Aleksander Bjerkøy and Mikael Kvalvær

Replicating Financial Markets using Reinforcement Learning

An Agent Based Approach

Master's thesis in MTIØT
Supervisor: Alexei Gaivoronski and Peter Molnar
June 2019

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Abstract

We present and implement an order-driven artificial stock market populated by learning agents. Agents adapt to their environment, mimicking the investor behaviour of real stock markets. To avoid making assumptions on complex investor behaviour, we model agents as neural networks. Agent learning is implemented using state-of-the-art reinforcement learning techniques. An advantage of this approach is that few exogenous parameters need to be calibrated. We demonstrate the feasibility of translating real investment strategies represented as high-level goals, to objective functions suitable for agents to learn. Tradable assets in the model are stocks in companies with an earnings process dependent on macroeconomic factors. The companies undertake income-generating projects partly financed by equity and debt. Each company has an associated credit rating, indicating the stability of the company. The credit rating governs credit spread above a floating interest rate as well as maximum allowed leverage. We show that the model produces the stylised facts of financial markets as well as time-varying and sector dependent trading volumes. Evidence is found for time-varying volatility with occasional clustering. We also find asymmetries in the return distribution, consistent with empirical distributions of real financial markets.

We propose a factor-dependency framework where macroeconomic time-series serve as drivers of the underlying economy. Input to the model can be any macroeconomic time-series. We run experiments where the macroeconomic time-series are the oil price, interest rate and GDP growth. The framework makes the model fit for scenario analysis. To illustrate the usefulness of the model's ability to perform scenario analysis, we replicate the financial crisis of 2008 for the Norwegian market. In our analysis, we lower the key deposit rate earlier than Norges Bank to investigate whether the crisis could have been avoided. We believe the possibility to conduct scenario analysis to be a useful tool for policy makers.

Sammendrag

Vi presenterer og implementerer et ordredrevet, kunstig aksjemarked med lærende agenter. Agenter som tilpasser seg miljøet etterligner investeringsoppførsel man kan observere i ekte aksjemarkeder. For å unngå å gjøre antakelser om komplisert investeringsadferd, modellerer vi agenter som nevrale nettverk. Læringsprosessen som finner sted hos agentene er implementert ved hjelp av avanserte, moderne teknikker for forsterket læring (*reinforcement learing*). En fordel med denne metoden er at man ikke behøver å kalibrere et stort antall eksogene parametere til empiriske datasett. Vi viser muligheten til å representere investeringsstrategier som høynivåspesifiserte mål. Agenter som forsøker å nå disse målene, oversetter målene til objektivfunksjoner. Investerbare instrumenter i modellen er selskaper med en tilhørende inntjeningsprosess som avhenger av makroøkonomiske faktorer. Selskapene påtar seg prosjekter som genererer inntjening. Prosjektene finansieres delvis med egenkapital og gjeld. Hvert selskap har en tilhørende kredittrating som indikerer selskapets økonomiske stabilitet. Kredittratingen bestemmer selskapets rentenivå beregnet utifra antall basispunkter over en flytende rente, i tillegg til maksimum belåning. Modellen fremskaper de stiliserte fakta for finansielle markeder (*the stylised facts of financial markets*). Vi presenterer bevis for tidsvarierende volatilitet med tidvis volatilitetsgruppering. Modellens avkastningsdistribusjon er asymmetrisk, hvilket er konsistent med empiriske distribusjoner fra ekte finansielle markeder.

Vi fremsetter et rammeverk der makroøkonomiske faktoravhengigheter fungerer som drivere for den underligende økonomien. Rammeverket gjør modellen egnet til å utføre analyser av ulike scenarioer. Inndata til modellen kan være hvilken som helst makroøkonomisk tidsserie. For å illustrere modellens brukbarhet replikerer vi finanskrisen fra 2008 for det norske markedet. I vårt forsøk endrer vi styringsrenten på et tidlgere tidspunkt enn hva Norges Bank gjorde i 2009 for å undersøke om krisen kunne vært avverget. Vi anser muligheten til å undersøke ulike scenarioer for å være et nyttig verktøy for offentlige beslutningstakere.

Preface

This dissertation is the final part of our Master Programme in Industrial Economics and Technology at NTNU. We reflect the mix between economics and technology in our thesis by having a financial focus combined with computer engineering. The thesis was written during the Spring of 2019 as a part in accordance with our specialisation in Investment, Finance and Management. It is a continuation of our project thesis written during the Fall of 2018.

We extend our sincere gratitude to our supervisors; Alexei Gaivoronski (NTNU) and Peter Molnar (University of Stavanger) for their inputs and discussions. Their insight has been extremely valuable to us during model design and academic writing. Your willingness to review and engage in our work has been an essential contribution to the final result. We appreciate your enthusiasm and involvement in our research. We also want to extend our appreciation to RagnaRock Geo, an up-and-coming Norwegian startup company, for letting us use their computer resources in our experiments.

Trondheim, June 9th, 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Economic policies are influenced by empirical observations and theoretical models. Often, these models are based on assumptions on human behaviour and the presence of certain relationships in the economy. What they lack, however, is the ability to provide information about different scenarios: How will the stock market react to changes in the key policy rate? How will an oil price crash impact different business sectors? How sensitive is the stock market to a continuous decline in the gross domestic product? Researchers have attempted to create artificial stock markets that assess the impact of large macroeconomic changes. Unfortunately, they must often resort to model complex human behaviour as a simple set of rules and assumptions (LeBaron 2002; Bjerkøy and Kvalvær 2018) in order to study these issues. There are two main challenges: Firstly, performing large simulations require considerable amounts of computing power. This limits the scale and level of details on the experiments that can be conducted. Secondly, a large number of exogenous parameters must be defined prior to the simulation. One such example is defining the human behaviour. What are the investor's overall goals? How do they decide which stocks to buy and which to sell? What happens when the market regime changes?

Earlier attempts of creating artificial markets have modelled human behaviour as agents with static sets of rules. Lo (2004) argues for an evolutionary approach to human behaviour in his works on the adaptive market hypothesis. Briefly, the adaptive market hypothesis is the idea that individuals form strategies through trial and error. Dominant trading strategies consequently change over time. Rules are not static; when the environment changes, strategies evolve. To incorporate this idea in artificial stock markets it is necessary to have agents that adjust their trading strategies based on market conditions and other agents' behaviour.

Our model takes inspiration from the adaptive market hypothesis and use reinforcement learning to model agent behaviour. Reinforcement learning is well-suited for artificial stock markets as agents learn from past experiences. They learn from trades that benefited them economically and forego strategies which resulted in losses. Through thousands of simulations the agents can learn to maximise their own reward function[1]. The reward function they seek to maximise might be to attain the highest possible portfolio return while keeping a reasonable sector exposure and low portfolio volatility. Our model differs from the existing literature as agents develop heuristics instead of utilising a set of predefined trading strategies. An advantage of this approach is the opportunity to easily specify high-level goals for the agents simultaneously as keeping the number of exogenous parameters limited.

Research Goals & Contributions

Bjerkøy and Kvalvær (2018) developed an order-driven artificial stock market populated by heterogeneous agents; Asset prices were driven exclusively by aggregated demand and supply. Although the model successfully demonstrated several stylised facts of financial markets, agent strategies were a fixed set of rules. We extend the model by introducing learning agents and reducing the number of exogenous parameters. Learning agents in an order-driven environment are in line with suggestions offered by LeBaron (2006) and Lo (2004). Their strategy are formed from experience and not by an external model designer. By reducing the number of exogenous parameters, the model dynamics are derived endogenously. Consequently, parameters cannot be used to "tune" the model into fitting desired outcomes such as the stylised facts of financial markets[2]. When aggregated behaviour is derived endogenously by specifying high-level goals, researchers cannot design agents which induce market failures. Instead, financial bubbles and crashes occur from the complex interaction among agents and changes in the macroeconomic environment. An extended model of Bjerkøy and Kvalvær (2018) lead to increased model realism and insight on dynamics caused by changes in the macroeconomic environment. Based on the aforementioned discussion we design a model which:

[1] The reward function of an agent is discussed in section 3.4. The reward function corresponds loosely to an investor's utility function as defined by Tversky and Kahneman (1974).

[2] Fitting of parameters to match the stylised facts has been an issue of earlier models (LeBaron 2006)

(i) **Captures Stylised Facts**
A valid model must foremost capture the stylised facts of financial markets. The stylised facts we consider are listed in appendix B. Any artificial stock market must exhibit the same statistical features as the ones found in real markets. Demonstration of the stylised facts is necessary to create realistic simulations.

(ii) **Framework for Scenario Analysis**
Upon achieving goal (i), the model can be used to explain dynamics observed in real financial markets. Exogenous parameters, such as the interest rate, can be time-varying to investigate the effect of different interest rate policies on the overall macroeconomic conditions. Inputs can be calibrated to data from real markets.

We build heavily on the framework of Bjerkøy and Kvalvær (2018) and incorporate changes that were suggested as concluding remarks. These changes generally increase the realism of the model. To achieve the goals, the following contributions are made:

(i) **Reinforcement Learning Agents**
Intelligent agents able to learn from experience using reinforcement learning and neural networks are introduced in the model. Whereas Routledge in LeBaron (2006) uses a genetic algorithm to model learning, we leverage on advancements in the fields of artificial intelligence. Agents learn using state-of-the-art reinforcement learning optimisation algorithms. These agents develop both long-term and short-term strategies, with a mix of fundamental[3] and momentum[4] approaches to asset valuation. Mixing of strategies was identified as one of the key steps towards increased realism in Bjerkøy and Kvalvær (2018). As Turrell (2016) notes, artificial intelligence reduces the set of assumptions researchers have to make on behaviour. Agents are learning and adapting as the system evolves and can respond more realistically to changes in policy and the macroeconomic environment. According to Marchesi et al. (2003), the employment of neural network techniques and learning agents are key research fields for contemporary agent based modelling.

(ii) **Factor Dependency Framework for Policy Analysis**
A factor dependency framework inspired by Fama and French (1993) is implemented for modelling company earnings. The factors are drivers of the overall economy. From the real world, one knows that oil companies are highly influenced bythe oil price, indebted companies on the interest rate and retail companies on the growth in gross domestic product. These economic factors are key performance indices which function as drivers of the global economy. The framework offers to model companies depending on several of these factors. One can use these dependencies to study transitions between market regimes or sector-specific bubbles as proposed by LeBaron (2011a).

(iii) **Investment Philosophies Framework**
The model offers opportunities to specify high-level goals for agents[5]. By defining goals we avoid static rules for investing. An investment philosophy is a translation from real world investment goals to a reward function in reinforcement learning. Agents observe their environment continuously. The mapping from observation of the environment to interaction with the environment is processed by a neural network[6]. Conceptually, the neural network mimics the cognitive process of real investors. An advantage of this contribution is the opportunity to intuitively model different risk preferences. This makes it possible to state high-level investment philosophies inspired from real world strategies.

(iv) **Company Debt and Credit Profiles**
Debt is vital to financial markets. Real-world companies use different levels, and types, of debt to finance their business. Different levels of leverage impact company profitability as well as stability (Myers 1984). Modelling debt supply and implications of debt on the macroeconomic environment is crucial to produce a realistic context. As companies have different access to debt for various reasons, we take a practical approach to model leverage and credit spread using what we coin *debt profiles*. In essence, each company has an associated debt profile governing the maximum level of leverage and credit spread, based on how stable the company is. Debt profiles draws on inspiration from credit ratings in real financial markets as well as insight on maximum leverage policies offered by Jarrow (2013). Different capital structures for companies is a novel contribution in the research field of artificial stock markets. In addition to simplicity, modelling companies' access to debt in this way yields an elegant entrance for the interest rate to impact the financial markets as suggested by Bjerkøy and Kvalvær (2018).

We depart from existing research on single-asset markets and extend the multi-asset model of Bjerkøy and Kvalvær (2018). Trading can occur during any period of the day as opposed to a single trading period per day. This makes the price process completely order-driven based on agent decisions, compared to the approach by Farmer and Joshi (2001). Consequently, the model depends less on exogenously defined parameters, encouraged by LeBaron (2002).

[3] Fundamental value is the total value of a company's assets in perpetuity (Koller, Goedhart, Wessels, et al. 2010).

[4] Momentum, or trend, investors believe in correlations in returns.

[5] For example, we specify a series of different investment philosophies that are increasingly risk-averse. Another investment philosophy we design is to find stocks that have a highest possible dividend yield.

[6] Neural networks are used because they are models that are capable of learning complex relationships when properly implemented. We motivate the use of neural networks in chapters 2 and 4.

We achieve this property by building on the model of Rutkauskas and Ramanauskas (2009) employing reinforcement learning. Learning agents takes inspiration from the adaptive market hypothesis of Lo (2004). The agents will employ the Proximal Policy Optimisation algorithm of Schulman, Wolski, et al. (2017b) for learning. A novelty of our model is the inspiration drawn from Fischer and Riedler (2014) on modelling different capital structures. State-of-the-art reinforcement learning techniques makes the learning process versatile and realistic. Combined, the design of the model is founded in previous research all the while making more realistic assumptions on capital markets and agent behaviour. We show these alterations are able to capture time-varying and sector dependent trading volume in addition to the stylised facts of financial markets. Ultimately, the model is a close-to-reality environment supporting the possibility for assessing different types of economic policies. To the authors' knowledge, a framework for scenario analysis using artificial stock markets is not present in existing research.

In conclusion, these contributions aim to explain observed behaviour by making more realistic assumptions than previous research. Fewer assumptions are made on agent behaviour using learning agents. Agents able to sense the environment, adapt and evolve are an integral part of Lo (2004)'s theory on adaptive markets. The price development is fully driven by agents forming their own beliefs in a stock market with multiple sectors and assets. Founded in LeBaron (2006)'s criticism of the limitations of single-asset systems, we believe the contributions offered by this thesis capture trading dynamics of real financial markets and stylised facts to a higher degree. Novelties are the combination of learning agents and a realistic factor-dependency framework for companies' business cycles and leverage levels. The fitness of the proposed model is evaluated according to how well it demonstrates the stylised facts of financial markets as described in appendix B.

The thesis is structured as follows: Chapter 2 reviews relevant previous and contemporary research. Based on the experiences and results of these models, chapter 3 describes the model implementation. Chapters 4 and 5 describe the simulation procedure and model parameters, respectively. In chapter 6, we present and discuss the results obtained from various model-runs. We conclude the thesis by reconstructing the financial crisis of 2008 and calibrate the model to real financial time-series from the Norwegian market. To demonstrate the power of the model, we conduct scenario analysis where the central bank of Norway responds more quickly to the global crisis and lowers the key policy rate earlier.

Chapter 2

Review of Earlier Models

The research on financial agent based modelling has transitioned from models with static rules and simple dynamics to adaptive, order-driven environments with learning agents. The section starts with an introduction to agent based models and its building blocks. Later, the section covers some of the founding contributions to the area as well as recent developments. As a concluding remark, we describe advancements made in neighbouring research areas relating to reinforcement learning. Experiences and techniques developed in these fields are extensively used in development of agent learning and cognition. In what follows, we discuss several relevant models from existing literature along with their main contributions and shortcomings. The review also highlights what our proposed model have in common with previously developed models and where it diverges from traditional research.

2.1 A Brief Introduction to Financial Agent Based Modelling

This section will give a quick introduction to the basics of agent based modelling (ABM). Agent based modelling attempts to simulate the actions and interaction between different agents in an environment. The assumption of homogeneity and rationality of investors is relaxed by allowing different trading strategies, investment philosophies, information sets, endowments and goals. This opens up for complex and more interesting interactions in the financial markets which could help explaining certain characteristics of empirical data, such as volatility clustering, fat tails and changing market regimes (see appendix B).

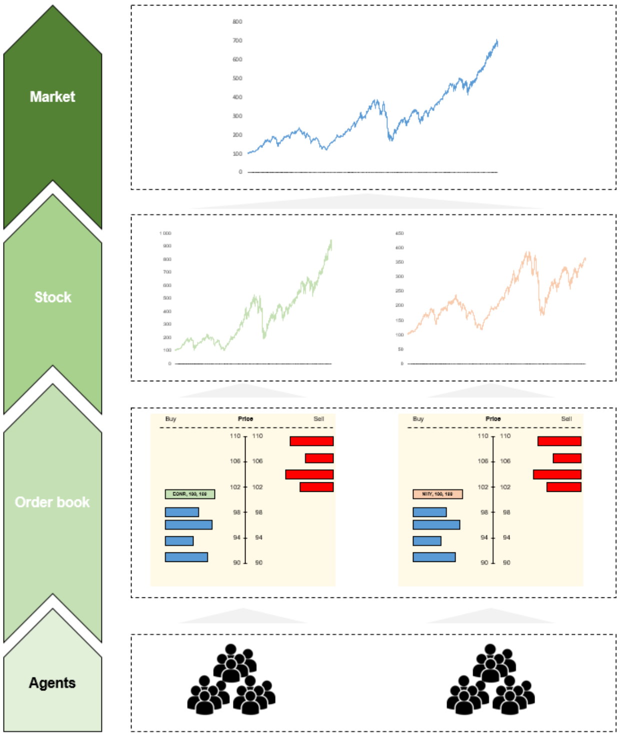


Figure 2.1: Structural overview of an agent based model. Agents take small and independent actions on the agent level. These actions can be be buy, sell or hold. Actions are materialised as orders at orderbook level. Fluctuation in demand and supply causes price fluctuations on the stock level. These aggregated fluctuations cause changes in the market sentiment over time. An aggregated picture of the economy, i.e. the aggregated status of the stock market, can be studied at the market level.

Applications of ABM in financial economics gained momentum with the early works of LeBaron[1] and the Santa Fe Artificial Stock Market model (LeBaron 2002). The model is one of the most famous attempts to construct an artificial financial market with heterogeneously learning agents[2]. Together with a team of psychologists, economists, physicists and computer scientists, LeBaron wanted to address important and controversial questions not answered by contemporary financial research. The artificial stock market was created in a bottom-up manner where trading strategies and interactions were defined for each agent. Trading strategies developed over time, such that a predominant strategy could vanish as time passed by[3]. The rule set, i.e., the predefined strategies which could be utilised, were however kept constant. Agents could choose between a mix of the predefined rules, synthesising their own strategy (LeBaron 2002). Asset price dynamics displayed by the model were shown to reflect several characteristics found in empirical data for real financial markets (LeBaron 2002).

ABM of financial markets showed steadily increase in popularity in the later part of the 20th century. Computational models were fueled by increasingly more powerful machines. Cristelli (2013) provides a thorough comparison of different models developed over time[4]. Starting from the onset of the 21st century, little to no research has been carried out on the ABM scene. Due to the very computational nature of ABM, more powerful machines, better simulation tools and more flexible programming languages are inarguably favourable for the discipline. As research interest declined, this lack of focus created a gap in contemporary research. The tools desired by pioneers became available, but the field was left unstudied. Pioneers in computational ABM expressed desire for more powerful computers in order to avoid making simplifications on dynamics (LeBaron 2011a). Simplifications was often in the form of extracting one property that was to be simulated, while the rest of the dynamics were derived from theoretical frameworks[5]. Progress in artificial intelligence has once again spiked the interest in financial agent based modelling. Previously, agent behaviour was at best adapting to the current market conditions by choosing from a predefined rule set. Only a few models allowed the agents to *develop* their own strategies as time went by. A such predefined rule set is a major simplification limiting the realism of the artificial market. Models where agent cognition is changed from a static rule set to a neural network open up for fewer simplifications and assumptions on behaviour. We aim to utilise advancements made in the fields of artificial intelligence to create an artificial stock market populated by heterogeneously learning agents.

Agent based models have recently found new applications in other disciplines such as social sciences (Chakraborti et al. 2009). Progress in these areas has sparked the interest for a modern approach to agent based modelling of financial markets. Researchers with a more financial focus can leverage extensively on the achievements made in other research fields. Additionally, access to high-resolution real financial data are now readily available to researchers. Combined with easy access to relatively powerful computers, agent based models with an empirical focus are feasible. With the progress of ABM in other research fields combined with better computation models, we believe the scene is set for a newfound interest of agent based financial models.

2.2 Traditional Artificial Stock Markets with Static Rules

Numerous approaches to creating artificial markets have emerged over time. The general trend has been going from simplistic models attempt to capture some aspects of real financial markets without caring too much about the plausibility of the underlying processes, to elaborate models aiming to capture complex aspects of financial markets. The research focus has also shifted slightly towards creating more realistic agent behaviour. Any agent based model is critically dependent on valid agent behaviour. Previous models have also relied heavily on static rules sets.

Perhaps one of the first models introduced that can be seen as an agent based model for financial markets is Garman (1976). In his model, investors place orders at random according to a Poisson distribution with the goal to describe and capture dynamics of price fluctuations and volume fluctuations of a risky stock. The goal is not to provide a realistic description of agent behaviour. Rather, the author focuses on: "[...] the aggregate market behaviour and shall adopt the attitude of the physicists who cares not whether his individual particles possess rationality" (Garman 1976, p. 258). A contribution of Garman (1976)'s model is that it departs from traditional market equilibrium models. Letting agents trade at random introduces the possibility for temporal disequilibrium; a situation where there is a large imbalance between supply and demand with corresponding large share price fluctuations. Criticisms of Garman (1976)'s model is mainly concerned with agent implementation. The agents trade completely at random, which is not a very realistic assumption[6]. Furthermore, an order submitted by an agent was not bounded by any budget constraint; hence an agent could, in principle, still trade while bankrupt.

Another notable model is the works of Chiarella (1992) which introduces agents with heterogeneous trading strategies.

[1] LeBaron is one of the pioneers of the Santa Fe artificial stock market. The model continues to be the main work of reference for a large part of research on agent based models. His contributions will be described in details later.

[2] We will use financial agent-based modelling and artificial stock markets as interchangeable terms in this thesis.

[3] In the event of development of new dominant strategies.

[4] For other comparisons of different models see LeBaron (2006) and Chakraborti et al. (2009).

[5] One such example might be Farmer and Joshi (2001) where beliefs on the asset value are simulated, but the price update is calculated using a theoretical approach.

[6] For an empirical review on how investors trade, see for example Fama (1970).

Two strategies are presented; fundamentalists[7] and chartists[8]. Excess demand is used to calculate the asset price. The model partition excess demand as the sum of the aggregate chartist demand and the aggregate fundamentalist demand of the stock. As Chiarella (1992) noted, the aggregate demand from the fundamentalists was more stable[9] whereas the chartist aggregate demand varied to a much greater extent. Consequently, Chiarella (1992) was able to capture much of the same dynamics as Garman (1976) did, with the addition that agents made informed decisions instead of trading according to a probability distribution. However, as the author noted, a better model of the asset market was needed. In particular, only one risky asset was traded and the investment universe of real financial markets consists of a significantly larger amount of risky assets. Additionally, the model relied on several exogenous parameters that could be adjusted by the researcher. Although the model proved fruitful, a large set of exogenous parameters left the door open for tuning the model to achieving the desired results. A great deal of subsequent research developed the approach of having heterogeneous trading strategies. Our model is inspired by Chiarella (1992) in the sense that there exist several trading strategies, some of which focus on short term trends and others that emphasise the fundamental value of a given asset. We also extend the investment universe to multiple assets.

LeBaron (2002), and more recently LeBaron (2006), extended the works of Chiarella (1992) and other earlier models. The key contribution of LeBaron (2006) was the problem of validation. A shared concern by nearly all agent based models is that they are difficult to validate; numerous exogenous parameters can be tweaked by the researcher. Moreover, the internal mechanisms can be set to fit sets of stylised facts (i.e., alter the exogenous parameters until the model output matches the stylised facts of financial markets). To ameliorate these issues, the researcher should aim to replicate difficult empirical observations by putting exogenous parameters under evolutionary control[10] (LeBaron 2006, p.32). The model provides a trading environment consisting of a risk-free asset with fixed return and a risky asset paying stochastic dividends according to an autoregressive process. All agents have a *set* of strategies, and the different strategies are evaluated according to how well they perform in the market. Better performance makes it more likely for the agent to use that particular strategy later. Furthermore, the strategy set is evolving over time, which makes LeBaron (2006)'s model less dependent on exogenous parameters[11] compared to earlier models. The idea of genetic trading strategies is indeed interesting, and our model takes inspiration from LeBaron (2006)'s model and Lo (2004). Agents are able to sense, learn and adapt to their environment. However, they have not a predefined set of strategies available. They learn from interactions with their environment and develop heuristics[12]. Strategies which proved successfully earlier will be repeated. We will depart from genetic trading strategies and incorporate learning using state-of-the-art reinforcement learning techniques.

2.3 Towards Learning and Adaptive Agents

The transition in artificial stock markets towards models with learning agents is becoming increasingly present. LeBaron (2011b) describes the notion of learning as a form of passive or active learning. Passive learning occurs when strategies survive through wealth accumulation as in LeBaron (2002). Active learning, on the other hand, is when agents switch actively between strategies. Strategies can be developed through exploration or formed from synthesis of existing strategies. LeBaron (2011b) argues for a form of mix: real financial markets consist of agents utilising both passive and active learning. Adapting agents, i.e., agents which learn actively, are coherent with the hypothesis of adaptive markets by Lo (2004). Agents which responds dynamically to new information represents a real trait of financial markets but modelling this type of agent is not trivial. Problems include the construction of objective functions, time-horizons and information set. Our model incorporates active learning: Agents adapt and evolve utilising heuristics formed from past observations and combine different trading strategies. The information set is the whole economy; we let the agents observe an unprocessed state. Objective functions are parameterised as different reward function[13].

According to Tesfatsion and Judd (2006), economists have long been unable to model feedback effects between the macro- and microenvironment quantitatively. Local interactions between subsets of agents give rise to macroeconomic regularities as market sentiments, strategies and innovations, in turn influencing the microeconomic conditions. Heavy importance has been placed on fixed decision rules, common information[14], representable agents and equilibrium

[7] A fundamentalist bases its trading decisions on the fundamental value of the company.

[8] A chartist agent is an agent that uses momentum as indicator, i.e., return in prices. Other implementations often name this agent trend agent.

[9] The demand at time t, D_t, was given by $D_t = a(W_t - P_t)$ where a was a positive constant, P_t the log-price of the risky asset at time t and $W(t)$ the log-price that clears a Walrasian auction.

[10] That is, agent and system parameters may change during a simulation; they are not fixed. Consequently, the parameters will evolve during the simulation and cannot be set by the researcher to fit the stylised facts.

[11] To see this, notice that in early works most of the parameters related to trading was set by the researcher. Here, good strategies will have a high probability of "survival", whereas strategies that perform poorly will have a low probability of "survival". This approach to changing trading strategies is very similar to the idea by Lo (2004) about adaptive markets.

[12] A heuristic is a strategy which fits empirical observations.

[13] Reward functions are defined in section 3.4.7. Conceptually, one can view these functions as a mapping between actions and goals for a learning agent.

[14] Schulenburg and Ross (1999) argue for a form of information segregation. Agents do not have access to the same sets of information. They view real markets as a diverse scene populated by agents with different cognitive capabilities and information supply. In real markets, information is distributed in an asynchronous fashion with propagational delays.

constraints. Tesfatsion and Judd (2006) motivate the advent, and arrival, of models with inductive learning - agents co-evolve with the markets and learn from their environment. The economy should evolve over time from an initial state without the modeller interacting. Tesfatsion and Judd (2006) consequently reiterate LeBaron (2006)'s call for removal of exogenous variables.

A model focusing on the removal of exogenous parameters is Tay and Linn (2001). They extend the model of LeBaron (2002) with inductive learning agents. Rather than a defined and complex rule-set as in LeBaron (2002), agents compress and assimilate information into what Tay and Linn (2001) refer to as fuzzy logic. Fuzzy logic is some condensed form of reduced information, comprehensible for humans. The inductive reasoning for agents take form as a stepwise process. Firstly, the agents reduce the set of information to hypothesises of plausible alternatives based on experiences. The hypothesises are tested on how well they fit the observations. Observations include movements in market prices and hypothesises include changes in dividend policy. The best hypothesis is chosen as a strategy. When the outcome becomes known, the strategy is evaluated. Unreliable hypotheses, i.e., failed strategies, are removed from the pool of possible strategies in subsequent periods. Removed strategies are replaced by modified alternatives. Individuals will consequently adapt and learn from the constantly evolving market. They show that the compressing of agent rule set produce excess kurtosis, similar to data from real financial markets. The authors argue that inductive reasoning and bounded rationality give rise to dynamics more close to the descriptive statistics observed in common stock markets than earlier attempts.

Problems with their model relates partly the asset universe. Only two assets are offered; one stock and one risk free bond. We take inspiration from Tay and Linn (2001) and create a model with few exogenous parameters as well as imposing no limitations on trading strategies. Learning forms from experiences. However, we do not remove strategies which proved to be bad. Agents receive a punishment in the form of a negative reward, encouraging them to avoid the strategy later. Although fuzzy-logic is analytically more tractable, we do not incorporate it in our model as it breaches with the model-free assumptions of reinforcement learning. Information should be presented uncondensed to the network.

Rutkauskas and Ramanauskas (2009) introduce the usage of reinforcement learning in agent based modelling of artificial stock markets. They utilise a reinforcement learning approach called Q-learning developed by Watkins (1989). As previously noted, Tesfatsion and Judd (2006) and LeBaron (2006) have suggested to model all agents in an artificial stock market as learning entities to increase realism. Rutkauskas and Ramanauskas (2009) follow suit and model agent cognition using reinforcement learning. An advantage of reinforcement learning is the explicit model-free nature[15]; it is not necessary to have an explicit model of the environment. The authors show the model to produce dynamics similar to real financial markets. They demonstrate that strategies developed by the learning agents exhibit a balance of economic rationale and optimisation techniques. A problem with the model relates partly to the emphasis on dividends. As discussed in Bjerkøy and Kvalvær (2018), there may be other fundamental drivers than only dividends. The fundamental value of a stock is too complex to be formed realistically using solely a forecast on the dividend process. Furthermore, there is only one stock in the asset universe and agents can only hold zero or one share at any time. We take inspiration from the works by Rutkauskas and Ramanauskas (2009) and pursue creating artificial markets populated by reinforcement learning agents. However, we consider Q-learning to be unsuited for models of financial markets. A central assumption of Q-learning is stationarity in the model environment. Stock markets are in general not stationary, which makes direct usage of Q-learning unfit for the problem. Agents in our model use a different form of reinforcement learning called policy gradients more fit for stochastic environments.

2.4 Pursuit for Realistic Agent and Environment Modelling

The trading scene in the model of Bjerkøy and Kvalvær (2018) is populated by heterogeneous agents with bounded rationality in an order-driven environment[16] inspired from Marchesi et al. (2003) and LeBaron (2002). The model demonstrate several of the stylised facts of financial markets, although the leverage effect is lacking. Their model also demonstrated the ability to generate time-varying trading volume, causing sector specific bubbles. This property is a step towards one of the goals of contemporary ABM suggested by LeBaron (2006). The authors suggested to extend the model with learning agents to limit exogenous parameters, following in the spirit of LeBaron (2006). In our model, time-varying interest rate is incorporated. We model inter alia the interest rate as a macroeconomic factor which company earnings and debt payments depend on. The factor-dependency framework make the model a viable tool for policy analysis[17].

Modelling of debt is a somewhat novel study in the ABM field (Fischer and Riedler 2014). Of the few examples identified by the authors, debt is often found on the agent side in the form of leveraged positions. Fischer and Riedler (2014) develop a model populated by heterogeneous agents with bounded rationality able to lever their position in a risky asset. Their goal is not to construct an artificial market, but to demonstrate the credit leverage cycle found in

[15] There also exist reinforcement learning methods that is model based. We do not discuss them here.

[16] Compared to e.g. Farmer and Joshi (2001) using a central market maker and a price-update function.

[17] By policy analysis we mean tweaking one or more of the model parameters and simulating how the parameter change affect the environment.

real financial markets. We draw on inspiration from their work and model debt as a financing source for the companies in our model. The companies can use debt to partially finance projects which in turn generate income for the company. Consequently, assets can have different capital structures. According to Koller, Goedhart, Wessels, et al. (2010), the choice of leverage and capital structure is an important part of a company's strategy. Debt increases the liquidity in the markets and can result in higher return on equity, but can destabilise markets (Fischer and Riedler 2014). Geanakoplos (2010) shows that changes in leverage can lead to fluctuations in asset prices. Cont (2001) observes a similar empirical finding.

