order-to-trade ratio.

Circuit breaker, when triggered, stops trading in the market completely either for a predefined period of time or until it is manually resumed by human operators controlling the market. It has been claimed that the circuit breaker at NYSE was a positive tool to cool down the pressure during the 2010 flash crash. The study experimented with two setups: single-market circuit breaker, and cross-market circuit breaker. In a single market circuit breaker, one of the

markets triggers a circuit breaker, and the rest of the markets continue to trade. Our results indicate that the triggering of the circuit breaker in a single market worsens the effects of a liquidity crisis rather than improving it. The study then examined a similar setup by triggering circuit breakers across all markets at the same time, and it was observed that this is a helpful tool to control the effects of a liquidity crisis.

Minimum quote life puts a lower bound on the time that an incoming order has to stay in the market before it can be cancelled or modified. This will give other market participants with higher latency the chance to receive and process market data for this incoming order and take action if needed. The results indicated that introducing minimum quote life can reduce the speed of market crashes but increasing the time window for this control slowed down the whole market with no significant effect on its response to the liquidity crisis. In reality, one needs to remember that, market participants will adjust their strategy behaviours by considering market rules and limitations. So, although in this model high-frequency traders get a large number of their cancel and replace requests rejected, such traders have considered this into their trading model and market connectivity software platform and unlikely to try sending replace or cancel request at the time they know for sure this is going to be rejected because of minimum quote life. On the other hand, this simple model reflects this change in behaviour in terms of the successful orders, cancel and replace request applied to the exchange matching engine should reflect their real-world counterparts.

Order-to-trade ratio limit is a softer version of the control which puts a higher bound on the number of order updates that can be applied by a participant compared to the number of trades that is done by that participants. As it does not have a limit per specific order, it allows some order to be cancelled or modified very quickly. The results confirmed this could also slow down the market crash, but its effect is less visible compared to minimum quote life. This is expected as this measure is designed to provide market participants with more flexibility.

8.1.4 Interaction between Markets

Flash crash happened not only because of the problem with the liquidity in the E-Mini future contract on CME but because the problem on E-Mini expanded to ETF representing that ETF and followed up to constituting stocks that form that index. A similar situation happened during ETF crash in 2014. There was a liquidity issue on some of the ETFs. The traders of those ETFs were forced to sell the underlying stocks to resolve the issues. This, in turn, created liquidity issues on the underlying stock. In Chapter 7 the study investigated how a liquidity crisis from one asset in one market can expand to other securities and markets.

We first experienced with the same asset trading on multiple electronic markets. This setup is similar to US and European stock markets. In the first scenario, this study looked into how the same instrument trading on multiple venues can be affected when one of the trading venues affected by liquidity issues. We observed that liquidity issue in one venue expands very rapidly into other venues trading the same instrument, but it is still slightly better than the case where

everything is trading on the same trading venue. This is due to the fact that not all market participants are connected to the same venue and interpret everything the same, so this would provide a smaller window for prices to fluctuate.

In the second experiment, this study investigated securities that are not the same but highly correlated and how liquidity issues can expand from one to another in such scenarios. The study has shown how correlation in the future price movement between two instruments can contribute to price fluctuations. The level of fluctuation is considerably smaller than the previous experiment.

8.2 Future Work

This section discusses several directions for future work which are motivated by the findings of this thesis.

8.2.1 Study of stop-less

One of the factors that has reported to have affected flash crash is the additional flow of retail stop-loss orders. Stop-loss orders are designed to limit the financial loss for retail and other investors. As most of these limits are set by humans, their distribution is biased toward round numbers and specific price points. As a result, market trading price crossing those points have an adverse and sudden affect on the volume of liquidity available in the market. In a scenario where there is already a problem with liquidity, this additional flow provides another extra push in the direction of the crash. It would be interesting to study how setting of these limits distributed more evenly can affect this phenomena. There has been also discussion on soft stop-loss orders that instead of waiting for the price to hit a trigger point and then trading all of the position at once, they start trading out of the position slowly when the price gets closer to pre-defined limit. This can be further studies using this platform.

8.2.2 Modelling Dynamic Delay in the Platform

Market participants and communicate with each other using computer networks. They also use software systems to process incoming orders and market data and make a decision. Experiments in this search have used latency information from flash crash to model speed of trading agents which implicitly includes the latency of their communication with exchange and processing power. In reality, both communication speed and response time can be affected during a crisis as the amount of traffic is high and can cause larger than normal delays or even missing data and slow down computers running trading strategies or trading venue matching engine. Our framework is capable of modelling delay elements in the communication channels connecting market participants to trading venues. Its current random but fixed speed of order handling can be modified to include a traffic or trading volume based delay model.

