Trading Strategies for Autonomous

Agents in Financial Markets

Nabeel M Shah

School of Computing, Engineering and Mathematics

University of Western Sydney

A thesis submitted for the degree of

Masters of Computer Science (Honours)

2013
I would like to dedicate this thesis to my loving parents ... Dr Akhlaq M Shah and Samar Akhlaq
First of all, I would like to thank my principal supervisor, Associate Prof Dongmo Zhang, for all the opportunities, support, understanding and academic advice he gave me over the years. I would not have been able to complete this thesis without his great support as well as his patience. Most of the theories, approaches and results in this thesis were spawned from our discussions and joint work. I am also thankful to my co-supervisor Prof Yan Zhang and Prof Michael Thielscher for their valuable time and giving me opportunity to learn. Associate Professor Dongmo Zhang is also the supervisor for design and development of Jackaroo Trading Agent Platform (JTAP) team and he is the main contributor to Strategic Trading Game (STG) Competition, further he taught me to build trading agents and develop new ideas and solutions for trading automation throughout my thesis work. I would like to thank the rest of my research colleagues in Artificial Intelligence Research Group (AIRG), University of Western Sydney for creating an educational, enjoyable and challenging atmosphere. Meetings with members of the AIRG team led to the development of many new ideas presented in this thesis. One of the most unique and important contribution was of Miss Hua Li, who has completed her degree in economics from Boston and she is now pursuing her master
degree in Sydney. Hua had a main role in my learning curve of finan-
cial trading techniques and methods. Especially I would like to thank
Dengji Zhao and Chun Gao who implemented a substantial portion
of our software agent JTAP, and also their support and advices help
me to clarify many problems. I am highly grateful for the support
of my dear friends Mr Shahbaz Chattha and Miss Upasana Rai in
keeping me motivated and looking after me like a family. I could not
have completed my thesis work without the prayers of my parents
and family members Mr Noman M Shah, Mr Naqqash M Shah, Miss
Nazziha Akhlaq and Mr Nayyer M Shah - and to all my friends who
inspired me in many ways.
Abstract

Electronic trading is relatively new to the long history of financial markets. The typical traditional way of trading in a financial market was that traders gathered in the market floor and exchanged their financial assets in a manner of open outcry. In the past twenty years or so, financial markets have been reshaped dramatically due to the use of electronic trading. Innovations in computing and Internet eliminate the need for direct person-to-person contact via trading floors and make possible global electronic order routing, broad dissemination of trade information, and effective market operations. Electronic trading systems for financial trading allow to match buy and sell orders automatically without human intervention. Such a system lowers trading costs, bypasses human intermediaries, such as dealers, and also offers faster trade execution.

The automation of market operations makes possible for traders to trade automatically. The advances of agent technology have resulted in the emergence of agent-based systems that automate business activities. Ambitious attempts have been made by academia and a number of companies to create automated programs, known as trading agents, which are capable of autonomously trading on behalf of human traders in financial markets. It has been widely recognised that
the automation of traders’ decision making creates more challenges than machinizing market making. The most crucial decision-making for either a human trader or an autonomous trading agent is when, how much and what to buy or sell an asset in a financial market, known as trading strategies.

This thesis aimed to shade a light on the research of autonomous trading agents by introducing an approach of trading strategy design and analysis. To mimic the way of decision-making by human traders, we investigated a number of financial technical indicators, including Simple Moving Average, Exponential Moving Average, Moving Average Convergence and Divergence, Relative Strength Index and Stochastic Oscillator, that have been widely used by human traders for market analysis and used them as the major signals for design of bidding and investment strategies. By utilising our Jackaroo Trading Agent Platform (JTAP), we first analysed the statistic properties of each technical indicator using the real data of stock exchange taking from the Australian Stock Exchange and then implemented a range of trading strategies based on the technical indicators we have investigated. The trading strategies we designed can be divided into two categories: bidding strategies and investment strategies. A bidding strategy deals with when, how many and how much to buy or sell a financial product, whereas, an investment strategy decides how much fund is allocated to each financial product. A simple bidding strategy can be easily designed based on either trend based indicators or momentum based indicators, and an investment strategy can be
implemented based on the enforcement learning with returns as rewards. However, our experiments showed that a trading strategy can perform much better if we combine a number of technical indicators and market returns. This is because such a combined signal can be more stable under volatile market conditions and more responsive to market returns. Each trading strategy we implemented has been fully tested against the real market data and simulated market data.

As part of research training during my master honours study, I participated in the design and development of Jackaroo Trading Agent Platform, mainly focusing on the interfaces of trader and trading strategies. I was also one of the co-organisers of the First Australasian Strategic Trading Game (http://ai12.org/challenges.html).
# Contents

List of Figures xi

List of Tables xiv

Nomenclature xiv

1 Introduction 1

1.1 Background 1

1.2 Motivation 3

1.3 Related Work 4

1.4 Trading Agent Design 5

1.5 Simulation Environment 7

1.5.1 The Market 7

1.5.2 Jackaroo Trading Agent Platform 8

1.5.2.1 Server 9

1.5.2.2 Traders 11

1.5.2.3 Markets 11
1.6 Major Contributions ........................................... 12

2 The Model of Markets and Traders .......................... 14
  2.1 Introduction .................................................... 14
  2.2 The Model of Financial Markets ......................... 15
    2.2.1 Market Mechanism .................................... 16
    2.2.2 Market Policies ....................................... 18
  2.3 Technical Indicators ...................................... 18
    2.3.1 Simple Moving Average ............................... 20
    2.3.2 Exponential Moving Average ....................... 21
    2.3.3 Moving Average Convergence-Divergence .......... 24
    2.3.4 Relative Strength Index ............................ 27
    2.3.5 Stochastic Oscillator ............................... 30
  2.4 Market Simulation .......................................... 32
    2.4.1 Double Sided Pricing Model ....................... 33
    2.4.2 Random Pricing Model ............................... 34
    2.4.3 Market Clear Pricing Model ....................... 35
  2.5 Summary .................................................... 36

3 Trading Agent: Strategy Design and Analysis ............... 38
  3.1 Introduction ................................................ 38
  3.2 Trading Agent Strategy Design .......................... 39
  3.3 Bidding Strategies ........................................ 41
    3.3.1 Simple Moving Average Based Strategies ........ 42
    3.3.2 Exponential Moving Average Based Strategies ... 47
      3.3.2.1 Crossover Strategy ............................ 48
3.3.2.2 Rainbow Strategy ........................................ 49
3.3.3 Moving Average Convergence-Divergence Based Strategies .... 52
  3.3.3.1 Double-cross Strategy ................................. 52
  3.3.3.2 PPO vs MACD based Strategy ......................... 54
3.3.4 Relative Strength Index Based Strategies ...................... 56
3.3.5 Stochastic Oscillator Based Strategies ....................... 60
  3.3.5.1 Varying Stochastic Oscillator Strategy .............. 60
  3.3.5.2 Double Check Strategy ............................... 62
3.4 Investment Strategies ...................................... 63
  3.4.1 Fixed Investment Strategy ................................ 64
    3.4.1.1 Fixed Investment Strategy ......................... 65
  3.4.2 Rank Based Investment Strategy .......................... 65
3.5 Summary .................................................. 66

4 Implementation of Trading Strategies ............................ 68
  4.1 Introduction ............................................... 68
  4.2 Jackaroo Trading Agent Platform Configuration ................ 69
  4.3 Trading Agents .......................................... 70
  4.4 Strategies Implementation .................................. 71
    4.4.1 SMA Based Strategy Implementation .................... 71
    4.4.2 EMA Based Strategy Implementation .................... 73
    4.4.3 MACD Based Strategy Implementation .................. 76
    4.4.4 RSI Pseudo-code ..................................... 77
    4.4.5 SO Based Strategy Implementation .................... 79
  4.5 Summary ................................................ 81
5 Conclusion and Future Work

5.1 Conclusion ................................................. 83
5.2 Future Work ............................................... 85

A

A.1 Trading Strategies Configuration ............................... 87
   A.1.1 SMA based strategy configuration ......................... 88
   A.1.2 EMA based strategy configuration ......................... 88
   A.1.3 MACD based strategy configuration ....................... 89
   A.1.4 RSI based strategy configuration ......................... 90
   A.1.5 SO based strategy configuration ......................... 91

A.2 Market Side Settings ........................................ 91
   A.2.1 Double Sided Pricing Model ............................... 91
   A.2.2 Random Pricing Model ................................. 92
   A.2.3 Market Clear Pricing Model .............................. 92

Bibliography .................................................. 94
List of Figures

1.1 Jackaroo Trading Agent Platform ............................................. 8
1.2 Overview of Jackaroo Trading Agent Platform; left side represents autonomous trading agents; right side are the markets available for trading ................................................................. 12
2.1 Market Structure Model with Double Auction ............................... 17
2.2 SMA 1-day and SMA 15-Days generated using JTAP .................... 20
2.3 EMA 12-days and EMA 26-Days generated using JTAP ................. 23
2.4 MACD and Signal lines generated using 12-day short and 26 day long moving average generated using JTAP ....................................... 25
2.5 MACD minus Signal is the histogram generated using JTAP .......... 26
2.6 Relative Strength Index based on 14-day Price generated using JTAP 29
2.7 Market Price of stock generated using JTAP .................................. 29
2.8 Stochastic Oscillator based on 5-days price generated using JTAP 31
3.1 Chevron Process Providing an Overview of Trading Agent Strategies Design ................................................................. 41
3.2 Flow Chart for Bidding Strategies ............................................. 43
3.4 SMA 15 days upto SMA 120 Days Returns .................................. 44
3.5 SMA Short vs Long History SMA Performance ................. 44
3.3 SMA 15 days upto SMA 120 Days Indicator Mapping .......... 45
3.6 EMA 12-days to EMA-120 Days generated using JTAP ........ 50
3.7 EMA 12-days to EMA 180-Days Returns ..................... 51
3.8 EMA 12-days to EMA 120-Days ................................ 51
3.9 PPO EMA 9 days minus EMA 26 Days and 9 Days EMA of PPO expressed in percentage ............................. 54
3.10 Relative Strength Index 10-70 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP ............................. 58
3.11 Relative Strength Index 20-70 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP ............................. 58
3.12 Relative Strength Index 30-60 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP ............................. 59
3.13 Relative Strength Index 30-80 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP ............................. 60
3.14 Stochastic Oscillator based on a 5, 7, 9, 12 and 14 days price .... 61
3.15 Stochastic Oscillator based on a 5, 7 days price and MACD based on 12, 26 days moving average ............................. 63
3.16 Flow Chart for Investment Strategies .............................. 64
3.17 Fixed Investment Strategy with three stocks and five trading agents strategies ........................................ 65
3.18 Rank Based Investment Strategy with three stocks and five trading agents strategies .............................................. 66
List of Tables

3.1 SMA Based Strategy Productivity: This table provides useful information for number of signals (buy and sell) generated by each trading agent. This information can be further used to perform statistical analysis, whereas we can see that as the window of historical data increases, the signals tend to decrease. But on the other hand profits increase until a certain length of historical data. 47

3.2 EMA based Strategy Productivity Table. Buy and sell signals count and their respective profits show us that buy signals for EMA keep on decreasing as the length increases but sell signals are higher for 120-period EMA as compared to 90-period EMA, meaning that the signals can show irregular patterns based on the moment of underlying price. 49
Chapter 1

Introduction

1.1 Background

Originally trade was conducted manually in open floor environment, where a specialist’s judgement and experience would allow traditional auction. There are two ways of trading in financial exchange markets i.e. manually or electronically. Whereas, in late 20th century with the development in communication and technology trading has been moved to electronic systems. Traders from remote locations can connect to exchange markets and main reason for the traders in to adapt to change is the speed of trade. Comparatively, now it is easier for traders to maintain their inventory, clear their stock and earn profits right away. New York Stock Exchange is a most widely presented example of such a hybrid market. Markets with fully electronic execution started appearing in the 1980’s. In late 1990’s, Electronic Communication Network (ECN) was developed facilitating remote trading, which then became one of the reasons that encouraged the use of algorithms to execute buy and sell orders also known as automated trading. Fur-
ther, automated trading for financial markets came to light in 2001 when a paper was published by a team of IBM researchers at the International Joint Conference on Artificial Intelligence [11]. They demonstrated that in experimental versions of the electronic auctions used for financial markets, two algorithmic strategies (IBM’s own MGD (Modified Gjerstad Dickhaut) and Hewlett-Packard’s ZIP) could consistently out-perform human traders [10]. MGD was a modified version of the “GD” algorithm invented by Steven Gjerstad & John Dickhaut in 1996/7 [18]; the ZIP algorithm had been invented at HP by Dave Cliff (Professor) in 1996 [9]. In their research, the IBM team discovered that the financial impact of their results showing MGD and ZIP outperforming human traders. This research caught the attention of media and various investment companies. Furthermore, machine learning technologies have been part of other experimentation [2], [31], [19].