2.5 State-of-the-Art Reinforcement Learning Methods

Artificial intelligence has led to advancements in neighbouring fields. Researchers have been able to develop models which systematically outperforms humans. Silver et al. (2018) of the DeepMind project AlphaZero[18] demonstrate superhuman performance in games such as Go and chess. The model has no domain knowledge and starts learning the game knowing only the rules. AlphaZero utilises reinforcement learning methods and is able to achieve similar results in other games such as Shogi without altering the design. This suggests that reinforcement learning can be appropriate in many different problems. Disruptions caused by artificial intelligence have also occurred in the medical and healthcare field. According to Cruz and Wishart (2006), artificial intelligence plays an increasingly important role in early discovery and diagnostic of cancer. Zhao et al. (2011) have developed an reinforcement learning algorithm which selects patients for first- and second-line treatments in clinical trials. Furthermore, machine learning is an important part of individualised treatments (Cruz and Wishart 2006). The common property by the achievements in these neighbouring fields is performance which beats humans both in terms of speed and precision. Our model takes inspiration from advancements in these fields and apply the insight gained from the results on artificial stock markets.

Moody and Saffell (2001) propose an algorithm based on policy gradients (PG) which they coin recurrent reinforcement learning (RRL). The goal is to adjust the trading policies to maximise some performance metric. In practice, the performance metric of Moody and Saffell (2001) is a path-dependent utility function taking either wealth or the Sharpe ratio as input. The trading system is applied on empirical data from the S&P 500 index as well as the US/GBP currency exchange rate. RRL outperforms standard trading techniques such as buy-and-hold in addition to a Q-learning trader. Moody and Saffell (2001)'s work is interesting from an ABM point of view due to its performance and ability to discover hidden structures in real data. The policy gradient method approach is able to develop strategies fit to different states. This adds support for the suitability of policy gradient methods in stock selection problems compared to Q-learning. As the goal of this thesis is to build a realistic environment which can be used for economic policy analysis, a learning model showing sound results when tested on empirical data is key.

We draw on inspiration from the works of Moody and Saffell (2001) and employ policy gradient methods for agent learning and cognition. Schulman, Wolski, et al. (2017b) propose a new gradient method for reinforcement learning called Proximal Policy Optimisation. The algorithm is designed for simplicity and is easy to implement. Empirical tests show encouraging results compared to other state-of-the-art reinforcement learning algorithms on a wide set of different problems. Combined with the natural suitability of policy gradients methods, the strong features of Proximal Policy Optimisation makes it an appealing candidate for for the learning process of the agents in our model.

[18] DeepMind is a British AI company owned by Google.

Chapter 3

Model Description

The model proposed in this thesis differs substantially from previous implementation of artificial stock markets such as LeBaron (2002) and Farmer and Joshi (2001). Rather than agents with a static set of rules, we implement learning agents using reinforcement learning[1]. The chapter is structured as follows: Firstly, the financial modelling of companies and capital structure are discussed. This section covers the types of assets available in the model. We proceed to discuss the implementation of the financial statement of the companies. A description of how profit is generated from projects is provided. Projects with factor and sector dependency is a vital part of contribution (ii) on a framework for policy analysis. The discussion naturally proceeds to implementation of debt in the model. Debt is implemented using what we coin *debt profiles*; an important part of contribution (iv) about different levels of leverage and associated credit spreads. Next, order placement and pricing is discussed in section 3.2. We make no changes to the original orderbook implementation as of Bjerkøy and Kvalvær (2018). Finally, section 3.4 describes reinforcement learning and specify how the agents learn from experience. The section covers how agents sense the environment, how they form opinions and how they decide to perform actions on the environment.

Before describing the various parts of the model, we make a point of the complexity of the artificial market. To create an artificial stock market a large amount of small parts needs to be defined. These parts range from specifics on how orders are submitted to the exchange and eventually transacted, to implementation of agents submitting the orders. Consequently, there are a lot of "moving parts" that need to work together to form a representable and realistic market. The drawback of this complexity is that the model description can appear a bit fragmented. However, all of the described parts are important to the model as a whole.

3.1 Financial Modelling and Capital Structure

This section describes the financial modelling and capital structures in the model. First, the different types of assets in the model is described. Next, capital structure and income generating projects are presented. Projects generating income and capital structure are important to create a realistic financial market. Next, we argue for a distinction between equity (attributable to shareholders) and debt (issued from a bank). Debt financing allows the companies to pursue a larger number of projects and generate higher returns than would be possible with equity alone. The cash-flows generated by the investments relate partly to the overall economy, driven by macroeconomic factors.

3.1.1 Assets

As mentioned in section 2, a major disadvantage in earlier attempts of constructing artificial stock markets has been the limitations on number of assets. Notable works such as LeBaron (2002) and Farmer and Joshi (2001) limit agent trading to one risky asset only. To create a more realistic investment universe, agents are allowed to trade several risky stocks. The motivation for creating a universe with several assets available is primarily to observe dynamics between assets of different sectors (LeBaron 2006). According to LeBaron (2006), single asset markets puts extreme restrictions on the amount of trading volume that can be generated in the simulated market. The author motivates the statement by referring to the technology bubble of 1990. Extreme events, such as bubbles, are very often sector dependent. Modelling multi-asset markets with different structural parameters is therefore important to create realistic models. Additionally, having multiple risky assets makes it possible to create agents that optimise a *portfolio* of risky assets. In this way, different assets with different return characteristics are available to the agents. These modifications are important parts of creating a realistic trading environment for learning agents.

[1] Reinforcement learning is properly defined in chapter 4. Briefly, agents observe the state of an environment. They perform actions on it (such as buying shares on a specific asset) and get a reward for it (such as the daily profit and loss of the agent's portfolio). The agent choosees its actions in a way that they maximise expected value of their reward function. The agents learn by repeating this process very many times and updating its parameters.

Stocks

An arbitrary number of stocks can be made available to the agents. The stocks are meant to exhibit certain characteristics that resemble real stocks. A universe of assets can be used to study dynamics under certain market environments. Companies undertake projects, each with a correlation to the macroeconomic environment. Section 3.1.4 discuss the importance of this feature. Furthermore, each asset belongs to a sector. Sectors are characterised by similar exposure to the macroeconomic environment.

Cash & Bank Deposit

In Bjerkøy and Kvalvær (2018), agents could invest in a risk-free bond. As the bond was redeemable at any time agents did not have any incentive to hold cash. We remove the bond in the model as it is conceptually similar to a bank deposit. On the other hand, we introduce interest rate on bank deposits. In this way, a bank deposit is conceptually similar to the bond position in Bjerkøy and Kvalvær (2018).

Interests on cash positions are accrued daily. Formally, the cash position X_{iC}^{t+1} for agent i on day $t+1$:

$$X_{iC}^{t+1} = X_{iC}^{t}(1 + i_d^t) \tag{3.1.1}$$

where i_d^t is the daily floating interest rate.

As stocks become relatively more expensive, bank deposits become a more attractive investment[2]. This will dampen diverging asset prices caused by the fact that bank deposits yield interest payments. Additionally, deposit rates can be used as a tool to investigate changing macroeconomic environments. Dynamics related to the pricing of assets and term structure of interest rates can be studied.

3.1.2 The Financial Statement of a Company

The financial statement gives an indicative overview of the financial health of the company. Normally, in an income statement, one starts with revenues and works towards the earnings per share. Modelling all of the different posts in an income statement is a tedious task and not in line with the goals of this thesis. Additionally, as there exists different kind of sectors in the model, modelling sector-specific posts would have been necessary. Firstly, we describe the overall structure of the financial statements in the model. Secondly, an in-depth discussion of the model implementation for each element is carried out.

We take a practical approach to model the capital inflow and outflow for the companies. Each company has an EBIT[3]-process. One loses the possibility to model different depreciation and amortisation policies, which are important aspects in capital intensive businesses. The argument for avoiding to model depreciation and amortisation is that it adds unnecessary complexity to the model and are often highly sector-dependent. The EBIT process is described in section 3.1.4.

Interest expenses are tax deductible[4]. Consequently, they enter the income statement before taxes. Different countries have different rules for interests deduction. The parameter τ_d governs how much of the interests expenses that are subject to deduction. Taxes are calculated as a fixed percentage of the taxable income. All companies are subject to taxation. The parameter τ describes the marginal tax rate.

A table overview of the income statement can be found in table 3.1. An important thing to realise from table 3.1 is that only two processes need to be modelled: the EBIT process and the interests expenses. The EBIT process is modelled closely to the earnings process in Bjerkøy and Kvalvær (2018), although with a slight modification to incorporate investment opportunities. Consequently, we are left with just one new element to model. As one of the goals of this thesis is to perform scenario analysis, it is convenient that interests paid affect directly the profitability of the companies through tax deduction.

[2] Relatively more expensive stocks relates, for example, to the concept of growing P/E multiples in this context. An interpretation of the P/E multiple is the dollar amount an agent can invest in order to receive a dollar of the company's earnings. Higher multiples translate to larger investments for the same share of the earnings.

[3] EBIT is an abbreviation for *earnings before interests and taxes.*

[4] Commonly, a percentage of the interests are deductible. The exact amount vary from country to country.

Financial Post	Model Implementation	Parameter
EBIT	Realisation of set of projects following an AR-process with factor dependency	n.a.
Interest expenses	Floating interest	i_d
EBT	EBIT Less Interest expenses	n.a.
Taxes	Fixed corporate tax	τ, τ_{i_c}
EAT	EBT Taxes	n.a.

Table 3.1: Structural overview

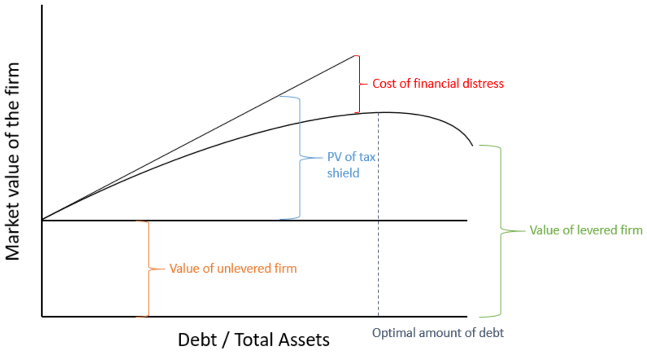


Figure 3.1: Myers (1984) describes this level as the point where one maximises the total value of the firm subject to debt. At a certain point, increased debt causes higher probability of default. Higher probability of default makes debt investments more risky, raising the required rate of return and affecting the total value of the company negatively.

3.1.3 Capital Structure

Capital structure affects the profits and stability of companies. In a world without taxes, the Modigliani-Miller Theorem states that the capital structure of a firm is irrelevant to company value (Modigliani and Miller 1958). As the companies in the model are subject to taxation, there exist benefits of having debt. Different capital structures are affected differently by changes in the interest rate and tax policies. As one of the goals of this thesis is to provide a framework for (ii) scenario analysis, debt needs to be taken into account. Debt is also a natural part of the modern economy (Koller, Goedhart, Wessels, et al. 2010), so a realistic model should in some way model debt.

According to Myers (1984) and the *Static Trade-off Hypothesis*, there exists an optimal level of leverage for a company. Usually, the level is determined as by a trade-off between costs and benefits of borrowing; namely the benefits acquired from tax shield and the harm from bankruptcy costs. Interest tax shield, as described in section 3.1.2, is a reduction in income taxes originating from deductible interests. Lower taxes reflect higher company valuation, as less taxes are paid indirectly by the owners. In a world without bankruptcy costs and other disadvantages associated to debt, a firm should be close to all-debt financed (Berk and DeMarzo 2007). However, as there indeed are costs related to debt, firms should substitute debt for equity until the value of the firm is maximised. The costs related to debt originates from increased probability of default. If a company default, debt holders lose their investments. Increased probability of default makes debt more risky, raising the required rate of return and affecting the total value of the company negatively (Myers 1984). Figure 3.1 indicates the trade-off between leverage and market value of the firm.

With Myers (1984)'s insight on optimal level of leverage in mind, we postulate that modelling debt for companies is a challenging task. At least three elements demand discussion and justification:

(i) Debt capacity of the firm.

(ii) Interest rate on the debt.

(iii) Supply of debt.

In reality, elements (i)-(ii) are determined by individual agreements between companies and financial institutions. (iii) Supply of debt, is taken care of by both private and public institutions. Banks issue debt and private individuals and entities purchase corporate bonds. As the goal of this thesis is not to model debt, we take a practical approach and introduce debt profiles. The role of debt profiles is discussed in section 3.1.5. Supply of debt is provided solely by a bank with unlimited lending capacity. Each firm have a debt capacity defined by their debt profile. Companies are allowed to borrow money up to their maximum leverage. In the event of default, a defaulting company has no effect on the bank; the bank cannot go bankrupt. The sole function of the bank in the model is to provide external financing for companies.

3.1.4 Equity

Equity is the value of the assets attributable to the owners. Companies have an earnings process which should cover debt payments, used to pursue investment opportunities and to be distributed as dividends to the equity owners. In essence, the stock price should reflect this value. This section describes how equity is generated; that is, how profit-generating investments are modelled.

EBIT

Modelling company earnings is a complex task. Although there exist sound frameworks for stock prices and dividend processes[5] (McDonald, Cassano, and Fahlenbrach 2006), a solid foundation for earnings is lacking. Griffin (1977)

[5] Examples includes the random walk process, geometric Brownian motion and the Ornstein-Uhlenbeck mean-reversion process.

presents a framework for the time-series behaviour of quarterly earnings. He finds that a first order ARIMA[6] model captures most of the dynamics of an individual earnings process for companies on the New York Stock Exchange (Griffin 1977). A limitation with this framework is that it lacks the possibility for factor dependency. That said, the model of the earnings itself and the absolute values of the process are irrelevant to the artificial market we develop. Hence, the implementation of the earnings process is not an issue, merely a design choice one can address in specific implementations. The focus of the earnings process is the dynamics between individual stocks, sectors and market. We take inspiration from the Griffin (1977) model and extend it with factor dependency inspired by Fama and French (1993) and Guerard Jr, Markowitz, and Xu (2015). Fama and French (1993) is a stock return model, whereas Guerard Jr, Markowitz, and Xu (2015) is a portfolio optimisation model for returns. The models have in common that returns for individual assets are dependent on several factors. We model the EBIT-process as a sum of cash flows generated by different projects undertaken by the company.

Modelling company earnings as an EBIT-process with the framework we propose have the following advantages:

(i) One avoids making assumptions on income, related costs of doing business and what kind of margin the business operates with.

(ii) No assumptions are made on the different types of depreciation and amortisation policies. These policies are important when it comes to capital structure.

(iii) It captures that earnings are variable[7]. There may exist different types of market regimes; some periods of high earnings, some periods of low earnings.

(iv) Earnings are connected to the sector and overall market. Commonly, company earnings are connected to how well the economy, and especially the sector, is performing.

Projects and Investments

Investments provide opportunities for the companies. They are projects that the companies can undertake in order to realise an uncertain profit in the future. Consequently, they will have impact on the company's EBIT in the future[8]. Investments are sector specific, meaning that the investments have different characteristics between each sector. Only companies in the same sector can bid for a sector-specific investment. The investments are dependent on different factors and consequently an integral part of contribution (ii) about a factor framework for scenario analysis.

Formally, a project I_p is specified by the following attributes:

(i) s, the sector

(ii) C_p, the upfront, fixed investment cost.

(iii) $\{T_1, \dots, T_T\}$, set of time steps when profit is realised.

(iv) α_r autoregressive term relating to previous earnings of earnings return process.

(v) α_f term relating to factor dependency of earnings return process.

(vi) $[w_1 \dots w_F]$ for $f \in F$; exposures to each of the F macroeconomic factors.

(vii) μ_0 first period expected return.

(viii) σ_p^2 idiosyncratic variance in project earnings return on timestep t

(ix) t_0 timestep where the project is undertaken.

Denote the earnings return for a project I_p on time $t_0 + t$ as $R^p_{t_0+t}$. The projects have an earnings return process which follows

$$R^p_{t_0+t} = \alpha_r R^p_{t_0+t-1} + \alpha_f \beta_{t_0+t} + \epsilon_{t_0+t} \tag{3.1.2}$$

β_t denotes the weighted sum of factor dependency. Formally,

$$\beta_{t_0+t} = \sum_{k=1}^{F} w_k f_{k,t_0+t} \tag{3.1.3}$$

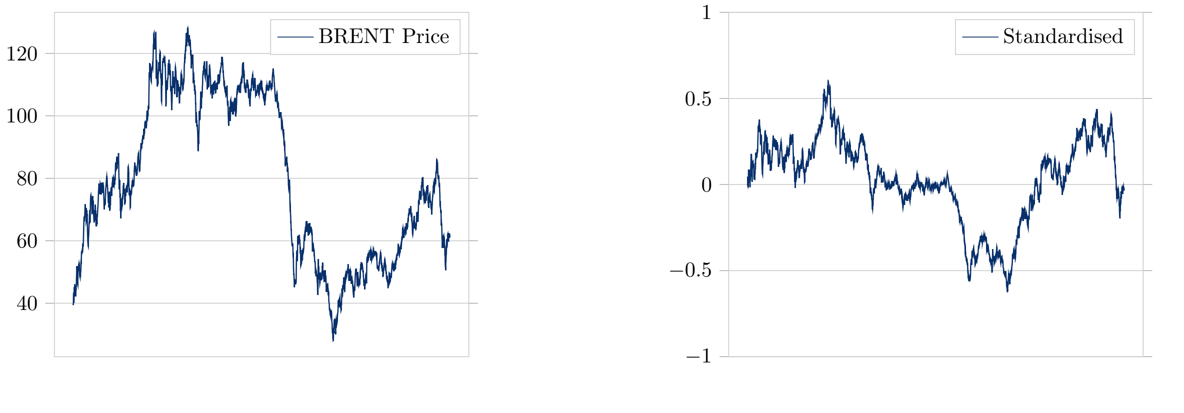
where F is the number of factors and f_{k,t_0+t} is the current value for factor k on day $t_0 + t$. ϵ_{t_0+t} in equation (3.1.2) denote the idiosyncratic part of the earnings process:

$$\epsilon_{t_0+t} \sim \mathcal{N}(0, \sigma_p^2) \tag{3.1.4}$$

[6] An ARIMA model is an autoregressive integrated moving average for non-stationary time series, meaning that evolving variable is regressed on its own lags.

[7] For a discussion on the topic of variable earnings, we refer to the McKinsey study *Who's afraid of variable earnings* of 2002 for a discussion. Available from: https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/whos-afraid-of-variable-earnings.

[8] We model income at an EBIT-level for practical reasons.



(a) BRENT index price in US dollars for 2001-2018. (b) Standardised index with $\theta = 2$ years rolling average.

Figure 3.2: Illustration of standardisation of factors.

The term $\alpha_f \beta_{t_0+t}$ in equation 3.1.2 is the project's dependency on the macroeconomic factors. α_f is a scalar value and β_{t_0+t} is a weighted sum of factor exposures, defined in equation (3.1.3). Project revenues are affected by a set of factors, constituting contribution (ii) regarding a factor-dependency framework. These factors are key performance indicators (KPI) which convey information about the overall state of the economy. From real financial markets, KPIs such as GDP growth, the oil price and the key policy interest rate are examples of such factors. w_k indicates how exposed the project is to factor f_k. The term α_f denote how sensitive the project I_p is to the factors on aggregate.

A special case for $R^p_{t_0+t}$ occurs for $t_0 + t = T_1$ (i.e. the first cash flow received from project I_p):

$$R^p_{T_1} = \mu_0 + \epsilon_1 \tag{3.1.5}$$

with ϵ as defined in equation 3.1.4. The motivation is that earnings are variable and uncertain; the first period return is no exception. μ_0 is just an estimate of the first period earnings return. The project disappears after the last earning cash flow on day T_T. To keep the model as simple as possible, we introduce no costs relating to termination of projects and we don't allow projects to be terminated early.

The exact modelling of investments is outside the scope of this thesis. The focus is on modelling companies in an environment influenced by macroeconomic factors. The exact implementation of investments does not matter as nominal numbers on value and income do not have any interpretation; only the interdependency between the projects do. Consequently, the factor model we propose provides an easy-to-implement and elegant way of achieving the desired macroeconomic dependency and is an integral part of contribution (ii).

The cash flow (CF) generated at timestep $t_0 + t$ is given by:

$$CF^p_{t_0+t} = C \times R^p_{t_0+t} \tag{3.1.6}$$

Macroeconomic Factors

We implement a factor dependency framework motivated in contribution (ii). Each factor $f_k, k \in \{1, \ldots, F\}$ is an indicator for the general state of a specific part of the economy. Factors can vary in range and have different units[9] In order to avoid interpreting the value the current observation, factors are standardised to

$$f^p_k = \frac{f^{p*}_k}{\bar{f}^p_k} - 1 \quad \text{for k} \in F \tag{3.1.7}$$

f^{p*}_k is the current observation of the indicator at time t in nominal units and $\bar{f}^p_k$ is the average over the period $[t - \theta_f, t]$. The value θ_f is some period one averages the observation over[10]. Conceptually, one can view the standardised factors as how much the current observation deviates from the long-term average. An interesting property of modelling factors this way is that one can input any time-series to the model. The standardisation of the oil price index BRENT for 2001-2018 is shown in figure 3.2a.

For completeness, the total cash flow for a company j on time t is

$$CF_{t,j} = \sum_{p=1}^{P} CF_{t,p} \tag{3.1.8}$$

where P is the number of projects undertaken by the company.

[9] For example, the oil price is commonly quoted in US dollars while the interest rate is in percent.
[10] In our model implementation, this parameter is set to 180 days.

Arrival of Projects

The arrival of projects per sector draws on inspiration from queuing theory[11]. Investments arrive as opportunities for all of the firms in the same sector according to a Poisson process. The Poisson process is chosen as it is memoryless and the arrival of events are proportional to the length of the time interval (Cox 2017). Poisson process arrivals are used to model everything from natural disasters and wars to customers at a service centre. The common property is independent events which one assumes arrive at a certain rate over a time interval, making it suitable for our model.

Formally, the probability of k events occurring in a time interval λ is

$$P(k) = \exp(-\lambda)\frac{\lambda^k}{k!} \tag{3.1.9}$$

where λ is the average number of events per interval. Each eligible firm is given the opportunity to participate in an auction for the project. Denote B_{pj} as a bid on project I_p by firm j. The winner of the auction is awarded the investment and pays the bid B_{pj}. The winner is chosen according to:

$$j = \underset{j}{\operatorname{argmax}} \quad B_{pj} \tag{3.1.10}$$

if and only if the following conditions hold

$$B_{pj} \geq C_p \tag{3.1.11}$$

$$B_{pj} \leq C_j \tag{3.1.12}$$

where C_j is the cash reserve (including maximum debt level) for company j.

The NPV of the investment is given by

$$\mathrm{NPV}^p = \sum_{t=0}^{T_t} PV(CF_t) = \mathrm{disc}\left(\sum_{t=0}^{T_t} CF_t\right) \tag{3.1.13}$$

where CF_t as in equation 3.1.4

The expected present value of the project cashflow at timestep $t+s$ is computed as follows

$$\mathbb{E}[PV(CF_{t+s})] = \mathbb{E}[\exp((-(t+s))r_j)CR^p_{t+s}] = \exp((-t+s)r_j)C\mathbb{E}[R_{t+s}] \tag{3.1.14}$$

where r_j is the company's discount factor. The expectation of the return is

$$\mathbb{E}[R_{t+s}] = \alpha_1^s R^p_{t+1} + \alpha_2 \sum_{i=0}^{t+s} \alpha_1^i \mathbb{E}[\beta_t] \tag{3.1.15}$$

The expectation of β_t is

$$\mathbb{E}[\beta_t] = \mathbb{E}\left[\sum_{i=1}^{F} w_i f_{it}\right] = \sum_{i=1}^{F} w_i \mathbb{E}\left[\frac{f_k^{p*} - f_k^p}{f_k^p}\right] = \sum_{i=1}^{F} w_i \left[\frac{f_k^p - f_k^p}{f_k^p}\right] = 0 \tag{3.1.16}$$

Lastly, the expectation of R_t is

$$\mathbb{E}[R_t] = \mathbb{E}[\mu_0 + \epsilon_t] = \mu_0 \tag{3.1.17}$$

We can now collect the expectations and proceed to write equation 3.1.4 as

$$\mathbb{E}[PV(CF_{t+s})] = \exp((-(t+s))r_j)C\alpha_1^s\mu_0 \tag{3.1.18}$$

Consequently, the NPV at time t is

$$\mathrm{NPV}^p_t = -C_p + C_p\mu_0 \sum_{i=1}^{s} \exp(-r_j i)\alpha_1^i \tag{3.1.19}$$

Each firm is willing to bid up to the discounted expected return, i.e., where the NPV is zero. The discount factor is individual as companies have different expertise, different equipment, ease of implementation and levels of experience. Consequently, projects appear with different risk characteristics for each company. The interpretation of the discount rate can therefore be a comparative advantage.

Modelling investments in this fashion yields some interesting properties:

[11] Queuing theory is a branch of operation research modelling the arrival, processing and departure of events.

(i) Firms with large cash reserves will, on average, win a larger portion of the auctions. Consequently, they will on average invest in more projects. Ensuring a flow of projects is vital to maintain the status as market leader and build up the cash reserves.

(ii) Market leaders, that is, firms which win several projects, will on average receive the expected return Γ_{pj}.

(iii) As projects have a probability of failing, the rise and fall of market leaders can be observed.

(iv) The process is stochastic. Notice that investments can have different variances and means depending on the project. Large variance indicates large fluctuations in earnings, in which case the company might be considered more risky. A positive mean indicates that the company seems to generate larger returns than what would otherwise be expected from the industry and overall market.

(v) Standardised factors render a versatile framework. The model makes no interpretation of the factor; only whether the current observation is high or low historically. Time-series of different factors can be fed to the model.

(vi) The factors are drivers of the overall economy. Investments are dependent on the current market climate, and consequently the profit flow for each company.

Company Factor Exposure

Each company has a net exposure to the macroeconomic factors. Denote by F_{jkt} the total exposure to factor k by asset j on day t:

$$F_{jkt} = \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} w_{pk} \tag{3.1.20}$$

where w_{pk} is project p's dependency on factor k. The set $\mathcal{P}_i$ is the set of project undertaken by agent i. The significance of equation (3.1.20) is that the exposure of company j to factor k is an average of the projects that company j has undertaken.

Initialisation

Initially, each company is assigned a first project. This project signifies the $t = 0$ base income. Initial projects are used to differentiate companies and their starting points. Formally, the following parameters are defined for each company:

$$CF_{t=0,j} = CF_{0,j} \tag{3.1.21}$$

as well as $\mu_{0,j}$ as in equation 3.1.5. μ_0 in this context is the last return of the company j before initialisation (from real return series).

Goal (ii) is to build a framework for scenario analysis. We aim to investigate scenarios based on real events. Different base incomes for different companies can be used to model the current state of the economy. We stress that the model is not suited for assessing different trajectories for individual companies. The focus is to model different scenarios and derived dynamics between assets and factors.

Bidding Process on Projects

This section covers how bids are calculated, increased, submitted and chosen as winners. Recall that projects arrive according to a Poisson process. The arrival of one project is consequently independent on any earlier arrived projects. Projects are available in a "now-or-never"-fashion: either at least one company submits bids, or the project vanishes. Companies participate in an auction until a winner is chosen.

The bidding process starts by an auctioneer notifying eligible companies (i.e., companies within a given sector) about an investment opportunity. The companies gain information about the parameters described in section 3.1.4. Each company proceed to calculate their individual NPV which serves as an upper limit for their bid. The range that a given company participates in the bidding process will be given as

$$B_{pj} \in [C_p, I_j^{p*}] \tag{3.1.22}$$

where C_p is the project's announced cost. No bids lower than C_p will be accepted by the auctioneer.

Let δ_j denote the bid increase:

$$\delta_j = (I_j^{p*} - C_p)\kappa_j \tag{3.1.23}$$

where κ_j is the company's predefined relative bid increase.

Define the following:

- I_j^p company j's last bid
- $\bar{I}^p$ the current highest bid

- $I_j^{p'}$ company j's next bid

The next bid a company will submit is given by

$$I_j^{p'} = \min(I_j^{p*}, I_j^p + \delta_j | I_j^p + \delta_j > \bar{I}^p, \bar{I}^p + \delta_j) \tag{3.1.24}$$

The next bid will be constrained by the company's NPV. Each bid will also follow the company's bidding strategy, dependent on the next bid being higher than the current. If the company has to break their strategy they will increase the current max bid with their bid increase δ_j

Bids will be submitted in a circular fashion. When the auctioneer has completed one full rotation of the auction circle and no new bids are submitted, a winner is announced. The auction process is an open English auction (open bids, highest bid wins and is payable). Figure 3.3 gives a conceptual overview of the auction process.

Dividends

Agents holding a particular stock may be attributable to dividend payments. Dividends are realised as a percentage of net income. The amount of net income not distributed to the owners is used to build capital buffer and pursue new growth opportunities[12]. Companies in the model have a dividend policy. Policies contain a dividend payout ratio. To introduce stochasticity[13], the realisation is randomised. Formally, let D_j denote the maximum payout ratio for a company j. Then

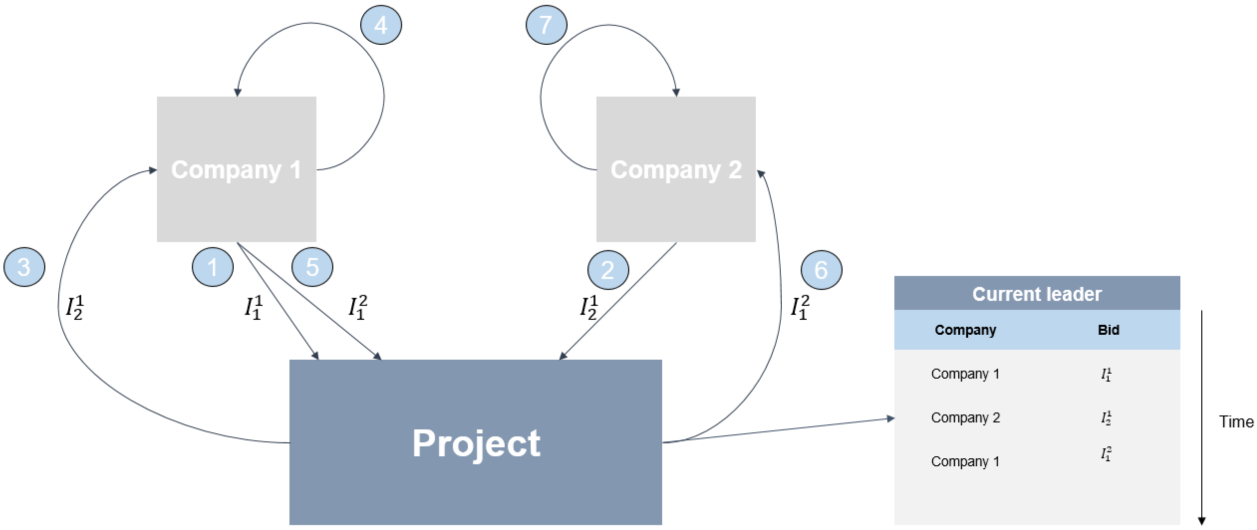
$$\tilde{d}_j \sim \mathcal{U}(0, D_j) \tag{3.1.25}$$

where $\tilde{d}_j$ is the random realised payout ratio from a uniform distribution between 0 and D_j.

Dividends are paid out according to a dividend payment schedule. This schedule is derived from the earnings reporting days. Realisation of dividends occur on the same day as the realisation of the company's earnings. Practically, dividends are consequently paid out quarterly.