8.2.3 Swarm Model of Trading Agents

Each market participant is a complex organisation with different teams and trading desks that have different trading strategies (for an example see Figure 3.1). Instead of modelling each participant as a simple agent, one should experiment with a multi-level structure where each market participant is in itself a system compromised of smaller trading agents interacting with each other in a limited-way and interact externally with other market participants.

8.2.4 Adaptive Trading Strategy

The trading strategies used in this research have fixed behavioural parameters that come from the analysis of real-life trading strategies. When performing experiments with changes to market trading rules and regulations, these new rules are not considered into strategies of trading agents. One can study a feedback mechanism that strategy can adapt to market limitation and analyse the response of the market regulation on the strategy itself.

8.2.5 Incorporating Market Impact into Mixed-mode Simulation

To analyse the behaviour of artificial trading agents this study has used a mixed-mode simulation platform. This model and platform can be extended to include market impact and can be used for applications outside the analysis of market regulation assessment.

8.2.6 Adding external signals into trading agents

We envision a simulation model where a central signal is fed to all the agent with an error depending on the agent type and prediction horizon. For example, a fundamental agent can predict long term price movement with high-probability but its prediction of short time horizon would be more noisy. On the other hand, a market maker has a more accurate prediction of short term price move but noisy perdition on the long term movement of the stock. The point about all agent having similar prediction of the future, *e.g.* all market makers predicting similar price move is an interesting question to investigate.

8.2.7 Studying high-frequency trader's budget constraints

High-frequency traders trade large volumes but they keep their positions small and near zero and mostly try to close the trading day flat. It has been reported that during flash-crash these type of traders became aggressive liquidity takers when they hit their position limit. It would be interesting to study how each of the position and PnL limit parameters affect the behaviour of these types of trading agents.

Bibliography

Abadi, Daniel J., Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, et al. "The Design of the Borealis Stream Processing Engine". In: *Proceedings of Biennial Conference on Innovative Data Systems Research*. 2005.

Aite Group. *New World Order: The High Frequency Trading Community and Its Impact on Market Structure*. Ed. by Sang Lee. Feb. 2009.

Aldridge, Irene. "High-Frequency Runs and Flash Crash Predictability". In: *Journal of Portfolio Management* 40.3 (2014).

Andersen, Torben and Oleg Bondarenko. "VPIN and the flash crash". In: *Journal of Financial Markets* 17.0 (2014), pp. 1–46.

Angel, James, Lawrence Harris, and Chester Spatt. *Equity Trading in the 21st Century*. Tech. rep. FBE 09-10. USC Marshall School of Business, Feb. 2010.

Arasu, Arvind, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, et al. "STREAM: The Stanford Stream Data Manager". In: *IEEE Data Engineering Bulletin* 26.1 (2003), pp. 19–26.

Bates, John. "Post Flash Crash, Regulators Still Use Bicycles To Catch Ferraris". In: *Traders Magazine* (Apr. 2015). 24$^{\text{th}}$.

BATS Europe. *Multicast PITCH Specification version 6.20*. Aug. 2017.

BATS Europe. *PITCH Specification Version 4.13*. July 2017.

BBC. *Pound plunges after Leave vote*. 24$^{\text{th}}$ Jul. 2016.

Becker, Gary S. "Irrational Behaviour and Economic Theory". In: *Journal of Political Economy* 70.1 (Feb. 1962), pp. 1–13.

Bellifemine, Fabio, Agostino Poggi, Giovanni Rimassa, and Paola Turci. "An Object Oriented Framework to Realize Agent Systems". In: *Proceedings of WOA 2000 Workshop*. 2000, pp. 52–57.

Bloomberg. *Knight Shows How to Lose $440 Million in 30 Minutes*. Ed. by Matthew Philips. 2$^{\text{nd}}$ Aug. 2012.

Bloomberg. *Trade Disorder Plagues Nasdaq Handling $16 Billion Facebook IPO*. Ed. by Matthew Philips. 19$^{\text{th}}$ May. 2012.

Bloomberg. *Goldman Sachs Said to Send Stock-Option Orders by Mistake*. Ed. by Nikolaj Gammeltoft and Donal Griffin. 20$^{\text{th}}$ Aug. 2013.