Most recent research interest is based on designing strategies for trading agents utilizing financial analysis techniques such as technical indicators, e.g. [36], [1], [13], [12], [21]. A framework was introduced by Vytelingum which can be useful for strategy design [40]. Our goal in this research is to design strategies that are flexible enough to trade in multiple financial markets, not necessarily optimal. We will start by designing strategies using financial technical indicators, these indicators are also used by human traders for markets like stock exchange and then simulate these strategies into software based automation platform called Jackaroo Trading Agent Platform (JTAP).
1.2 Motivation

Our question here is, how trading strategies can be designed for autonomous agents in financial markets, so that they reflect the human trading techniques whilst having the ability to perform in various markets conditions? Method proposed to achieve the goal is by dividing the trading agent design into modules based on the decision, when to place shout in markets, and how much to invest using financial technical trading rules. These decisions will be made by two different strategies.

Although electronic financial markets are software assisted in these days, but traders on the other hand are still human beings. This means that they are not fully capable of handling large amounts of data and processing it in an efficient way. Secondly, the fact that markets are electronic, puts a trader in a situation where data is changing rapidly, allowing the analysis for traders to be immensely hectic. Thirdly, the information generated with the aid of software is massive which in return makes decision to be complicated, where a trader may not be able to keep up with the speed of changing trading scenarios. Therefore, to overcome above mentioned limitations autonomous agents can be used. Autonomous agents utilise algorithms to trade, eliminating human intervention. Unlike humans agent-mediated traders do not require to rest, so they can trade around the clock, contributing to the business profits.

To test trading agent strategies in a simulation with financial markets we needed a simulation system. Therefore, JTAP simulation system (Details of implementation are in section 1.5.2) was developed by Artificial Intelligence Research Group at University of Western Sydney under the supervision of Dr.
Dongmo Zhang. JTAP simulation environment is able to simulate traders strategies and has a moulder design so that various strategies can interact with each other. In parallel this system has the capability to simulate multiple markets. Simulation platforms need to be comprehensive enough to adapt complicated market environments so that the trading can be sampled, and at the same time simple enough for researchers to easily implement models of markets and trading strategies.

Another influence on this research is the annual Trading Agent Competition (TAC) [42]. The competition has been run successfully for over twelve years and has made significant impact on the research of trading agents and markets, especially for double auction, supply-chain management, ad auction and energy trading [29], [23].

1.3 Related Work

The foremost concern while designing trading strategies for autonomous agents is its decision making capabilities. Various ways have been adopted in multiple research efforts initially starting with machine learning approaches [11]. Most recently AI experts have started deploying techniques used by human traders, an example that relates to our work is [15]. They have attempted to simulate strategies in an automated market simulation, although their work is closely related in a sense that they design trend following trading strategies for exchange markets, but this thesis tends to explore a wider range of technical indicators (including trend following and momentum based) in a multi-agent simulation environment. So, by simulating trading strategies for autonomous trading agents
on a multi-agent platform sets this research apart from any other work done in autonomous agent based computing.

Over the period of time, our research team has made considerable amount of contributions to the above mentioned research area through strong publications, including the top journals and conferences in Artificial Intelligence (AIJ, JAIR, IJCAI, AAAI, AAMAS) [47], [44], [46], [45] and has a track record of outstanding performance in the Trading Agent Competition.

JTAP has been recently used as a game server for a trading agent competition known as Strategic Traders Game. The First Australasian Strategic Trading Game (http://www.jackarooMarket.org), which was held in December 2012 in conjunction with the 25th Australasian Joint Conference on Artificial Intelligence (http://ai12.org). The game is designed for people who want to practice their trading skills of investment in financial markets. With the same amount of capital, time, and simulated market conditions, each participant in the game designs and implements a trading agent (acts as a trader) based on the provided Jackaroo Trading Agent Platform (JTAP) to buy or sell assets in multiple markets. The competition environment of markets offers three different stocks for exchange with the real price data from the Australian Stock Exchange (ASX) and multiple teams from various research groups across the globe connect with their own design of trading agent strategies.

1.4 Trading Agent Design

Trading agent is a term used for software agent that will perform in a software environment. In our case it will be a autonomous trading agent capable of decision
making. Automated trading also known as algorithmic trading, it is the use of software system for trading. An algorithm deciding on properties of transactions such as the time, price and quantity, or in many cases initiating transactions without human intervention [25].

With the available technology and the demands of fast changing financial conditions, a large amount of today’s trading is conducted electronically [22], [14]. Markets are run by protocols that define the rules for trading agents. Decisions are made by trading agents that might be well optimized in some market environments but that same decision might not be an optimized solution in other market. For example, market mechanisms are modified into complex mechanisms, such as eBay (proxy bidding as an implementation of English Auction and Vickery auction with exceptions) where we keep on bidding so that one’s strategy is no more optimal.

Broadly stating the logical design of autonomous trading agent can be divided into logical modules:

- Market Data module is where the market generates its price, quotes and other information (e.g. news) that is useful for the traders.

- Decision Making module from the traders point of view enables the trading agents to perform certain analysis based on strategy to achieve their trading goal. Whereas on the other hand allows the markets to run under set of rules that govern the marketplace.

- Processing module will take the decision of the traders and process it with the market to achieve the outcome, which may not be the same as the trader expects. For example bids from trading agents may get rejected.
1. Introduction

- Reporting module will store and present the information that is the outcome of processing module.

1.5 Simulation Environment

Simulation environment has an important role to play in experimentation, it is critical to have the environment configured in an appropriate manner so that it depicts the real world counterpart. Simulating financial market has its own challenges, where configuring a market simulation is one task and the mechanism under which this market will run is another. A particular financial market we have chosen to simulate is the stock exchange, other simulation efforts include market like derivatives [24], [15].

1.5.1 The Market

Stock markets are markets in which shares are issued and traded either through exchanges or over-the-counter markets. Also known as the equity market, it is one of the most vital areas of a market economy as it provides companies with access to capital and investors with a slice of ownership in the company and the potential of gains based on the company’s future performance. Stocks are the foundation of nearly every portfolio. Historically, they have outperformed most other investments over the long run. Stocks are generally traded in stock markets. A holder of stock (a shareholder) has a claim to a part of the corporation’s assets and earnings. In other words, a shareholder is an owner of a company. Ownership is determined by the number of shares a person owns relative to the number of outstanding shares. For example, if a company has 1,000 shares of
stock outstanding and one person owns 100 shares, that person would own and have claim to 10% of the company’s assets. Stock market has been subject to experimentation in the past for agent based trading [6].

1.5.2 Jackaroo Trading Agent Platform

JTAP is a multi-agent autonomous software system implemented in a client-server structure. The purpose of this system is for the design, analysis, implementation and evaluation of trading agent systems [17]. JTAP has the flexibility of simulating the properties of various market models and strategies for the traders.

![JTAP Diagram](image)

Figure 1.1: Jackaroo Trading Agent Platform

JTAP consists of 3 components: the server, traders and markets.
1.5.2.1 Server

The server is composed of 5 components: central controller, clock, connection manager, registry and report utility. A server listens on port 9090 upon startup. Clients connect to the port and establish connections. Connections work in a way similar to a persistent HTTP connection. Normally they should be kept alive from the beginning of a game through the end and process all the request and response sessions. Let us describe the functionality of servers components:

The central controller is the coordinator of the system. In normal operation of the game\textsuperscript{1}, all traders and markets are required to check in, i.e., make their presence known to the server, and this is normally done just prior to the start of each game (i.e., before the first Trading Day). Each trader and specialist\textsuperscript{2} is then assigned a unique ID at this point, and this ID will remain unchanged throughout the course of a game. Typically, the ID is the same as the short name suggested by the entrant; the server will assign the suggested short name as the ID, provided that this short name does not conflict with existing IDs in the system. After all the specialists and traders have checked in (or until a time out), the server announces that the game is starting and informs the clients of the number of days, rounds per day, and the length of each round (in milliseconds).

The clock synchronizes actions among clients and the server. It is divided into 3 sections: Days, Rounds and milliseconds. Days represent a day of trading during a simulation. It is associated with events such as day opening, day open, day closing and day closed. Each day can be configured to have multiple rounds depending on the needs of a particular simulation. Rounds are created to sub-

\textsuperscript{1}game is a logical representation of one successful simulation completed
\textsuperscript{2}Specialist is a market specialized in trading a single commodity
divide a day in trading sessions, for example days can represent the number of
hours of trading reflecting a scenario similar to a real world market or it can any
number of logical sub-divisions. Each round has to last for a certain amount of
time period. This time period is given in milliseconds and it is the smallest level
of controllable variable in the servers clock module.

The connection manager of JTAP supports three different methods of commu-
nication between the server and clients: asynchronous socket based, asynchronous
message-queue based and method invocation based. Multiple methods of commu-
nication are for the sake of convenience in doing experiments. Method invocation
based communication is usually faster to simulate and the easiest way to debug.
The connection manager is implemented using socket based connection. It man-
ages the connection and communication requests between clients and server. In
case a trader loses connection with the server at any time during the game, it is
the responsibility of the traders to recognize this event and to reconnect again
with the server. When reconnecting, the traders will log-in again, using the ID
previously assigned by the server, and re-enter the ongoing competition from the
next trading day. Once a game starts, the server will only allow existing clients
to re-connect or new clients to connect if these clients arrive from machines with
a registered IP address and using an approved ID (i.e, an IP address and ID
previously approved by the tournament organizers).

The registry records all the information and activities for the system. Server
can keep on posting all this information for storage. Stored information can be
used in various ways, particularly, it is used for generating web page reports.
This is needed in a scenario when remote clients are connected and want to view
results on the web in real-time.
1. Introduction

The report utility can print out the information needed for future analysis. This information is stored in comma separated values (CSV).

1.5.2.2 Traders

Each trader is an intelligent agent. There are two layers of trading strategies for each trader. The first layer of trading strategy decides which market to trade in and how much to invest, second layer of trading strategy performs analysis on the commodity to buy or sell. The second layer trading strategies trade in determined markets independently from the first layer trading strategy. The first layer strategy will evaluate the performance of the current second layer trading strategies and make adjustment where it is necessary. All of the traders first layer strategies and second layer strategies have access to the same information received from markets.

1.5.2.3 Markets

Each market is an intelligent agent, which can be programmed to represent various markets. There are a number of policies implemented for each market. By editing the functions in these policies, different market mechanisms can be implemented. Markets also send out information to traders for analysis.
1. Introduction

A server listens on port 9090 upon start-up. Clients connect to the port and establish connections. Connections work in a way similar to a persistent HTTP connection. Normally they should be kept alive from the beginning of a game through the end and process all the request and response sessions.

1.6 Major Contributions

This thesis contributes to design and development of trading agent strategies in the following ways:

- Introduce number of trading strategies using technical financial indicators.
- Implementation of above mentioned strategies with JTAP system.
1. Introduction

- Conduct a set of experiments for testing the efficiency of trading agent strategies.

- Introduce market asset pricing models to simulate the real market price movement.

- Finally, this research conducts experiments to propose and implement a new game for Trading Agent Competition (TAC). Whereas the trading agent competition is designed to help researchers solve problem relating to trading agents, market mechanism and multi-agent systems. Which also contributes towards the design enhancement and testing of autonomous multi-agent system, that is JTAP.