The importance of dividends relates to capital influx. Without dividends, the overall amount of capital is fixed in the model. As Bjerkøy and Kvalvær (2018) discuss, the artificial economy[14] does not grow when there is no influx of capital. The trading environment becomes stable with only small fluctuations in both trading volume and asset prices. An undesired side effect is value aggregation on a few agents. At one point, wealthy agents with successful strategies systematically outperforms agents with less wealth. The result is a few agents dominating the economy, whereas the rest of the agent population is either bankrupt or perform insignificant trades. LeBaron (2006) describes an approach to avoid this situation by having a stable inflow of noise agents. The new agents are endowed with an initial wealth, consequently increasing the wealth of the system over time. As companies are profit-generating entities as in Bjerkøy and Kvalvær (2018), we model capital inflow through dividends instead.

[12] Growth opportunities are in form of investments, as described in section 3.1.4.
[13] Even though investments introduce stochasticity in the dividends, we do not want to let the agent be able to learn the payout ratio.
[14] That is, there is no value appreciation. Stock prices evolve within a limit.



Current leader	
Company	Bid
Company 1	I_1^1
Company 2	I_2^1
Company 1	I_1^2

Figure 3.3: Illustrative overview of a bidding process between two companies. (1) Company 1 submits their bid. (2) Company 2 submits their bid. (3) The auctioneer notifies Company 1 that Company 2's bid is the highest. (4) Company 1 chooses between increasing their bid or step out of the auction. (5) Company 1 decides to increase their bid. (6) The auctioneer notifies Company 2 that Company 1's bid is the highest. (7) Company 2 choose between increasing their bid or step out of the auction. No further bids are submitted and Company 1 is awarded the project at a cost of I_1^2.

3.1.5 Debt

Modelling different capital structures is a novel addition to agent based models. Earlier ABMs such as Thurner, J. D. Farmer, and Geanakoplos (2012) and Fischer and Riedler (2014) modelled debt at the agent side: Agents are allowed to lever their portfolio. In our model, agents are unable to lever. Companies, i.e., the assets an agent can purchase, can instead have leverage. Leverage are used to finance investments as described in section 3.1.4. Companies may realise higher profits compared to what would have been possible with equity alone, although they may go bankrupt if said investments fail. An agent holding a share in the company which goes bankrupt, lose their position in the stock. The stock price is forcibly set to zero and no dividend can be paid out.

Incorporating debt in the model connects companies more tightly to the overall market conditions. In Bjerkøy and Kvalvær (2018), the interest rate influenced only the model through the agents' discount rates. By allowing companies to borrow, we expect to observe the following:

(i) **Bankruptcies**
Debt require instalments and interest payments. Insufficient cash flows can therefore render the companies unable to pay back the debt. Additionally, changes in the interest rates impact the financial health of the companies.

(ii) **Volatility in earnings**
In Bjerkøy and Kvalvær (2018), the earnings process was dependent on previous earnings in addition to a stochastic term. The companies had no way of influencing their earnings. Undertaking investment opportunities are introduced as one such way to let companies endogenously influence earnings. Investments cost a fixed sum of cash and have an uncertain payoff. Companies without the required cash available may borrow in order to finance the project. We expect levered companies to have greater volatility in net income compared to companies that do not borrow to finance their projects due to variability in interest payments.

Debt is repaid according to a debt schedule. Instalments, as well as interests, are paid on the earnings realisation day[15]. Interests are deducted from the quarterly earnings and the company's cash reserve is reduced by the debt amortisation.

Debt Profiles

A debt profile is a generalised profile for the debt classification for an individual company. Debt profiles are imperative for contribution (iv) regarding a model framework for debt and credit spread. Profiles are similar to credit ratings given by institutions such as Moody's, Standard & Poor and Fitch[16]. Credit ratings are used in practice to capture the credit worthiness of a company (Kronwald 2009). These metrics can be used to rank risks between companies. Assigning a credit ranking to a company involves advanced and complex processes. Once compiled, they convey a large set of information in a few letters. In addition to provide information to investors about the risks associated with the debt, credit ratings determines what interests that can be charged. More risky debt demands higher interest rate. Usually, the interest rates are usually determined by some reference index[17] and a credit spread. The credit spread is the interest rate basis points above the reference index. Formally, for a company j, the interest rate paid on their debt at time t is defined as

$$r^t_{cj} = i^t + CS_j \tag{3.1.26}$$

where i^t is the floating base interest rate at time t and CS_j the credit spread. CS_j is defined by the company j debt profile (i.e., credit rating) and r is an exogenous parameter.

The different profiles, description on profiles and structural parameters are shown in table 3.2. A higher profile (in lexicographic order), indicates a more stable company. Modelling credit spread using debt profiles have the following advantages:

(i) **Time-varying base interest rate**
The interest rate is connected to the overall macroeconomic market climate. In real life, interest rates are time-varying and subject to change. Instead of having a fixed interest rate, one allows the market to transition between different regimes. The credit spread holds information about the relative risk of the company.

(ii) **Convenient and generalised**
One effectively avoids modelling the credits spread individually and can assign companies to a debt profile. The profile holds information for companies with similar characteristics.

(iii) **Useful way to provide real cases for scenario testing**
By structuring the profiles similarly to credit ratings from the real world, one can model risk structures and credit spreads conveniently and supply real data.

[15] This is convenient as the model cash flow statement and profit-and-loss statement will be conceptually similar to real statements.

[16] A description of Standard $ Poor's credit ratings is given in appendix E

[17] Inter alia LIBOR and NIBOR; floating money market interest rate indices.

Profile	Description	Credit spread	Maximum leverage
A	Highest quality and lowest degree of risk. Extremely stable and close to no risk of default.	r_{cA}	D_A
B	Medium quality and speculative fundamentals. Currently healthy, but future payments may be unreliable in the long term.	r_{cB}	D_B
C	Poor quality and highly speculative. Distressed companies and unreliable payments.	r_{cA}	D_C
D	Companies in default.	r_{cA}	D_D

Table 3.2: Description of debt profiles in the model. Each profile have a description of the riskiness of the business as well as a structural parameter containing information about the credit spread of the profile.

The maximum leverage column indicates how much debt a company can have relative to their equity. Stable companies, i.e., higher profiles, should have larger maximum leverage. Leverage is defined as

$$Leverage_j = \frac{D_j}{E_j} \tag{3.1.27}$$

for a company j. E_j is the current equity and D_j the current debt. A limit on maximum leverage draws on inspiration from real financial markets as well as insight on capital adequacy offered by Jarrow (2013). According to Jarrow (2013), maximum leverage ratio rules have a long history in financial markets. Additionally, they are intuitively easy to understand and simple to compare across companies and industries.

With reference to section 3.1.4, the maximum bid I_j^p a firm at any point can submit is

$$I_j^p = C_j + (D_j^* - \frac{D_j}{E_j})E_j \tag{3.1.28}$$

where C_j is the current cash reserve and D_j^* is the maximum leverage for the company. This restriction is inspired from how banks limit leverage in practice and often applied in research papers as well (Chan and Steiglitz 2008).

The debt profile for a company is determined endogenously. Initially, each company is assigned a profile randomly. As the model runs, the credit profile is a function of volatility in relative earnings. A company with stable earnings should be rewarded with a lower interest rate. As there are four profiles in the model, we divide the companies in quantiles based on the volatility in relative earnings. The 25% companies with the lowest volatility is assigned to profile A and so on. Profile assignment is done annually.

Interests Expenses and Tax

Interests on debt are tax deductible. In order to account for the fact that different countries[18] have different tax policies, the marginal tax deductible rate is defined as τ_d.

Denote the tax deductible for company j as T_j. Then,

$$T_j^d = r_j D_j^t \tau_d \tag{3.1.29}$$

for the debt D^t that matures on day t. The tax expense T_j is

$$T_j = (\text{EBT}_j - T_j^d)\tau \tag{3.1.30}$$

where τ is the marginal tax rate.

Consequently, the EAT becomes

$$\text{EAT}_j = \text{EBIT}_j(1-\tau) + r_j D_j^t \tau_d \tag{3.1.31}$$

where τ is the marginal tax rate.

EAT is the financial result of the company. If the business is profitable, the profit is distributed to equity owners as dividends. Should the EAT be negative, the company will decrease its cash reserve or, eventually, go bankrupt. A bankruptcy makes the company default on its debt obligations. This has no additional side-effects in our model except that the share price goes to zero and no dividends can ever be paid out[19].

A waterfall diagram of the process for calculating EAT from EBIT is shown in figure 3.4

[18] Examples include different corporate tax levels between the US and Norway. Additionally, Norway has different tax levels for companies in the oil industry.

[19] Thus, from the shareholders perspective a bankrupt company is worthless.

3.2 Portfolio Composition and Order Submission

We incorporate the orderbook and market clearing mechanism as of Bjerkøy and Kvalvær (2018). The discussion on implementation is included for completeness. From the discussion in section 2 a clear limitation in earlier research has been that the models only support end-of-day trading . Agents can update their portfolio once each day - there are no possibilities for intraday trading (Marchesi et al. 2003)[20]. This practice is not very realistic as trading is done continuously and asynchronously in real financial markets. The importance of this is most present in periods of turmoil. Large intraday movements have impact on the observed bid-ask spreads, consequently impacting the demand and supply and partial equilibrium (Andersen, Bollerslev, et al. 1997). An artificial stock market should therefore strive to offer trading on different assets multiple times per day. The model proposed in this thesis offers trading to all agents several times each day. This approach is favourable as agents can take into account the last market price and the current bid-ask spread before making their trading decision.

This section first describes the portfolio of the agents. Next, orders are order matching is presented. Finally, price setting of orders is discussed.

3.2.1 Portfolio

The portfolio represent the composition of shares in different assets held by an agent. We omit the time subscript t for clarity. For agent i the portfolio is given by the following vector:

$$[q_{i1}, q_{i2}, \ldots, q_{iJ}, q_{iC}] \tag{3.2.1}$$

where q_{ij} is the number of shares in asset j held by agent i. q_{iC} represent how much cash the agent has. Currently, no agent can hold a negative position in stocks or cash. Negativity constraints on stocks indicate that shorting is not allowed. The advantage of short constraints is that it makes the model much simpler as no issues with large short positions will occur. One effectively avoids assuming how much an agent must place as collateral[21] and what interest to charge for short positions. However, a clear limitation is that one loses important dynamics of efficient capital markets. In addition to be used for speculative positions (i.e., investing in depreciation of stock value), short positions serve as hedges. Investors can also short sell stocks they believe to be severely overpriced. Negativity constraints on cash mean that no agent can borrow money. This constraint is reasonable as the cash is risk-free, and lending out privately to other agents typically is not considered risk-free. Hence, the following two restrictions must hold:

$$q_{ij} \geq 0, \quad \text{for } j = 1, \ldots, J \tag{3.2.2}$$

$$q_{iC} \geq 0 \tag{3.2.3}$$

The market value of agent i's position in asset j is given by

$$X_{ij} = q_{ij} P_j \tag{3.2.4}$$

where P_j represent the current price. $P_C = 1$ as we define a unit of cash to have value 1. Similarly, the portfolio weight of asset j is given by[22]

$$x_{ij} = \frac{X_{ij}}{X_{iC} + \sum_{j=1}^{J} X_{ij}} \tag{3.2.5}$$

[20] Observe that our current implementation of learning agents support only one action (i.e., buy, sell or hold) for each day. The order framework, however, supports multiple actions per agent. On the other hand, orders from different agents can arrive at different steps during the day, generating the desired intraday trading effect.

[21] Normally, to be allowed to short, one must place a certain amount of funds as collateral in an account.

[22] x_{iC} is defined similarly.

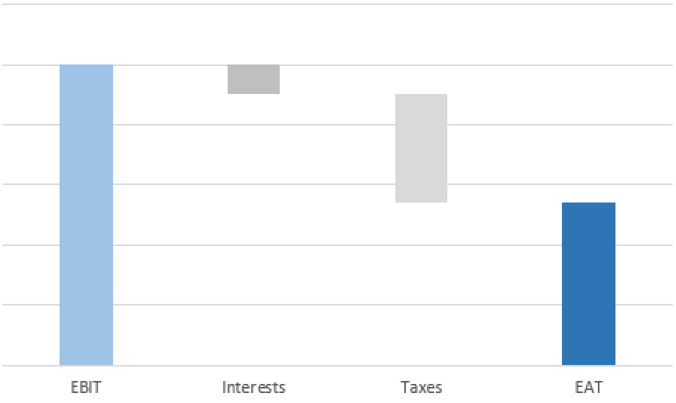


Figure 3.4: EBIT-EAT bridge showing the effect of interests and taxes.

3.2.2 Orders

An order ω submitted to an exchange is a sextuple

$$\omega = (i, j, \text{type}, q, P, \text{timestamp}) \tag{3.2.6}$$

where i and j refer to the agent and asset, respectively. The type is any of $\{\text{buy}, \text{sell}\}$. If the agent wishes to sell, then the type is sell (ask order). If the agent wishes to buy shares, then the type is sell (bid order). The quantity and price is given by q and P, respectively. Finally, a unique timestamp is provided. An illustrative order is shown in figure 3.5. The order is processed by the exchange and matched according to the matching rules. If the order is not immediately completely matched with another order, the order is placed in an orderbook.

3.2.3 Matching Rules

All orders are placed on a public exchange, so every agent is able to see all non-completed and completed orders. Let $\omega = (i, j, \text{buy}, q_j^{bid}, P_j^{bid}, t)$ and $\omega' = (i', j, \text{sell}, q_j^{ask}, P_j^{ask}, t')$ be a buy and sell order on the same asset, respectively. Moreover, let P_j^{ask} and q_i^{ask} be the price and quantity for a sell order, respectively. Similarly, let P_j^{bid} and q_j^{bid} be the buy price and quantity. Let t and t' be two different timestamps. Which assets to place orders on and what quantity is determined by the agent's trading strategy. The price is set to be approximately equal to the last close, as described later in section 3.2.4

A *match* is said to occur if $P_j^{bid} \geq P_j^{ask}$. For order completion, the following rules apply:

(i) If a bid order is placed and no ask orders exist, then no match occurs and the bid order is placed in the orderbook.

(ii) If a bid order is placed and $P_j^{bid} < P_j^{ask}$ then no match occurs and the bid order is placed in the orderbook.

(iii) If $P_j^{bid} \geq P_j^{ask}$ and $q_j^{bid} = q_j^{ask}$ then both orders are completely executed. Agent i will receive q_j^{bid} shares of stock j at a price of P_j^{ask} each[23].

(iv) If $P_j^{bid} \geq P_j^{ask}$ and $q_j^{bid} < q_j^{ask}$, then the bid order is completely executed and the ask order is partially executed. Agent i will receive q_j^{ask} shares of stock j at a price of P_j^{ask} each and a new ask order with price P_j^{ask} and quantity $q_j^{ask} - q_j^{bid}$ will be placed on behalf of agent i'.

(v) If $P_j^{bid} \geq P_j^{ask}$ and $q_j^{bid} > q_j^{ask}$, then the bid order is partially executed and the ask order completely executed. Agent i will receive q_j^{ask} shares of stock j at a price of P_j^{ask} each and a new bid order with price P_j^{bid} and quantity $q_j^{bid} - q_j^{ask}$ will be placed on behalf of agent i.

Rules for an incoming sell order are analogous to the above matching rules. Cases (iii)-(v) are denoted as a transaction. Orders that are not matched (i.e., no transaction occurs) will be placed in a queue. The matching algorithm will always try to match overlapping orders according to which order that has the lowest timestamp. These trading mechanisms are comparable to a continuous order system with automatic order matching between anonymous agents (Chiarella, Iori, et al. 2002). An illustration of placing orders is shown in figure 3.6.

3.2.4 Price Formation

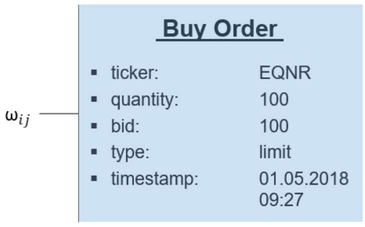
The price of each order on shares on asset j is typically close to the last transacted price. To induce dynamics in the submission of orders, each order submitted by any agent is adjusted according to the following rules:

(a) If the submitted order was a bid order, the price will be set to $P_j^{bid} = P_j \times z$ with $z \sim \times\mathcal{N}(\mu_o, \sigma_o)$

(b) If the submitted order was an ask order, the price will be set to$P_j^{ask} = P_j \times z$ with $z \sim \mathcal{N}(\frac{1}{\mu_o}, \sigma_o)$

Where P_j is the most recent price of shares in asset j. This is the same approach as in Marchesi et al. (2003) in order to generate small fluctuations in bid and ask prices around the bid-ask spread. One does this to avoid dead-locks

[23] Similarly, the agent i' will hand away its q_j^{bid} shares of asset j and receive P_j^{ask} each.



Buy Order

- ticker: EQNR
- quantity: 100
- bid: 100
- type: limit
- timestamp: 01.05.2018 09:27

Figure 3.5: Structure for a buy order on EQNR. An order consists of a ticker, quantity, price, order type and a timestamp.

where each subsequent order is placed *at* the spread and no matching occurs. μ_o is the average increase and decrease in current bid-ask for respectively a buy and sell orders. Larger values of this parameter generates larger spreads. Efficient markets, i.e. markets with liquid stocks, encourage smaller bid-ask spreads. σ_o is the standard deviation of the described increase or decrease. This parameter is of relative importance: smaller σ_o make it less likely that order prices significantly differ from the last price. These parameters can be specified as input to the model. With a similar discussion of the implication of these parameters, Marchesi et al. (2003) fixes $\mu_o = 1.01$ and $\sigma_o = 0.01$. One can reduce our model to the model of Marchesi et al. (2003) having only noise agents submitting orders at random.

3.3 Agents

This section and section 3.4 describes the agents that populate the system. Agents are heterogeneous with different strategies. In Bjerkøy and Kvalvær (2018) agents had a static set of rules. We extend the model with learning agents that are able to adapt to the environment. Consequently, the rule set for these agents are heuristics which constantly evolves and shapes to the current market conditions. In Bjerkøy and Kvalvær (2018) there were four types of agents; noise, trend, fundamental value and portfolio optimising agents. We discard agents with a fixed rule set, but keep noise agents. Noise agents are liquidity providers and necessary for avoiding abnormal situations with market failures in the form of large discrepancies between supply and demand (LeBaron 2002).

3.3.1 Noise Agent

Noise agents provides liquidity and fluctuations in the asset prices. They are agents which utilise no strategies to increase their portfolio value. Consequently, they do not have any high-level goals. They essentially trade assets at random limited only by cash and holding constraints. Farmer and Joshi (2001) view these traders as the stochastic component in a random walk process. As prices are updated directly in their model, noise is added to the price and liquidity is provided by a market maker[24]. We create an order-driven market, hence prices cannot be modified directly. Consequently, the approach in Marchesi et al. (2003) to model noise as an agent is chosen. Regardless of which approach being chosen, noise agents or stochastic components are required to demonstrate small fluctuations in prices (LeBaron 2006).

Noise agents are initialised with a trading intensity denoted as I. This parameter is a probability of performing a trade at each round and reflects how often the agent will trade. Formally, the following is done for each asset j:

- Let $z \sim \mathcal{U}(0;1)$ be a realisation from a uniform distribution distributed between 0 and 1. Let d be the number of periods[25] in a single day. The trading decision $\phi_{ij}(z; I, d)$ to buy, sell or hold is defined according to equation (3.3.3). The motivation for equation (3.3.3) is that, on average, the agent performs dI trades per round[26] and the probability of a buy is equal to the probability of a sell.
- If ϕ_{ij} = buy and the agent has sufficient funds to buy at least one stock:

$$q \sim \mathcal{U}(1, \lfloor \frac{X_{iC}}{P_j} \rfloor) \tag{3.3.1}$$

  where q is the submitted number of shares, X_{iC} the available cash (equation (3.2.4) and P_j the current price of asset j.

[24] Farmer and Joshi (2001) uses what is called a price-impact function to update the prices. Our model is order-driven.
[25] A single trading day is divided into multiple rounds approaching continuous time. Section 4 discusses this in detail.
[26] As this scales with the number of rounds approaching continuous time, I needs to be adjusted when increasing the number of rounds.

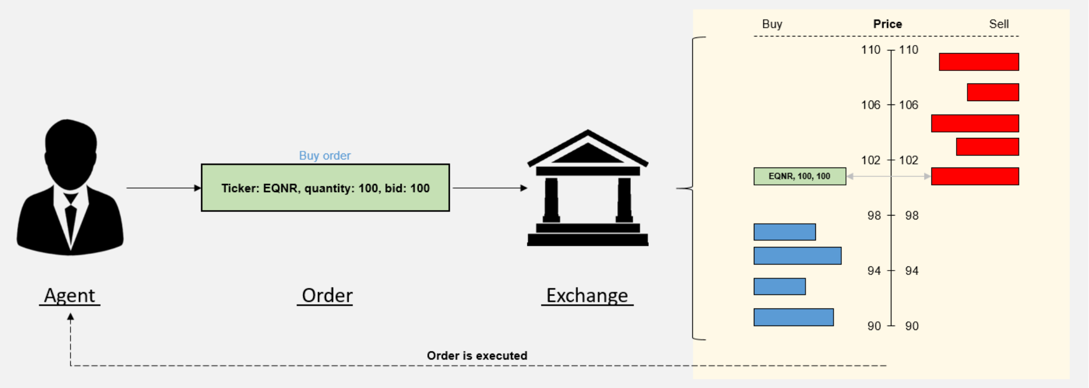


Figure 3.6: Order placement procedure. An agent places an order on the "EQNR" stock with a quantity of 100 and bid price 100. The order is sent to the exchange for matching. A matching sell order is found at the same price and quantity, so both the buy order and the corresponding sell order are completely executed.

- If $\phi_{ij} = \text{sell}$ and the agent has at least one share of asset j available for sale:

  $$q \sim \mathcal{U}_I(1, q_{ij}) \tag{3.3.2}$$

  where q_{ij} is agent i's holding of asset j and $\mathcal{U}_I$ the uniform integer distribution bounded below and above by 1 and x_{ij}.

$$\phi_{ij}(z; I, d) = \begin{cases} \text{buy}, & \text{if } z > 1 - \frac{I}{2d} \\ \text{sell}, & \text{if } z < \frac{I}{2d} \\ \text{none}, & \text{else} \end{cases} \tag{3.3.3}$$

If $\phi_{ij} = \text{buy}$, the agent will submit an order with price $P_j = P_j^{bid}$ and quantity q. Conversely, if $\phi_{ij} = \text{sell}$, the agent will submit an order with $P_j = P_j^{ask}$ and quantity q. P_j^{bid} and P_j^{ask} is the observed bid and ask price, respectively.

3.4 Learning Agents Implementation

Learning agents constitute contribution (i) regarding reinforcement learning agents. In Bjerkøy and Kvalvær (2018) agents had a static set of rules. Value agents based their investment decisions on the fundamental value; the discounted earnings of the company. Trend agents assessed the momentum exhibited by the stocks and took positions that were in line with the overall trend. Portfolio optimising agents used statistical measures such as mean-variance analysis to circle out investment opportunities. Although the agents had the possibility to adjust their behaviour by how much weight that was placed on different factors[27], their decision where constrained to the use of one investment strategy (i.e., value, trend or portfolio). In reality, investors use a variety of strategies. To model this realistically, agents should be able to combine several types of strategies to identify investment opportunities. Investment strategies evolve over time with the market climate. Agents must consequentially be able to learn from past experiences. Rutkauskas and Ramanauskas (2009) suggest to use reinforcement learning in the form of Q-learning. A limitation of their framework relates to the observation space. The authors impose assumptions on behaviour related to the fundamental value and stock momentum. The learning agents in our model can observe the full environment and generate actions using past observations. Each agent has their own neural network and learning procedure. Consequently, the agent may have different views on the same stock and different time horizons. This represent the heterogeneity of investors in real financial markets. Agents sense the environment and performs actions asynchronously and indecently of each other.

The learning agents are represented as parameterised models that map a set of input observations to a set of actions. The parameters of the model are updated by performing a large number of simulations and adjusting the parameters so they maximise the expected value of a reward function. That is, the agents trade for a couple of years in an artificial stock market. If they do well (measured by the reward function), the parameters are adjusted so that they will do similar actions in similar situations. If they perform badly, the parameters are adjusted so there is a smaller probability of carrying out similar actions in similar situations later. An optimisation algorithm performs simulations and parameter updates to the agents so, over time, the performance of the agents improve.

The two next sections briefly introduce neural networks and discuss some of the preliminaries of reinforcement learning. Section 3.4.3 discusses some key issues associated with reinforcement learning that need to be assessed. Next, the Proximal Policy Optimisation (PPO) algorithm is briefly introduced in section 3.4.4. PPO is the algorithm that is responsible of adjusting the parameters of the agents so they learn from experience. Finally, sections 3.4.5, 3.4.6 and 3.4.7 discuss specifications for the observation space, action space and rewards in the artificial stock market, respectively.

3.4.1 Neural Networks

We model agent cognition using neural networks. The network mimics the design of the human brain. Neural networks are frameworks for information process systems; not algorithm implementations. They are used for various tasks, including natural language processing and image processing. We briefly describe how neural networks work. We limit the discussion to a high level description of the nomenclature, overall implementation and key components for completeness. For a good introduction and discussion of neural networks we refer to Bishop (2006).

The network is made up of layers. Each layer consists of several nodes, commonly referred to as *neurons*. A neuron does a small calculation based on the input - stimulus - it receives and generates an output. The process is analogous to the stimulus-response pattern in human brain neurons. The network takes observations of the environment as input at the input layer. The observations are encoded as vector of numerical values. Signals propagate through the network

[27] Note that this is not the same factors in our model. Factors in Bjerkøy and Kvalvær (2018) are observable properties of the stocks such as price returns and earnings.

through one or several *hidden layers*. Hidden layers take some processed input and apply a nonlinear function on the signal. In a feed-forward network a series of functions are applied to the signal.

More formally, the network is made up of $n \geq 1$ layers. We let layer j and $j+1$ have D_j and D_{j+1} neurons, respectively. Each neuron in layer j is connected to every neuron in layer $j+1$ by means of a weight[28]. Thus, a weight matrix $W_{j,j+1} \in \mathbb{R}^{D_j \times D_{j+1}}$ represents the connection weights between every pair of neurons in the two layers.

If the input to layer j is a vector $h_j \in \mathbb{R}^{D_j}$, the corresponding output of layer j is given by

$$h_{j+1} = \sigma(W_{j,j+1} h_j) \tag{3.4.1}$$

Where $\sigma(\cdot)$ represents a differentiable non-linear function[29], commonly referred to as the activation function. The output of layer j is used as input to layer $j+1$.

There are two special cases: (1) The first layer, the input layer, receives the raw observation state denoted as $s \in \mathbb{R}^{D_0}$. (2) Layer n, the last layer, apply the identity function $g(x) = x$ instead of a non-linear activation function $\sigma(\cdot)$.

Consequently, the full computation from a single vector of input s to a vector of output $y \in \mathbb{R}^{D_n}$ is given by

$$y = g(W_{n-1,n}\sigma(W_{n-2,n-1} \cdots (\sigma(W_{1,2}\sigma(W_{0,1}s))))) \tag{3.4.2}$$

In our model, we have J assets. For each asset, the agent is allowed to either (1) do nothing; (2) place a buy order; or (3) place a sell order. Hence, there are $3 \times J$ available actions in total. Therefore, the final layer should have $3 \times J$ neurons, where the three first neurons roughly correspond to the probability of doing nothing, buying and selling on the first asset, respectively. Correspondingly, the output of the three next neurons represent the probability of doing nothing, buying and selling on the next asset, and so on. A schematic overview of the network is shown in figure 3.7.

The network parameters are defined as a vector θ with a total number of $\Pi_{j=0}^{J-1} D_j D_{j+1}$ parameters which is the concatenation weights in the network. Formally,

$$\theta = \text{concatenate}(\text{flatten}(W_{01}), \cdots, \text{flatten}(W_{J-1,J})) \tag{3.4.3}$$

Where *concatenate* concatenates a set of vectors and *flatten* transforms a matrix with dimensions (m, n) to a vector of length mn.

Finally, the network parameters are updated by adjusting the parameters in θ. In the context of reinforcement learning, an optimisation algorithm adjusts the parameters in θ so, on average, the probability of actions that lead to high total expected reward should be favoured to actions that lead to mediocre or bad rewards.

3.4.2 Reinforcement Learning Preliminaries

Reinforcement learning is a form of machine learning where an agent performs actions in an environment and attempts to maximise the expected value of a reward function (Sutton and Barto 2018). There are several techniques of reinforcement learning[30]. One example is trust region policy optimisation (TRPO) using policy gradients. Policy

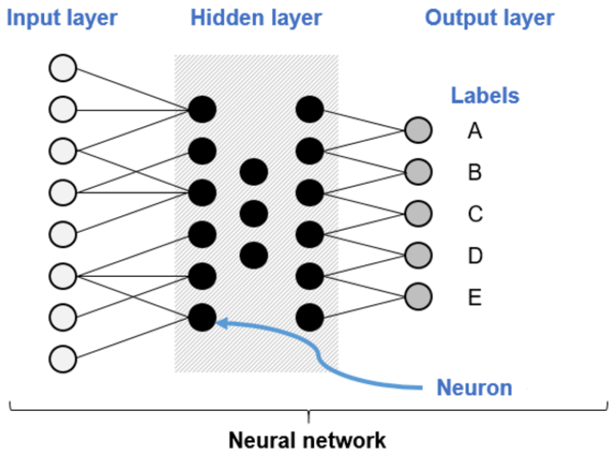


Figure 3.7: High-level illustration of a neural network and key components. In the hidden layer, there can be arbitrarily many connections between different layers. All of the circles are nodes.

[28] This kind of network architecture is very common and is known as a feed-forward network.

[29] In our experiments, we set $\sigma(x) = \max(0, x)$. Empirically, it has shown to be a good function (Kaiming He and Sun 2015).

[30] To keep the discussion short, we only discuss policy optimisation techniques.

gradients are methods to learn a parameterised policy π, which is a function mapping from an observation space $\mathcal{O}$ to an action space $\mathcal{A}$. The policy is approximated by a neural network with parameters $\theta \in \mathbb{R}^d$ with d parameters. Hence, denote

$$\pi(a|s, \theta) = Pr\{A_t = a|S_t = t, \theta_t = \theta\} \tag{3.4.4}$$

as the probability of choosing action a at time t given the environment state o at time t (Sutton and Barto 2018). Once action a_t is chosen, the agent ends up in state s_{t+1} with a probability governed by the transition probability distribution $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. In addition, the agent receives a reward r_t governed by the function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ for action a_t in state s_t.

At each timestep t, the agents observe[31] the environment, choose an action a_t, receive a reward r_t and a transition to a new state s_{t+1} occurs. This process continues until a termination criterion occurs[32]. The *trajectory* τ of length T is defined as the sequence of state-action-rewards triplet T periods ahead:

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{T-1}, a_{T-1}, r_{T-1}, s_T) \tag{3.4.5}$$

Where the initial state s_0 is drawn from a distribution μ_0 of possible initial states. From this, we can find the cumulative discounted reward η, discounted by a factor $\gamma \in (0, 1]$[33] as:

$$\eta(\tau, \gamma) = \sum_{t=0}^{T-1} \gamma^t r_t \tag{3.4.6}$$

The overall goal is to optimise the expected reward by following the policy π_θ:

$$\max_\theta \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T-1} \gamma^t r_t] \tag{3.4.7}$$

The economic interpretation is as follows: the agent observes the state of the market (such as macroeconomic factors, asset prices, leverage ratios, company earnings and its own portfolio holdings). The action is what the agent chooses to do based on the observations. Finally, at the end of the day, the agent receives an immediate reward (such as daily portfolio change, volatility, dividend received or a combination of those). The policy π_θ represents *how* the agent should act in different situations, parameterised by the vector θ. The discount factor γ weighs immediate rewards against expected rewards further in the future. Conceptually, γ can be viewed as the agent's investment horizon of the agent.