Bloomberg. *Trading on Speed: Man, Machine and the Stock Market*. Ed. by Sam Mamudi and Annie Massa. 15$^{\text{th}}$ Feb. 2017.

Booth, Ash. *PyLOBsim: Python package for simulating Intra-day orderflow in a limit order book*. 2013.

URL: https://github.com/DrAshBooth/PyLOBsim.

Brogaard, Jonathan A. "Minimum quote life and maximum order message-to-trade ratio". In: *UK Government's Foresight Project, The Future of Computer Trading in Financial Markets* (Nov. 2011).

Brogaard, Jonathan A., Terrence Hendershott, Stefan Hunt, Torben Latza, Lucas Pedace, et al. "High-Frequency Trading and the Execution Costs of Institutional Investors". In: *UK Government's Foresight Project, The Future of Computer Trading in Financial Markets* (Jan. 2013).

Carney, Don, Uur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, et al. "Monitoring streams: a new class of data management applications". In: *Proceedings of the International Conference on Very Large Data Bases*. Aug. 2002, pp. 215–226.

CFTC. *Sub-Committee on Automated and High Frequency Trading - Working Group 1*. Jan. 2012.

CFTC and SEC. *Findings Regarding the Market Events of May 6, 2010*. Sept. 2010.

CFTC and SEC. *Preliminary Findings Regarding the Market Events of May 6, 2010*. Mar. 2010.

Chandrasekaran, Sirish, Owen Cooper, Amol Deshpande, Michael Franklin, Joseph Hellerstein, et al. "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World". In: *Proceedings of Biennial Conference on Innovative Data Systems Research*. Jan. 2003.

Cherniack, Mitch, Hari Balakrishnan, Magdalena Balazinsk Don Carney, Ugur Cetintemel, Ying Xing, et al. "Scalable Distributed Stream Processing". In: *Proceedings of Biennial Conference on Innovative Data Systems Research*. Jan. 2003, pp. 1445–1447.

Cliff, Dave. *Bristol Stock Exchange market simulator*. 2012.

URL: https://github.com/davecliff/BristolStockExchange.

CNBC. *What happened during the Aug 24 'flash crash'*. 25th September. 2015.

Coakley, Simon, Marian Gheorghe, Mike Holcombe, Shawn Chin, David Worth, et al. "Exploitation of High Performance Computing in the FLAME Agent-Based Simulation Framework". In: *Proceedings of the 14^{th} International Conference on High Performance Computing and Communications*. 2012.

Cont, Rama. "Empirical properties of asset returns: stylized facts and statistical issues". In: *Quantitative Finance* 1.2 (2001), pp. 223–236.

Deissenberg, Christophe, Sander van der Hoog, and Herbert Dawid. "EURACE: A massively parallel agent-based model of the European economy". In: *Applied Mathematics and Computation* 204.2 (2008), pp. 541–552.

Demers, Alan, Johannes Gehrke, Mingsheng Hong, Biswanath Panda, Mirek Riedewald, et al. "Cayuga: A General Purpose Event Monitoring System". In: *Proceedings of Biennial Conference on Innovative Data Systems Research*. 2007.

Easley, David, Marcos López de Prado, and Maureen O'Hara. "The Microstructure of the Flash Crash: Flow Toxicity, Liquidity Crashes and the Probability of Informed Trading". In: *The Journal of Portfolio Management* 37.2 (2010), pp. 118–128.

Easley, David, Marcos López de Prado, and Maureen O'Hara. "VPIN and the Flash Crash: A rejoinder". In: *Journal of Financial Markets* 17 (2014), pp. 47–52.

EsperTech. *Esper and EsperIO 7.0.0 Reference*. 29th. Nov. 2017.

Financial Times. *Citi trader deepened October's pound "flash crash"*. Ed. by Katie Martin and Caroline Binham. 7th Dec. 2016.

FIPA. *Agent Communication Language*. 2002.

FIX Protocol Limited. *FIX 4.2 Specification*. Mar. 2000.

FIX Protocol Limited. *FIX 5.0 Specification*. Dec. 2006.

FIX Protocol Limited. *Simple Binary Encoding Specification - Draft Standard v1.0*. 29th. June 2016.

Free Software Foundation. *GNU General Public License, Version 3*. 29th. June 2007.

Free Software Foundation. *GNU Lesser General Public License, Version 3*. 29th. June 2007.

Gode, Dhananjay K. and Shyam Sunder. "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality". In: *Journal of Political Economy* 101.1 (Feb. 1993), pp. 119–137.