So, we can briefly introduce the chapters of this thesis in the form of an outline. Chapter 1 provides the motivation, background and the initial environment of trading strategies for autonomous agents, and is summarised with the major contributions. Chapter 2, goes into the details of financial market models and trading indicators which will be used for trading strategy design, finally Chapter 2 introduces pricing models for markets, these will be used in simulations. Chapter 3 contributes to the trading agents strategy design and analysis of simulation results. Chapter 4 contains algorithms relating to the design and simulation of trading strategies discussed in Chapter 3. Lastly, Chapter 5 is the conclusion of this thesis and the future work.
Chapter 2

The Model of Markets and Traders

In this chapter we first introduce the basic concepts about a financial market. Then we define the technical indicators that we will use in our trading agents. Finally we introduce our price model which has been used in market simulations with our JTAP system.

2.1 Introduction

To design an autonomous software agent that is capable of trading in financial markets, the agent must have a model of that market so that it can understand the market movement and make right decision. In this chapter, we will focus on how autonomous trading agents model a financial market for financial analysis and trading strategy design. Section 2.2 gives details of the model of financial markets.
To trade in numerous market scenario’s, autonomous trading agents need to adapt certain technical methods of analysis [3]. Therefore, we will develop a deeper understanding of various financial technical indicators as techniques for trading. Technical indicators are tools used by human traders in real world financial markets for financial analysis, market movement prediction and its strength. Technical Indicators have long been in practice by financial experts for their returns on investment and future price prediction properties. Many research studies have proved the importance of technical indicators, particularly focusing on trading strategy design and analysing the statistical characteristics for performance in real world markets [26], [7].

In section 2.3, we introduce five technical indicators: Simple Moving Average, Exponential Moving Average, Moving Average Convergence Divergence, Relative Strength Index and Stochastic Oscillator. All of them have their unique properties and some are related to one another. Some of this work relies on using price history available from various exchange markets across the globe. We will use daily prices of stock exchange and simulate market price movement within a day.

Section 2.4 gives details of price simulation using our market policies and pricing models between the bounds of the high and low price for the real market price. Then our trading agent strategies will be designed by using Indicators and trade autonomously in simulated market environment.

2.2 The Model of Financial Markets

A Financial market is an institution where traders buy and sell financial commodities. Typically financial commodities include securities, future contracts,
gold, agricultural goods and other fungible items. Conventional models of financial markets are based on assumptions of rationality and market efficiency, therefore, they are extremely well-designed in practice. Unfortunately, no single model to date has been capable of explaining the basic characteristics of real financial markets by experimentation [38]. A financial market can normally be characterised by factors such as; open pricing, basic policies on trading, costs and fees, other than that market transactions determine the prices of securities that are traded. Financial market models are based on a particular trading mechanism such as auction. The design of a mechanism is comprised of a set of policies. These policies administer the trading in that particular financial market. In the subsections we will describe the market model based on double auction mechanism.

2.2.1 Market Mechanism

Traders can trade in multiple markets; collectively these markets are known as marketplace. For the transactions to occur in a marketplace between markets and traders, there needs to be a trading mechanism in-place. The mechanism proposed for the market modelling will be the double auction. Almost all financial markets use double auction mechanism as a way of trading. It is particularly popular because it is operationally simple, efficient and it can adapt to dynamic market conditions.

The way in which double auction works is that, there are three participants i.e., buyers, sellers and the market maker. Based on certain policies buyers submit bids (buy orders) and sellers submit asks (sell orders) until they come to a match
2. The Model of Markets and Traders

this is when the market chooses some price at which the market is cleared, [16].

Figure 2.1: Market Structure Model with Double Auction

Figure 2.1 shows a structural model of a double auction trading mechanism with multiple polices. Initially buyers and sellers submit their shouts, after accepting shouts, the market matches certain asks and bids. These matched shouts are cleared according to a particular clearing policy. Meanwhile the price is decided by the pricing policy and the charging policy will be implemented while the transaction is processed to make profit.
2.2.2 Market Policies

A double auction is operated with the following major policies.

- Accepting policy: Determines which shouts are chosen for further processing and which shouts are immediately rejected.

- Matching policy: Determines which asks and bids are matched for transaction.

- Clearing Policy: Determines when to clear the matched pair of shouts.

- Pricing Policy: Determines the price of a transaction that is assigned to a matched pair of bid and ask orders as clearing price.

- Charging Policy: Determines fees to be charged from traders for transactions. Indeed there are multiple transaction fees that a market can consider applying. These may include registration, information, shout, transaction and profit fees.

Considering the fact that price is a dynamic component of trading, hence forth we will proceed into details of the pricing models in section 2.4. Note that pricing is closely related to matching policy and clearing policy.

2.3 Technical Indicators

Financial technical indicators derive their values from the price of financial markets, such as stock exchange, these values are used to anticipate price movements
2. The Model of Markets and Traders

of that stock in future. Historical prices are considered in the process of calculating the financial technical indicators. Financial technical indicators allow the human traders to perform financial analysis and follow market trends.

Financial indicators can be divided into two categories: momentum indicators and trend indicators. Momentum defines the strength measurement of a trend. In other words, if the market is increasing its value this means trend is upwards and can be referred to as bullish market and if it is decreasing in value then the market trend is bearish. Hence, if the market tends to keep on increasing or decreasing this measurement of strength is the momentum. Financial technical indicators can be used in combination with one another, specifically based on the fact that whether it is a trend indicator or a momentum indicator.

Simple Moving Average (SMA), Exponential Moving Average (EMA) and Moving Average Convergence Divergence (MACD) are trend indicators. Details of these indicators will be discussed along with Relative Strength Index (RSI) and Stochastic oscillator (SO), where RSI and SO are momentum indicators. Particular interest in technical indicators can be seen from the fact that they have long been a topic of debate for research to justify their performance in trading [7]. The use of financial technical indicators for designing trading strategies in agent-mediated market simulation can be different as compared to how human traders are utilizing them. One reason for this is the fact that the computers are good for calculations but on the other hand lacking the capability of a human trader sense of financial analysis, for example to recognise a false indication.
2. The model of markets and traders

2.3.1 Simple Moving Average

Simple Moving Average (SMA) is one of the most widely used and basic type of technical indicators. The idea behind computation of SMA is to smooth out the volatility of price movement. Simple Moving Average is calculated by summing up the price $p_n$ of a stock for $N$ periods price and then dividing this sum by $n$ period.

\[ SMA_n = \frac{p_1 + p_2 + p_3 + \cdots + p_n}{n} \]

where, $p_1, p_2, p_3, \cdots, p_n$ are the $n$ period historical prices.

For the purpose of technical analysis of price trends different SMAs are calculated in terms of length of history of data. An SMA is called a short-term SMA with a smaller value of $n$, whereas the long term SMA is based on a larger value of $n$, meaning that the short term SMA will be more reactive to the price change and the long term SMA having a slower effect.
Figure 2.2: SMA 1-day and SMA 15-Days generated using JTAP

Fig 2.2 demonstrates the way SMA works. This graph has been generated using a popular SMA period, with 1-day as the short-term moving average and 15-days long-term moving average. The data of underlying price was taken from Australian Stock Exchange for the period of 2009 - 2012.

Both the SMAs confirm upward or downward trends, by comparing the short term moving average and long term moving average. We can see that when the short-term and long-term moving averages cross, the buy and sell signals are triggered. If the short-term moving average has crossed above the longer-term moving average a buy signal is triggered for traders. Conversely, if the short-term moving average has crossed below the long-term moving average, a sell signal is triggered. SMA has been used in technical analysis by traders from different international financial markets [8].

2.3.2 Exponential Moving Average

Exponential Moving average (EMA) is a technical indicator for financial analysis, we have used this as a trigger for buying and selling in the implementation of our strategies (refer to section 3.3), it is also known as exponentially weighed moving average. EMA works best in trending markets. It is not to be confused with Simple Moving Average, as it behaves quite differently from a Simple Moving Average because it is a function of the weighting factor on length of the average. EMA is calculated by increasing the weight to the most recent days. Proportionally decreasing weight as we move on by a factor $\alpha$ known as the smoothing factor. The main attribute of the exponential moving average is to understand the use
2. The model of markets and traders

of smoothing factor. The smoothing factor is also a reason why EMA performs better than SMA, it removes short-term fluctuations and provides a market trend indication.

\[ \alpha = \frac{2}{n + 1} \]

let \( n \) be the number of periods of history prices EMA would be looking back into. In the following equation \( EMA(C, n) \) represents the exponential moving average of current day, similarly \( EMA(C, n - 1) \) is the EMA of yesterday.

\[ EMA(C, n) = EMA(C, n - 1) + \alpha \times (C - EMA(C, n - 1)) \]

\( EMA(C, n) \) can be rewritten as follows, which gives us the idea how EMA moves towards the most recent price but rather by a small fraction.

\[ EMA(C, n) = (\alpha \times C) + (1 - \alpha) \times EMA(C, n - 1) \]

\( EMA(C, n - 1) \) is a recursive calculation, so to simplify the equation for the first \( n \) days a simple moving average can be calculated.
2. The model of markets and traders

Figure 2.3: EMA 12-days and EMA 26-Days generated using JTAP

Figure 2.3 illustrates the properties of EMA. A buy signal occurs when the short-term EMA (12-day EMA as in figure) crosses above the longer-term EMA (26-days EMA, as in figure). Conversely, a sell signal is generated when the short-term EMA (12-day EMA) crosses below the longer-term EMA (26-day EMA). Given these facts, following is a description of some pros and cons that are associated with EMA.

- Exponential Moving Average is a easy to follow in-terms of strategy design as signals can be picked clearly.

- EMA neatly specifies average price changes over time and smooths out trading noise.

- It is a good trading indicator which can be used to study the past performance of stock exchange and will give signals on market trends for future trading.
• As all moving averages lag, this also affects EMA indications.

• Ineffective for the flat market prices, or in other case if the market is too volatile it might give delayed signals or even false triggers.

Trading strategies based on EMA signals will be presented in section 3.3.2 of chapter 3. Its implementation will be briefly described in chapter 4.

### 2.3.3 Moving Average Convergence-Divergence

As its name implies, Moving Average Convergence-Divergence (MACD) is all about the convergence and divergence of moving averages. When the moving averages move towards each other, it is known as convergence. Divergence is when the moving averages move away from one another. The short-term moving average (e.g. 12-day EMA) is faster and most MACD movements occur because of this. The longer moving average (26-Day EMA) is less reactive to price changes in the financial market, hence is slow to react.

Figure 2.4 and 2.5 represent two calculations of MACD which are used by traders for financial market analysis. Figure 2.4 shows that MACD line oscillates above and below the signal line, if the difference between the short-term moving average is not significant then it is hard to pick a signal. Therefore MACD utilizes second method to indicate the signal which can be seen in figure 2.5. It is a histogram generated by taking the difference between the MACD line and signal line in figure 2.4. Mathematically this calculation can be written as:

\[
MACD = EMA(C, n_x) - EMA(C, n_y)
\]
Where, $x < y$

$$Signal = EMA(MACD, n)$$

Where $C$ represents the closing price of that stock at the day under consideration. $MACD$ is a difference of the $EMA(C, n_x)$, $n_x$-day EMA (short moving average) and $EMA(C, n_y)$, $n_y$-Day EMA (longer moving average), represented by the blue line in graph 2.4. The red line in graph 2.4 represents the $Signal$, which is the n-day exponential moving average of $MACD$. Accordingly, the traders can get three meaningful indications from the above mentioned calculations:

- $MACD - Signal = 0$: This crossover indicates that the $MACD$ has crossed the $Signal$, the direction surely represents the trend of the market. Positive MACD indicates that $MACD$ is above the $Signal$ line. Negative MACD means that $MACD$ is below the $Signal$ line.
2. The model of markets and traders

Figure 2.5: MACD minus Signal is the histogram generated using JTAP

- $EMA(C, n_x) - EMA(C, n_y) = 0$: Positive values increase as the short-term $EMA(C, 12)$ diverges further upwards from the longer $EMA(C, n_y)$, this means upside momentum is increasing. Negative values increase as $EMA(C, n_x)$ diverges further below $EMA(C, n_y)$, this means downside momentum is increasing.

- relative $Price \neq$ relative $MACD$: higher highs or lower lows on the price graph but not on the MACD line (blue line). Higher highs or lower lows on the price graph but not on the histogram, represent strength of momentum.

Closing prices are used to form the moving averages. A n-day EMA of MACD is plotted along side to act as a signal line to identify turns in the indicator. The MACD-Histogram represents the difference between MACD and its 9-day EMA, the signal line. The histogram is positive when MACD is above its n-day EMA and negative when MACD is below its n-day EMA, graph in Figure 2.5 is a histogram.
2. The model of markets and traders

The MACD-Histogram is an indicator based MACD indicator. In fact, MACD is also an indicator of an indicator. This means that the MACD-Histogram is four steps from the price of the underlying stock. In other words, it is the fourth derivative of price.