The parameters of the policy is updated using gradient ascent. That is, the parameters are adjusted in the direction of greatest increased expected reward. Specifically, the following parameter update is done:

$$\theta \leftarrow \theta + \alpha \nabla \mathbb{E}_\pi[\eta(\tau, \gamma)] \tag{3.4.8}$$

where α represents a small step size[34]. The ∇ operator finds the partial derivative of each parameter in θ subject to the expected reward η dependent on following policy π_θ. It is not in the scope of this thesis to derive the policy update procedure to a form suitable for optimisation. We refer to Sutton, McAllester, et al. (1999) for a theoretical justification of the policy gradients and Schulman, Moritz, et al. (2016) for more recent modifications to the policy update procedure.

3.4.3 Issues with Reinforcement Learning

In this section we discuss some key issues that need to be addressed to successfully train the agents. Some issues are unique to the problem we try to solve while some are common to other reinforcement learning tasks.

(i) **Non-stationarity in stock markets**
Stock markets do not repeat themselves. Otherwise it would be possible to create profitable trading strategies which exploit the deterministic aspects of the stock market (Lo 2004). Because of the non-stationarity of stock markets, we rule out all learning methods that relies heavily on stationarity, such as Q-learning. As discussed in chapter 2, Rutkauskas and Ramanauskas (2009) developed agents using Q-learning[35] to predict dividend payouts.

[31] The observation space might not be complete; that is, the agents might not observe all relevant factors necessary to do optimal decisions. Indeed, investors in real financial markets do not have a complete view of all relevant factors either.

[32] A natural termination criterion when simulating financial markets might be reaching a specific date. .

[33] The discount factor specifies how important immediate rewards are compared to expected future rewards. For example, $\gamma = 1$ specifies immediate reward to be equally important as a reward 100 days in the future. The value $\gamma = 0.87$ specifies the reward after 5 days to be $0.87^5 \approx 0.50$ times as important as the reward today.

[34] The motivation for using a small step size is to limit the magnitude of parameter value changes; the gradient of steepest ascent is not valid for large changes. Step size is also known as learning rate.

[35] To limit the scope of this thesis we omit a discussion on the fundamentals of Q-learning. We refer to (Watkins 1989) for an introduction to Q-learning

We deviate from their model by letting the agents learn a *policy* of what to do, instead of learning the value of taking action a in state s[36]. Hence, policy gradients methods discussed in the previous section is a more natural way to model the agent cognition in a stochastic environment.

(ii) **Challenging to learn across bull and bear markets**
Learning in a stock market can be significantly harder than other problems due to the stochastic nature of stock markets[37]. In a favourable market environment where all stocks are rising (a "bull market"), an agent seeking to maximise portfolio returns can basically do whatever it wants and still end up with a positive total return at the end of the year. Updating parameters based on rewards from a bull market can be misleading because the corresponding reward in normal market environments could be low. The converse also holds in unfavourable market conditions where stock prices generally fall ("bear markets"); Rewards are low, but the actions could still be considered appropriate given the market conditions.

A way to mitigate the challenges with learning across regimes is to run multiple simulations in parallel and aggregate the trajectories when performing parameter updates[38]. In this way, the likelihood of all simulations being in bull or bear markets decrease sharply. In our experiments, we run between 16 and 32 simulations in parallel. Additionally, the step size α in equation (3.4.8) is set to a small size to limit the effect of large parameter updates which would reduce agent performance.

(iii) **Noisy gradient updates**
In general, policy gradient methods suffer from two main challenges: the large number of experience required to train agents and the difficulty of obtaining stable improvements in the policy due to highly stochastic gradients. We address the former challenge by using an exponentially-weighted estimator of the value function[39] as introduced by Schulman, Moritz, et al. (2016). Instead of using the raw rewards when computing gradients, we estimate how much better it would be to do a different action a instead of following the default policy. The value difference, known as the advantage, can be used as a way to reduce variance of the gradients at the cost of some bias in the estimation of the gradient update[40]. We address the latter challenge by using an optimisation scheme which limits parameter updates to be within a trusted region, known as trust region policy optimisation. The next section will go into some more detail of Proximal Policy Optimisation (PPO), which can be seen as a variant of trust region policy optimisation.

3.4.4 Proximal Policy Optimisation

Proximal Policy Optimisation (PPO) is an optimisation algorithm[41] introduced by Schulman, Wolski, et al. (2017a) used to optimise a "surrogate" objective function[42]. We discuss the PPO algorithm for the reader of this thesis to get an overview of how the optimisation algorithm works, how it affects the model design and how it affects the simulation procedure. The algorithm has a few important technical details we omit in this discussion. For complete implementation details we refer to the original paper. Briefly, the optimisation algorithm alternates between two phases: The first phase samples trajectories from different simulated environments. The second phase performs parameter updates on the policy parameters θ. An overview of the optimisation procedure is illustrated in figure 3.8. In the following section we introduce some necessary terminology. Later, the two phases of PPO are briefly discussed.

Terminology

Denote $Q(s_t, a_t)$ as the expected cumulative discounted reward of taking action a_t in state s_t and subsequently sampling[43] from the policy π until termination:

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1},a_{t+1},\ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})] \tag{3.4.9}$$

[36] Recall, there are no guarantees that one observes the same state twice.

[37] For example, consider learning Tetris. In Tetris, the state transition distribution P and distribution of initial states μ_o is not very large, whereas the state space is significantly larger in a stock market. For example, the initial state s_0 could be in the middle of a recession. An agent seeking to maximise dividend payouts will have a hard time getting any reward at all.

[38] Recall from section 3.4.2 that a trajectory is a sequence of state-action-rewards.

[39] The value function $V^\pi(s_t)$ of the policy π starting in state s_t is the discounted expected cumulative reward obtained by following the policy until termination; $V^\pi(s_t) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T-1} \gamma^t r_t]$

[40] See Schulman, Moritz, et al. (2016) for a discussion on the effects of bias.

[41] Actually, PPO is a family of optimisation algorithms. To keep the discussion brief we discuss only one of them.

[42] A surrogate objective function approximates the real objective function. The real function can either be time-consuming to evaluate or unknown.

[43] The policy π is stochastic. For a given observation s_t and parameters θ, equation (3.4.4) gives a *distribution* of actions a_t. The actual action taken at time t is sampled from this distribution.

Additionally, define the value function $V_\pi(s_t)$ as the expected cumulative reward by following the policy π starting in state s_t at time t:

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots}[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})] \tag{3.4.10}$$

Finally, define the advantage $A(s_t, a_t)$ as the difference between the state-action value function and value function:

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t) \tag{3.4.11}$$

Intuitively, the advantage tells how much better (or worse) it is to take the action a_t at time t instead of sampling from the policy π. In general it is not possible to get an exact value of the advantage $A(s_t, a_t)$. However, we can get a good empirical estimate $\hat{A}(s_t, a_t)$ of the advantage by training another neural network whose only job is to estimate the advantage of actions a_t in state s_t[44]. This network is usually called the *critic*[45](Mnih et al. 2016). The critic is trained parallel to the agents. Hence, they become better and better at estimating the advantage of an action. But in order to train the networks, trajectories need to be collected from the simulations.

Phase 1: Experience Collection

When collecting experiences, S simulations are run in parallel, each as independent processes. Each such simulation collects B trajectories where each trajectory has length T. If each simulation has N learning agents from a single policy π, the optimisation algorithm will have a pool of a total SBN trajectories for each parameter update step. In our experiments we have $S \in [18, 32]$. I.e., between 16 and 32 experiments are run in parallel. We also have $B \in [1, 50]$ and $T = 90$ (i.e., trajectories are 90 days). The left part of figure 3.8 illustrates the experience collection phase.

If there are k distinct policies $\pi_1, \ldots, \pi_k$ in the simulation, then the same experience collection will be done independently for each of the k distinct policies. The PPO algorithm does not use the experience from a policy $\pi_{k'}, k' \neq k$ when updating policy k[46]. Once all trajectories are collected, they are passed to a central optimiser and parameter update is done.

Phase 2: Parameter Updates

The optimiser is given SBN trajectories from the simulations. It randomly selects a subset of these trajectories and estimate the advantage $\hat{A}$ for each of the state-action pairs in the selected trajectories. Next, an estimate of the gradients of the policy π with regards to the trajectories can be found by (Schulman, Moritz, et al. 2016):

$$\hat{g} = \hat{\mathbb{E}}_t[\nabla_\theta log \pi_\theta(a_t|s_t)\hat{A}_t] \tag{3.4.12}$$

Finally, the optimisation objective of PPO is conceptually the same as (3.4.4). The parameters are updated by maximising the likelihood ratio of actions:

$$\max_\theta \hat{\mathbb{E}}_t \left[\frac{\pi_{\theta_{\texttt{new}}}(a_t|s_t)}{\pi_{\theta_{\texttt{old}}}(a_t|s_t)} \hat{A}_t \right] \tag{3.4.13}$$

Subject to

$$\hat{\mathbb{E}}_t[\texttt{KL}[\pi_{\theta_{\texttt{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]] \leq \delta \tag{3.4.14}$$

where θ_{old} is the vector of policy parameters before the parameter update. Intuitively, equation (3.4.13) states that the optimiser should adjust the parameters θ so the probability of choosing actions with high advantage $\hat{A}_t$ is increased relative to previous probabilities (with parameters θ_{old}). Additionally, the new probability distribution should not deviate largely from the original distribution in a single step, as enforced by equation (3.4.14). $\texttt{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence[47]. The parameter δ limits how large the expected KL-divergence can be in a single optimisation step. In effect, the KL restriction limits the effect of large, adverse parameter updates. Conceptually, large parameter updates are unfortunate if there in reality was no causal relationship between the performed action and the received reward. The agent will in this case weigh the incorrect experience too heavily in the future.

Once done, the optimisation will select a new random subset of trajectories. As was discussed in the previous section, policy gradient methods suffer from noisy gradient updates and poor utilisation of experience. The PPO algorithm repeatedly selects random subsets of trajectories. This selection strategy reduces the problem of noisy gradient updates (because trajectories are sampled randomly from a pool of parallel and independent simulations) and increases data efficiency (because trajectories might be used multiple times).

[44] We create another neural network whose task is solely to estimate the value of a state given the observation; i.e., it is a mapping $\mathcal{O} \rightarrow \mathbb{R}$. We implement it as a network with similar architecture as the policy network π, except that the output layer has a Tg neuron instead of $3 \times n$ as discussed in section 3.4.1. The network is trained in parallel with the policy π.

[45] Intuitively, the advantage tells how good an action is. Hence, the name critic.

[46] Indeed, it does not make sense to use experience from policy k' when updating policy k. The reward function for policy k' is not necessarily the same as the reward function of policy k, For example, agents with policy k' might be much more risk-averse than agents with policy k so that generally low volatility leads to higher rewards. Using k''s trajectory can in that situation would be misleading.

[47] Kullback-Leibler (KL) divergence is a measure for how much one probability distribution differs from a reference distribution. For two distributions P and Q, it is defined as $KL[P, Q] = \sum_x P(x)(log(P(x)) - log(Q(x))$.

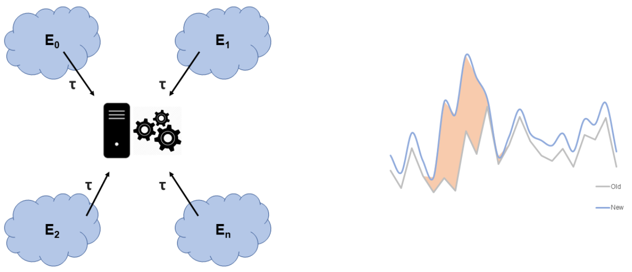


Figure 3.8: Schematic overview of the PPO algorithm. *Left*: Trajectories from different environments are collected. *Right*: Trust region policy optimisation is performed; Parameters $\theta_{\texttt{new}}$ are found from $\theta_{\texttt{old}}$. The red area indicates large deviations in the probability distributions.

3.4.5 Specification of Observation Space

The observation space $\mathcal{O}$ is the set of all possible observations in the environment. It is not complete, meaning all attributes that can be considered relevant is not necessarily included[48]. The agent might also receive information about the environment that is essentially irrelevant[49]. It is up to the parameters θ in the policy π_θ to determine which attributes are important and which are not. We draw on inspiration from T. Ramanauskas and A. Rutkauskas (2009) where they give processed information to the agents in form of perceived fundamental values and the EWMA-average in prices. However, we stay true to the model-free assumption of reinforcement learning advocated by Sutton and Barto (2018) and provide raw data of the full observation space. A formal and mathematical precise version of the observation space can be found in appendix D

For brevity, the notation with subscript θ_j^Q and θ_j^D is backward looking. As earnings and prices are realised at different time scales, denote θ_j^Q as how many past quarterly realisations the agent uses in its calculations. Similarly, θ_j^D denotes how many past daily realisations that are used. For simplicity the current time is $t = 0$ and components of matrix and vectors range backwards in time. For a given observation $s_t \in \mathcal{O}$ on day t, the following attributes are provided:

(i) **Company earnings**
For each asset, the θ_j^Q most recent returns in quarterly company earnings. The earnings are represented as percentage change between subsequent quarters. Relative values are provided instead of absolute as it provides natural limitations on the input range of company earnings[50].

(ii) **PE-ratios**
For each asset, the θ_j^D most recent price-earnings ratios. PE measures of how relatively expensive a stock is as a multiple between its price and earnings. However, the stock price is non-stationary. Non-stationary input to neural network makes training harder, so we use the PE-ratios to make it (somewhat) more stationary. Additionally, PE-ratios can be used to indicate how expensive a stock is compared to a peer group[51].

(iii) **Relative trading volume**
For each asset, θ_j^D most recent observations on trading volume measured in number of shares traded on a given day divided by total number of shares outstanding.

(iv) **Volatility**
For each asset, the equal-weighted daily volatility of the asset the last θ_j^Q days.

(v) **Macroeconomic factors**
The most θ_j^D most recent observations for each of the macroeconomic factors that influence the earnings of companies as described in section 3.1.4.

(vi) **Debt profile**
For each asset, the debt profile of the company, as defined in section 3.1.5.

(vii) **Portfolio holdings**
The current portfolio for the agent as defined in section 3.2.1.

[48] The good news is even though the environment does not reflect all relevant attributes, i.e., it is a partial observation, the theory on policy gradients still apply (Sutton and Barto 2018)

[49] For example, we could let the agents observe the Zodiac signs of the companies' CEOs.

[50] Neural networks tend to work better if the input range is constrained. For example, if a company has earnings 1000, 1200, 1500, 2000, 1500 in subsequent quarters the corresponding returns used as input are 20%, 25%, 33% and -25%.

[51] A peer group is a group of similar stocks. In our model a natural peer group would be assets in the same sector.

(viii) **Dividend yield**
For each asset, the θ_j^Q most recent dividend yields.

(ix) **Auxiliary information**
For each asset, information about whether the current day is a dividend payout day, earnings realisation day, days until the next dividend and days until the next earnings realisation. The first two columns are binary: a 1 indicates that the current day is a dividend day or earnings realisation day.

We emphasise that other attributes can be added to the observation space as well. In principle, there is no limit on what attributes can be provided to the agent. However, larger observation space generally mean more parameters in θ which in turn lead to longer training times. Therefore, to limit number of parameters and implementation effort, only the above attributes are provided in the observation space. We also believe they should be sufficient for the agents to create policies that handles the environment well.

3.4.6 Specification of Action Space

In this section the action space $\mathcal{A}$ is motivated and defined. An action $a_t \in \mathcal{A}$ on day t is a vector of $3J$ elements, where J is the number of assets in the simulation. If the vector is reshaped to a matrix with three entries per row it will have the following form:

$$a_t = \begin{bmatrix} a_t^1 \\ a_t^2 \\ \vdots \\ a_t^J \end{bmatrix} = \begin{bmatrix} l_N^1 & l_A^1 & l_B^1 \\ l_N^2 & l_A^2 & l_B^2 \\ \vdots & \vdots & \vdots \\ l_N^J & l_A^J & l_B^J \end{bmatrix} \tag{3.4.15}$$

The first row roughly correspond doing nothing (l_N^1), placing buy order (l_A^1) and placing a sell order (l_B^1) on the first asset. Similar interpretations holds for rows $2, \ldots, n$. As discussed in section 3.4.1 the output of the final layer is given by $g(W_J h_{J-1}) = W_J h_{J-1}$, where g is the identity function. Hence, the entries in a_t are not bounded between 0 and 1, and the rows generally do not sum to 1. A transformation of the rows using the softmax-function on each row of a transforms the raw values to probabilities. The softmax-function is defined as

$$S(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{3.4.16}$$

for a vector x. The transformation has the property that all $x_j \geq 0$ and x_j sums to one, i.e. $\sum_j x_j = 1$.

We apply the softmax-function to each row of the matrix and end up with values that can be interpreted as probabilities:

$$\begin{bmatrix} S(a_t^1) \\ \vdots \\ S(a_t^n) \end{bmatrix} = \begin{bmatrix} p_N^1 & p_A^1 & p_B^1 \\ p_N^2 & p_A^2 & p_B^2 \\ \vdots & \vdots & \vdots \\ p_N^J & p_A^J & p_B^J \end{bmatrix} \tag{3.4.17}$$

where p_N^j, p_A^j and p_B^j now correspond to probabilities of doing nothing, placing a buy order and placing a sell order on asset j on day t, respectively. The actual selected action for each asset is selected from a distribution where the probabilities match the rows in matrix 3.4.17.

Some considerations about the action space are in order:

(i) **The action space is a probability distribution**
Hence, to make a decision it is necessary to choose an action from the probability distribution $\{p_N^j, p_A^j, p_B^j\}$ for asset j. This has several implications. Firstly, if the agent has no preference for a single action, we expect the probabilities of doing nothing and placing bid or ask orders to be relatively equal. Thus, it lets the agent explore the consequences of its actions early in the simulation. After a while, the agent might learn that some actions generally are preferred in some situations. Hence, the probability of choosing that action increases. The agent's actions gradually shift from exploration to exploitation as it gains experience. This is characterised by the agent choosing a single good action with high probability and other relatively bad actions with very low probabilities.

(ii) **An agent can place multiple orders each day, but only one order per asset**
Because a single action is chosen for each asset, it is possible for an agent to perform a large reweighting of its portfolio on a single day[52]. However, a clear disadvantage with the proposed action space is the inability to perform multiple actions on a single asset the same day. For example, it is not possible to place a buy and sell

[52] For example, the agent might sell shares in assets 1, 2 and 3 while buying shares in assets 4. The exact procedure on how trading occurs is discussed in section 4.

order simultaneously on the same asset at different prices, making intraday trading impossible[53]. However, we believe multiple actions per asset should be considered in future models.

3.4.7 Specification of Rewards

Reward functions shapes the behaviour of the agents. A reward function can therefore be loosely translated to a form of investment philosophy (contribution (iii)). In this section we specify different investment philosophies and design rewards functions suitable to promote behaviour consistent with the philosophy. We demonstrate that it is possible to model complex investment behaviour simply by defining reward functions that stimulate certain behaviour.

Formally, the reward function is a function that maps a state s_t, action a_t and next state s_{t+1} to a real value (Sutton and Barto 2018);

$$r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R} \tag{3.4.18}$$

In our model the reward function is specified as the following function:

$$r(s_t) = w_P P(s_t) + w_d D(s_t) + w_F F(s_t) + w_v V(s_t) + w_{PE} PE(s_t) \tag{3.4.19}$$

The components of the reward function are as follows (where the subscript t and i for day and agent is judiciously omitted):

(i) $P(s_t)$: The portfolio return of the agent's portfolio (cash included) on day t[54].

(ii) $D(s_t)$: The dividend received on day t as a fraction of total portfolio market value[55].

(iii) $F(s_t)$: Net portfolio sector exposure[56]. The reward from portfolio sector exposure is found by summing across all assets and all factors:

$$F(s_t) = \sum_k \sum_j F_{jkt} x_j f_{kt} \tag{3.4.20}$$

where F_{jkt} denotes the net exposure to factor k from asset j on day t (equation (3.1.20)). f_{kt} denotes the current value of factor k on day t. x_j is the holding of asset j as a fraction of total portfolio value (equation (3.2.5)).

(iv) $V(s_t)$: Negative equal-weighted volatility of portfolio logreturns the last v_T[57] days. The volatility is negated as the agents seek to maximise rewards[58]. Thus, $V(s_t)$ seeks to construct portfolios with low volatility in daily logreturns.

(v) $PE(s_t)$: This metric is a measure of how expensive agent i's holdings are, as measured by the price-earnings ratio. In a given sector, it compares the weighted price-earnings assets in the same sector held by the agent to the sector average. Intuitively, agents maximising rewards from this function would prefer to buy assets that are cheap (i.e. have low PE) compared to to the sector average and sell assets that are expensive. The metric is given by

$$PE(s_t) = \sum_{s \in S} \left[\sum_{j \in s} x_j \left[\sum_{j \in s} \frac{P_j}{E_j} \left(\frac{1}{|s|} - \frac{x_j}{\sum_{j \in s} x_j} \right) \right] \right] \tag{3.4.21}$$

where S is the set of sectors, and $j \in s$ are assets in sector s. P_j is the stock price for asset j on day t and E_j is the most recent quarterly earning per share for the asset. The expression in the innermost bracket compare the average PE in sector s with the weighted PE average of agent i in the same sector. The agent is penalised hard if the relative holdings of expensive assets is high compared to a sector average. Finally, the middle summation weights the penalties by portfolio sector weights $\sum_{j \in s} x_{ij}$ for each sector s.

Finally, table 3.3 specifies example parameterisations of the reward function 3.4.19. The parameterisations construct reward functions encouraging a certain investment philosophy (contribution (iii)). In our experiments, we train agents that train variants of these reward functions.

[53] Indeed, it would be interesting to allow agents to perform multiple actions on single assets on a given day. We decided not allow this as it would increase the implementation effort significantly. For example, what should be the maximum number of actions an agent could do on a single day? Another issue is concerned with illiquid stocks: If the agent seek to maximise daily returns, it could place a sell order on an illiquid asset at a very high price and subsequently place a corresponding buy, artificially inflating the prices indefinitely.

[54] For example, if the initial market value of the agent's portfolio value is 1000 and ends at 995, the return is $1 - 995/1000 = -0.5\%$.

[55] For example, if the dividend received is 25 and the agent's portfolio value portfolio is 1000 on day t, the reward from this component is $25/1000 = 0.25\%$.

[56] For example, if the agent has 10% of its portfolio value in a stock with 60% exposure in the oil sector and the oil sector at time t has a relative value of 0.56, the reward from the oil factor is $0.10 \times 0.60 \times 0.56 = 0.0366$.

[57] In our experiments, $v_T = 30$ days, meaning that the portfolio volatility is calculated based on the last 30 days of portfolio returns. days. For example, if $v_T = 3$ and the three last portfolio returns are -1%, -2% and 1% the reward from this component is approximately -1.24%.

[58] If the volatility is not negated, then agents will try to maximise volatility.

[59] Holding cash only is a safe bet; the value do not change on a daily basis, so the volatility is zero.

Reward Name	**Description**	**Weights**				
		w_P	w_D	w_F	w_V	w_{PE}
MaxDailyReturn	Maximise the daily return of portfolio holdings. Conceptually it is similar to trend investors as in LeBaron (2002) and Farmer and Joshi (2001)	**0.60**	0.10	0.10	0.10	0.10
MaxDividendPayouts	Emphasises finding assets with high dividend payouts.	0.10	**0.60**	0.10	0.10	0.10
FactorFollower	Advocates holding a portfolio having large exposure to factors with high relative value. In other words, it attempts to construct a portfolio where the factor exposure of the portfolio matches the macroeconomic factors. For example, if the oil price is relatively high and the agent has a portfolio consisting largely of oil stocks, the reward will be high. Rapid portfolio reweightings based on changes int the macroeconomic environment is expected to happen frequently with this reward function.	0.10	0.10	**0.60**	0.10	0.10
MinDailyVolatility	Minimises the daily portfolio volatility. The optimal solution is to hold only cash[59]. However, other parts of the reward function will advocate holding at least some assets.	0.10	0.10	0.10	**0.60**	0.10
Combination	This kind of reward function consider all factors to be equally important. A good policy is therefore to maximise rewards with a reasonable volatility level, factor exposure, price level and dividend yield.	0.20	0.20	0.20	0.20	0.20

Table 3.3: Specification of various reward functions. The first column provide mnemonical names to the reward functions. The second column describe the function and the final column provides corresponding weights of equation (3.4.19). Most significant weight are written in bold.

Parameterisations are not limited to the examples provided in table 3.3. Researches stand free to introduce new reward functions as well as weights placed on different rewards. To illustrate the versatility of the parameterisation, consider the following: A commonly applied technique in portfolio optimisation is mean-variance analysis. The portfolio selection problem for agent i is to choose the optimal weights ψ_j of each asset j that maximise expected portfolio return for a given level of risk. More formally, each asset allocation weight ψ_j is chosen to minimise the target function (3.4.22):

$$\min_{w} \quad f(\psi; \lambda_i, \theta_i, H) = \lambda_i \psi^T Q \psi - (1-\lambda_i)\psi^T \mu \tag{3.4.22}$$

where Q is the variance-covariance matrix and μ the expected return. Subject to

$$\sum_{j=1}^{J} \psi_j = 1 \tag{3.4.23}$$

Which makes sure that all weights sum to one. The value of λ_i should be between 0 and 1. A high λ_i indicates high aversity for volatility, whereas a low λ_i accepts potentially higher volatility in order to get higher expected daily portfolio return.

Loosely speaking, the portfolio selection problem of mean-variance analysis can be translated to a combination of focus on rewards from daily return and volatility. We can write

$$\max_{\lambda} r = \lambda_i V(s_t) - (1-\lambda_i)P(s_t) \tag{3.4.24}$$

so that

$$w_V = \lambda \tag{3.4.25}$$

and

$$w_P = 1 - \lambda \tag{3.4.26}$$

where w_V and w_P are as defined earlier.

Chapter 4

Simulation Procedure

In this section the overall simulation procedure is presented. The simulation procedure is described as simple as possible. Algorithm 1 demonstrate an entire trading day, the distribution of dividends (if any) and issuance of projects for companies (if any). Algorithm 2 specifies at what price and quantity an agent decides to buy or sell shares in a stock (if any). Finally, algorithm 3 specifies how a portfolio reweighting is done. The algorithms are written to provide a high-level overview of what is going on during simulation. It is in no way complete. A lot of bookkeeping needs to be done and special circumstances needs to be considered[1]. We omit such details in favour of highlighting the major bits and bobs of the simulation procedure.

The algorithms use notation that is presented in previous chapters. As a recap, each agent i can trade in any asset $j \in \{1, \ldots, \text{J}\}$. The number of shares in asset j held by agent i is given by q_{ij}. The cash available is X_{iC}. Last transacted price of a share in asset j is given by P_j. Finally, a_i is a vector of decisions: $a_i \in \{\text{buy}, \text{sell}, \text{nothing}\}^n$ for all n assets. The current simulation day is given by t. A small time increment is given by dt.

Algorithm 1 Performs a single day of trading along with pre-trading and after-trading hours.

```
 1: procedure SINGLE_DAY_OF_TRADING([])
 2:     for $s \in sectors$ do                 ▷ Pre-hour trading; companies submit bids for projects
 3:         if $u \geq \lambda_s, u \sim \mathcal{U}(0,1)$ then
 4:             spawn new project that companies in sector $s$ can bid for.
 5:
 6:     for $n \in \{1, \ldots, \text{num\_rounds}\}$ do      ▷ Noise agents initially fill up the order book
 7:         for $i \in shuffle(noise\_agents)$ do   ▷ Shuffle so no agent consecutively gets to submit orders first
 8:             REWEIGHT_PORTFOLIO($i$)
 9:     for $i \in shuffle(learning\_agents)$ do
10:         Update observations for agent $i$
11:         REWEIGHT_PORTFOLIO($i$)
12:     for $n \in \{1, \ldots, \text{num\_rounds}\}$ do
13:         for $i \in shuffle(noise\_agents)$ do
14:             REWEIGHT_PORTFOLIO($i$)
15:
16:     for $j \in assets$ do                                  ▷ End of day, update prices
17:         Update closing price and volume for $j$
18:         if $j$ has earnings day and dividend $> 0$ then
19:             for $i \in learning\_agents \cup noise\_agents$ do
20:                 Let $d$ be the dividend per share
21:                 Let $q_{ij}$ be the number of shares agent $i$ has in asset $j$
22:                 Pay out $d \times q_{ij}$ to agent $i$
23:
24:     for $i \in learning\_agents$ do
25:         Calculate rewards for agent $i$
26:         Update observations for agent $i$
27:     return rewards, new_observations
```

Notes on the algorithm
Lines 2 - 4 randomly spawn new projects according to a Poisson distribution (as discussed in section 3.1.4). If a project

[1] For the interested reader, the complete code is available on https://github.com/mikaelkvalvaer/reinforced_rhapsody.git.

is spawned, companies can bid for it. The highest bidder undertakes the project. Next, noise agents do *initial_rounds* rounds of trading. The motivation is to fill up the orderbooks before letting the learning agents trade[2]. The agents trade in a random order each round. This ensures that no agent consistently trades first or last. For noise agents, this is not really important (since they only trade at random so the order does not matter). For learning agents the trading order might be important on days of earnings announcements and dividend payouts. An agent that consistently is able to trade first could get an unfair advantage. Finally, on lines 12 - 14 the noise agents perform more trades. The motivation is to provide more liquidity and let orders placed by learning agents transact if they have not already done so.

Lines 16 - 27 are done at the end of the day when trading is finished. Closing prices and trading volumes are updated. The new data is fed to the observation space for each agent on lines 26 and 10[3]. Additionally, we check whether each asset has earnings day. If they do, and if they pay out dividends, agents holding shares will receive dividends proportional to the number of shares they own. Finally, new observations and rewards are calculated for each learning agent. The optimisation algorithm collects the experience and uses the experience during training as discussed in section 3.4.4.

Algorithm 2 Agent i buys, sells or does nothing on asset j

```
 1: procedure DO_ACTION($i$, $j$, $decision$, $t$)
 2:     $P_j \leftarrow$ the last price for a share in asset $j$
 3:     if $decision$ = buy then
 4:         $P_{bj} \leftarrow$ random number drawn from $\mathcal{N}(\mu_o P_j, \sigma_o P_j)$      ▷ Buy price per share
 5:         $q_{\max} \leftarrow \lfloor \frac{X_{iC}}{P_{bj}} \rfloor$      ▷ Maximum possible shares to buy
 6:         if $q_{\max} \geq 1$ then
 7:             $q_j \leftarrow$ random number drawn uniformly from $\{1, \ldots, q_{\max}\}$
 8:             Place a buy order of $q_j$ shares on asset $j$ for $P_{bj}$ per share
 9:     else if $decision$ = sell then
10:         $P_{aj} \leftarrow$ random number drawn from $\mathcal{N}(\frac{1}{\mu_o} P_j, \sigma_o P_j)$      ▷ Sell price per share
11:         $q_{\max} \leftarrow q_{ij}$      ▷ Maximum possible shares to sell
12:         if $q_{\max} \geq 1$ then
13:             $q_j \leftarrow$ random number drawn uniformly from $\{1, \ldots, q_{\max}\}$
14:             Place a sell order of $q_j$ shares on asset $j$ for $P_{aj}$ per share
15:     Update holdings and cash available for agent $i$
```

Notes on the Algorithm

Decision is one of $\{\text{buy}, \text{sell}, \text{nothing}\}$. Lines 3 - 8 consider the buy situation. A price is selected randomly close, but typically above, the most recent price P_j. Next, the quantity is chosen as a number between 1 and $q_{\max}$ and an order is placed. No order is placed in case the agent cannot afford to buy any shares . A similar description holds for the sell case (lines 9 - 14). The maximum number of shares agent i can sell in asset j is the number of shares it holds, X_{ij}. The price is randomly set close, but typically below, the most recent price P_j similar to (Marchesi et al. 2003).