Greenough, Chris, David Worth, Shawn Chin, Mike Holcombe, and Simon Coakley. *The exploitation of parallel high performance systems in the FLAME agent-based simulation framework*. Tech. rep. RAL-TR-2008-022. Rutherford Appleton Laboratory, 2008.

Hagstrmer, Bjrn and LarsNordn. "The diversity of high-frequency traders". In: *Journal of Financial Markets* 16.4 (2013), pp. 741–770.

Hanson, Thomas. "High frequency traders in a simulated market". In: *Review of Accounting and Finance* 15.3 (2016), pp. 329–351.

Independent. *Mysterious pound sterling 'flash crash' of 7 October not due to trader manipulation or error, finds report*. 13th Jan. 2017.

Jacob Leal, Sandrine, Mauro Napoletano, Andrea Roventini, and Giorgio Fagiolo. "Rock around the clock: An agent-based model of low- and high-frequency trading". In: *Journal of Evolutionary Economics* 26.1 (2015), pp. 49–76.

JADE. *Java Agent DEvelopment Framework*. Aug. 2015.
URL: http://jade.tilab.com.

Kefalas, Petros, George Eleftherakis, and Evangelos Kehris. "Communicating X-Machines: From Theory to Practice". In: *Advances in Informatics: 8th Panhellenic Conference on Informatics*. LNCS 2563 (2003), pp. 316–335.

Kiran, Mariam, Simon Coakley, Neil Walkinshaw, Phil McMinn, and Mike Holcombe. "Validation and discovery from computational biology models". In: *Biosystems* 93.1–2 (2008), pp. 141–150.

Kirilenko, Andrei, Mehrdad Samadi, Albert S. Kyle, and Tugkan Tuzun. *The Flash Crash: The Impact of High Frequency Trading on an Electronic Market*. SSRN. May 2011.

Lawson, Nigel, Michael Snyder, Douglas McWilliams, Jonathan Said, Andrew Hilton, et al. *Big Bang 20 years on*. Centre for Policy Studies, Oct. 2006.

Lewis, Michael. *Flash Boys: A Wall Street Revolt*. W. W. Norton, 2014.

Luke, Sean, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. "MASON: A Multi-Agent Simulation Environment". In: *Simulation* 81.7 (July 2005), pp. 517–527.

Macal, Charles and Michael North. "Tutorial on agent-based modelling and simulation". In: *Journal of Simulation* 4.3 (2010), pp. 151–162.

Maslov, Sergei. "Simple model of a limit order-driven market". In: *Physica A: Statistical Mechanics and its Applications* 278.3–4 (2000), pp. 571–578.

McGroarty, Frank, Ash Booth, Enrico Gerding, and V.L. Raju Chinthalapati. "High frequency trading strategies, market fragility and price spikes: an agent based model perspective". In: *Annals of Operations Research* (2018).

Miller, Darius P. "The market reaction to international cross-listings:" in: *Journal of Financial Economics* 51.1 (1999), pp. 103–123.

Minar, Nelson, Roger Burkhart, Christopher Langton, and Manor Askenazi. *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*. Working Paper 96-06-042. Santa Fe Institute, June 1996.

Motwani, Rajeev, Jennifer Widom, Arvind Arasu, Brian Babcock, Shivnath Babu, et al. "Query Processing, Approximation, and Resource Management in a Data Stream Management System". In: *Proceedings of Biennial Conference on Innovative Data Systems Research*. Jan. 2003.

Nanex. *Nanex Flash Crash Summary Report*. Sept. 2010.

New York Times. *Nasdaq Is Fined $10 Million Over Mishandled Facebook Public Offering*. 29th May. 2013.

New York Times. *Best Sellers, Hardcover Non-fiction*. 20th April. 2015.

Niazi, Muaz and Amir Hussain. "Agent-based Computing from Multi-agent Systems to Agent-Based Models: A Visual Survey". In: *Scientometrics* 89.2 (2011), pp. 479–499.

NSE. *Press release*. 5th. Oct. 2012.

URL: https://www.nseindia.com/content/press/05102012.htm.

Nuti, Giuseppe, Mahnoosh Mirghaemi, Philip Treleaven, and Chaiyakorn Yingsaeree. "Algorithmic Trading". In: *Computer* 44.11 (Nov. 2011), pp. 61–69.

NYSE. *Data Products*. 2016.

URL: http://www.nyxdata.com/Data-Products.