1. First derivative: short-term EMA and long-term EMA
2. Second derivative: MACD (short-term EMA minus the long-term EMA)
3. Third derivative: MACD signal line (n-day EMA of MACD)
4. Fourth derivative: MACD-Histogram (MACD minus signal line)

Therefore, MACD is designed to anticipate changes in the price momentum of the underlying stock. Strategies utilizing the MACD indicator will be presented in 3.3.3 of chapter 3 and the implementation of these strategies will be given in chapter 4.

2.3.4 Relative Strength Index

Developed by J. Welles Wilder, the Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100. Original model of RSI considers overbought when above 70 and oversold when below 30. Signals can also be generated by looking for divergences, failure swings and centerline crossovers. RSI can also be used to identify the general trend [43].

RSI is an extremely popular momentum indicator that has been featured in a number of articles and books over the years [8].
2. The model of markets and traders

Mathematically RSI is represented as following:

\[ RSI = \frac{100}{1 + \frac{\text{gain}(t)_{\text{avg}}}{\text{loss}(t)_{\text{avg}}}} \]  

(2.1)

To simplify the calculation explanation of equation 2.1, RSI has been broken down into its basic components: Average Gain (Equation 2.4) and Average Loss (Equation 2.5). This RSI calculation is based on \( n \) periods of historical prices, whereas 14-days is the default period suggested by Wilder in his book. Losses are expressed as positive values not negative values. The very first calculations for average gain and average loss are simple \( n \) period averages, where \( t \leq n \), \( t \) is the current period for which the RSI is being calculated.

\[ \text{gain}(t)_{\text{avg}} = \frac{\sum_{t=1}^{n} \text{gain}(t)}{n} \]  

(2.2)

\[ \text{loss}(t)_{\text{avg}} = \frac{\sum_{t=1}^{n} \text{loss}(t)}{n} \]  

(2.3)

Where,

\[ \text{gain}(t) = C(t) - C(t - 1) \]

\[ \text{loss}(t) = C(t - 1) - C(t) \]

The second and subsequent calculations are based on the prior averages and the current gain loss, where \( t \geq n \) current period is enough to calculate \( n \) period RSI:

\[ \text{gain}(t)_{\text{avg}} = ((\text{gain}(t - 1)_{\text{avg}}) \times (n - 1) + \text{gain}(t))/n \]  

(2.4)
2. The model of markets and traders

\[
\text{loss}(t)_{\text{avg}} = ((\text{loss}(t - 1)_{\text{avg}}) \times (n - 1) + \text{loss}(t))/n
\]  \hspace{1cm} (2.5)

Taking the prior value plus the current value is a smoothing technique similar to that used in exponential moving average calculation. This also means that RSI values become more accurate as the calculation period extends.

Figure 2.6: Relative Strength Index based on 14-day Price generated using JTAP
2. The model of markets and traders

Figure 2.7: Market Price of stock generated using JTAP

Figure 2.6 is a RSI generated where \( n = 14 \). Figure 2.7 is the market price movement, comparing the two we can easily conclude that the indications being generated by RSI are useful for analysis on price momentum.

2.3.5 Stochastic Oscillator

The theory behind SO indicator is that in an upward-trending market prices tend to close near their high and during a downward-trending market prices tend to close near their low. Transaction signals occur when the \( %K \) value crosses through a three-period moving average called \( %D \). As a bound oscillator, the Stochastic Oscillator makes it easy to identify overbought and oversold levels. The oscillator ranges from zero to one hundred. Traditional settings use 80 as the overbought threshold and 20 as the oversold threshold. Following is the list of convention of symbols that will be used by SO:

- \( P \): Market Price
- \( C \): last closing price
- \( L \): lowest low price for last \( n \) days
- \( H \): highest high price for last \( n \) days
- \( n \): number of days of statistical price

\[
%K = 100 \frac{(C - L)}{(H - L)}
\]

\[
%D : 3 - EMA(%K)
\]
2. The model of markets and traders

Here we can see that the value of n is from the stylised facts and has a direct effect on the SO value $\%K$. $\%K$ will be used to determine the direction of stock price movement.

Figure 2.8: Stochastic Oscillator based on 5-days price generated using JTAP

The Stochastic Oscillator measures the level of the closing price relative to the high minus low price range over a given period of time. Assume that the highest high equals 110, the lowest low equals 100 and the close price equals 108. The high minus low range is 10 which is the denominator in the $\%K$ formula. The close less the lowest low equals 8, which is the numerator, 8 divided by 10 equals 0.80 or 80%. Multiply this number by 100 to find $\%K$ would equal 30 if the close was at 103 (.30 x 100). The Stochastic Oscillator is above 50 when the close is in the upper half of the range and below 50 when the close is in the lower half. Low readings (below 20) indicate that price is near its low for the given time period. High readings (above 80) indicate that price is near its high for the given time period. Figure 2.8 is a representation of fast Stochastic Oscillator because it represents the value of $\%D$ as a 3-period moving average of the $\%K$. 
2. The model of markets and traders

2.4 Market Simulation

Market simulations are virtual marketplaces that allow transactions between market maker and autonomous trading agents. Ideally, simulated markets should have similarities in characteristics with the real markets [28]. Our focus will be to simulate financial markets, so that trading agents can simulate strategies based on financial technical indicators discussed in section 2.3[35]. The earliest artificial agent-based financial market, including the highly prominent Santa Fe artificial stock market has been studies and surveyed in detail by [4]. The latter study develops a dynamic theory of asset pricing based on heterogeneous stock market traders who update their price expectations individually [37].

There are two challenges when we simulate a financial market such as stock exchange. The first one is to implement the market mechanism that is used in the financial market. As we have described in Section 2.2.1, most financial markets use double auction as the mechanism for market clearing. In our JTAP system, a built-in double auction mechanism has been implemented with variety of market policies. The other challenge is that how to mimic the price movement of the real stock market. Which is part of market making known as pricing policy. For the purpose of simulation we used stock prices from the Australian Stock Exchange (ASX). Since the history data for a share in a stock market are mostly limited to daily prices, such as day average, day high, day low as well as day open and day close, their arises a need to simulate the market price following certain pricing model meanwhile reflecting the real world financial market price. In this section, we details our approach of simulating market price using daily market low price and daily market high price. The price generated is between certain high and low
2. The model of markets and traders

values, where these high and low value comes from the real stock prices historical
data. After reproducing this data each day of the market simulation will have its
own high $H$ and low $L$.

The following market information is available to trading agents from the mar-
kets. Trading agents will acquire necessary information depending on the statistical
needs of trading strategy.

2.4.1 Double Sided Pricing Model

In this section, we describe the model we use to regenerate stock exchange market
price using daily market low and market high prices from the real stock exchange
market. This model uses the information of market balance of demand and supply
for market clearing. It clears traders asks first than the bids when the market is
going up (demand $\geq$ supply), and clear traders’ bids then asks when the market
is going down (demand $\leq$ supply). In each day, we have three prices, High ($H$),
Low ($L$) and Average ($Ave$). Let $B_m$ be randomly selected from $[L, Ave]$ and $A_m$
be randomly selected from $[Ave, H]$. $B_m$ and $A_m$ will be the bid and ask prices
for the market to play demand and supply role respectively.

If, Demand $\geq$ Supply: That is, the market price is going up. Since demand
is more than supply, so we will use bid prices to set up the market/transaction
prices. Market will play demand role in this case, where $B_m$ is the bid price given
by the market. The match price $P$ is defined as

$$P = \begin{cases} 
\min(B_{\text{max}}, H), & \text{if } B_{\text{max}} \geq L \\
B_m, & \text{otherwise}
\end{cases}$$
Where, $B_{\text{max}}$ is the maximum bid price among all bids. The matching procedure will be: if the current minimum ask price $A_{\text{min}} \leq P$, match $A_{\text{min}}$ with transaction price $P$ (and update $P$).

**If, Demand < Supply:** That is, the market price is going down. The market will play supply role. Market price $P$ is defined as:

$$P = \begin{cases} 
  \max(A_{\text{min}}, L), & \text{if } A_{\text{max}} \leq H \\
  A_{\text{m}}, & \text{otherwise}
\end{cases}$$

Where $A_{\text{min}}$ is the minimum ask price among all asks. The matching procedure will be: if the current maximum bid price $B_{\text{max}} \geq P$, match $B_{\text{max}}$ with transaction price $P$ (and update $P$).

### 2.4.2 Random Pricing Model

After each day, we have two prices, High ($H$) and Low ($L$). Let $C_{\text{m}}$ be the clear price of the market and be randomly selected from $[L; H]$. $C_{\text{m}}$ will be the bid prices for the market when the market is going up, and ask price for the market when the market is going down. In this market, the match priority is given to traders. The market only clears those shouts, which cannot be matched with other traders. This market uses PRNG to generate random factors and can give repeatable results under the same settings.

**When the market is moving up**, it means that demand is higher than supply. Thus we use the market to clear the asks sent by the traders, which means the
2. The model of markets and traders

market is using $C_m$ as the bid price. The match price $P$ is defined as:

$$P = C_m$$

When the market is moving down, it means that supply is higher than demand. Thus we use the market to clear the bids sent by the traders, which means the market is using $C_m$ as the ask price. The match price $P$ is defined as:

$$P = C_m$$

2.4.3 Market Clear Pricing Model

When the market is going up, market clearing pricing model uses market to clear traders asks ($demand > supply$), and clear traders bids when the market is going down ($demand < supply$).

Let $C_m$ be the clear price of the market and be randomly selected from $[L; H]$. $C_m$ will be the bid prices for the market when the market is going up and ask price for the market when the market is going down. In this implementation, $C_m$ is more smooth than the previous market model, so no sharp jumping during the same trading day. In addition, this market do not allow traders to trade among themselves, instead, all the shouts are cleared by the market. This market uses random number generator to generate random factors, and can give repeatable results under the same settings. When a market is moving up, it means that demand is higher than supply. Thus we use the market to clear the asks sent by
the traders, which means the market is using $C_m$ as the bid price. Denote $ask$ to be the price for the current ask from trader to be matched. The match price $P$ is defined as:

$$P = \begin{cases} 
    L, & \text{if } ask \leq L \\
    ask, & \text{if } L \leq ask \leq C_m 
\end{cases}$$

Note that if $C_m \leq ask$, this means that the ask price is too high for the round, and will be disregarded when clearing. When a market is moving down, it means that supply is higher than demand. Thus we use the market to clear the bids sent by the traders which means, the market is using $C_m$ as the ask price. Denote $bid$ to be the price for the current bid from trader to be matched. The match price $P$ is defined as:

$$P = \begin{cases} 
    H, & \text{if } H \leq bid \\
    bid, & \text{if } C_m \leq bid \leq H 
\end{cases}$$

Note that if $bid \leq C_m$, this means that the bid price is too low for that round and will be disregarded when clearing.

2.5 Summary

In this chapter we have gone through the details of how a trading agent models markets for trading. The market models discussed in this chapter for financial analysis of markets are requisite to create a decision based design for trading agent strategies similar to human traders. We started by giving an overview of what a financial market is, its types and internal mechanism then continuing on to double auction mechanism. Moving on to technical indicators we described
five technical indicators, from two different types that is trend indicators and momentum indicators. Three of trend indicators are SMA, EMA and MACD then two momentum indicators RSI and SO, the importance of these indicators lies in the fact that they will be the key to decision making in a trading agent strategy. To trade, the trading agent need to know the market model, hence, at the end market simulation model for trading agents needs to be expressed. Which was explained by understanding the fact on what policies or rules the market operates under in order for transactions to be completed. Three different pricing models gave the essence of the market policies or rules needed by trading agent for autonomous trading. Summing up we have given a model for the trading agents to follow. In other word financial markets can be modelled in a simulation from the traders point of view to design and implement trading strategies.
Chapter 3

Trading Agent: Strategy Design and Analysis

3.1 Introduction

This chapter will bring together trading strategies we have designed for autonomous agents in stock exchange market. Trading strategies will be categorized using technical indicators from human traders. Besides, we will also consider the fact that how well they can perform in an autonomous multi-agent simulation environment. Prior to designing the trading agent strategies we briefly describe how to simulate a stock market, each market represents a real stock in the Australian Stock Exchange. The simulation of real stocks is done to minimise the random effect reflected by trading agents during transactions and also for the reason to keep the simulation close to the real market. When the real data of the stock market is used as input to the system we only consider using the high and low price for that interval under consideration.
3. Trading Agent: Strategy Design and Analysis

Some concerns that arise during the design of trading agent strategies include decision of investment, taking advantage of a price difference between two or more markets (arbitrage) and trend following [5]. Keeping these concerns in mind our trading strategies design is based on technical indicators. To carry out number of experiments or generate multiple results for comparative study, variation in parameter settings for a strategy design is one way of optimizing strategies [27]. Another way is to test two different strategies based on the fact which technical indicator they are using, this allows comparison and analysis of two or more strategies with similar properties.