Algorithm 3 Agent i performs a portfolio reweighting

```
1: procedure REWEIGHT_PORTFOLIO($i$, $t$)
2:     $a_t \leftarrow$ action matrix for agent $i$ on day $t$      ▷ Equation (3.4.15).
3:     $current\_time \leftarrow t$
4:     for $j \in shuffle(\{1, \ldots, J\})$ do
5:         $decision \leftarrow$ realisation from $\{\text{nothing}, \text{buy}, \text{hold}\}$ with probabilities $\{p^j_N, p^j_A, p^j_B\}$      ▷ Equation (3.4.17)
6:         DO_ACTION($i$, $j$, $decision$, $current\_time$)
7:         $current\_time \leftarrow current\_time + dt$      ▷ $dt$ is a small unit of time
```

Notes on the Algorithm

Line 2 fetches the actions by agent i on day t . Lines 3 and 7 specify a *current_time* variable; all orders are submitted at a unique time. Consequently, if two buy (sell) orders are placed with the same bid (ask) price, the order placed first will be prioritised when matching orders[4]. Line 4 selects each of the J assets in random order. An example situation motivates why shuffling is necessary: Suppose $decision$ = buy for assets $j = 1, \ldots, J$. The DO_ACTION procedure places a random order based on how much cash agent i has available. The more cash agent i has, the larger the order size is on average. Consequently, if the order of actions is not initially shuffled asset 1 will, on average, have larger bids (measured in order size) compared to assets $2, \ldots, J$. The following relationship holds: $\mathbb{E}[q_1 P_{1b}] \geq \mathbb{E}[q_2 P_{2b}] \cdots \geq \mathbb{E}[q_n P_{bn}]$. where P_{jb} and q_j is the bid price and quantity, respectively. Shuffling is done to avoid this effect.

[2] Recall from section 3.3.1 that the purpose of noise agents is to provide liquidity to the market.

[3] We update the last price observation for agent i before it selects an action.

[4] Recall the matching procedure from section 3.2.3.

Chapter 5

Configuration of Model

This chapter describes the inputs used to generate the results. We motivate the different sectors and their exposure to different factors as in contribution (ii) in section 5.1. The different factors used in the factor framework is presented in section 5.2. Finally, the agent composition and various other simulation parameters are presented in section 5.3.

5.1 Sectors

We design three sectors: banking, oil and retail. The sectors have different exposure to the factors, summarised in table 5.1. A positive correlation between the sector and the factor indicate that an increase in the factor yields higher income in the sector and vice versa. The sectors are chosen arbitrarily and there are no empirically data behind the correlations. We motivate the relationships intuitively:

(i) The banking sector has a positive correlation to the interest rate as their profits are closely connected to the net interest rate margin[1]. Increasing oil prices does only influence the financial sector to a small degree. A growth in the domestic product signifies a growing economy, increasing the lending provided by banks.

(ii) Oil is influenced negatively by increasing interest rates as investments become more expensive. An increase in the oil price directly increases the profits. Growth in domestic product increases the companies' revenues as the economy consumes more products[2].

(iii) Retail is negatively connected to the interest rate as higher interests make the consumers' disposable income fall. Consumption of retail goods consequently fall. Retail is negatively connected to the oil price as an increasing oil price make transportation more expensive. A growing economy, i.e. a growth in the domestic product, increases consumption. Retail is therefore positively connected to GDP growth.

5.2 Factors

As the model is designed to evaluate different economic policies, one has the ability to define the current economic conditions. Contribution (ii) defines a factor dependency framework for investments which drives the economy. The factors are essential to how profitable the companies become. In our factor framework we have three economic factors:

(i) **Interest rate**
The interest rate affects the debt schedules of section 3.1.5. Jumps in the interest rate render debt payments more expensive or cheaper for the companies. The interest rate does also affects the cash flows as described in section 5.1.

(ii) **Oil price**
In real financial markets, the oil price is one of the fundamental drivers of the global economy as one of the

Sector	Oil price	Interest rate	GDP growth
Oil	$+0.80$	-0.30	$+0.20$
Banking	$+0.20$	$+0.70$	$+0.30$
Retail	-0.20	-0.40	$+0.50$

Table 5.1: Earning sensitivities to various macroeconomic factors.

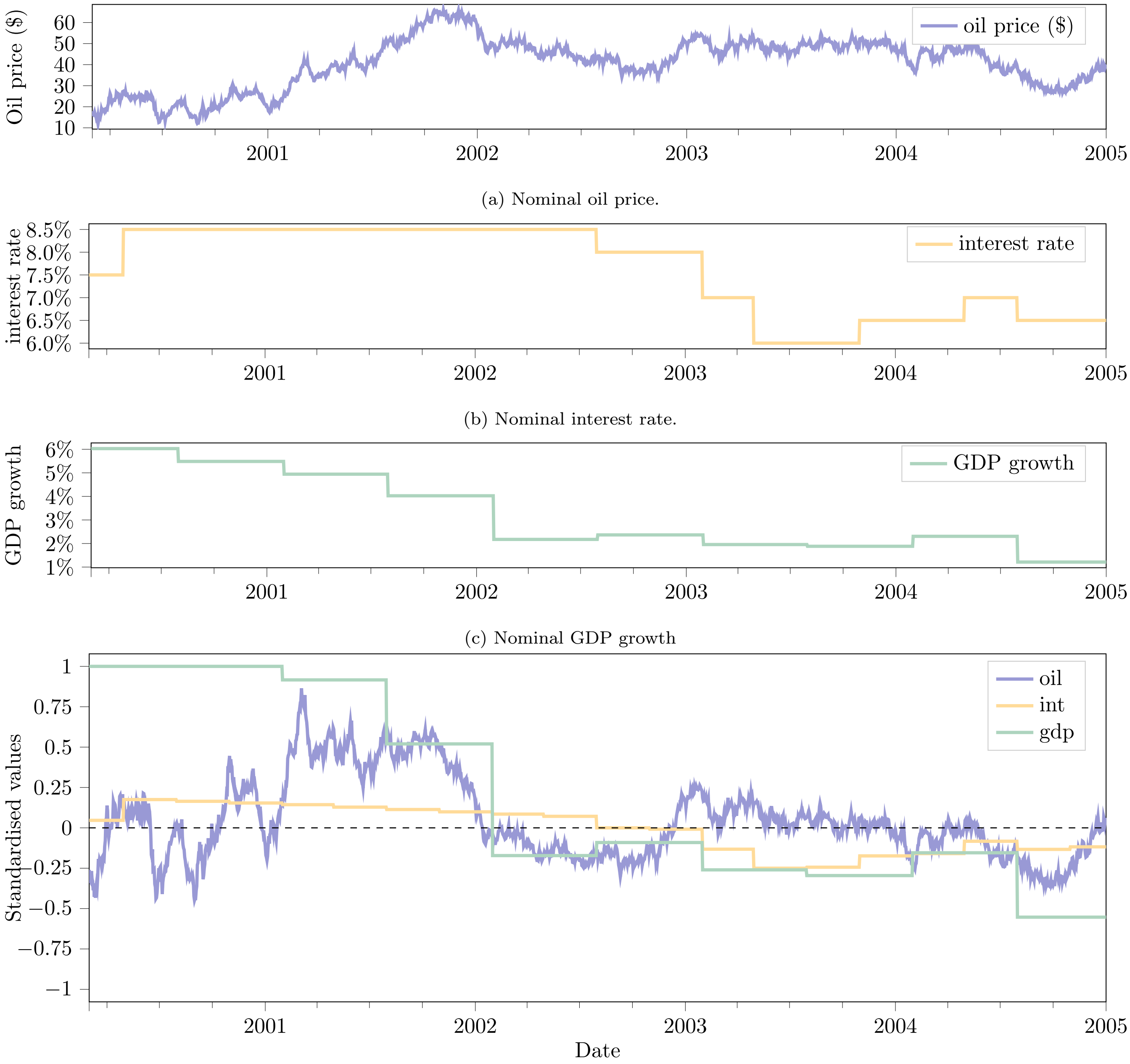
[1] The difference between their lending and borrowing rate.

[2] We assume that the environment depends on fossil fuel.

primary commodities.

(iii) **Gross domestic product growth**
The growth in gross domestic product signifies how much the economy is growing as a whole. A growth is typically associated to favourable conditions, growing economy and attractive investment environments. A decline is commonly associated to economic stagnation and less favourable conditions.

Table 5.1 shows the ranges for the economic factors of the model run. Similarly, figure 5.1 shows the time-series of the factors in both nominal values as well as their standardised counterparts.



(a) Nominal oil price.

(b) Nominal interest rate.

(c) Nominal GDP growth

(d) Standardised time-series used in the model.

Figure 5.1: Conversion from absolute values to standardised values.

Name	w_P	w_D	w_F	w_V	w_{PE}	**Number of agents**
MaxDailyReturn 1	0.20	0.20	0.20	0.20	0.20	5
MaxDailyReturn 2	0.30	0.175	0.175	0.175	0.175	5
MaxDailyReturn 3	0.40	0.15	0.15	0.15	0.15	5
FactorFollower 1	0.20	0.20	0.20	0.20	0.20	5
FactorFollower 2	0.175	0.175	0.30	0.175	0.175	5
FactorFollower 2	0.15	0.15	0.40	0.15	0.15	5
MinDailyVolatility 1	0.20	0.20	0.20	0.20	0.20	5
MinDailyVolatility 2	0.175	0.175	0.175	0.30	0.175	5
MinDailyVolatility 3	0.15	0.15	0.15	0.40	0.15	5
MinDailyVolatility 4	0.10	0.10	0.10	0.60	0.10	5
noise	-	-	-	-	-	10

Table 5.2: Agent composition for experiments in sections 6.2.2 and 6.2.2.

5.3 Simulation Parameters

The agent composition is shown in table 5.2. If not otherwise specified, the simulation run from January 1st 2000 to December 31st 2001, in total 2 years of trading. The first 3 months are typically not displayed. The reason is that the equilibrium[3] price is not known when the simulation begins[4]. Therefore, the prices fluctuate significantly the first couple of the simulation days before they stabilise. 24 simulations are run in parallel collecting the experience. For training, 32 cores and a Tesla P100 GPU are used. We trained the agents for 72 hours. For each training episode, a new set of macroeconomic time-series was generated. For the complete set of parameters used in the simulation, see appendix A.

[3] That is, a sound starting price for the stock where there are no large discrepancies in either demand or supply.

[4] For example, if the standardised macroeconomic factors are all negative, the prices typically fall sharply.

Chapter 6

Discussion

This chapter discusses the results of the model. We start by testing the model's validity and robustness. Testing for validity is done by evaluating whether the model demonstrates the stylised facts of financial markets as given in appendix B. This part evaluates whether goal (i) capture stylised facts is achieved[1]. We asses the model's robustness by altering the agent population. This includes model runs populated solely by learning agents as well runs with no learning agents. We show that an environment consisting of only noise agents generates a random walk price process. In section 6.2 we conduct experiments indicating that the the agents are indeed learning to maximise their reward function. Evaluating the agents' ability to learn is important as a step in validating the model design. We conclude this chapter by performing scenario analysis, as discussed in 1. We demonstrate the following experiments:

1. **Replicating the stock market from 2006 through 2010 from real time-series**
   We carry out a simulation using the real interest rate, the oil price and and growth in domestic product[2]. The model display similar dynamics as observed in the real market.

2. **What-if analysis of the financial crisis**
   We assess the implications of a more rapid change in the key policy rate. The scenario is inspired from the changes in interest rate by the Federal Reserve[3] in the same time period. We investigate a scenario where Norges Bank responds more quickly to the changing economy and adjusts the key policy rate in May 2008 instead of January 2009. Our experiments indicate that the crisis was inevitable; however, the magnitude of the crisis appears smaller.

6.1 Validation and Dynamics

This section tests the model's validity and dynamics when the agent composition is changed. The first part of the section discusses the descriptive statistics from a representable run with a composition of different learning agents[4]. We conclude the section by assessing the model's dynamics. Dynamics is determined through changing the agent composition and investigating the corresponding model behaviour.

6.1.1 Validation

The primary goal of the model is to exhibit the stylised facts of financial markets (goal (i)). Demonstration of the stylised facts is essential for the model to behave realistically. Contributions (i)-(ii) are designed to increase the realism of the model of Bjerkøy and Kvalvær (2018). This section combines results obtained quantitatively through descriptive statistics and qualitatively using plots from individual runs of the model. We assess the presence of each of the stylised facts given in appendix B.

Readers of this thesis should bear in mind that individual price series has no meaning viewed out of context. The model attempts to capture dynamics between assets, sectors and different debt profiles in an constantly evolving macroeconomic environment. The nominal values for each price series (i.e., the daily price for the stock) has no interpretation[6]. Plots used to validate the model are usually shown for an aggregated index. We construct an equal-

[1] It is important to test whether the results produced by the model are meaningful and demonstrate the same properties as real financial markets for several reasons. Most importantly, if the model should be a contribution to the research field it has to be valid. Secondly, in order to perform scenario analysis in a meaningful way, the model dynamics should be representable of real financial markets.

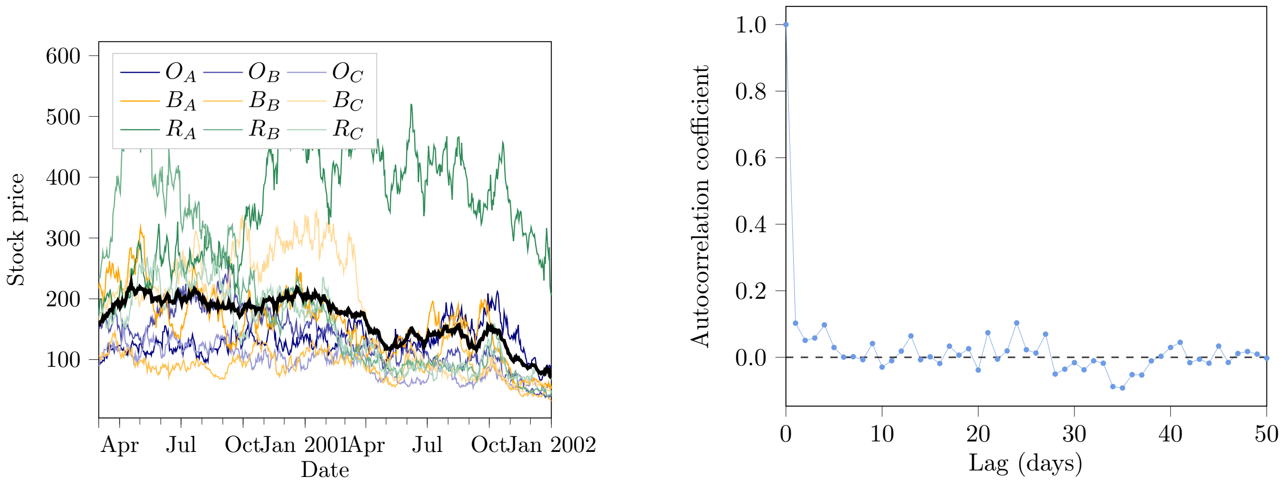
[2] The interest rate is the Norwegian key policy rate and Norway's growth in domestic product.

[3] The US Central bank.

[4] The model is run several times over. A representable run is chosen from the subset of runs when we group the runs on the descriptive statistics.

[5] Annualised volatility is found by multiplying the daily volatility by $\sqrt{365}$. In our model, the stock market is open every day of the year instead of the usual 270.

[6] As we have no notion of monetary value in the model.



(a) Equal-weighted price index in black with the underlying individual price processes. Stocks are grouped on sector. The subscript indicates the stock's debt profile from section 3.1.5.

(b) Autocorrelation in returns with lag-order $\tau = [1, 50]$ for an equal-weighted index.

Figure 6.1: Price-series and autocorrelation in returns for lag-order $\tau = [1, 50]$ for an equal-weighted index.

weighted index by averaging each of the prices. Consequently, each stock carry equal weight when we assess dynamics. We use an equal-weighted index instead of a market-capitalisation (MCAP) weighted as we want to report how the economy performs as a whole. However, when one uses the model for scenario analysis and provides real data it is sometimes more natural to use an MCAP-weighted index. We circle back to this remark in the experiment section of this chapter.

Absence of Autocorrelation

The model does not exhibit autocorrelation in returns. Figure 6.1b shows the autocorrelation for different lag orders. One can observe the returns to display close to no autocorrelation for different lag orders, except for the first observation[7]. The same conclusion can be reached utilising the data in table 6.2. Table 6.2 shows the ACF-values, Q-statistics for the Ljung-Box test and the corresponding P-value for the test[8]. Except for lag-order $\tau = 10$, the test suggests that the data is independently distributed on a confidence level $\alpha = 95\%$. As in Christoffersen (2011), we will take this as evidence for a close to constant conditional daily mean.

According to Cont (2001), the absence of autocorrelation is a key trait of financial markets. Returns in prices do not exhibit autocorrelation in real financial markets (Cont 2001). Any presence of autocorrelation in returns could be utilised by a trader and open up for arbitrage situations. A model attempting to create an artificial stock market should therefore show no autocorrelation in returns. As the agents of our model are able to learn and adapt to their environment, autocorrelation in returns would have been discovered and utilised. Interestingly, the learning ability of the agents naturally prevent returns from exhibiting autocorrelation. If the returns indeed exhibited signs of autocorrelation, the agents would discover this phenomenon. A good strategy would be to buy and hold the stock; not sell it (if the price appreciates and autocorrelation is positive). No learning agent would therefore supply the stock. In this situation, supply is provided by noise agents only. A system populated by an overweight of noise agents will generate a random walk price process[9]. Contribution (i) learning agents effectively limit autocorrelation in returns,

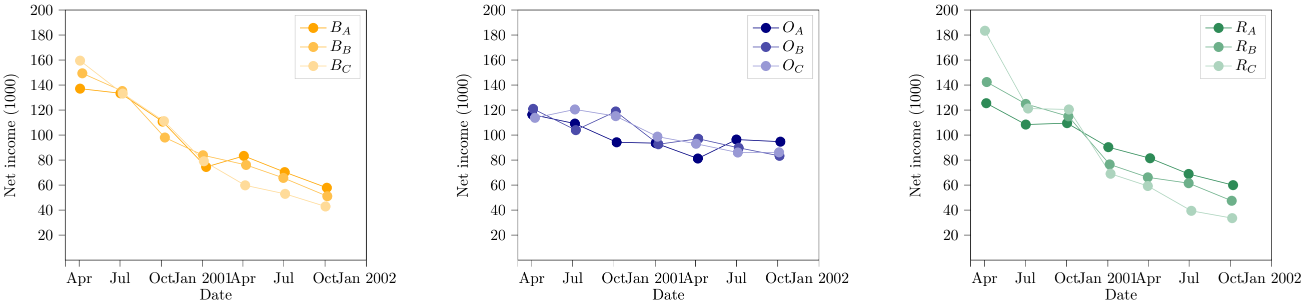
[7] This is the correlation between the time-series and itself, by definition it equals 1.

[8] Ljung-Box tests whether the autocorrelation of a time-series is statistically different from zero (Christoffersen 2011). It tests the overall randomness of the time-series; not for specific lags.

[9] We show this is section 6.1.4

Statistics	Value
Daily mean	-0.00118
Daily volatility	0.0186
Annualised volatility	0.3568[5]
Skew	-0.3274
Kurtosis	5.602
Min	-0.1176
Max	0.0668
# observations	671
Jarque-Bera	197.1

Table 6.1: Descriptive statistics for an equal-weighted index of daily returns for the model run.



(a) Quarterly net income for companies B_A, B_B and B_C in the banking sector during the simulation.

(b) Quarterly net income for companies O_A, O_B and O_C in the oil sector during the simulation.

(c) Quarterly net income for companies R_A, R_B and R_C in the retail sector during the simulation.

Figure 6.2: Quarterly net income for grouped by sector. Observe the negative earnings pattern for all sectors primarily driven by declining growth in GDP.

supporting the usefulness of the contribution.

Absence of autocorrelation is widely viewed as support for the efficient market hypothesis (Cont 2001). Asset prices in our model are fully order-driven. At any time, the price of the stocks are determined through supply and demand. Contribution (i) on learning agents limits autocorrelation in returns, consequently making the market efficient.

Heavy Tails

The model produces a return distribution with heavy tails. One can affirm the heavy tails property using figure 6.4b which shows a histogram for the return in prices with a normal overlay. Combined with descriptive statistics shown in table 6.1 one can confirm the distribution to be leptokurtic. A leptokurtic return distribution is consistent with empirical data from real financial markets (LeBaron 2006) as described in appendix C.1.

The presence of heavy tails is an important characteristics of real stock markets. There is a non-negligible probability of extreme events. Large market movements happen too frequently to be discarded as simple outliers (Cont 2001). In real markets, there are numerous factors contributing to large market movements. Human behaviour, macroeconomic incidents and performance of individual companies are some examples. The factor-dependency framework we introduce in contribution (ii) is designed to induce these movements. The factors are drivers of the overall economy, influencing the earnings of the companies. Extreme observations occur when the market regimes changes dramatically.

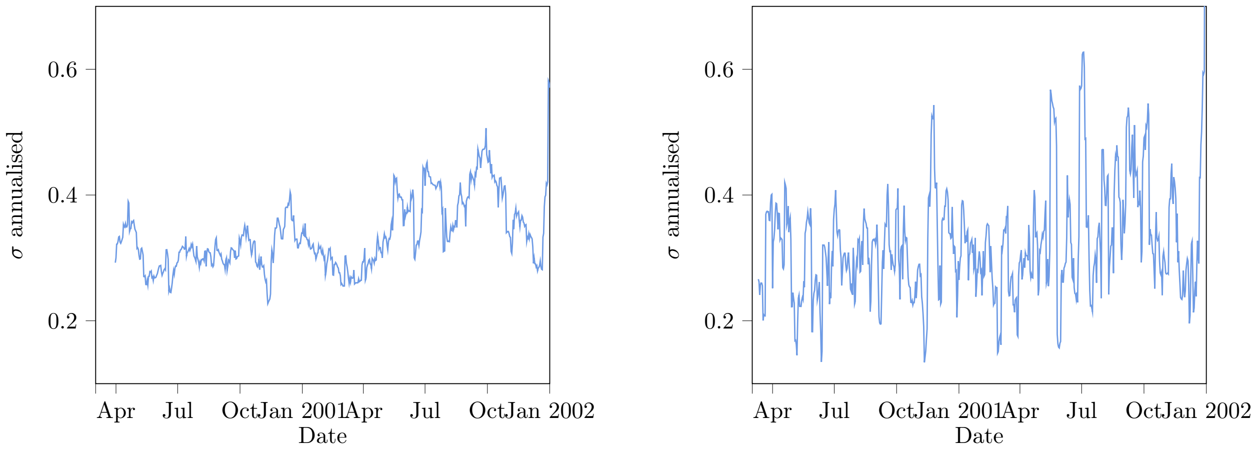
Figure 6.5 shows the prices and the factors as time-series. One can observe from the stock that some stocks exhibit large jumps and falls after the macroeconomic environment changes. One such example can be asset R_C in Q1-2001 after GDP experiences a drop. We view this as support for contribution (ii) factor-dependency framework. From figure 6.5 one can observe intuitive relationships between the factors and the price changes. For example, one can see that the oil price begins to appreciate. In the subsequent time period, oil stocks appreciate as well. On the other hand, when the interest rate starts to increase, the market as a whole is affected negatively. Similar to real stock markets, access to debt becomes relatively more expensive for the stock market. This materialises as lower earnings for the companies. On the agent side, an increased interest rate yields higher interests on cash positions. Stocks become relatively less favourable. The fact that agents are able to discover these relationships, observe the economic factors and transform the observations to trading actions is taken as further justification for contribution (i) learning agents.

Gain / Loss Asymmetry

The model produces a negatively skewed distribution; suggesting that the left tail is longer than the right. Negative skew is demonstrated by table 6.1. This property satisfies (iii) gain/loss asymmetry as one does not observe equally large up-moves as down-moves in price returns. We believe this to be an interesting property of our model. Negatively skewed return distributions are found in real stock markets (Cont 2001). Commonly, fear induced by negative news and overreactions are quoted as the main reasons for negatively skewed return distributions in literature (Damodaran et al. 2013).

Lag order	ACF-value	Q-statistics	P-value
1	0.059	2.35	0.13
10	-0.064	22.50	0.01
30	-0.001	36.97	0.18
50	0.024	50.04	0.47

Table 6.2: Table of autocorrelation values, Q-statistics (Ljung-Box test) and the corresponding P-value.



(a) Rolling 30-day volatility for an equal-weighted index.

(b) Rolling 10-day volatility for an equal-weighted index.

Figure 6.3: Rolling window volatility for an index comprised of equal weights.

From table 6.1 one can observe the largest negative return to be greater than the highest increase, adding further evidence to the gain/loss asymmetry[10]. From real financial markets, market crashes are typically much more dramatic compared to the days with the largest increase. The same pattern can be observed in the model; some days are characterised by large declines not matched by an equally large increase (in relative terms, i.e., percent).

We attribute the negatively skewed return distribution to contribution (iii), on investment philosophies. Some agents are rewarded for withdrawing from the market in the event of large increases in volatility. Agents which attempt to maximise daily return will impact the supply and demand equally in both large increases and decreases. Volatility-minimising agents, however, will impact individual stocks more when the share prices are falling than when they are increasing. We provide two explanations:

1. When an agent discover a change in the economic factors and believe the change to influence stocks positively, it has a whole sector to spread its exposure on. It will generate demand for all the stocks in the sector and the prices may appreciate. In the event of a negative observation, it can only generate supply for the stocks it holds. Positions are not equal in size. On a sector level, the agent can hold a relatively larger position in one of the stocks. This causes excess supply, driving the price of individual stocks down.
2. The agents have limited cash available for buying. Normally, agents are encouraged to hold stocks through their reward policies. When they discover a positive signal, they must choose between reweighing or maintain their current positions due to cash restrictions. In the event of a negative signal, they can liquidate their positions immediately (dependent on demand). Noise agents will absorb some of the supply, but prices will ultimately be driven down.

Aggregational Gaussanity

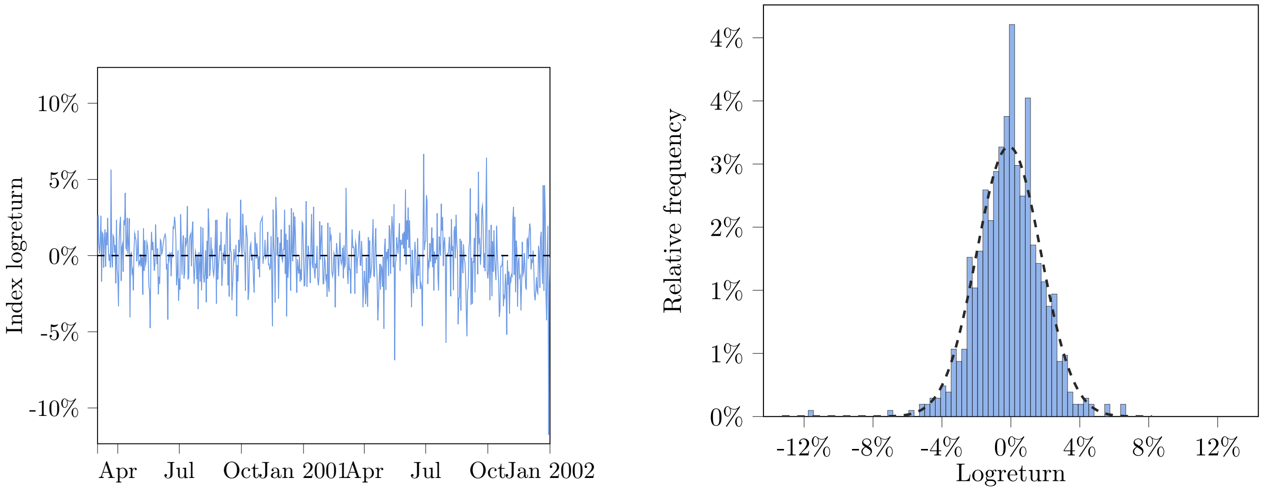
We observe a return distribution similar to the normal distribution, although skewed and with fatter tails. According to Cont (2001), the daily return distributions of stock markets approach the normal distribution over time. The model successfully demonstrate this property. Table 6.1 indicates that the average daily return of the equal-weighted index is close to zero. This is consistent with empirical findings as demonstrated in appendix C showing the empirical distribution of the OSEBX-index. From the histogram in figure 6.4b one can observe the distribution to approach the Gaussian distribution. The normal overlay, combined with the QQ-plot in figure 6.6b suggest that the model satisfies the aggregational Gaussanity property. Although, one can observe significant different tail behaviour, especially in the left tail. The interpretation of the deviation found in the left part of figure 6.6b is a fat tailed, negatively skewed distribution (Alexander 2009). This strengthens the evidence for (ii) heavy tails and (iii) gain / loss asymmetry property.

Volatility Clustering

The model produces time-varying volatility with occasional clustering. The concept of volatility clustering relates to periods of occasional consecutive peaks. Figure 3.2b shows the rolling volatility for the equal-weighted index. One can observe from the figure that the volatility is evolving around 30% annualised but has times of both increased and decreased volatility.

The spikes in volatility can connected to changes in the economic drivers (macroeconomic factors). One such example can be seen around Jul-2001 when GDP drops. In the following time period the volatility increases. We believe the rationale to be two-fold:

[10] For an excellent source of interpretation of descriptive statistics and return data, see Alexander (2009).



(a) Logreturn time-series for the equal-weighted index.

(b) Histogram of logreturns with normal distribution overlay.

Figure 6.4: Logreturn time-series for the equal weighted index and histogram of the returns.

(i) Changes in economic drivers cause the earnings of the companies to change. This causes a delayed effect. Earnings are only realised at certain time steps for each company. When the realisation occurs, the agents updates their input vector presented in section 3.4.5. If the change in earnings has been sufficiently large, the agents can adjust their appetite for the stock significantly. The large change in demand causes a price appreciation or depreciation. We believe one such example of a delayed effect can be seen in stock R_C late Q1-2001. The factors changed in the time period before the earnings realisation (medio Q1), later influencing the earnings of the company. Figure 6.3b shows a short-term temporal increase in the 10-day rolling volatility.

(ii) The agents observe a change in the macroeconomic factors. From past experience they have learned that this will affect both the future income of the companies and asset prices. Figure 6.5 supports this claim: when one of the macroeconomic factors changes, some agents reweight their portfolio immediately. This can be seen around Q1-2001 in figure 6.5

The agents' ability to learn from past experience and adapt to the current environment justifies contribution (i) learning agents. From real financial markets we know that the market is constantly evolving and adapting to new information. Models of artificial markets should therefore aim to mimic this procedure. Additionally, company earnings driven by some fundamental macroeconomic factors are able to explain temporary changes in volatility. We conclude that both contribution (i) about a learning agent and contribution (ii) help to explain temporal clusters in volatility.

Notes on Validation

This section has provided justification for the contributions made in section 1. However, we do not provide support for contribution (iv) on debt. The contribution is in no way irrelevant for the dynamics we observe in the model. It is, however, interwoven with contribution (ii) on a factor-dependency framework. Consequently, it is difficult to show exact situations where the contribution is valuable. The contribution exposes the companies in the model to changes in the interest rate. The interest rate is a potential economic factor in our model. This makes it difficult to assess which of the contributions that induces the dynamics we observe. We have focused on the contribution (ii) factor-dependency framework as one of our goals is to perform scenario analysis[11] (goal (ii)). With that said, we expect considerable amount of dynamics to be lost if the model did not offer a direct relationship between the interest rate and the company stability as offered by contribution (iv).

6.1.2 Order-Driven Dynamics

The model produces order-dynamics similar to observed behaviour from real financial markets. Earlier attempts of creating artificial stock markets have typically been limited to single-asset markets with a risk-free bond (LeBaron 2006). Although these models are analytically more tractable, important dynamics between assets, sectors and macroeconomic factors are lost. Contribution (ii) on a factor-dependency framework is designed to alleviate these challenges. Founded in LeBaron (2006)'s criticism of single-asset markets, Bjerkøy and Kvalvær (2018) constructed an order-driven environment composed of several assets. The price processes are fully driven by aggregated agent behaviour and not influenced by a central entity such as a market-maker. The trading framework in this thesis is similar to Bjerkøy and Kvalvær (2018), but the agents are different. In this section we provide details on the observed order-dynamics and compare it to empirical data.

[11] Recall that it is the factors that ultimately open up for scenario analysis.