Paddrik, M., R. Hayes, A. Todd, S. Yang, P. Beling, et al. "An agent based model of the E-Mini S&P 500 applied to flash crash analysis". In: *Proceedings of IEEE Conference*

onComputational Intelligence for Financial Engineering Economics (CIFEr). Mar. 2012, pp. 1–8.

Paddrik, Mark, Roy Hayes, William Scherer, and Peter Beling. "Effects of limit order book information level on market stability metrics". In: *Journal of Economic Interaction and Coordination* 12.2 (2017), pp. 221–247.

Paddrik, Mark E., Richard Haynes, Andrew E. Todd, William T. Scherer, and Peter A. Beling. "Visual analysis to support regulators in electronic order book markets". In: *Environment Systems & Decisions* 36.2 (June 2016), pp. 167–182.

Poslad, Stefan. "Specifying Protocols for Multi-Agent Systems Interaction". In: *ACM Transactions on Autonomous and Adaptive Systems* 2.4 (Nov. 2007).

Reuters. *Waddell is mystery trader in market plunge*. Ed. by Herbert Lash and Jonathan Spicer. 14th May. 2010.

Reuters. *Emkay Global's bad orders trigger brief halt on NSE*. 5th Oct. 2012.

Reuters. *Swiss central bank stuns market with policy U-turn*. 1st Jan. 2015.

Reuters. *Top ETF issuers ask U.S. SEC to prevent repeat of Aug 2015 turmoil*. 10th. Mar. 2016.

Richmond, Paul, Dawn Walker, Simon Coakley, and Daniela Romano. "High performance cellular level agent-based simulation with FLAME for the GPU". In: *Briefings in Bioinformatics* 11.3 (2010), pp. 334–347.

Roosenboom, Peter and Mathijs A. van Dijk. "The market reaction to cross-listings: Does the destination market matter?" In: *Journal of Banking & Finance* 33.10 (2009), pp. 1898–1908.

Schoreels, Cyril and Jonathan M. Garibaldi. "A Comparison of Adaptive and Static Agents in Equity Market Trading". In: *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. 2005, pp. 393–399.

Schoreels, Cyril, Brian Logan, and Jonathan M. Garibaldi. "Agent based Genetic Algorithm Employing Financial Technical Analysis for Making Trading Decisions Using Historical Equity Market Data". In: *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. 2004, pp. 421–424.

Schultz-Moeller, Nicholas Poul, Matteo Migliavacca, and Peter Pietzuch. "Distributed Complex Event Processing with Query Optimisation". In: *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*. June 2009.

SEC. *Regulation NMS*. Rule RIN 3235-AJ18. Securities and Exchange Commission, 2005.

SEC. *Knight Capital Americas LLC*. Order 70694. Securities and Exchange Commission, Oct. 2013.

SEC. *SEC Charges Goldman Sachs With Violating Market Access Rule*. Press Release 2015-133. Securities and Exchange Commission, June 2015.

SEC. *Merrill Lynch Charged With Trading Controls Failures That Led to Mini-Flash Crashes*. 26th Sep. 2016.

Statistica. *Largest derivatives exchanges worldwide in 2016, by number of contracts traded*. 2016. URL: https://www.statista.com/statistics/272832/largest-international-futures-exchanges-by-number-of-contracts-traded.

Tabb Group. *FIND ME, FIX ME*. Ed. by TO FIND. Feb. 2010.

The Bank for International Settlement. *The sterling "flash event" of 7 October 2016*. Jan. 2017.

The Guardian. *Pound slumps to 31-year low following Brexit vote*. 23rd Jul. 2016.

The Guardian. *What caused the pound's flash crash?* 7th Oct. 2016.

The Telegraph. *Swiss franc surges after scrapping euro ceiling*. 1st Jan. 2015.

The Telegraph. *Citi trader exacerbated sterling flash crash*. 7th Dec. 2016.

Todd, Andrew, William Scherer, Peter Beling, Mark Paddrik, and Richard Haynes. "Visualizations for sense-making in financial market regulation". In: *IEEE International Conference on Big Data*. IEEE, 2014, pp. 730–735.

UK Gov. *The Future of Computer Trading in Financial Mmarkets: An International Perspective (Final Project Report)*. Tech. rep. Foresight 12-1086. The Governgment Office for Science, 2012.

Wall Street Journal. *Multiple Buyers, Not One, Influenced Most Active E-Mini Move of 2016*. 27th December. 2016.

Zhang, Frank. *High-Frequency Trading, Stock Volatility, and Price Discovery*. SSRN. 2010.