There are various other research efforts related to designing trading strategies for autonomous agent systems [1], [20], [39], [26]. What makes our work unique is that all our trading strategies have been fully implemented and tested with real stock exchange market with demonstrative performance.

3.2 Trading Agent Strategy Design

Trading agents strategy design needs to demonstrate flexibility in decision making. Following questions need to be answered, that have been identified earlier by Greenwald where they describe bidding agents for autonomous agents and later by Vytelingum, in which they purpose a framework for designing trading agent strategies [20], [41]. We will be answering similar questions with the help of technical financial indicators.

- How much to invest?
- What to buy or sell?
3. Trading Agent: Strategy Design and Analysis

- When to buy or sell?

- How much to buy or sell?

To answer these questions we have divided the trading agent strategies in two categories; first the bidding strategies followed by investment strategies.

1. First category is of bidding strategies. It has three parts in terms of decision making: when to buy or sell from markets, at what price and How much to buy or sell for. Details of each decision type are given in section 3.3.

2. Second category covers details of what should be investment strategies of trading agents? Investment strategies are part of the trading where the trader decides their investment portfolio. As of initial settings during the experimentation, the traders have pre-allocated funds. So to make use of this fund trading agent should be capable of making decisions on which market to invest in, as the agent has a number of choices for consideration.

The next question regarding the investment strategy after the agent has selected to invest in a certain stock is, how much percentage of its total available fund should go towards a certain selected market? To address this question each strategy can have a custom portfolio so that the funds can be moved around between the available markets as required.

The following diagram explains the flow of information as input is available to the traders for processing, this input information can be utilized by trading strategies for analysis. Based on the decisions made during the processing, output is generated, i.e., whether to buy or sell, at what price and quantity? It is processed in module through the trading agent.
3. Bidding Strategies

This section describes the design details of bidding strategies. Designs of various trading agent bidding strategies will be discussed. Strategy designs will be based on the technical indicators along with the experimental results carried out using Jackaroo Trading Agent Platform (section 1.5.2). There are three phases in bidding strategy design using technical indicators:

- The first phase to decide which technical indicator to use borrowed from finance, we have designed strategies to buy and sell stocks from agent-mediated markets. These indicators help the trading agents in understanding what is the trend or momentum of various stock markets at a certain time period.
• Once this decision has been made whether to buy or sell, then trading agents have to decide a price for placing shout in markets. Regarding the price of shout, in certain cases trading agents can just proceed with what is called a “market order”. Market order is when traders are not concerned about prices of the stocks, main concern here is to buy or sell that stock. The other option at this step is to go with a “limit order”. For a limit order the trading agent will have to decide the price of that shout which can be based on the margin of profit a trader wants to earn during transactions.

• Last but not least, is the quantity of a particular commodity. Trading Agents have at least two ways to decide on this, firstly, they can check their investment amount and invest a certain percentage or all of it. Secondly, the trading agent can make a decision of how much to buy or sell by analysing the intensity of movement of that stock.

3.3.1 Simple Moving Average Based Strategies

Short vs Long Simple Moving Average Strategy: The design and analysis of SMA based strategy in an automated simulation environment will provide performance comparison for variations in parameters. Details on model and use of SMA were discussed in 2.3.1. The challenge of designing a SMA based strategy is to find out what periods of historical data should be considered. This length of period might not guarantee us the optimal solution in different market conditions, but nevertheless it can be a strategy which competes under varying market conditions. Figure 2.2 gives us an overview of the representation that SMA short-term average (1-day) almost overlaps the market price of the stock. Whereas the
Figure 3.2: Flow Chart for Bidding Strategies
3. Trading Agent: Strategy Design and Analysis

long-term moving average removes the sharp-trend movements (smooths volatile series). So, when the SMA1 (calculation based on 1 day) is above the SMA15 (calculation based on 15 days history), signals that the market will have bullish trend, and can be used as a buy signal. Similarly, SMA1 crossing below the SMA15 will give an indication of bearish trend and will be used as a sell signal.

Figure 3.4: SMA 15 days upto SMA 120 Days Returns
Figure 3.3: SMA 15 days upto SMA 120 Days Indicator Mapping

figure
In this experiment we have five trading agents, all using the simple moving average indicator. SMA15, SMA30, SMA60 are considered to be short history trading agents meaning that they are considering the 15, 30 and 60 days historical data to compute the market trend. Whereas SMA 90 and SMA 120 are considered to be the long history based trading agents, evaluating their decision to buy or sell based on the 90 and 120 days of historical data. Keeping in mind that SMA 120 will not start performing completely until we have reached day 120, as the historical data becomes available to the trading agents their performance is updated, similarly rest of the agents will have to wait till their is a certain amount of historical data available. Other than the difference in long-term history of above mentioned five trading agents, rest of the strategy will be identical. Modifying a single parameter for these trading agent strategies will allow us to compare and judge performance of trading agents, that are trading in a similar environment. Moving on with the experiments we use simple moving average as a strategy indicator to buy and sell the stocks. As seen in figure 3.5 we have 5 traders based on strategies using popular SMA trading averages of 1-15, 1-30, 1-60, 1-90, 1-120. The goal in this experiment is to investigate the properties of simple moving average with respect to a stock trending in various directions. Figure 3.5 shows that transactions rate of SMA 1-15 is the highest of 140 with an inventory of zero stocks. Whereas, profitability wise, the results show that it does not make the most of the returns, which is 945.887. With a return rate of -5.411% it is the least profitable SMA strategy. SMA 1-120 is the most successful of the strategies suggesting that the lower number of transactions yielding a comparatively fractional increase in profit margin, with a value of
1122.713 (12.271% return rate).

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Buy Signals</th>
<th>Sell Signals</th>
<th>Inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMA 15</td>
<td>924</td>
<td>627</td>
<td>993.39</td>
</tr>
<tr>
<td>SMA 30</td>
<td>793</td>
<td>592</td>
<td>1004.864</td>
</tr>
<tr>
<td>SMA 60</td>
<td>647</td>
<td>402</td>
<td>1045.426</td>
</tr>
<tr>
<td>SMA 90</td>
<td>490</td>
<td>314</td>
<td>1067.546</td>
</tr>
<tr>
<td>SMA 120</td>
<td>501</td>
<td>270</td>
<td>1077.025</td>
</tr>
<tr>
<td>SMA 150</td>
<td>305</td>
<td>241</td>
<td>1146.799</td>
</tr>
<tr>
<td>SMA 180</td>
<td>241</td>
<td>221</td>
<td>1127.87</td>
</tr>
</tbody>
</table>

Table 3.1: SMA Based Strategy Productivity: This table provides useful information for number of signals (buy and sell) generated by each trading agent. This information can be further used to perform statistical analysis, whereas we can see that as the window of historical data increases, the signals tend to decrease. But on the other hand profits increase until a certain length of historical data.

3.3.2 Exponential Moving Average Based Strategies

We move on to a rather more interesting type of trading strategy design by utilizing the properties of exponential moving average, described in section 2.3.2. Our goal is to keep the trading environment similar for all the indicators. Whereas for the short and long-term periods of moving average we will use the following values: 15-26, 15-35, 15-60, 15-90, 15-120. First two of these periods are quite common in practice for day trading, while the other values are recommended for long term trading positions. The challenge with designing a trading agent strategy based on exponential moving average technical indicator is to understand
the use of smoothing factor and how this smoothing factor will give us appropriate information to conduct trading. In this way, exponential moving average removes short-term fluctuations and provides a market trend indication. Another reason to design strategy based on exponential moving average is that it will remove the lag in buy and sell signals by giving more weight to the recent price movements.

3.3.2.1 Crossover Strategy

In graph 3.6 we see the EMA as a crossover trading strategy design. For a crossover to occur, we are using two different Exponential Moving Averages. Generally, the lengths for these Moving Averages are short and long-term periods. A commonly used length of historical prices is 12, 26 intervals or periods. An interval may be in minutes, hours, days or some certain measure of time. Our graph is generated by representing 365 days, furthermore the days are sub-divided into 10 rounds per day. Another interpretation of crossover strategy design using the exponential moving average is to use the current price concept. If the current price is above the Exponential Moving Averages, you buy and then, liquidate that position when the current price crosses below either Moving Average. For a short-sell position, when the current price is below the Exponential Moving Average. Liquidate that position when the current price rises above the Exponential Moving Averages. So, based on the figure 2.3 we are able to conclude from the graph of short-term EMA and long-term EMA crossover gives us signals at various intervals as the market trends swing from being bullish or bearish. A clear movement of the EMA-12 line turns below the EMA-26 line. Clearly justifying the indicators calculation and usefulness in understanding market trends.
3. Trading Agent: Strategy Design and Analysis

3.3.2.2 Rainbow Strategy

Following the results given in figure 3.6, we observe that EMA 15-26 and EMA 15-35 seem to have done twice as many transactions compared to other competing EMA trading agents. Looking into the returns, the results for EMA 15-26 and EMA 15-35 are not so profitable because the inventory held by these two agents are least in number but they have incurred twice as many transaction costing them to decrease profits to a 1.434%. EMA 15-90 with the smoother long term average turns out to yield profits slightly better than the best of SMA trading agent, at a rate of 4.42%. The investment distribution for these EMA trading agents was kept similar to SMA, so total of the initial capital is divided in three equal parts (which is the number of stocks available to the trading agent for investment). Looking at figure 3.6 we can see the color of the averages from where the strategy got its name. Figure 3.7 is a graph of returns based on exponential moving average crossover strategy for 26, 35, 60, 90, 120, 150 and 180 days. It is clear that the returns are maximum when the EMA is calculated for 90 days and gradually starts to decrease as we move towards a longer period of historical pricing.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Buy Signals</th>
<th>Sell Signals</th>
<th>Inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMA 15, 26</td>
<td>667</td>
<td>350</td>
<td>1001.715</td>
</tr>
<tr>
<td>EMA 15, 35</td>
<td>419</td>
<td>289</td>
<td>1062.374</td>
</tr>
<tr>
<td>EMA 15, 60</td>
<td>441</td>
<td>240</td>
<td>1080.186</td>
</tr>
<tr>
<td>EMA 15, 90</td>
<td>278</td>
<td>179</td>
<td>1131.388</td>
</tr>
<tr>
<td>EMA 15, 120</td>
<td>237</td>
<td>200</td>
<td>1116.944</td>
</tr>
</tbody>
</table>
Figure 3.6: EMA 12-days to EMA-120 Days generated using JTAP
3. Trading Agent: Strategy Design and Analysis

Figure 3.7: EMA 12-days to EMA 180-Days Returns

Figure 3.8: EMA 12-days to EMA 120-Days
Table 3.2: EMA based Strategy Productivity Table. Buy and sell signals count and their respective profits show us that buy signals for EMA keep on decreasing as the length increases but sell signals are higher for 120-period EMA as compared to 90-period EMA, meaning that the signals can show irregular patterns based on the moment of underlying price.

3.3.3 Moving Average Convergence-Divergence Based Strategies

Moving Average Convergence and Divergence possesses an attribute to quickly indicate the price momentum for short term trading, various other properties of MACD have also been described in section 2.3.3. This is the reason what makes the MACD so popular.