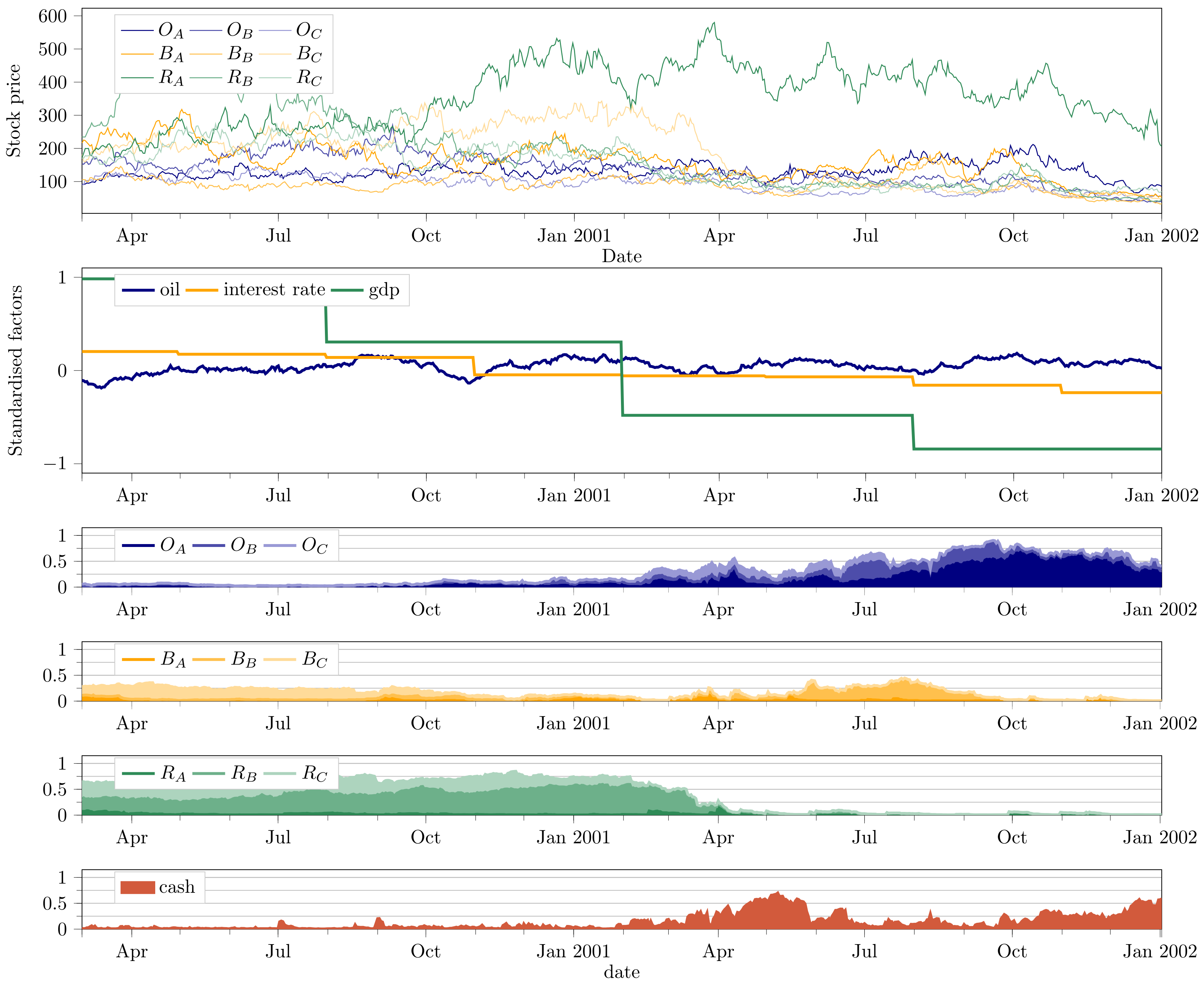
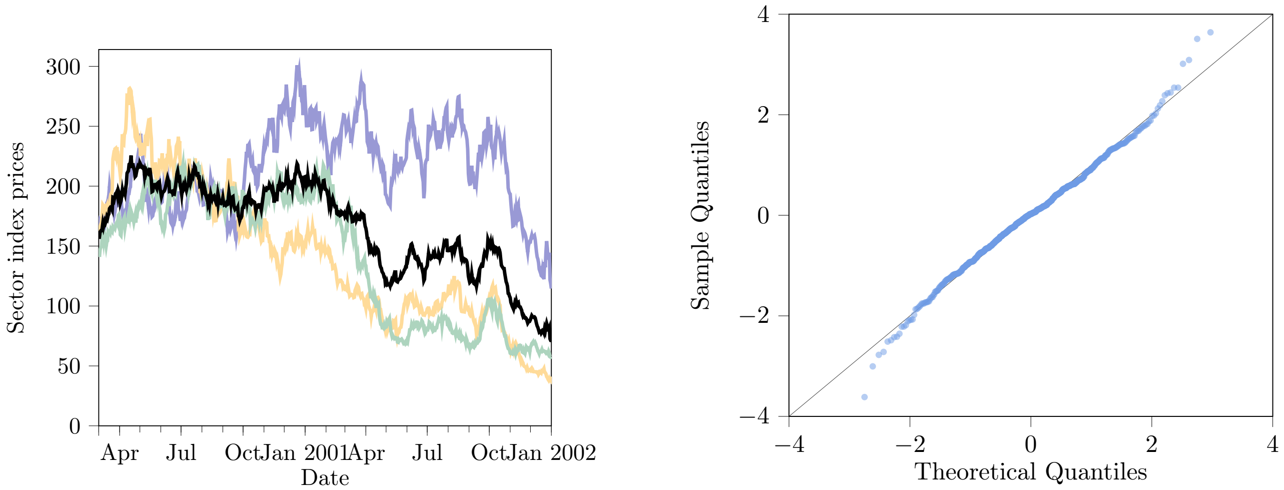


Figure 6.5: From top to bottom chart: share prices; standardised macroeconomic factors; portfolio holdings in the oil price sectors; portfolio holdings in the banking sector; portfolio holdings in the retail sector; cash holdings (all for a single agent). All portfolio holdings are in fraction of total portfolio value.

.



(a) Sector price indices.

(b) QQ-plot showing the ratio of the quantiles of the cumulative probability distribution for the log returns compared to a standardised normal distribution.

Figure 6.6: Sector price indices and QQ-plot for the index return distribution.

The volume dynamics of an index for each sector are shown in figure 6.7a. When we examine the plot in combination with the price series in figure 6.1a, three insights can be realised:

(i) **The model generates time-varying turnover**
Although it is standard to show trading volume in financial agent based models, we show turnover. We use turnover as one can naturally expect higher trading volume (in shares) if the stock price declines[12]. By using turnover one can compare activity levels in absolute terms. One can observe that the turnover decreases as the stock prices increases. We believe this effect to be traceable to the fact that agents gradually perceive the stock to be overvalued. Consequently, fewer agents attempt to buy the expensive stock.

(ii) **Tendencies to sector dependent turnover**
One can observe from figure 6.7a that the agents are able to produce time-varying trading volume and endogenously drive the price process for different sectors. Although the effect appears to be small, the presence of different climates in different sectors is important. In real financial markets one periodically observes bubbles and crashes in individual sectors (LeBaron 2011a). Examples include oil stocks during the sharp decrease in oil price as of 2014, and technology stocks in the Dot-com bubble of 2000. A realistic artificial stock market model should therefore to the least show tendencies to sector-dependent turnover.

(iii) **Major increases or decreases in the short term prices are driven by temporal disequilibriums in supply and demand, indicated by lower turnover**
From figure 6.5 showing earnings one can observe this to be particularly true for situations where the earnings process is unstable. For example, notice the sharp price decline in oil stocks during Q2/Q3-2000, correlated with a oil price decline in the same period. Unstable earnings correlated with price declines are similar to the findings of Garman (1976) and Farmer and Joshi (2001).

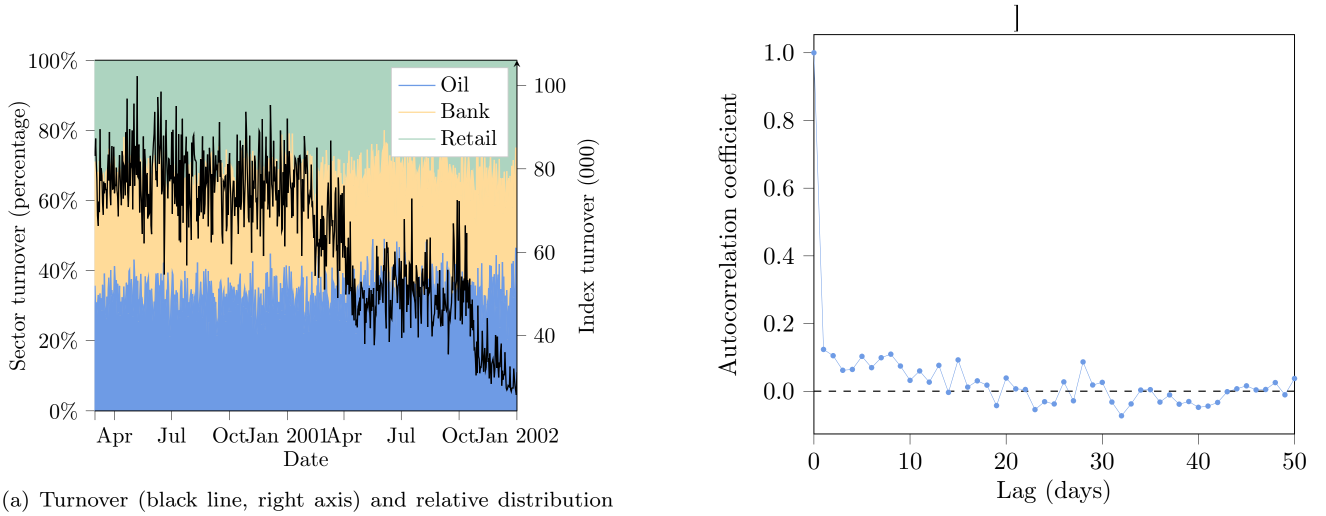
As one can observe in figure 6.7a, trading volume is correlated with the macroeconomic environment. With reference to LeBaron (2006)'s criticism of single-stock markets, we conclude that contribution (ii) factor-dependency framework for company earnings adds value to models for artificial stock markets. Even though the overall market climate can be healthy, one can observe bubbles and recessions in individual stocks and sectors in real financial markets. As shown in figure 6.6a, there might be some sectors that perform differently than the overall market. One such example can be the banking sector during December 2000. Large and sudden changes in the macroeconomic environment alters the driving economic factors and cause bubbles and steep decreases in the price.

Time-varying sector dependent trading volumes is an important feature when it comes to policy analysis. Changes in macroeconomic factors will influence sectors differently. A sound framework for investigating relationships between the stock market and the factors which drives the economy is quintessential for the validity of the experiments carried out in section 6.3. As the model exhibit the stylised facts of financial markets and exhibit promising order dynamics, we believe the model to be well-suited for policy analysis.

6.1.3 Dynamics

We change the agent population and investigate the corresponding model dynamics. The model should behave similarly when changing the number of agents, although one should expect different dynamics when the number of agents is

[12] For a given level of wealth, the agents can purchase more shares when the price is low.



(a) Turnover (black line, right axis) and relative distribution between each sector (left axis).

(b) Autocorrelation in trading volume with lag-order $\tau = [1, 50]$ for an equal-weighted index.

Figure 6.7: Trading volume grouped on different sector, total turnover and autocorrelation in trading volume

very low. Additionally, a run comprised of noise agents only should yield different dynamics. Noise agents perform trades at random, consequently we expect behaviour similar to a random walk. The general observation from these experiments is that increasing the agent population size increases the realism of the model. A larger population, both in terms of noise agents and learning agents, exhibit return distributions similar to the stylised facts of financial markets.

6.1.4 Noise Agents Only

Noise agents are agents which utilise no strategy. They have an equal probability of selling and buying, dependent on owning the stock. On average, the total demand and supply generated by a system comprised of noise agents only should be zero. On specific days there may be temporal differences in supply and demand causing fluctuations in the price. Over time one expects the dynamics of a system with only noise agents to be similar to a random walk process (Bjerkøy and Kvalvær 2018).

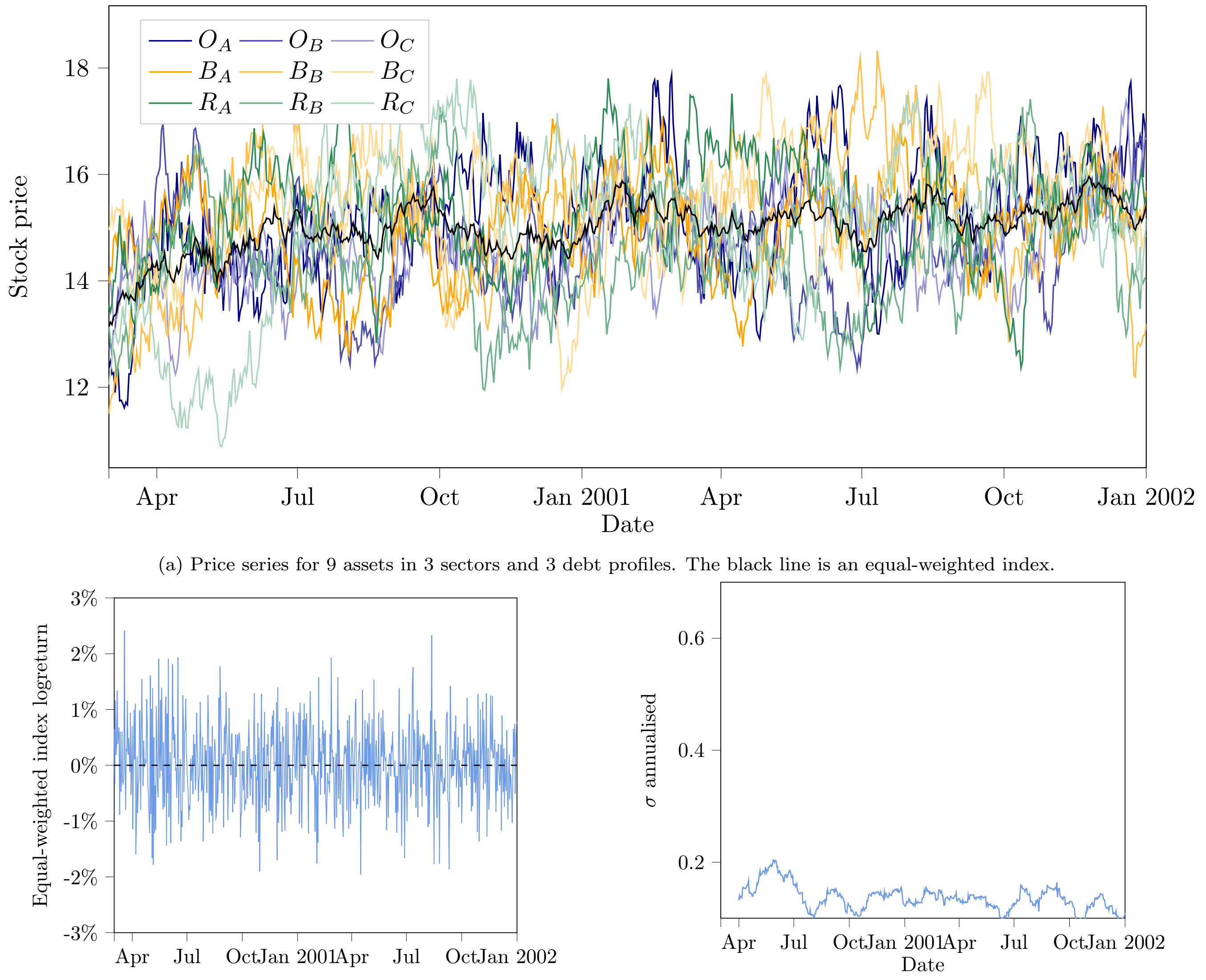
Figure 6.8a shows the price process for an equal-weighted index. The logreturn process is shown in figure 6.8b. One can observe that there are no major increases or decreases in the price process. Neither does the logreturn process exhibit any successive period of increased return. If noise agents indeed produce a random walk process, this is expected. Consequently, we take these observations as evidence for the randomness in price dynamics induced by noise agents. Although the price process appear to be overall increasing, this can be attributed to the fact that the overall wealth in the system is increasing. As earnings are realised, stocks pay dividends. These dividends are distributed to the stock owners, making them more wealthy. Increased wealth drives the price of individual stocks (Bjerkøy and Kvalvær 2018).

Figure 6.8c shows the rolling 30-days average volatility for a simulation with only noise agents. The volatility appears to be constant with no persistent period of heightened volatility. This is consistent with expected behaviour as noise agents have an equal probability of buying and selling. The volatility is significantly lower compared to the normal run. As noise agents trade at random, there is no reason to believe a stock to experience continuous periods of increased volatility.

Table 6.3 shows the descriptive statistics for a noise-only run. Similar to the results in section 6.1.1, one observes a daily mean around zero. This is consistent with the hypothesis of a random walk process. We observe a lower annualised

Statistics	**Value**
Daily mean	0.000
Daily volatility	0.007
Annualised volatility	0.136
Skew	0.005
Kurtosis	3.1578
Min	-0.025
Max	0.026
# observations	3,593

Table 6.3: Descriptive statistics for an equal-weighted index of daily returns for the a run consisting of noise agents only.



(a) Price series for 9 assets in 3 sectors and 3 debt profiles. The black line is an equal-weighted index.

(b) Logreturns for an equal-weighted index.

(c) 30-days rolling volatility for an equal weighted index.

Figure 6.8: Model run with $N_N = 40$ noise agents and no learning agents.

volatility. If the agents utilise no strategy, they buy and sell with equal probability. They make no observations which should yield a significant surplus in either demand or supply. On average, one expects equal amounts of supply and demand which will neutralise each other. The net supply or demand will be low as a fraction of total trading volume on any given day. Lower volatility is therefore expected. Both skew and kurtosis are close to the standard normal values (0 and 3, respectively).

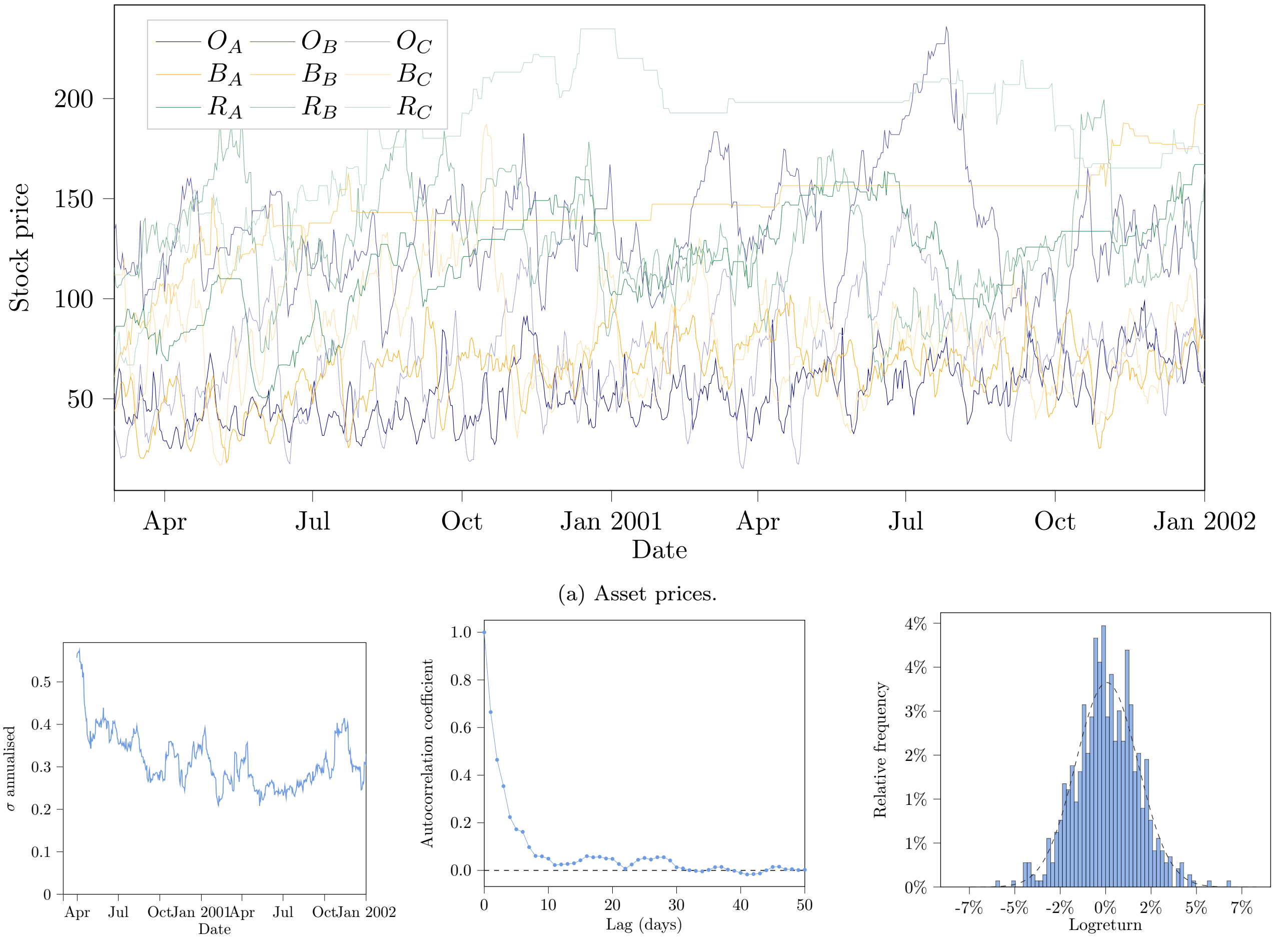
Combined, these observations provide supporting evidence for the random walk hypothesis of a noise agent-only system. Consequently, the model design does not itself attribute to the dynamics: Price processes and overall dynamics are fully driven by the agents populating the system. Similar to the findings of LeBaron (2006) and Bjerkøy and Kvalvær (2018), noise traders alone are not able to produce dynamics similar to the one found in real financial markets.

6.1.5 Learning Agents Only

A system populated solely by learning agents does not yield realistic trading dynamics. Most significantly, the model shows clear signs of autocorrelation in price returns. The return distribution is slightly positively skewed and mesokurtic.

We performed a model run with a total of 500 learning agents and no noise trading agents. 100 learning agents maximised each of the 5 different reward functions in table 3.3. Descriptive statistics for the run is shown in table 6.4.

Figure 6.9c shows the autocorrelation function up to lag order 50. We observe clear signs for autocorrelation in return. Without noise agents, no stochastic agents absorb excess demand and supply. The agents learn to utilise each other to push prices upwards. The best example of this phenomenon can be observed in figure 6.9a for asset O_B. At one point, the price begins to rise. We are not certain of the exact reason for the initial increase. The agents observe the positive trend and demand for the stock increases. We view this as an interesting property, as the model design implicitly



(a) Asset prices.

(b) Rolling 30-days volatility

(c) Autocorrelation for lags $1, \ldots, 50$ on an equal-weighted price index.

(d) Histogram of logreturns

Figure 6.9: Learning Agents only

discourage high-frequency trading[13]. This form of trading is much less present in normal runs of the model. At a later time step, the bubble bursts and the price declines fast. This kind of bubble behaviour is repeatedly observed in the simulation.

As demand is funneled into the stock showing signs of autocorrelation, trading is significantly reduced in the other assets. The agents are adapting to the current environment and evaluate the chance of return to be higher in the stock exhibiting autocorrelation. Although autocorrelation is a property one does not want to be present in an artificial stock market (LeBaron 2006), we view it as positive in a run with only learning agents. The agents are able to learn and adapt to the current environment, which is stated as the main motivation for contribution (i) learning agents. We take autocorrelation in a learning-agents only environment as evidence for the agents' abilities to learn patterns and utilise discoveries. The versatility of the learning agents is a step towards increased realism of agent based models of financial stock markets - one of the key motivations of this thesis.

Figure 6.9b shows the 30-day rolling volatility. The volatility process is time-varying with periods of significantly higher volatility. Interestingly, these periods coincides with periods of high autocorrelation. Increasing prices naturally causes

[13] High-frequency in this context relates to e.g. daily rebalancings.

Statistics	**Value**
Daily mean	0.001
Daily volatility	0.016
Annualised volatility	0.328
Skew	0.014
Kurtosis	3.188
Min	-0.060
Max	0.069
# observations	732

Table 6.4: Descriptive statistics for an equal-weighted index of daily returns for the a run consisting of learning agents only.

an increase in volatility. However, we observe that neither prices nor volatility is persistent. Subsequent to the price increase and increase in volatility, a fall is always observed. This observation is normally referred as a correction in real financial markets.

Compared to a model with noise agents only, the volatility is much higher. One can observe the volatility to be in the same order of magnitude in the normal model. We take this as evidence of the learning agents ability to induce time-varying volatility. Consequently, evidence is further added to Farmer and Joshi (2001)'s observation on noise agents' inability to alone describe real financial markets: agent based artificial stock market models must incorporate heterogeneous agents with different strategies. This observation further justifies contribution (i) regarding learning agents.

Table 6.4 shows the descriptive statistics. Similar to both the normal run and noise-only run, the daily mean is around zero. Annualised volatility is in line with empirical observations of real financial markets (Cont 2001). We observe a slightly positively skewed and mesokurtic distribution. The same observation can be seen qualitatively in the histogram of returns in figure 6.9d. Interestingly, the learning-agents only model does not exhibit equally large up and down movements as the normal case. We believe this observation can be accounted to the lack of noise agents. The learning agents are able to learn each other's strategy and are much more sensitive to larger price movements. Additionally, the market clears less often. Noise agents trade at random and consequently absorbs excess supply and demand. When the prices increase, the learning agents are unwilling to trade with each other. Consequently, there is a gap between demand and supply causing the price increase to stop. In the normal run, this gap is closed by noise agents and the price is allowed to increase.

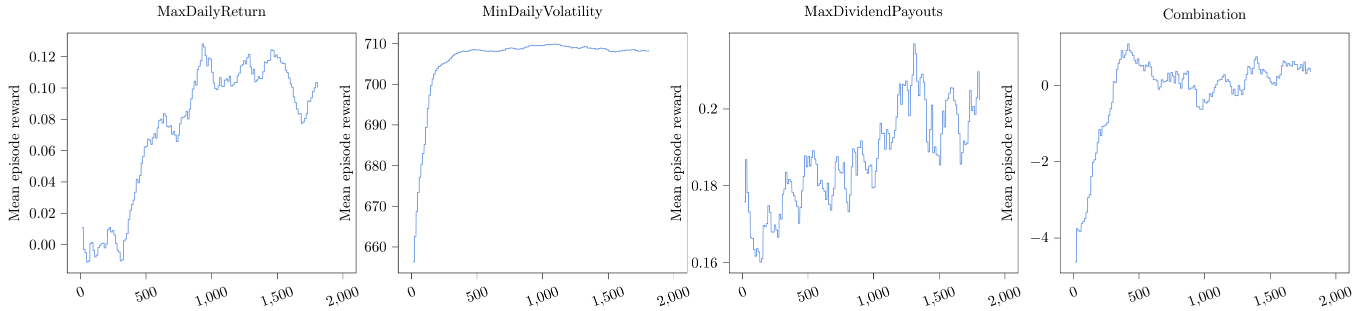


Figure 6.10: Mean episode reward for different policies. The x-axis displays the episode number and the y-axis shows the total mean episode reward for an episode. Note that the y-axis values are in different scales. In general, rewards obtained from different policies cannot be directly compared. The different policies are specified in table 3.3.

6.2 Learning

This section shows that the agents are indeed learning. We keep the discussion limited as the focus of this thesis is economical properties. A discussion on the agents' ability to learn is carried out to demonstrate the validity of our implementation of agents utilising reinforcement learning.

6.2.1 Overall Learning Trend

The agents are learning from their environment and adapt to the current market climate. Figure 6.10 shows the mean episode reward for different policies across simulations[14]. An increasing mean episode reward indicates that the agents become, on average, better at maximising their rewards. To successfully train the agents, several aspects needs to be considered:

(i) The observation space must contain enough relevant information in order for the agent to make informed decisions. For example, we noticed that if an agent seeking to maximise dividend yields did not observe the macroeconomic environment and time to quarterly earnings, then the agent's performance did not improve.

(ii) For smooth increase in performance, one needs to run many parallel simulations. As was discussed in section 3.4.3 the motivation to run many simulations in parallel was to even out the extreme rewards attained from bull and bear markets.

The market dynamics change as the parameters of the learning agents change. The change in policy of one agent might affect the macroeconomic environment[15]. When the market environment changes, this might force other learning agents to adapt as well. Therefore, we do not consider learning to be a static process; agents constantly have to adapt their trading strategies as a response to the overall environment and how other agents responds. This is highly in line with the adaptive market hypothesis by Lo (2004). Based on this insight, we do not expect a monotonically increasing reward function. It may be interrupted by sharp declines before it continues to increase. Figure 6.10 supports this hypothesis. It shows that (1) some reward functions are much easier to learn[16]; and (2) when the mean episode reward of one agent increases, it might also mean that the policy of another agent becomes worse, forcing the other agent to adapt its policy.

We believe the differences in ease of learning is affected by the following aspects:

(i) Maximising return on daily basis is hard. On a day-to-day basis the prices exhibit much noise, as in real financial markets. We have proved the model to exhibit close to zero autocorrelation in returns, making this policy challenging to learn: there are no underlying statistical properties supporting the agent's learning model.

(ii) Minimising daily volatility seem to be less challenging. The model is shown to exhibit time-varying volatility with periods of clustering. If the agent perceives the current volatility of individual stocks to be abnormal, it can utilise statistical relationships to construct a portfolio with overall low volatility.

(iii) Maximising dividends is difficult. Company earnings are stochastic in the model and so are the corresponding dividends. The agent can attempt to learn how the macroeconomic factors influence individual company earnings. However, this relationship is hidden to the agents. Extensive training and exploration have to be performed for an agent to discover the correlation between earnings and macroeconomic factors. There are two major challenges associated to learning this relationship: (1) Macroeconomic factors are updated daily but earnings are only revealed quarterly. This causes a delayed effect where the agent have to find relationships in the history. (2) The rewards are sparse: Agents attempting to maximise dividends receive rewards on a quarterly basis; there

[14] The episode reward is the total discounted reward of an episode (equation (3.4.6)).
[15] A potential scenario could be the following: the policy might say to buy bank stocks more often because, on average, their earnings are more stable
[16] Learning different reward functions is discussed in section 6.2.

are a lot of time between realisation periods where no dividends are distributed. In the period between dividend pay-outs the agent can perform reweighting of their portfolios. As a consequence they can have a hard time identifying which of these reweightings led to a reward. In general, learning a policy based on sparse rewards is much more difficult than learning policies with frequent rewards (Sutton and Barto 2018).

(iv) Combination is relatively easy. The agent has simply more than one leg to stand on. They can combine techniques from the different pure strategies to form a policy yielding the highest reward. The agent becomes more versatile in an ever-changing market.

6.2.2 Learning Different Reward Functions

Sections 6.2.2 and 6.2.2 examine experiments where agents learns how to (1) minimise volatility and (2) maximise factor exposure. The motivation is to observe how weights of the reward function in equation (3.4.19) affect agent behaviour. The experimental setup is the same as in chapter 5. That is, the macro factors are the oil price, interest rate and GDP growth. Agent composition is also the same as in table 5.2 in chapter 5.

Learning Volatility

The agent names and reward function weights are found in table 5.2. Policy $MinDailyVolatility$ 4 has higher weights on daily volatility compared to $MinDailyVolatility$ 4. Consequently, one expects lower portfolio volatility for agents following implementing first policy. The same relationship holds for $MinDailyVolatility$ 2 and $MinDailyVolatility$ 1.

Figure 6.11 shows annualised average equal-weighted rolling 30 day portfolio volatility for agents with reward functions $MinDailyVolatility$ 1, $MinDailyVolatility$ 2, $MinDailyVolatility$ 3 and $MinDailyVolatility$ 4. The annualised equal-weighted 30 day index portfolio is included for comparison. The figure shows that agents with $MinDailyVolatility$ 2, $MinDailyVolatility$ 3 and $MinDailyVolatility$ 4 reward functions are able to create portfolios with significantly lower volatility compared to the index. We take this as evidence for the agents' ability to learn different reward policies and achieve goals in line with the policy. It also indicates that the agents are able to learn statistical relationships between assets and utilise those relationships when constructing portfolios. Interestingly, $MinDailyVolatility$ 1 do in general have higher volatility than the index. We believe this can be traced to the fact that portfolio reward weight $w_V = 0.20$ is relatively low: Agents in $MinDailyVolatility$ 1 choose to accept risk (and receive punishments from the volatility reward function) in favour of rewards from other factors, such as portfolio return and factor exposure.

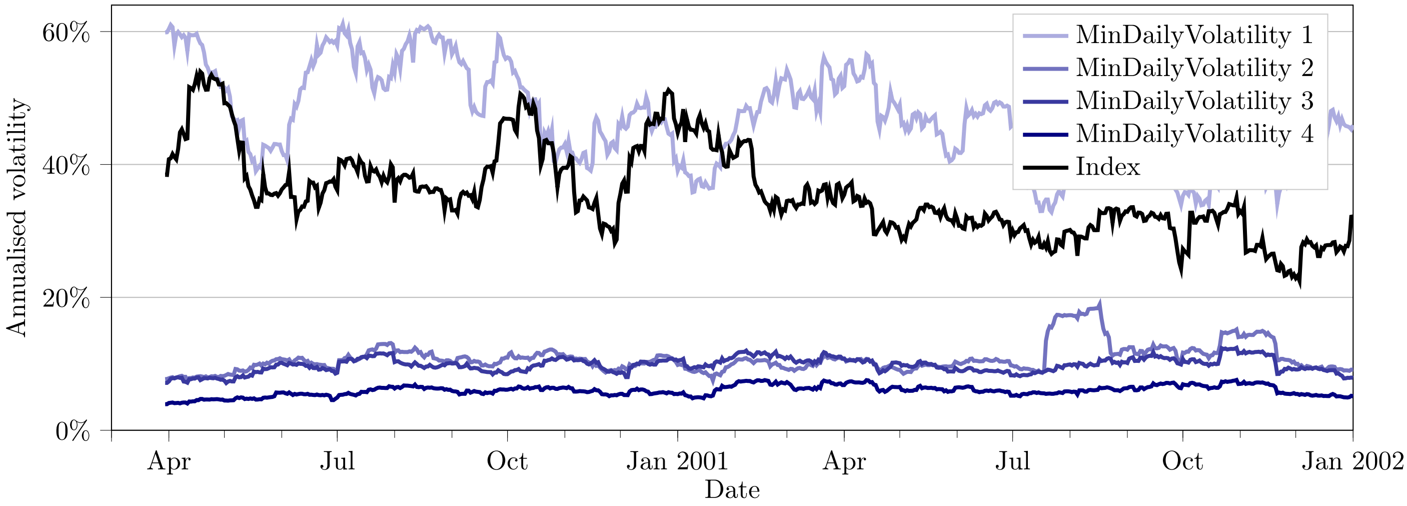


Figure 6.11: Annualised volatility for the different volatility minimising policies in table 5.2.

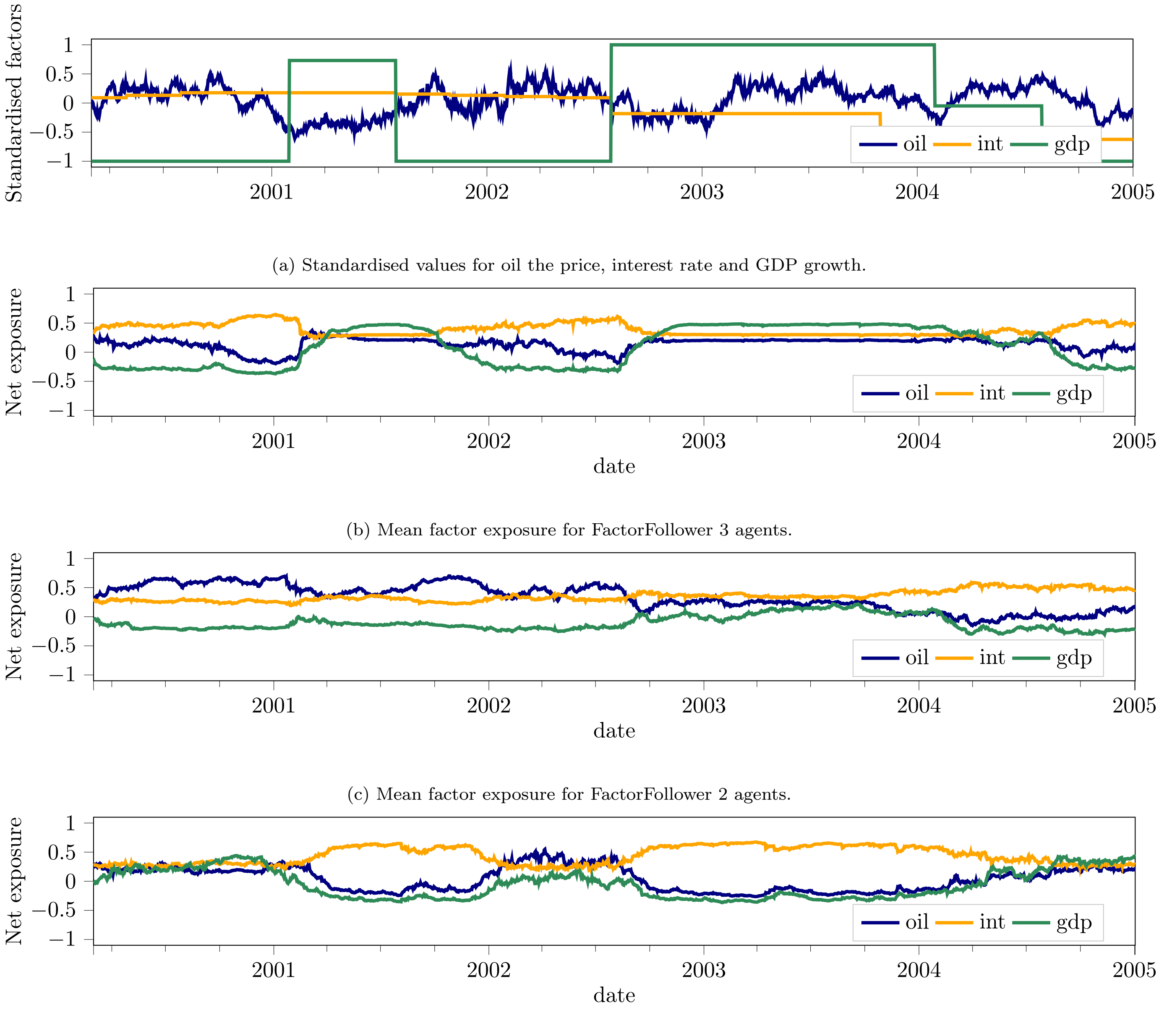
Learning Factors

We assess different variations of the FactorFollower strategy in table 3.3. The motivation is to examine how different values of w_F affect trading behaviour and how successful agents are at constructing portfolios which replicate the macroeconomic factors. We hypothesise that higher w_F makes agents construct portfolios with higher net exposure to the macroeconomic environment. For example, if the standardised oil price is high, we expect an agent to have a portfolio that contains an overweight of oil stocks.

Bear in mind that companies' direct exposure to factors are hidden to the agents. They have to explore and learn how different company earnings correlate with the macroeconomic factors. In addition, net exposure is also unobservable.

We, who know the underlying correlations, can assess the net exposure. Formally, net exposure for a factor f at time t is defined as:

$$\zeta_{if} = \sum_{j}^{J} w_{ij} f_t \tag{6.2.1}$$



(a) Standardised values for oil the price, interest rate and GDP growth.

(b) Mean factor exposure for FactorFollower 3 agents.

(c) Mean factor exposure for FactorFollower 2 agents.

(d) Mean factor exposure for FactorFollower 1 agents.

Figure 6.12: Mean factor exposure for different reward functions.

The results are shown in figure 6.12. Figure 6.12a shows that over a 2 year period both the oil price, interest rate and GDP fluctuates. Agents with FactorFollower 3 reward functions (figure 6.12b) track the factors well. For example, in the beginning of 2001 the standardised GDP spikes from -1 to about 0.5. Shortly, the net exposure to GDP growth increases dramatically for FactorFollower 3 agents. The same applies for FactorFollower 2 agents, however to a smaller degree. It is also interesting to note that FactorFollower 1 agents actually seem to have a negative net exposure when the market factors are positive and vice versa. We believe the origin of the phenomenon to be that agents maximise their reward function where factor exposure comprises small part only of the total policy ($w_F = 0.20$). Demand is driven significantly up for stocks with positive exposure by agents with FactorFollower 3 and FactorFollower 2 reward functions. Hence, agents in FactorFollower 1 sell their shares to lock in an immediate reward. In conclusion, the experiment shows that the agents are able to learn how to create portfolios positively exposed to the macroeconomic environment.

6.3 Experiments

This section demonstrates the versatility and power of our model. We conduct a few experiments to indicate possible applications of the model. The first experiment relates to goal (ii) on scenario analysis. We replicate the financial crisis of 2008 using real data from the Norwegian market. Furthermore, we experiment with alternative changes in the key deposit rate.

6.3.1 Replicating the Financial Crisis of 2008

The global financial market experienced a widespread crisis starting late 2007 and spanning through 2008. Markets slowly began to recover in 2010 after joint global efforts from authorities and central banks. American markets were hit especially hard, with S&P500 losing about 50% of its value.

We input real economic time-series to the model and simulate asset prices. Economic factors used in the experiment are the oil price, the key policy rate and the growth in domestic product. Data are Norwegian key performance indices. Figure 6.13 shows these time-series. As previously, we have three sectors comprised of three stocks per sector. The stocks have different debt profiles.

Before we conduct the experiment, a few motivations on the design are in order:

(i) We use data for the Norwegian market. Factor time-series are Norwegian sizes. OSEBX index is used for comparing the model results with real data. The rationale for using the OSEBX index is that it is an index with a high exposure to the oil price. Therefore, the oil price is an important macroeconomic factor that helps explain why the index fell sharply during the crisis.

(ii) A market capitalisation (MCAP) weighted index[17] is used instead of an equal weighted index. There are two reasons for this: Firstly, OSEBX, which is used for comparison, is an MCAP-weighted index. For comparability the index we use should be MCAP-weighted. Secondly, large stocks falling sharply during turmoil is more dramatic than smaller stocks. Such events should be reflected in the index.

Figure 6.14a shows the MCAP-weighted index for the model run with individual assets in blue. We observe a steady increase in index price up until late 2007. In figure 6.13 showing the economic factors, the oil price starts to decrease suddenly from all-time high levels. The decrease is sharp and almost continuous. Oil stocks in the model are affected

[17] An MCAP-weighted index uses relative market values as weights instead of equal-weights.

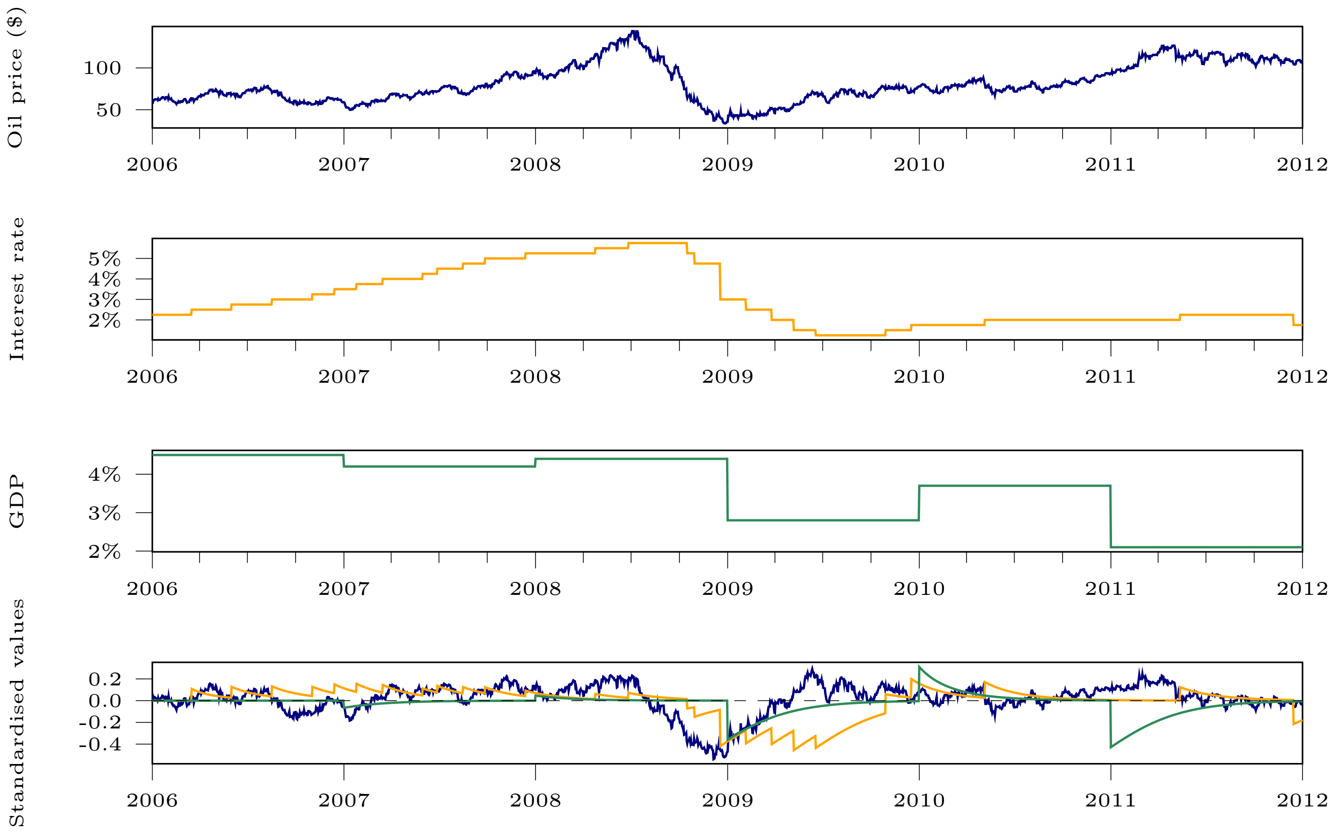
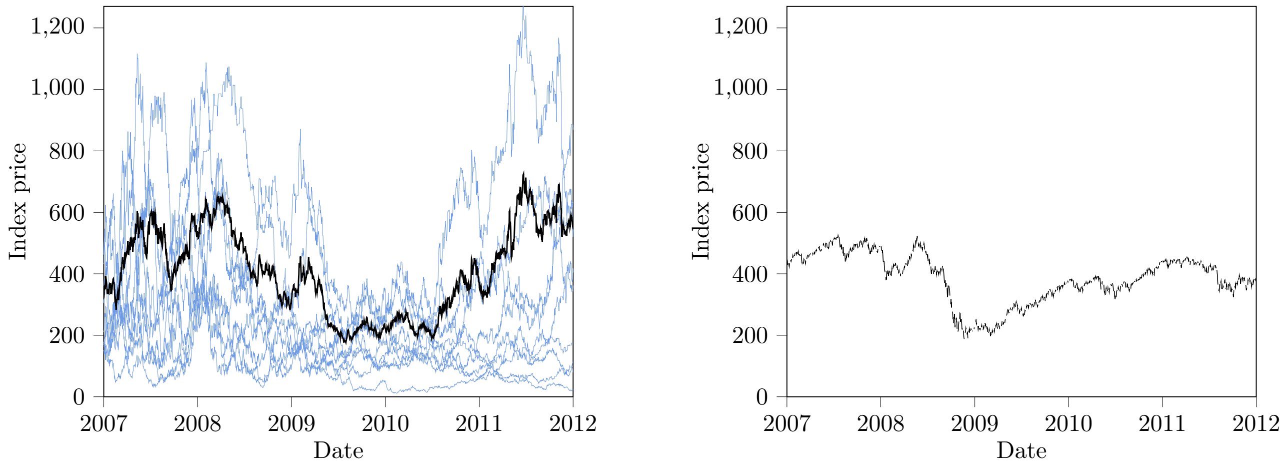


Figure 6.13: Time-series of the oil-price, the interest rate and the growth in domestic product during and in the aftermath of the financial crisis of 2008. Bottom graph show the standardised time series.



(a) Market index for the model run in black with individual assets in blue.

(b) OSEBX index during the period.

Figure 6.14: Price indices for the model run and real OSEBX data during the financial crisis.

by this macroeconomic change and asset values begin to deteriorate. The other sectors are affected as well; both retail and banking stocks decline heavily. Ultimately, these underlying processes are reflected in the aggregated price index: the index falls sharply in the period, reaching a period all-time low in early-2009. The same observations hold for the real data. OSEBX experienced strong markets prior to the financial crises. Upon changes in the global macroeconomic environment, the index falls heavily. Figure 6.14b showing the OSEBX price index during the financial crisis reflects this.

As a response to a stagnant economy with lower growth in domestic product, the authorities launched several incentive programmes. Norges Bank (the central bank of Norway) reduced the key policy rate[18] by nearly half in late 2008. By this time, the OSEBX index had reached its period all time low. From the onset of 2009, prices begun to appreciate once more. Volatility was starting to decrease as well, as can be seen in figure 6.15d.

Similar observations on dynamics can be made in the model output. The logreturn diagram for OSEBX in figure 6.15c is slightly more jagged compared to the model logreturn in figure 6.15a. Although, one observes the same order of magnitude; meaning the returns fluctuate equally in the model and in the real series. The model has higher amplitude in the logreturn both prior to and after the financial crisis, however. One interpretation of this observation can be that the crises spans longer in the model; the markets do not recover as quickly as in the real data. We believe this period of extended crisis originate from the fact that the agents observe a subset of the factors influencing the economy. All of the factors in the model experience declines. However, in the real market the authorities and Norges Bank performed actions not reflected directly in the factor time-series. Examples include fiscal measures intended to stimulate GDP growth in the future (which can be anticipated by investors of real financial markets), extending the maturity of F-loans[19] and supplying US dollars to private banks (Gram 2017). These forms of commitment to stabilising the economy can be communicated by policy makers and authorities, but takes time to materialise in real results. Consequently, we believe the stock market recovered more quickly than the model due to optimism and reliable actions by the government.

Rolling 30-day volatility is shown in 6.15b. The same observation of increased duration of the crisis is present. Similar to figure 6.15d showing the 30-days rolling volatility for OSEBX, one can observe a period of increased volatility during 2008-2009. In conclusion, the model exhibit similar dynamics patterns as real financial data for Norway.

What if Norges Bank acted more quickly?

The financial crisis did not hit the Norwegian markets as hard as other Western counterparts[20]. Norges Bank responded to the liquidity losses in the market by supplying the banking sector with excess capital. The government launched fiscal measures to cope with lower rates of employment and decreased export. However, the key policy rate was not affected initially. Norges Bank assessed the Norwegian economy to perform better compared to other industrialised economies (Gram 2017)[21]. Consequently, the key policy rate was increased a few times through 2008, before eventually being decreased in the start 2009.

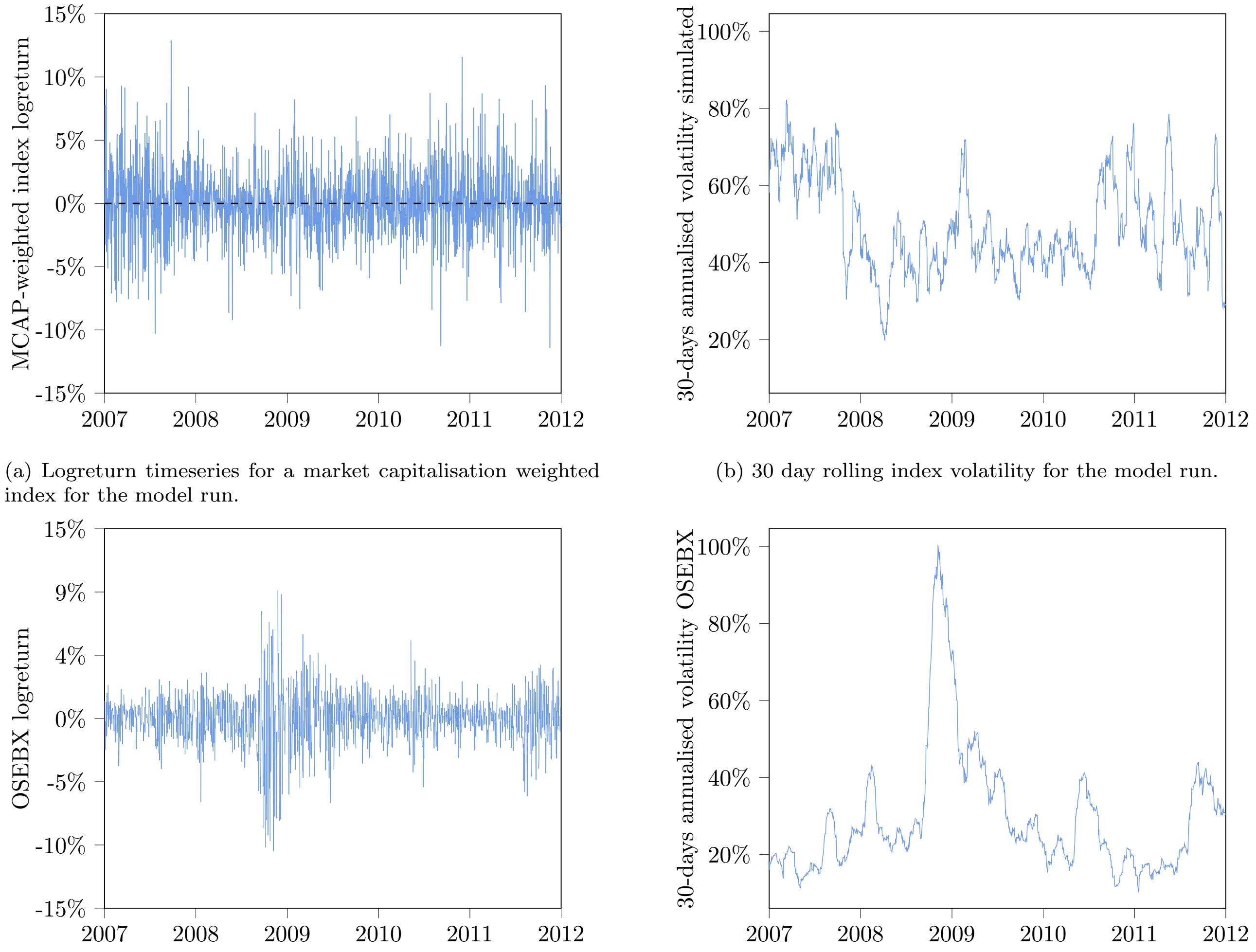
Even though Norges Bank undertook actions to ensure liquidity in the private debt sector and the government initiated

[18] The key policy rate is the interest rate offered by Norges Bank. Key policy rate and sight deposit rate are interchangable terms.

[19] F-loans are instruments issued by the central bank to provide liquidity to banks.

[20] In terms of duration and side-effects. The financial crisis spanned for a longer duration of timed, had greater influence on the private economy and caused a large debt overhang which many countries still wrestle with today (Gram 2017).

[21] Their assessment was, in retrospect, quite correct. The Norwegian market was not as *badly* regulated with lack of transparency. More strict regulations and less tail risk made Norway avoid the domino-effect one observed in e.g. the US when Lehman-Brothers went bankrupt (Holden 2009).



(a) Logreturn timeseries for a market capitalisation weighted index for the model run.

(b) 30 day rolling index volatility for the model run.

(c) Logreturn timeseries for OSEBX in the period.

(d) 30 day rolling volatility for the OSEBX index.

Figure 6.15: Comparison of model and real data logreturn time-series and volatility

fiscal programmes, there is no broad consensus on the actual effects of these initiatives. The financial crisis was indeed less severe in Norway, but researches debate whether this stems from structural relationships of the economy or the public incentive programmes (Gram 2017). Norway's economy is highly exposed to the oil price; much less to global sentiments. Additionally, Norway had the opportunity to finance fiscal measures through income from the petroleum sector instead of issuing debt[22]. Financial flexibility was higher in Norway compared to other industrialised countries. Nevertheless, the Norwegian stock market was hit hard. The OSEBX index fell by more than 60% between 2008 and 2009.

A natural question then arises: what if Norges Bank had acted more quickly? Would the market decline as sharply if the central bank had turned around and lowered the key policy rate earlier? We perform an experiment where Norges Bank begins to lower the key policy rate in May 2008, instead of January 2009 as was the case[23].

Simulations of the experiments indicate that the crash was imminent, independent of Norges Bank's interest rate policy. The fall appears softer, however. Figure 6.16a shows the price index simulation where Norges Bank responded to the crises earlier and lowered the key policy rate. As one can observe from the figure, a crash occurs later in 2009. The lowered key policy rate was able to stabilise the markets for some extended period of time, but the stimulation was not strong enough to counteract the other macroeconomic changes. Eventually, the market crashed.

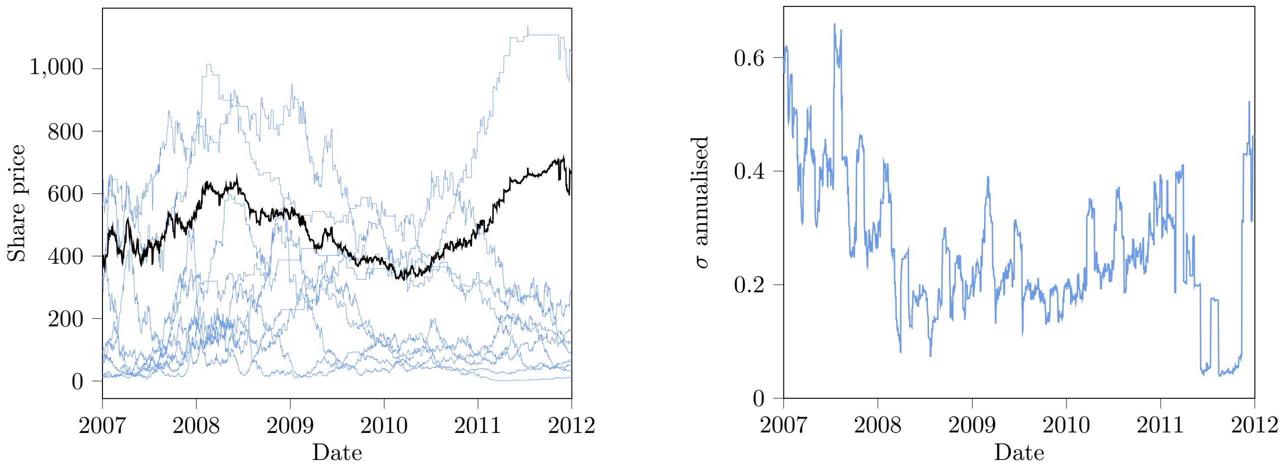
A prolonged stabilisation of the market can also be found in figure 6.16b showing the model volatility with a shifted change in the key policy rate. The experiment exhibits the same dynamics as the volatility of the model run of the crisis found in figure 6.15b: The spike in volatility occurs later and is not as high as in the normal case.

It is not in the scope of this thesis to speculate in underlying factors for the inevitable crash of OSEBX. However, we believe the fall was imminent from a model standpoint for numerous reasons, including:

(i) Correlated drops in economic factors. All factors experience a correlated decrease. In our model, this influences company earnings. Lowered earnings are normally reflected in lower stock prices.

[22] Accrued income as well as current.

[23] We use the same time-series as the real key policy rate but clips the rates between May and January. The resulting series is consequently shifted negatively by 7 months.



(a) Price index for the experiment where Norges Bank lowers the key policy rate earlier.

(b) 30 day rolling index volatility for the model run.

Figure 6.16: Price index and 30-day rolling volatility for a model run where Norges Bank lowers the key policy rate in May 2008 instead of January 2009.

(ii) Agents perceive the environment to be changing. From extensive training they may have developed heuristics for identifying bear markets. As previously noted, there exists a causal relationship between falling economic factors and stock prices. A natural implication of correlated factor drops is the gradual shift towards a bear climate. Agents start to withdraw from the market and increase their cash holdings. Withdrawal of funds lays pressure on supply, driving stock prices down. When the perception of a significantly weaker markets is shared by the majority of agents, the market crashes.

6.4 Improvements & Further Research

This section describes design choices that could have been different along with the motivation for a different implementation approach. Some of the limitations come as a consequence of deliberate choices, while others were discovered during experiments.

6.4.1 Calibrating to Empirical Parameters

In section 6.3.1 we replicated the financial crisis of 2008 using empirical time-series and conducted and experiment where the interest rate was changed earlier. The model was able to successfully replicate the crisis. In order to increase the explanatory power of the experiments one should calibrate other data than the economic factors to empirical data. The framework for company earnings in section 3.1.4 as well as the framework for debt in section 3.1.5 support fitting of company data.

Specifically, section 3.1.4 described the earnings process proposed in the model. As noted, an improvement would be to imitate real company earnings processes. These processes would be closely related to the market one tries to model. Consequently, empirical studies on how earnings are distributed and what kind of process they follow need be carried out. We considered this to be outside the scope of the thesis. Sadly, this is often the case for financial agent based models. Rutkauskas and Ramanauskas (2009), developing an ABM with Q-learning agents, do not attempt to fit the model to empirical parameters either. We rationalise the decision of omitting calibration by two arguments: (1) It is within the field of empirical finance, wheras our model fits more closely in applied computer science. (2) The task is complicated and highly market specific. One would need to tailor the processes to individual stocks, sectors and markets. We believe that a simple framework free of assumptions is convenient when the goal is to *replicate* the stylised facts of financial markets.

Lastly, one can argue that calibration of parameters to real data would yield more realistic dynamics. This will increase the realism of the model and the validity of the results. We propose that further research should aim to calibrate earnings processes, dividend policies and other company specific parameters as described in table A.2 to the markets one tries to model.

6.4.2 Debt

Section 3.1.5 motivated the need for different capital structures. In section 3.1.4 we proposed a framework for modelling company earnings as a sum of undertaken revenue-generating projects where companies can use debt to finance the investment cost. As described in section 3.1.5, the motivation for including debt is to expose the companies to the interest rate. Although the interest rate successfully impacts company earnings and stability in the model, we propose to alter the implementation. We have found the arrival of projects, as well as bidding process in section 3.1.4, to be too stochastic and difficult to analyse. As a consequence we had to limit the amount of projects to keep the earnings of companies under control. A new framework for debt modelling should aim to achieve the same level of simplicity as debt profiles of section 3.1.5 offer while connecting debt closer to the economy as a whole. We view a such framework as a valuable contribution to developing a agent based model fit for scenario analysis.

6.4.3 Reward Policies

We suggested several reward policies in section 3.4.7 meant to represent investment philosophies found in the real world. Five generalised policies are defined. Future work utilising reinforcement learning in agent based modelling should focus on increasing the number of policies available to agents. Conceptually, one particular configuration (i.e., weighting of the different policies for an agent), represents an individual in real markets. Increasing the number of policies as well as the number of agents will make the model approach a real life distribution of investors.

6.4.4 Stock Prioritising and Reweighting Schemes

In chapter 4 we proposed a reweighting scheme. The purpose of the scheme was to translate the agent's perception of the stock to an explicit trading operation. An unfortunate side effect of the scheme is that it is a static rule set where the actual submitted order is somewhat stochastic. To increase realism of trading we suggest to develop a framework for intelligent (i.e., not static) reweightning. In a such framework output from the neural network should not be limited to $\{buy, sell, hold\}$. To the authors' knowledge this does not exists today.

An alternative approach to reweighting can be to develop an auxiliary neural network tasked with calculating exact relative positions. This will increase the output complexity significantly. A naive implementation could be to have different degrees of *buy* and *sell*, translating to different relative weights.