3.3.3.1 Double-cross Strategy

Indications to buy or sell are generated by the signal line and MACD crossover, signal line below the MACD represents a bullish trend whereas vice-versa is the bearish trend. In addition to this, MACD indicator is calculated by taking the difference between a short-term moving average (12-day EMA) and a longer-term moving average (26-day EMA). So it is clear from this sort of calculation based design that the value of the MACD indicator must be equal to zero each time the two moving averages cross over. A cross through the zero line is a very simple indication that is used to identify the direction of the trend and the key points when in momentum. Combining the movement of the MACD indicator above or below the centre line with the crossover of MACD and the signal line we have a
trading strategy that will confirm the momentum of the stock.

Figure 3.9 represent the result for the experiments carried out for MACD based double crossover strategy in our stock exchange simulation. This graph helps us to understand the momentum of the market, based on the fact that, rise or fall in the stocks price will consistently keep on rising or falling. As from the experiment conducted for analysis of MACD, it is not fairly clear whether the MACD is crossing the signal line. Although it can be seen at some points along the y-axis. So to further strengthen the outcome of the experimentation, we plot a histogram (difference of MACD-Signal). MACD histograms movement above or below the zero line will indicate a bullish or bearish trend. A crossing of the MACD histogram line through zero happens when there is no difference between the short-term and long-term EMAs. A move from positive to negative is bearish and from negative to positive, bullish. Zero crossovers provide evidence of a change in the direction of a trend but less confirmation of its momentum then a signal line crossover. So looking at the graph we can evaluate that the histogram tends to remain over the zero line. Meaning a upward or bullish trend can be observed.
3.3.3.2 PPO vs MACD based Strategy

Experimental results given in figure 3.9 are particularly interesting and unique in nature, as the Percentage Price Oscillator and the Moving Average Convergence Divergence work in a similar fashion by generating a signal which is the exponential moving average of the value composition formed as result of difference between the short term and long term moving average. The difference is that PPO is represented as the percentage, moving the decimal two places [33]. The Percentage Price Oscillator (PPO) generates the same signals at MACD, but provides an added dimension as a percentage version of MACD. The PPO levels of the Share A can be compared against the PPO levels of ShareB because the PPO levels are a percentage representation. In addition, PPO levels in one security can be compared over extended periods of time, even if the price has doubled or...
triplled. This is not the case for MACD. Figure 3.9 give us the indication that supports the advantages of percentage price oscillator over longer periods of time as compared to the MACD. In this experiment we are using two different investment strategies in combination to two different bidding strategies. As described earlier one of the investment strategy is to divide the initial capital equally based on the number of stocks available. This reduces the risk of loss in-case where a stock might fall in unimagined scenario. Second investment strategy is where trading agent rank the different stocks available from the market. Each day the investment is moved around based on the stock previous closing performance, known as rank based investment strategy. Going back to figure 3.9, we see that MACDR9-26 and PPOR9-26 represents the MACD and PPO indicators implemented with the rank based trading strategy. There is a very clear difference between the returns gained with using the rank based investment strategy and the investment strategy where capital was fixed. MACD has a return of 3.26%, then PPO doing better at 4.11%. Comparing this to the gains of using rank based investment strategy we find that returns are much higher. MACDR has 12.74% return rate, increasing approximately 3 times. PPOR leading the profits with a return on investment of 19.39%.

Despite its advantages, PPO is still not the best oscillator to identify overbought or oversold conditions because movements are unlimited (in theory). Levels for RSI and the Stochastic Oscillator are limited and this makes them better suited to identify overbought and oversold levels.
3. Trading Agent: Strategy Design and Analysis

3.3.4 Relative Strength Index Based Strategies

Relative Strength Index (RSI) based strategies are helpful to determine the market momentum and strength of trend irrespective to the prices of the stocks that are under analysis, so it is extremely helpful to perform a comparative study of multiple stocks. We will discuss a 2-period Strategy which will describe the use of short-term history price periods based on RSI indicator.

5-period Strategy: In the classical trading, traders use a 14-day RSI, but with the help of our design and experimentation we see that there is no actual need to go back in statistical data that far. However, when you shorten the time frame you start seeing some very impressive results. Our research shows that the most robust and consistent results are obtained by using a 2-period RSI and we have built many successful trading systems that incorporate the 2-period RSI. We then quantified overbought and oversold conditions as measured by n-periods RSI reading being above 70 (overbought) and below 40 (oversold). In other words we looked at stocks with a n-period RSI reading above 60, 70, 80 and 90, which was considered overbought; and all stocks with a n-period RSI reading below 40, 30, 20 and 10, which was considered oversold. We then compared these results, here is what we found:

For Overbought:

The average returns of stocks with a n-period RSI reading below 30 and above 90 are as follows: 2-day (+0.08%), 5-days (+1.32%), 7-days (+1.86%), and 14-days later (+0.49%).

The average returns of stocks with a n-period RSI reading below 30 and above 80 are as follows: 2-day (+0.22%), 5-days (+0.38%), 7-days (+0.06%), and 14-
days later (-0.85%).

The average returns of stocks with a n-period RSI reading below 30 and above 70 are as follows: 2-days later (+0.27%), 5-days (-0.88%), 7-days (+0.66%) and 14-days later (-0.66%).

The average returns of stocks with a n-period RSI reading below 30 and above 60 are as follows: 2-day (+0.64%), 5-days (+0.49%), 7-days (+0.18%), and 14-days later (-0.48%).

When looking at these results, it is important to understand that the performance improved dramatically. Increase the value of RSI. The average returns of stocks with a n-period RSI reading above 90 were significantly higher than those stocks with a n-period RSI reading around 70.

For Oversold:

The average returns of stocks with a n-period RSI reading above 70 and below 40 are as follows: 2-days (+0.37%), 5-days (+1.32%), 7-days (+0.43%) and 14-days later (+1.02%).

The average returns of stocks with a n-period RSI reading above 70 and below 20 are as follows: 2-day (+0.54%), 5-days (-0.72%), 7-days (-0.12%), and 14-days later (-0.14%).

The average returns of stocks with a n-period RSI reading above 70 and below 10 are as follows: 2-day (+0.54%), 5-days (-0.72%), 7-days (-1.1%), and 14-days later (+0.15%).

When looking at these results, it is important to understand that the performance decreased as we decreased the value of RSI. The average returns of stocks with a 5-period RSI reading below 40 were much greater than those stocks with a 14-period RSI reading. This means traders should look to build strategies around
stocks with a 5-period RSI reading above 90 and below 40.

Figure 3.10: Relative Strength Index 10-70 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP

Figure 3.11: Relative Strength Index 20-70 (oversold-overbought) value, this is
3. Trading Agent: Strategy Design and Analysis

a graphical representation of the results generated for RSI during the simulation based on JTAP

![Graphical representation of RSI results](image)

Figure 3.12: Relative Strength Index 30-60 (oversold-overbought) value, this is a graphical representation of the results generated for RSI during the simulation based on JTAP
3. Trading Agent: Strategy Design and Analysis

3.3.5 Stochastic Oscillator Based Strategies

3.3.5.1 Varying Stochastic Oscillator Strategy

Stochastic Oscillator is a bound oscillator, which makes it easy to indicate overbought and oversold levels. No matter how fast a security advances or declines, the Stochastic Oscillator will always fluctuate within zero to one hundred in percentage. Traditional settings use 80 as the overbought threshold and 20 as the oversold threshold. These levels can be adjusted to suit analytical needs and stock characteristics. Hence to start the experimentation we adjust the value of n. Each day of trading is subdivided into 10 rounds which facilitates the market to match and clear the prices each round of the day. In this context we increment
3. Trading Agent: Strategy Design and Analysis

Figure 3.14: Stochastic Oscillator based on a 5, 7, 9, 12 and 14 days price

the value of n by 10 so that it represents a day of trading in the simulation. And as the theory suggests the most common settings for n should be 14 days meaning 140 rounds. Figure 3.15 is a representation of SO %D level 75 as over-bought threshold and 25 as oversold threshold. Readings above 75 for the $d_n$ day Stochastic Oscillator would indicate that the underlying stock is trading near the top of its $n$ day high-low range. Readings below 25 occur when a security is trading at the low end of its high-low range. Analysis of figure 3.15 shows that profits seem to rely heavily on the fact that what value of n is. Based on strategies tested in experiment 2, the suitable value for n is 5 days. Calculating the returns we see that SO50 gives us 15.75% return on investment. Increase in the number of days of statistical data may not be useful according to these results and trading agents end up loosing Investment. Trading agent SO14 works on 14 days stochastic oscillator, ending up with a loss of 2.91% compared to initial
investment.

3.3.5.2 Double Check Strategy

These pair of indicators work to form a strategy because the stochastic is comparing a stock’s closing price to its price range over a certain period of time, while the MACD is the formation of two moving averages diverging from and converging with each other. SO and MACD integrate to establish a bullish MACD crossover and a bullish stochastic crossover into a trend-confirmation strategy, when a faster moving average crosses up over a slower moving average, creating market momentum and suggesting further price trend to increases. When MACD is bullish, this will occur when the histogram value is above the equilibrium line, and also when the MACD line is of a greater value than the nine-day EMA, known as the signal line. The stochastic’s bullish divergence occurs when %K value passes the %D, confirming a likely price turnaround. This strategy gives traders an opportunity to hold out for a better entry point on upward trending stocks or to be surer that any downward trend is actually truly reversing itself when buying for long-term trading. probably the only drawback of this strategy is that the stock generally takes a longer time to line up in the best buying position, the actual trading of the stock occurs less frequently, mostly it is a good idea to apply this strategy to more than one stock.
3.4 Investment Strategies

Investment is an important part of a trading agent design where trading agent need to decide on where to invest? and how much to invest? There are so many stocks in an exchange and this is where the traders can get confused on which stock to invest their funds into. Similar to the human traders, a trading agent has to make a decision based on some statistical facts. There can be various ways to design the investment strategy. Investment strategy can be based on the market analysis or the trading agent can analyse their own returns from trading and then decide whether to invest in a certain stock. Further we will explore how different investment strategies for a trading agent can be designed.
3.4.1 Fixed Investment Strategy

The most simplest of the ways to invest is by dividing the funds in equal portions. This does not seem to be a smart choice but has its benefits associated according to the market conditions. Fixed Investment Strategy gives less exposure to the trading agent from risk downside in a situation where a certain stock may be effected by some external factor causing the stock to change direction instantly. So investing equally can save the trader for market downside risk. For this ex-
3. Trading Agent: Strategy Design and Analysis

The investment strategy is to divide the fund by the number of stocks available for trading.

3.4.1.1 Fixed Investment Strategy

![Fixed Investment Strategy](image)

Figure 3.17: Fixed Investment Strategy with three stocks and five trading agents

3.4.2 Rank Based Investment Strategy

Using an investment strategy which can perform some analysis is a smart option, but requires the trader to do the calculations in-order to find options. Whereas, an autonomous trading agent can have several investment strategies so that it can opt for one according to the market requirements. Rank based strategy is designed to analyse the stocks and rank them in-order such that the share which has been trending upward recently is on top and the stock with bearish trend or
least bullish to be ranked last. Ranking of the stocks is carried out by taking the difference of current days opening price and a closing price of day $n$ from history market prices. This gives us a value indicating the trend of the stocks and using the value of price change we rank stocks so that a stock with highest positive price movement gets more investment and the stock with least or negative price change gets their investment decreased. This increase in investment for top ranked share is equal to the amount of investment decreased from the lowest ranked stock.

Figure 3.18: Rank Based Investment Strategy with three stocks and five trading agents strategies

### 3.5 Summary

Trading agents decision making ability is an integral part of the strategy design. In this Chapter we described eight different trading agent strategies derived from five technical indicators. Other than that, good investment leads to increased
3. Trading Agent: Strategy Design and Analysis

profits. So, to add to the strategies, two types of investment portfolio strategies are designed and the analysis of the results shows that by dynamically adjusting your investment fruitful results can be achieved.

Let us go back to the bidding strategies in Section 3.3. Based on the classification of bidding strategies into trend based and momentum based, we can conclude that if they are used in combination then it will definitely yield better results in-terms of return stability.
Chapter 4

Implementation of Trading Strategies

4.1 Introduction

Trading agent simulations provide the opportunity to test complex trading strategies in a controlled environment. With a simulation system, we can analyse trading outcomes over time, refine trading strategies and track the results before practising the strategies in the real world or even comparing them with the real world strategies.

With increasingly sophisticated technologies of multi-agent systems, simulations of trading in financial markets with fully autonomous agents are not only possible but also play a key role in trading strategy design, analysis and implementation. This is because we can simulate years of trading data in a fraction of time and test our trading strategies repeatedly.

Details of simulation environment and trading agent strategies will be de-
scribed as we go through this chapter. First of all we will introduce JTAP configuration, so that, it has a standard configuration throughout the simulation. Than a number of assumptions will be made, these assumptions help to refine the outcome. Further, algorithms will be used to explain the assembly of strategies in general. These strategy algorithms will be categorised according to technical financial indicators.

4.2 Jackaroo Trading Agent Platform Configuration

Jackaroo Trading Agent Platform (JTAP) provides us a simulation environment for experimenting autonomous trading agent strategies in parallel to simulating multiple traders and markets. Details of JTAP have been described in 1.5.2. We have designed and implemented various trading agent strategies utilizing JTAP, design details are discussed in section 3.3.

Recalling that the structure of JTAP is client server, now we move onto configuration details of JTAP server. JTAP’s server maintains a clock which controls the duration of the simulation. For our experimentation the clock was configured to 365 days similar to representing a year of trading. Each day has 10 rounds and each round will last for 500 milliseconds.

The market side of JTAP simulates double auction mechanism of trading. Each market represents a single non-divisible asset, assets that are traded in this market are stocks. The pricing of these stocks are taken from the Australian Stock Exchange, considering the high and low range of price for each stock from the
period of 2009 to 2012. As far as the traders configurations are concerned, bidding strategies are tested using a single stock price movement, reason for this is to study the outcome of strategies in a non-sophisticated manner. Whereas, investment strategies are tested with three stock price movement with independent trends and price ranges, because investment strategies require the trader to move the funds around in order to analyse the true effect. As we are using real world financial techniques to simulate trading and therefore strategies are based on the financial indicators. So, for each simulation we configure a single type of strategy with variations in parameters, which is a useful tactic to optimise strategies based on the variables. Number of trading agents in a simulation can vary between 2 to 8, depending on a particular strategy.

4.3 Trading Agents

In order to have an accurate understanding and comparison among the trading agents with variations of trading strategies and to clearly recognise their effect on trading results, we made a number of assumptions to make sure the experimental environment is fair to each trading agent:

Assumption 4.1. Each trading agent has the same amount of budget available at the beginning of the trading simulation. Initial budget is fixed to ensure a controlled environment and to perform a relative study of trading agent strategies.

Assumption 4.2. The commodities (stocks) in the stock market simulation available to the trading agents are heterogeneous and non-dividable.

Assumption 4.3. The trading agents may have different prices for each shout
order on a single stock in the market, but the market price $P$ of that certain stock is same at a particular time $t$ for all trading agents, e.g. Trader $x$ places a bid order on Share $A$ at 101.3 at the same time Trader $y$ can place a shout order at his price which is 100.5, but the market price offered to the traders is same i.e. 100.9. Basically it depends on the market matching policy to decide on clearing price of these two shouts.

4.4 Strategies Implementation

Using simple algorithms we can devise criteria that can assist us to select strategies that are most suited to the market conditions, whereas research shows models of algorithms that have been developed to meet complex market conditions [34], [30]. Recently there has been research involving algorithm parameter tuning to enhance trading agents output [32]. These strategies or their parameters can be updated on the basis of every day as the JTAP simulation progresses. Furthermore the trading agents can be modified to update the strategies for every round of the game, but this will cause the traders to have inconsistent behaviour as of frequent switching between the strategies it is hard to analyse what sort of affect one particular strategy had on a certain round or day.

4.4.1 SMA Based Strategy Implementation

Each trading agent strategy has three checks to bid on a stock.

- First Check is to see if there is a indication to buy? If SMA long-term average is above SMA short-term average trading agent performs the next check.
4. Implementation Trading Strategies

- Second Check is to see if the price is on the increase. If so we proceed to the next step.

- Third Check is that if the trading agents have enough fund to buy the stock, or if they have invested all their funds.

Once all three checks are true, then only the trading agent will proceed to place a bid in the market. Here the price needs to be decided, as for this experiment the trading agent places market order, which means buying at the market price, rather than waiting for the price to change. With every bid the quantity is restricted to one, this is mainly because the market is not volume capped. Selling a stock has its own check, mostly the opposite of buying a stock.

- First Check is to see if the SMA long-term average is below the SMA short-term average.

- Second Check is to see if the current price is lower than previous days price.

- Third Check is based on the availability of the commodity, to see that whether the trading agent holds that certain stock.

An ask is placed based on the fact that these three checks have been met. The price to sell is the same as the current market price. Each ask has a quantity of one stock associated. Trading agent can keep on selling if it holds a certain stock.
4. Implementation Trading Strategies

and the first two checks are true.

**Algorithm 1:** Simple Moving Average Based Strategy

**Input:** \( n, \text{currentPrice}, \text{buyQuantity}, \text{sellQuantity} \)

**Output:** buyShout, sellShout

1. \( \text{shortSMA} = \frac{\text{currentPrice} + \text{previousPrice}_1}{2} \);
2. \( \text{longSMA} = \frac{\text{currentPrice} + \text{previousPrice}_1 + \text{previousPrice}_2 + \cdots + \text{previousPrice}_n}{n} \);
3. if \( \text{traderBalance} > \text{currentPrice} \times \text{buyQuantity} \) & \( \text{shortSMA} > \text{longSMA} \) & \( \text{currentPrice} > \text{previousPrice} \) then
   4. create anew Shout;
   5. Shout.setBid(true);
   6. Shout.price = \text{currentPrice} + \text{marginal};
   7. Shout.quantity = \text{buyQuantity};

8. else if \( \text{Inventory} > 0 \) & \( \text{shortSMA} < \text{longSMA} \) & \( \text{currentPrice} < \text{previousPrice} \) then
   9. create anew Shout;
   10. Shout.setBid(false);
   11. Shout.price = \text{currentPrice} - \text{marginal};
   12. Shout.quantity = \text{sellQuantity};

4.4.2 EMA Based Strategy Implementation

Exponential moving average is a weight based average and more weight is given to the recent statistical data. Implementation of EMA is not complicated but also not straightforward like simple moving average, because the statistical data needs to be given weight while moving forward with time. If we want to get a \( n \)-days EMA, we start of by calculating the smoothing function \( k = 2/n + 1 \) where
4. Implementation Trading Strategies

$n$ is the number of historical prices. After calculating the value of $k$, we will have a moving average which is exponentially weighed. This will allow us to have a more precise indication on the recent trend, at the same time providing some feedback on the past. We implemented a combination of six moving averages of lengths, varying from 15 to 180. These particular length moving averages have been selected based on popularity and research of traders. Using the moving averages in a combination, the trading agent strategy can start to analyse what is called a rainbow effect. The idea behind is that when the fast moving averages are above the slow moving averages then this indicates an upward (bullish) trend in the market. Whereas if the fast moving averages move below, turning upside down will indicate a downward trend. Trading agent performs a check on the combined movement, these moving averages and then makes a decision to buy or sell, moving on to the next step to decide the price of the shout and then the
Algorithm 2: Exponential Moving Average Rainbow Strategy

**Input:** history, currentPrice, buyQuantity, sellQuantity

**Output:** buy, sell

previousPrice = getPriceHistory(2) \( k = \frac{2.00}{n + 1} \);

if history > availableHistory then
    ema_{t-1} = SMA(availableHistory, history)
else
    history = 3.45 \times (availableHistory + 1);

for \( i \) == history do
    currentPrice+ = \((1 - k)^{(i-1)} \times getPriceHistory(i)\);
    getPriceHistory(i);
    ema_{t-1} = k \times currentPrice;

shortEMA = (currentPrice \times k)+previousPrice \times (1-k);
longEMA = (currentPrice \times k)+ema_{t-1} \times (1-k);

if traderBalance > currentPrice \times buyQuantity \& shortEMA >
    longEMA \& currentPrice > previousPrice then
    createNewShout;
    Shout.setBid(true);
    Shout.price = currentPrice + margin;
    Shout.quantity = buyQuantity;

else if Inventory > 0 \& shortEMA < longEMA \& currentPrice <
    previousPrice then
    createNewShout;
    Shout.setBid(false);
    Shout.price = currentPrice - margin;
    Shout.quantity = sellQuantity;
4. Implementation Trading Strategies

4.4.3 MACD Based Strategy Implementation

Moving Average Convergence Divergence Based strategy implementation is derived from exponential moving average. In theory a 26-day EMA is subtracted from the 12-day EMA, the resulting calculation is a MACD. The 26-day EMA can be replaced by \( n \). This is because multiple trading agents can be simulated with different parameter settings to compare the performance. Following pseudo-code is an implementation of MACD. There are two check conditions in this implementation. First check is to see that if the MACD line has crossed below the signal and the second check is to make sure that the signal is in the positive range, this confirms that trading agent can bid to buy. Otherwise if the MACD line is above the signal line and the signal is in positive range trading agent can sell the item.
4. Implementation Trading Strategies

Algorithm 3: Moving Average Convergence-Divergence Based Strategy

**Input:** \( n, \text{currentPrice}, \text{buyQuantity}, \text{sellQuantity} \)

**Output:** buy, sell

1. \( \text{double macd} = \text{ema}(12) - \text{ema}(n); \)
2. \( \text{signal} = (\text{macd} * 0.2) + \text{macdy} * (0.8); \)
3. \( \text{if macd} < \text{signal} \& \text{signal} < 0 \& \text{traderBalance} > \) \( \text{currentPrice} \times \text{buyQuantity} \& \text{currentPrice} > \text{previousPrice} \) then
   4. createanewShout;
   5. Shout.setBid(true);
   6. Shout.price = currentPrice + margin;
   7. Shout.quantity = buyQuantity;
8. else if \( \text{macd} > \text{signal} \& \text{signal} > 0 \& \text{Inventory} > 0 \& \) \( \text{currentPrice} < \text{previousPrice} \) then
   9. createanewShout;
   10. Shout.setBid(false);
   11. Shout.price = currentPrice – margin;
   12. Shout.quantity = sellQuantity;

4.4.4 RSI Pseudo-code

Implementation of RSI needs more precision while defining functions for calculations. RSI calculation is based in a similar fashion as moving average. This is mainly because of the fact that moving average for past \( n \)-periods needs to be calculated recursively. This recursive calculation of the moving average is done to give more weight to the recent stock price which allows a better prediction of the market trend. The average of gain and loss are calculated sep-
Algorithm 4: Percentage Price Oscillator Based Strategy

**Input:** \( n, \text{currentPrice}, \text{buyQuantity}, \text{sellQuantity} \)

**Output:** buy, sell, Quantity, Price

1. \( \text{ppo} = \left(\frac{\text{ema}(12) - \text{ema}(n)}{\text{ema}26}\right) \times 100; \)
2. \( \text{ppoYESTERDAY} = \left(\frac{\text{ema}12YESTERDAY - \text{ema}26YESTERDAY}{\text{ema}26YESTERDAY}\right) \times 100; \)
3. \( \text{signal} = (\text{ppo} \times 0.2) + \text{ppoYESTERDAY} \times (0.8); \)
4. **if** \( \text{ppo} < \text{signal} \) \& \( \text{signal} < 0 \) \& \( \text{traderBalance} > \text{currentPrice} \times \text{buyQuantity} \) \& \( \text{currentPrice} > \text{previousPrice} \) **then**
   
   a. create a new Shout;
   b. Shout.setBid(true);
   c. Shout.price = currentPrice + margin;
   d. Shout.quantity = buyQuantity;

5. **else if** \( \text{ppo} > \text{signal} \) \& \( \text{signal} > 0 \) \& \( \text{Inventory} > 0 \) \& \( \text{currentPrice} < \text{previousPrice} \) **then**
   
   a. create a new Shout;
   b. Shout.setBid(false);
   c. Shout.price = currentPrice − margin;
   d. Shout.quantity = sellQuantity;

arately and then put to a function so that the percentage can be determined.