6.4.5 Perception of Fundamental Value

The agents in the model have no perception of fundamental value. When they compare the different stocks, they obtain a relative weight between each stock as defined in section 3.4.6. Their only notion of value is relative; how expensive is the stock compared to similar assets. A relative measurement is comparatively similar to a peer based evaluation, but our implementation lacks a final computation of share price. Commonly, in real life investment analysis, investors calculate a price range for the perceived value[24]. Consequently, there are no "bands" for the prices. Individual prices can, in principle, be severely disconnected from the overall market. While this is not a problem in the event of an economic bubble, it can pose challenges to the interactions between agents when only a small subset of the agents can afford the stock. Ultimately, this causes stale prices and illiquid stocks.

Bjerkøy and Kvalvær (2018) proposed a fundamental value agent implementation using a stochastic forecast on the discounted earnings as well as a multiple based approach to the continuation value[25]. This approach was abandoned when we developed the model presented in this thesis. Although the implementation in Bjerkøy and Kvalvær (2018) yielded promising results, an early design choice was to avoid all kinds of static trading rules. There was no natural entry point for share price calculation in the implementation given in chapter 3.4.

We suggest to extend the model with notions of fundamental value. A possible approach is to use auxiliary neural networks designed especially for this task.

6.4.6 Observation Space and Model Size

In line with other experiments utilising reinforcement learning, we let the agents of our model observe as many parameters as possible. We do not assign sentiment to any of the observations; the agents themselves stand free to develop their own heuristics based on the observations. A natural step towards increased realism is to extend the observation space for the agents. Examples include higher moment order statistics than volatility (i.e., kurtosis, skew) and analyst recommendations[26]

As shown in section 6.1.3, increasing the agent population yields dynamics which resemble real world data more closely. Due to computational and time limitations, we had to keep the agent population relatively small (500 agents or smaller). In real financial markets, a large number of individuals participate in the daily trading. Normally, individual actions undertaken by any individual does not affect the economy as a whole. In a model populated by few agents, individual agents can temporarily influence the price of an asset dramatically. Large increases or decreases in prices should originate from surplus in aggregated demand or supply. The source of this surplus should be underlying changes in the driving factors, company earnings, behaviour of a group of agents and so on; not caused by the actions of single agents. Future attempts of creating artificial stock markets should strive for increasing the number of agents to a realistic level (orders of magnitude larger than in section 6.

Increasing the number of agents is, as shown in section 6.1.3, advantageous for simulations where one changes the driving factors of the economy. Increasing the amount of training data is conceptually similar to increasing the cognitive capacity of the agents. Agents which have been trained more and on more extensive training data, generate more informed actions.

6.4.7 Tractability in Experiments

A common problem with neural networks is tractability. One can observe input, configurations and output, but the core of the network remains hidden. The process can essentially be viewed as a black-box. This is challenging for a researcher attempting to investigate relationships. In Tay and Linn (2001) the information set is reduced to something the authors call fuzzy logic, as described in section 2. Fuzzy logic is comprehensible to humans, while the interior of neural network calculations is difficult to analyse. In this way the authors are able to investigate relationships and trading dynamics at the cost of loss in complexity: One can say that human behaviour is encoded to a set of rules.

We wanted the agents to be as realistic as possible. This relates to assumptions on agent behaviour and tuneable exogenous parameters. Human behaviour is inherently unpredictable, being both inconsistent and irrational at times. To make trading occur as realistically as possible we did not want to make assumptions on human behaviour. Consequently, we modelled agent cognition using neural networks. It was a deliberate choice to lose the possibility to evaluate causal relationships between observable factors and trading decisions. We believe the choice is justified as the model remains assumption-free and has few exogenous parameters.

[24] Commonly applied methods include the DCF approach, peer based valuation and sum of the parts analysis(Koller, Goedhart, Wessels, et al. 2010)

[25] The continuation value, or the terminal value, is the perpetual value of a company's assets after the forecasting period (Koller, Goedhart, Wessels, et al. 2010)

[26] Analyst recommendations are not present in our model. However, as shown in Bjerkøy and Kvalvær (2018), analyst recommendations explain herding. Herding is a form of inconsistent behaviour; individuals attribute trust to the decisions of a group instead of their own reasoning.

The model offers the opportunity to provide background data such as time series prices and earnings as well as forecasts on economic data. In the resulting simulation one can observe the dynamics between stocks, sectors and factors. The ability to dig further into the dynamics is impossible. Again, the intractability of our cognition model impose challenges on evaluating direct relationships.

6.4.8 Limitations on Trading Model

A limitation of the trading algorithm in section 4 is the number of times agents are allowed to trade each day. On any given day, each agent are just allowed to perform one action per stock; they decide on whether to purchase, sell or do nothing. This reduces the intraday effects observed i real financial markets (Cont 2001). Agents which already have selected an action will be oblivious to large changes in the environment. Their observation space will be updated the next day. Additionally, this design limits the dynamic side of the order-driven model. A key trait of real stock markets is intraday fluctuations caused by temporary disequilibrium in supply and demand. As a consequence, our time-series become less jagged compared to real financial data[27].

6.4.9 Limitation on Reinforcement Learning Reward Distribution

The behaviour of the agents in a reinforcement learning model is highly dependent on the design of the reward functions. As Sutton and Barto (2018) note, future applications of reinforcement learning will benefit from research on how reward signals form learning. In our model, the design of reward functions and distribution of rewards pose the following challenges:

1. **Discounting future rewards**
   Investors of real financial markets must choose between locking in profits today or hope for further value appreciation in the future. The same can be said for agents of artificial markets. Agents discount expected future rewards and compares them to rewards that could be obtained today. If the discounting is set high, agents will become myopic. They will choose short-term gains over long-term rewards.

2. **Sparse rewards**
   Agents following policies where rewards are not distributed daily suffer from the problem of sparse rewards (Sutton and Barto 2018). For agents attempting to maximising total received dividends, the action that led to the reward may not be as easy to identify.

3. **Limitation on hardware**
   The optimiser algorithm of section 3.4.4 relies on large amounts of samples to be efficient. Consequently, a lot of simulations has to be performed before the training yields realistic results. This makes the model prone to hardware limitations. To exhibit dynamics resembling real financial markets, we had to transition from running the model on normal computers to a cloud-based farm.

In addition to imposing challenges, the design of reward functions can cause agents to exhibit unintended behaviour. Consider the following real example from one of the early iterations when we designed the model: The agent's primarily reward function was to maximise daily return. At one point, the agent held a significant amount of shares in one of the stocks. There was practically no supply of the stock and demand was solely by the agent. This lack of liquidity caused the stock price to increase sharply, although at a low volume. The agent identified that it could exploit the disequilibrium. It had to pay a high price for the shares it was able to purchase, but it received a large reward on the daily return of its position. To make things "worse", the agent was relatively wealthy. The agent continued to inflate the price of the stock until, eventually, its funds were depleted. Ultimately, the agent had received a large sum of rewards, all the while it practically bankrupted itself. No other agent was able, or willing, to participate in the market for the stock at the inflated price.

The aforementioned example tells us to thread carefully when one designs reward functions. If not done correctly, unforeseen side-effects can occur.

[27] One cannot observe this effect, however, as we only show closing prices.

Chapter 7

Conclusion

In conclusion, the contributions offered by this thesis increases the validity and realism of a model of an artificial stock market. Agents learn and constantly adapts to the market. By modelling agent cognition using reinforcement learning we have successfully reduced the set of exogenous parameters, as suggested by LeBaron (2006). One of the few remaining exogenous parameters in our model is the macroeconomic factor time-series. These series are used to examine certain chain of events, making the model suited for scenario analysis.

The model exhibits the stylised facts of financial markets, consequently achieving goal (i). We find no evidence of autocorrelation in price returns. The absence of autocorrelation in returns is a key trait of financial markets, vital to the efficient market hypothesis. Any signs of autocorrelation could be exploited by constructing a statistical trading strategy. We show that contribution (i) learning agents effectively limits the opportunities for autocorrelation in returns. The return distribution generated by the model is negatively skewed and leptokurtic. This type of distribution is often found in real financial markets (Cont 2001). A negatively skewed distribution indicates that the left tail of the return distribution is larger than the right. This can be translated to a higher probability of large down-movements than corresponding up-movements in price returns. One experiences more frequently days of large negative returns. In the real world, fear, human behaviour and negative news from the companies are attributed as the source of this phenomenon. In our model, we propose the effect to originate from contribution (iii) regarding different investment philosophies of the agents. The reward functions defined in section 3.4.7 encourage certain agents to withdraw from the market in the event of turmoil. This causes excess supply, driving the prices down. A leptokurtic distribution indicates a fat-tailed distribution. From real financial markets we know that large movements in stock prices can be caused by changes in the macroeconomic environment. Similarly, we show that contribution (ii) regarding a factor-dependency framework for company earnings produces the same effect. Lastly, the model produces time-varying volatility with occasional clustering. Time-varying volatility is important to experience different market climates such as bubbles and recessions. We show that the spikes in volatility is connected to the changes in macroeconomic factors, underpinning the value of contribution (ii) regarding a factor-dependency framework.

The model was shown to be robust to changes in the agent population. The general observation was that a larger and versatile population increased the realism of the model runs. Versatility in this context relates to contribution (iii) investment philosophies: the model behaved more realistically as one increased the number of different philosophies and the number of agents following a philosophy. Prices from runs consisting only of noise agents where shown to be random-walk processes, similar to the findings of Farmer and Joshi (2001). Furthermore, model runs comprised only of learning agents failed to generate realistic dynamics, indicating that noise agents must be present in the environment. The price series exhibited clear signs of autocorrelation and large continuous discrepancies in supply and demand. We believe the failure in market clearance occurs as the agents learn each other's trading patterns and forms the same opinions, consequently performing the same actions.

Reinforcement learning using Proximal Policy Optimisation is well-suited for modelling agent cognition in an artificial stock market. We show why optimising a stochastic policy using policy gradient methods is preferred to inter alia Q-learning. Most importantly, Proximal Policy Optimisation is appropriate in artificial stock markets due to the non-stationarity of real markets. The agents in the model are learning from their environment and evolve heuristics[1] to based on market conditions. The presence of a continuous adapting process for the agents draws on inspiration from Lo (2004)'s adaptive market hypothesis and constitute contribution (i). We show that the investment philosophy framework in contribution (iii) provides a reasonable mapping from real-life investment strategies to reward functions in reinforcement learning. Additionally, we demonstrate some reward functions to be more challenging to learn than others.

One of the goals of the thesis was to create a framework for scenario analysis (goal (ii)). We demonstrate the power of the model by replicating the financial crisis of 2008. Real financial time-series from the Norwegian market are used

[1] Recall, a heuristic is a trading strategy formed from empirical observations

as inputs. The model is able to successfully replicate the crisis; correlated drops in the macroeconomic factors induce a market-wide crash. Similar dynamics are observed in the model as in the real data. To demonstrate the model's power as a scenario analysis tool we conduct an experiment where Norges Bank lowers the key policy rate in May 2008 instead of January 2009. We investigate whether reduced interest rates would have stimulated the economy enough to avoid a crash. Results from the simulation indicate that the crash was inevitable. However, the crash appears softer when the drop in interest rate occurs in May 2008.

We conclude the thesis by describing improvements to the model for future research. Some of the improvements originate from deliberate design choices, while others were discovered during experimentation. The most important improvements relates to expanding the set of reward policies (i.e., specifying additional investment philosophies as defined in contribution (iii)); increasing the action space for the agents (i.e., develop schemes for intelligent stock prioritisation); calibration of earnings processes to real data; and arm the agents with a concept of fundamental value of companies.

Bibliography

Modigliani, Franco and Merton H. Miller (1958). "The Cost of Capital, Corporation Finance and the Theory of Investment". In: *The American Economic Review* 48.3, pp. 261–297. ISSN: 00028282. URL: http://www.jstor.org/stable/1809766.

Fama (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *The Journal of Finance* 2, pp. 383–417. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325486.

Tversky, Amos and Daniel Kahneman (1974). "Judgment under Uncertainty: Heuristics and Biases". In: *Science* 4157, pp. 1124–1131. ISSN: 00368075, 10959203. URL: http://www.jstor.org/stable/1738360.

Garman, Mark (June 1976). "Market Microstructure". In: *Journal of Financial Economics*, pp. 257–275.

Griffin, Paul A. (1977). "The Time-Series Behavior of Quarterly Earnings: Preliminary Evidence". In: *Journal of Accounting Research* 1, pp. 71–83. ISSN: 00218456, 1475679X. URL: http://www.jstor.org/stable/2490556.

Myers, Stewart C (1984). "The Capital Structure Puzzle". In: *The Journal of Finance* 3, pp. 574–592.

Watkins, Christopher John Cornish Hellaby (1989). "Learning from delayed rewards". PhD thesis. King's College, Cambridge.

Chiarella, Carl (Dec. 1992). "The dynamics of speculative behaviour". In: *Annals of Operations Research* 1, pp. 101–123. ISSN: 1572-9338. DOI: 10.1007/BF02071051. URL: https://doi.org/10.1007/BF02071051.

Fama and French (1993). "Common risk factors in the returns on stocks and bonds". In: *Journal of Financial Economics* 1, pp. 3–56.

Andersen, Torben G, Tim Bollerslev, et al. (1997). "Intraday periodicity and volatility persistence in financial markets". In: *Journal of Empirical Finance* 2-3, pp. 115–158.

Schulenburg, Sonia and Peter Ross (1999). "An adaptive agent based economic model". In: *International Workshop on Learning Classifier Systems*. Springer, pp. 263–282.

Sutton, Richard, David McAllester, et al. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, pp. 1057–1063. URL: http://dl.acm.org/citation.cfm?id=3009657.3009806.

Cont, Rama (2001). "Empirical properties of asset returns: stylized facts and statistical issues". In: *Taylor & Francis*.

Farmer and Joshi (Jan. 2001). "The Price Dynamics of Common Trading Strategies". In: *Journal of Economic Behavior & Organization*, pp. 149–171.

Moody, John and Matthew Saffell (2001). "Learning to trade via direct reinforcement". In: *IEEE transactions on neural Networks* 4, pp. 875–889.

Tay, Nicholas SP and Scott C Linn (2001). "Fuzzy inductive reasoning, expectation formation and the behavior of security prices". In: *Journal of Economic Dynamics and Control* 3-4, pp. 321–361.

Chiarella, Carl, Giulia Iori, et al. (2002). "A simulation analysis of the microstructure of double auction markets*". In: *Quantitative Finance* 5, pp. 346–353.

LeBaron, Blake (July 2002). "Building the Santa Fe artificial stock market". In: *Physica A*, pp. 1–20.

Marchesi, Michele et al. (2003). "The genoa artificial stock market: Microstructure and simulations". In: *Heterogenous Agents, Interactions and Economic Performance*. Springer, pp. 277–289.

Lo, Andrew W. (2004). "The Adaptive Markets Hypothesis". In: *The Journal of Portfolio Management* 5, pp. 15–29. ISSN: 0095-4918. DOI: 10.3905/jpm.2004.442611. eprint: http://jpm.iijournals.com/content/30/5/15.full.pdf. URL: http://jpm.iijournals.com/content/30/5/15.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.

Cruz, Joseph A and David S Wishart (2006). "Applications of machine learning in cancer prediction and prognosis". In: *Cancer informatics*, p. 117693510600200030.

LeBaron, Blake (2006). "Agent-based computational finance". In: *Handbook of Computational Economics*, pp. 1187–1233.

McDonald, Robert Lynch, Mark Cassano, and Rüdiger Fahlenbrach (2006). *Derivatives markets*. Addison-Wesley Boston.

Tesfatsion, Leigh and Kenneth L Judd (2006). *Handbook of computational economics: agent-based computational economics*. Elsevier.

Berk, Jonathan B and Peter M DeMarzo (2007). *Corporate finance*. Pearson Education.

Chan, Christopher K and Ken Steiglitz (2008). "An agent-based model of a minimal economy". In:

Alexander, Carol (2009). *Market Risk Analysis, Value at Risk Models*. John Wiley & Sons.

Chakraborti, Anirban et al. (2009). "Econophysics: Empirical facts and agent-based models". In: *arXiv preprint arXiv:0909.1974*.

Holden, Steinar (2009). *Finanskrisen–årsaker og mekanismer*.

Kronwald, Christian (2009). *Credit rating and the impact on capital structure*. GRIN Verlag.

Ramanauskas, Tomas and AV Rutkauskas (2009). "Empirical Version of an Artificalt Stock Market Model". In: *Pinigų studijos [Study of Money]* 1, pp. 5–21.

Rutkauskas and Ramanauskas (2009). "Building an artificial stock market populated by reinforcement-learning agents". In: *Journal of Business Economics and Management* 4, pp. 329–341.

Geanakoplos, John (2010). "The leverage cycle". In: *NBER macroeconomics annual* 1, pp. 1–66.

Koller, Tim, Marc Goedhart, David Wessels, et al. (2010). *Valuation: measuring and managing the value of companies.* John Wiley & sons.

Christoffersen, Peter (2011). *Elements of financial risk management.* Academic Press.

LeBaron, Blake (Jan. 2011a). “Active and Passive Learning in Agent-based Financial Markets”. In: *Eastern Economic Journal* 1, pp. 35–43. ISSN: 1939-4632. DOI: 10.1057/eej.2010.53. URL: https://doi.org/10.1057/eej.2010.53.

– (2011b). “Active and passive learning in agent-based financial markets”. In: *Eastern Economic Journal* 1, pp. 35–43.

Zhao, Yufan et al. (2011). “Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer”. In: *Biometrics* 4, pp. 1422–1433.

Thurner, Stefan, J Doyne Farmer, and John Geanakoplos (2012). “Leverage causes fat tails and clustered volatility”. In: *Quantitative Finance* 5, pp. 695–707.

Cristelli, Matthieu (2013). *Complexity in financial markets: Modelling Psychological Behavior In Agent-based Models and Order Book Models.* Springer Science & Business Media.

Damodaran, Aswath et al. (2013). “Equity risk premiums (ERP): Determinants, estimation and implications–The 2013 edition”. In: *Managing and Measuring Risk: Emerging Global Standards and Regulations After the Financial Crisis*, pp. 343–455.

Jarrow, Robert (2013). “A leverage ratio rule for capital adequacy”. In: *Journal of Banking & Finance* 3, pp. 973–976.

Fischer, Thomas and Jesper Riedler (2014). “Prices, debt and market structure in an agent-based model of the financial market”. In: *Journal of Economic Dynamics and Control*, pp. 95–120.

Guerard Jr, John B, Harry Markowitz, and GanLin Xu (2015). “Earnings forecasting in a global stock selection model and efficient portfolio construction and management”. In: *International Journal of Forecasting* 2, pp. 550–560.

Kaiming He Xiangyu Zhang, Shaoqing Ren and Jian Sun (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *arXiv.*

Mnih, Volodymyr et al. (Feb. 2016). “Asynchronous Methods for Deep Reinforcement Learning”. In: *arXiv e-prints*, arXiv:1602.01783, arXiv:1602.01783. arXiv: 1602.01783 [cs.LG].

Schulman, John, Philipp Moritz, et al. (2016). “High-Dimensional Continuous Control Using Generalized Advantage Estimation”. In: *CoRR.*

Turrell, Arthur (2016). *Agent-based models: understanding the economy from the bottom up.*

Cox, David Roxbee (2017). *The theory of stochastic processes.* Routledge.

Gram, Trond (2017). “Bankkriser i Norge”. In: *Norges Bank.*

Schulman, John, Filip Wolski, et al. (2017a). “Proximal Policy Optimization Algorithms”. In: *CoRR.*

– (2017b). “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347.*

Bjerkøy, Aleksander and Mikael Kvalvær (Dec. 2018). “Asynchronous Trading in an Order-Driven Market with Heterogeneous Agents”. MA thesis. Norwegian University of Science and Technology.

Silver, David et al. (2018). “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 6419, pp. 1140–1144.

Sutton, Richard and Andrew Barto (2018). *Reinforcement learning: An introduction.* MIT press.

Appendix A

Parameters

General parameters and company specific parameters relating to projects and earnings are shown in this appendix.

A.1 General Parameters

Table A.1 shows the general parameters used as input to the model.

Parameter	Value	Interpretation	Reference
N_N	10	Number of noise agents	-
N_L	50	Number of learning agents	-
N_S	3	Number of sectors	-
J	9	Number of assets	-
j_x	500	Shares outstanding	-
X_{C0}	$\mathcal{U}(500, 1500)$	Initial cash	-
μ_o	1.01	Mean order submission adjustment	3.2.4
σ_o	0.01	Variance in order submission adjustment	3.2.4
I	$I_j \sim \mathcal{U}(0, 1)$	Trading intensity for noise agents	3.3.1
τ	27%	Marginal corporate tax rate	3.1.30
τ_d	27%	Percentage of interests that are tax deductible	3.1.29
D_j	1	Upper limit on dividend policy (100% of net income)	3.1.25
N	5	Number of trading rounds	1
γ_i	0.93	Discount factor on future rewards	3.4.6
θ_j^D	$\mathcal{U}(10, 1000)$	Backward-looking daily observations used in observation space	3.4.5
θ_j^Q	$\mathcal{U}(4, 40)$	Backward-looking quarterly observations used in observation space	3.4.5
α	0.01	Learning rate	3.4.2
λ_s	$\mathcal{U}(\frac{1}{67}, \frac{10}{67})$	Expected arrival rate of new projects in sector s	3.1.9

Table A.1: Table showing an overview of configurable parameters in the model

A.2 Company Specific Parameters

Table A.2 shows the factor-dependencies for companies in different sectors and with different debt profiles, described in sections 3.1.4 and 3.1.5 (specifically, equation 3.1.3 and table 3.2).

			Weights f_i	
Debt Profile	**Asset type**	**Oil price**	**Interest rate**	**GDP growth**
A	Oil	0.8	-0.3	0.2
	Banking	0.2	0.7	0.3
	Retail	-0.2	-0.4	0.5
B	Oil	1.2	-0.45	0.3
	Banking	0.4	0.7	0.45
	Retail	-0.3	-0.6	0.75
C	Oil	1.6	-0.6	0.4
	Banking	0.4	0.7	0.6
	Retail	-0.4	-0.8	1

Table A.2: Factor-dependency for companies in different sectors with different debt profiles from section 3.1.5.

A.3 Debt Profiles

Specification for debt profiles from section 3.1.5, specifically table 3.2.

Debt Profile	**Credit Spread**	**Maximum Leverage**
A	0%	5
B	1%	4
C	2%	3

Table A.3: Debt profiles, credit spreads and maximum leverage for assets in the profile.

Appendix B

The Stylised Facts of Financial Markets

Empirical findings have shown that financial data exhibit certain characteristics often referred to as stylised facts. These statistical properties are fairly identical for different timescales and stock markets (Cont 2001). An appropriate model should therefore at least exhibit some of these stylised facts to a certain degree. Cont (2001), these properties are inter alia:

(i) **Absence of autocorrelations**
Correlation in returns are often insignificant except for very small timescales. In other words, for a given daily log-return $r(t)$,

$$\mathrm{Corr}(r(t), r(t-\tau)) \approx 0, \quad \tau = 1, 2, \ldots T$$

(ii) **Heavy tails**
Stock return distributions exhibit fatter tails than those belonging to a normal distribution. The distribution seems to display a Pareto-like tail with a finite tail index. The precise form of the tail is however difficult to determine.

(iii) **Gain / loss asymmetry**
One does not observe equally large up-movements as down-movements. Consequently, the distribution is often negatively skewed.

(iv) **Aggregational Gaussianity**
Increasing the timescale for calculating the returns makes the distribution become more Gaussian. Another consequence of this property is that the distribution varies with time.

(v) **Volatility clustering**
High-volatility events tend to cluster in time causing a positive autocorrelation in volatility over several days. Clustering are also more present in down-markets. Consequently, volatility appears to be time-varying.

Appendix C

OSEBX

For completion, descriptive statistics for the Oslo Stock Exchange is attached. These data are used for evaluating the performance of the model. Index returns for OSEBX[1] creates the foundation for the empirical evaluation of the model. Section C.1 carries out an empirical analysis of the return distribution. This analysis will be used to evaluate whether the model displays the stylised facts described in section B. Please bear in mind that empirical analysis of returns is not the subject of this paper. The analysis will hence be brief. There are no explicit reason for using OSEBX data. However, there has to be a connection between the company specific parameters and the index that one uses to evaluate the model.

C.1 Empirical Distribution OSEBX

With reference to appendix B, the OSEBX return distribution should fit the characteristics described in (i)-(v). Table C.1 shows the descriptive statistics for OSEBX for the period 1996-2017. As expected, the daily mean is close to zero. We also have a positive skew and high excess kurtosis. These two observations are vital in order to demonstrate the heavy tail(ii) and gain/loss asymmetry (iii) properties.

[1] OSEBX is an investable index on Oslo Stock Exchange with a representative selection of stocks for the Norwegian market.

[2] Test for normality.

Statistic	**Observation**
Daily mean	0.04 %
Daily volatility	1.40 %
Annualised volatility	22.12 %
Skew	-0.59
Kurtosis	6.74
Number of observations	5,726
Min	-10.48 %
Max	10.14 %
Jarque-Bera[2]	11,652
Jarque-Bera p-value	0.0

Table C.1: Descriptive statistics for intraday log-returns for OSEBX 1996-2017.

Appendix D

Formal Observation Space

A conceptual overview of the observation space was provided in section 3.4.5. Here we provide a mathematical precise formulation of the inputs as matrices and vectors.

(i) **Company earnings**
A matrix with dimensions (J, θ_j^Q) where J are the number of companies and θ_j^Q the agent's memory length. In other words, the matrix grows with time. The matrix holds information about the historical return in earnings for all of the companies. Formally,

$$\mathbf{E} = \begin{bmatrix} E_{00} & \dots & E_{0\theta_j^Q} \\ E_{10} & \dots & E_{1\theta_j^Q} \\ \vdots & \ddots & \vdots \\ E_{J0} & \dots & E_{J\theta_j^Q} \end{bmatrix} \tag{D.0.1}$$

(ii) **PE-ratios**
A matrix with dimensions (J, θ_j^D). The matrix holds information about the price-earnings ratio for all of the assets $j \in J$ available in the model. Denote $\rho_{jt} = \frac{P}{E}_{jt}$ as the price-earnings ratio for company j at time t. Then PE-matrix becomes,

$$\rho = \begin{bmatrix} \rho_{00} & \dots & \rho_{0\theta_j^D} \\ \rho_{10} & \dots & \rho_{1\theta_j^D} \\ \vdots & \ddots & \vdots \\ \rho_{J0} & \dots & \rho_{J\theta_j^D} \end{bmatrix} \tag{D.0.2}$$

(iii) **Relative trading volume**
A matrix with dimensions (J, θ_j^D). The matrix holds information about the percentage of shares traded for each day. For each day, the trading volume is divided by the shares outstanding for each asset. Formally,

$$\Omega = \begin{bmatrix} \Omega_{00} & \dots & \Omega_{0\theta_j^D} \\ \Omega_{10} & \dots & \Omega_{1\theta_j^D} \\ \vdots & \ddots & \vdots \\ \Omega_{J0} & \dots & \Omega_{J\theta_j^D} \end{bmatrix} \tag{D.0.3}$$

(iv) **Volatility**
A vector with length J. The vector holds information about the volatility with a time period of length θ_j^D. Formally,

$$\sigma = [\sigma_0 \dots \sigma_J]^T \tag{D.0.4}$$

(v) **Macroeconomic factors**
A matrix with dimensions (J, θ_j^D). The macro-factors are the same factors that influence the return of the investments described in section 3.1.4. Formally,

$$\mathbf{f} = \begin{bmatrix} f_{00} & \dots & f_{0\theta_j^D} \\ f_{10} & \dots & f_{1\theta_j^D} \\ \vdots & \ddots & \vdots \\ f_{F0} & \dots & f_{F\theta_j^D} \end{bmatrix} \tag{D.0.5}$$

(vi) **Debt profile** A vector with length J. The vector holds a numerical interpretation of the debt profile for the company. The following mapping holds between the profiles

$$p_j = \begin{cases} \text{A} & 4 \\ \text{B} & 3 \\ \text{C} & 2 \\ \text{D} & 1 \end{cases}$$

Formally, the vector p takes the form:

$$\mathbf{p} = [p_0 \dots p_J] \tag{D.0.6}$$

(vii) **Dividend yield** A matrix with dimensions (J,θ_j^Q). The matrix holds information about the current and historical dividend yield for all stocks in the environment. Formally,

$$\mathbf{Y} = \begin{bmatrix} y_{00} & \cdots & y_{0\theta_j^Q} \\ y_{10} & \cdots & y_{1\theta_j^Q} \\ \vdots & \ddots & \vdots \\ y_{J0} & \cdots & y_{J\theta_j^Q} \end{bmatrix} \tag{D.0.7}$$

(viii) **Current portfolio**
A vector with length $J+1$ values. The vector holds information about the agent's current portfolio. Formally, as defined in section 3.2.1, the portfolio for agent i is the vector

$$[x_{i1}, x_{i2}, \dots, x_{iJ}, x_{iC}] \tag{D.0.8}$$

(ix) **Auxiliary information**
A matrix Z with dimensions (J,4) holding information about whether the current day is a dividend payout day, earnings realisation day, days until the next dividend and days until the next earnings realisation day. The first to columns are binary: a 1 indicates that the current day is a dividend day or earnings realisation day.

Appendix E

S&Ps definition of different credit ratings

Category	Definition
AAA	An obligation rated 'AAA' has the highest rating assigned by S&P Global Ratings. The obligor's capacity to meet its financial commitments on the obligation is extremely strong.
AA	An obligation rated 'AA' differs from the highest-rated obligations only to a small degree. The obligor's capacity to meet its financial commitments on the obligation is very strong.
A	An obligation rated 'A' is somewhat more susceptible to the adverse effects of changes in circumstances and economic conditions than obligations in higher-rated categories. However, the obligor's capacity to meet its financial commitments on the obligation is still strong.
BBB	An obligation rated 'BBB' exhibits adequate protection parameters. However, adverse economic conditions or changing circumstances are more likely to weaken the obligor's capacity to meet its financial commitments on the obligation.
BB, B, CCC, CC, and C	Obligations rated 'BB', 'B', 'CCC', 'CC', and 'C' are regarded as having significant speculative characteristics. 'BB' indicates the least degree of speculation and 'C' the highest. While such obligations will likely have some quality and protective characteristics, these may be outweighed by large uncertainties or major exposure to adverse conditions.
BB	An obligation rated 'BB' is less vulnerable to nonpayment than other speculative issues. However, it faces major ongoing uncertainties or exposure to adverse business, financial, or economic conditions that could lead to the obligor's inadequate capacity to meet its financial commitments on the obligation.
B	An obligation rated 'B' is more vulnerable to nonpayment than obligations rated 'BB', but the obligor currently has the capacity to meet its financial commitments on the obligation. Adverse business, financial, or economic conditions will likely impair the obligor's capacity or willingness to meet its financial commitments on the obligation.
CCC	An obligation rated 'CCC' is currently vulnerable to nonpayment and is dependent upon favorable business, financial, and economic conditions for the obligor to meet its financial commitments on the obligation. In the event of adverse business, financial, or economic conditions, the obligor is not likely to have the capacity to meet its financial commitments on the obligation.
CC	An obligation rated 'CC' is currently highly vulnerable to nonpayment. The 'CC' rating is used when a default has not yet occurred but S&P Global Ratings expects default to be a virtual certainty, regardless of the anticipated time to default.
C	An obligation rated 'C' is currently highly vulnerable to nonpayment, and the obligation is expected to have lower relative seniority or lower ultimate recovery compared with obligations that are rated higher.
D	An obligation rated 'D' is in default or in breach of an imputed promise. For non-hybrid capital instruments, the 'D' rating category is used when payments on an obligation are not made on the date due, unless S&P Global Ratings believes that such payments will be made within five business days in the absence of a stated grace period or within the earlier of the stated grace period or 30 calendar days. The 'D' rating also will be used upon the filing of a bankruptcy petition or the taking of similar action and where default on an obligation is a virtual certainty, for example due to automatic stay provisions. A rating on an obligation is lowered to 'D' if it is subject to a distressed exchange offer.

Table E.1: Description of credit ratings by Standard & Poor's