RSI percentage representation oscillates between 0 to 100 where 80 is thought to be the over bought limit and 20 is the oversold limit. Whereas research shows that the value of RSI can be adjusted to market conditions. Algorithm 5 is a representation of the RSI based strategy. Where, RS is a calculation of average gain divided by average loss, which is then used to calculate the RSI.
4. Implementation Trading Strategies

**Algorithm 5:** Relative Strength Index Based Strategy

**Input:** \( n, \text{currentPrice}, \text{buyQuantity}, \text{sellQuantity} \)

**Output:** buy, sell, Quantity, Price

1. \( RS = \frac{\text{gain}(t)_{\text{avg}}}{\text{loss}(t)_{\text{avg}}}; \)
2. \( RSI = \frac{100}{1 + RS}; \)
3. if \( RSI > 65 \) & \( \text{traderBalance} > \text{currentPrice} \times \text{buyQuantity} \) then
   4. create anew Shout;
   5. Shout.setBid(true);
   6. Shout.price = currentPrice + margin;
   7. Shout.quantity = buyQuantity;
8. else if \( RSI < 35 \) & \( \text{Inventory} > 0 \) then
   9. create anew Shout;
   10. Shout.setBid(false);
   11. Shout.price = currentPrice − margin;
   12. Shout.quantity = sellQuantity;

4.4.5 SO Based Strategy Implementation

Strategies based on stochastic oscillator are particularly meaningful because they do not follow the price or the volume they speed or momentum of the price. Developed by George C. Lane in the late 1950s, the Stochastic Oscillator is a momentum based indicator that shows the high-low range over a \( n \) number of periods. Algorithm 6 is a fast SO, other strategy variations are slow SO and full SO. Where slow SO is the 3-period moving average of \( d \) and full SO uses the \( n \)-period moving average of \( d \). SO based strategies take the closing price in consideration for statistical analysis. This allows the indicator to calculate a value \( k \) which is the
percentage representing the momentum, as it can be seen from algorithm 6 that we have calculated three period moving average of $k$ in-order to compute $d$. There are several checks that can be performed on the value of $d$. These checks will indicate whether it is time to buy or sell. Following algorithm performs a check to see if the value of D is above 75, meaning that trading agent can place an ask shout to sell the stock. One other check is performed which tells the trading agent if it has the stock in inventory. Whereas if the stock is below 25, trading agent will trigger a bid shout to buy the stock, simultaneously checking to see if enough balance is available to complete the transaction. This strategy has been used with combination to other strategies in the experimentation, the results where promising.
4. Trading Strategy: Implementation

Algorithm 6: Stochastic Oscillator Based Strategy

**Input**: $n, \text{currentPrice}, \text{buyQuantity}, \text{sellQuantity}$

**Output**: buy, sell, Quantity, Price

1. $k = 100 \times \left(\frac{\text{marketClosingPrice} - \text{low}}{\text{high} - \text{low}}\right)$;
2. $k_1 = 100 \times \left(\frac{\text{marketClosingPrice} - \text{low}_1}{\text{high}_1 - \text{low}_1}\right)$;
3. $k_2 = 100 \times \left(\frac{\text{marketClosingPrice} - \text{low}_2}{\text{high}_2 - \text{low}_2}\right)$;
4. $d = (k + k_1 + k_2)/3$;
5. if $D > 25$ & traderBalance > currentPrice × buyQuantity then
   6. createNewShout;
   7. Shout.setBid(true);
   8. Shout.price = currentPrice + margin;
   9. Shout.quantity = buyQuantity;
10. else if $D < 75$ & Inventory > 0 then
    11. createNewShout;
    12. Shout.setBid(false);
    13. Shout.price = currentPrice − margin;
    14. Shout.quantity = sellQuantity;

4.5 Summary

In this chapter we gave implementation details and algorithms for trading agent strategies. Although some of the algorithms were a generalization of the many types of trading agent strategies discussed in 3.3, this was done to represent the flexibility in the structure of simulation implementation. Overall, it was observed that we can decide a common set of checks that serve as rules for trading agents.
strategies. These rules provide trading agent a standard criteria for implementing strategies in an autonomous simulation environment. If implementation is based on common standard checks, that means that these strategies and their outcomes can be compared for analysis (discussed in chapter 3). Mostly the multiple checks were introduced in detail in 4.4.1, later these served in other algorithms proceeding SMA based strategy. So in this chapter we had a look at the algorithms of trading agent strategies used in experimentation of this thesis. From these strategies details and algorithms we are able to summarize that financial technical indicators are the main component of the strategy design and this is where the decision information is generated from.
Chapter 5

Conclusion and Future Work

5.1 Conclusion

Manual floor trading seems to be a thing of the past, except in cases such as the New York Stock Exchange (NYSE), of course which is also a hybrid system of trading. As new technology is emerging, it needs to be integrated with human methodologies. Technical indicators have long been used in manual trading and recently been an important part of electronic and autonomous trading.

Following are the conclusions of this thesis: At first, we have introduced financial market models as perceived by the trading agents in an autonomous multi-agent environment. We have defined the mechanism that will serve as a core of agent-mediated markets, the policies that will govern the markets and models for pricing the assets.

Secondly, we describe financial technical indicators used by traders in real world financial markets for the purpose of financial analysis. These indicators are an integral part of real world trading and financial research. Strategies are
the most important aspect of the trading agent. The essence of trading agent strategy design is its decision making capability. So, strategies have been designed on technical indicators introduced earlier and keeping in view the market model. We represent strategy design by dividing it into two categories, bidding strategies and investment strategies. Bidding strategies and their design answers questions such as, what to buy and sell? what is the time to buy and sell? How much to buy and sell? The solution to these questions was answered by merging the technical indicators with computing. Investment strategies design and analysis are the key decision making component of trading agent to increase the profitability. Making decision on how to invest? Where to invest? How to move investment around markets?

Thirdly, trading simulation approach and its configurations have been considered in detail, so that the results generated from this environment can be understood correctly. Configurations of parameters for simulation platform JTAP demonstrates the way strategies can be optimised. With the help of these simulations, we where able to demonstrate that when momentum based bidding strategies are combined with trend following bidding strategies they yield better profits, plus the returns are stable under volatile market conditions. The results also prove that, rank based investment strategy was more profitable then the fixed investment strategy. The reason behind this is that trading agents are able to move around a certain proportion of investment on real-time basis leading to yield higher returns. Moreover this research investigates technical indicators and their properties in an autonomous multi-agent simulation.
5. Conclusion and Future Work

5.2 Future Work

1. Enhancement to the simulation environment, JTAP. JTAP is under constant improvement and has been updated as needs increase. Initially it was capable of simulating market side as a flexible testing bed. Our team enhanced its capabilities and now it is capable enough to simulate the traders strategies plus the markets. Nevertheless, there is room for improvement for introducing market volume as to cap the amount of stocks in the market. Further, bank as an institution can be introduced for loan services. Although most of the work has already been done to implement the bank.

2. With the increased capabilities and restructuring of JTAP, the testing of derivatives market would be an interesting feature. Another reason for this could be, the derivatives markets can be tested with the trading agent strategies which we have already developed as part of this research.

3. Application of the trading agent strategies to real world, rather than the simulations.

4. Trading Agent Competition is an important event, which not only encourages research in the field of agent based trading but also provides opportunity to research groups across the globe for interaction and innovation. Introducing Strategy design platform JTAP to TAC will provide a new competition environment to replace the JCAT platform (which was used for market simulation).

5. Model & Design charting indicators to enhance traders decisions making capabilities for strategy designing. Charting is another technique used by
financial experts to enhance their financial analysis capabilities. If charting is introduced to JTAP, trading agents will have an extended capability to understand the market movements.

6. Analysis based on news feed is an important factor reshaping the trends of a financial market. One way to do this is to add on a social media component that will allow the system to read news feed in some standard format and update itself accordingly.
Appendix A

In appendix A, we have listed Configurations for JTAP trading agent strategies and markets.

A.1 Trading Strategies Configuration

We configure the parameter file for the JTAP in the following way. This file is located in the “params/modules” folder, named “traders.params”. Each example is a generalized example of the trading strategies implemented in chapter 4. Firstly “jtap.agent.n” is the number of trading agents that will be part of the simulation. “size=1”, means that the number of copies of a trading agent, so if size is 2, it will be multiplied by 7 trading agent strategies, therefore 14 will be the total trading agents. Then the parameter “jtap.agent.0.id”, is the ID of the trading agent each trading agent has a unique identity described here, otherwise the system will generate an auto ID. “jtap.agent.0.investmentstrategy” is used to assign the investment strategy to be used and “jtap.agent.0.investmentstrategy.biddingstrategy” is the parameter for the bidding strategy.

Finally, “jtap.agent.0.investmentstrategy.biddingstrategy.windowsize” is the size
of the rounds for historical prices. This parameter can be varied to see the effect of trading strategies. The five trading strategy configurations utilise the following configurations.

A.1.1 SMA based strategy configuration

```
#############################################################
# trading agents

jtap.agent.n = 7
size = 1

jtap.agent.0 = jtap.TraderClient
jtap.agent.0.n = &size;
jtap.agent.0.id = SMA15
jtap.agent.0.investmentstrategy =
jtap.trader.strategy.SampleInvestmentStrategy
jtap.agent.0.investmentstrategy.biddingstrategy =
jtap.trader.strategy.SMABiddingStrategy
jtap.agent.0.investmentstrategy.biddingstrategy.windowsize = 15
```

A.1.2 EMA based strategy configuration

```
#############################################################
# trading agents
```
jtap.agent.n = 5
size = 1

jtap.agent.0 = jtap.TraderClient
jtap.agent.0.n = &size;
jtap.agent.0.id = EMA15 - 26
jtap.agent.0.investmentstrategy =
jtap.trader.strategy.SampleInvestmentStrategy
jtap.agent.0.investmentstrategy.biddingstrategy =
jtap.trader.strategy.EMABiddingStrategy
jtap.agent.0.investmentstrategy.biddingstrategy.windowsize = 26
...

A.1.3 MACD based strategy configuration

##########################################################################
# trading agents

jtap.agent.n = 4
size = 1

jtap.agent.0 = jtap.TraderClient
jtap.agent.0.n = &size;
jtap.agent.0.id = MACD12 - 26
A.1.4 RSI based strategy configuration

# trading agents

jtap.agent.n = 1
size = 1

jtap.agent.0 = jtap.TraderClient
jtap.agent.0.n = &size;
jtap.agent.0.id = RSI2
jtap.agent.0.investmentstrategy =
jtap.trader.strategy.RankInvestmentStrategy
jtap.agent.0.investmentstrategy.biddingstrategy =
jtap.trader.strategy.RSIBiddingStrategy
jtap.agent.0.investmentstrategy.biddingstrategy.windowsize = 20
...
A.1.5 SO based strategy configuration

# trading agents

jtap.agent.n = 5
size = 1

jtap.agent.0 = jtap.TraderClient
jtap.agent.0.n = &size;
jtap.agent.0.id = SO50
jtap.agent.0.investmentstrategy =
jtap.trader.strategy.RankInvestmentStrategy
jtap.agent.0.investmentstrategy.biddingstrategy =
jtap.trader.strategy.SOBiddingStrategy
jtap.agent.0.investmentstrategy.biddingstrategy.windowsize = 50
...

A.2 Market Side Settings

A.2.1 Double Sided Pricing Model

To use JTAP Double Sided Pricing Model, we have to set up the auctioneer and simulating policy in the “markets.params”, which is located in folder “params”. The parameters need to be used is shown below:

# trading agents
# specialists
...

tap.specialist.0.auctioneer = tap.market.JTAPDoubleAuctioneer
...

tap.specialist.0.auctioneer.simulating =
tap.market.simulating.DoubleSidedSimulatingPolicy
...

## A.2.2 Random Pricing Model

To use JTAP Double Sided Pricing Model, we have to set up the auctioneer and simulating policy in the “markets.params”, which is located in folder “params”. The parameters need to be used is shown below:

```
# specialists
...

tap.specialist.0.auctioneer = tap.market.JTAPRPDoubleAuctioneer
...

tap.specialist.0.auctioneer.simulating =
tap.market.simulating.RandomPriceSimulatingPolicy
...
```

## A.2.3 Market Clear Pricing Model

To use JTAP Double Sided Pricing Model, we have to set up the auctioneer and simulating policy in the “markets.params”, which is located in folder “params”.

92
The parameters need to be used is shown below:

```
# specialists
...
jtap.specialist.0.auctioneer = jtap.market.JTAMCPDoubleAuctioneer
...
jtap.specialist.0.auctioneer.simulating =
  jtap.market.simulating.MarketClearSimulatingPolicy
...
```
Bibliography


