Extracting Information from the Order Book in an Artificial Stock Market

P.M. Koornneef
276521pk@student.eur.nl

Supervised by dr. ir. Uzay Kaymak
kaymak@few.eur.nl

Bachelor Thesis of Informatics & Economics,
Erasmus University Rotterdam, The Netherlands

August 17, 2006

Abstract

In a stock market, the flow of orders depends on many factors, like expected future profits of the company, economic prospects but also historical price movements and even the moment of the day. Seen from the perspective of the market maker, analysing the arrival of orders could give him important information. We present a model of a market maker in an artificial stock market (ASM) that actually uses this information. This model is based on the market maker model developed by Sanmay Das that runs in an agent-based environment, but uses the order book to determine its future bid and ask prices.

Contents

1 Introduction

With computer processing power constantly growing, the possibilities for undertaking research by means of simulation have been greatly improved. This is especially the case for *agent-based simulation*, in which analysis of a system takes place while the behaviors of individual agents are modelled. Within the field of finance, a lot of research is done on the simulation of stock markets, and recently due to the emergence of agent-based models, by means of Agent-Based Artificial Stock Markets (ABASM).

An example of an ABASM is the simulation program ABSTRACTE, developed at the Department of Economics at Erasmus University, Rotterdam. This program provides a framework for an agent based stock market, based on the ideas of Boer, Polman, De Bruin and Kaymak [2]. Specific models can be implemented and experimented with. Basically, the system consists of a number of investors and a market maker, all with their own customizable behaviors.

One of the models implemented is the market maker model described by Das [6][13]. However, this is a discrete-time and synchronous model, while ABSTRACTE (being agent-based), is continuous and asynchronous. Because of this, a number of choices had to be made with respect to the implementation of the model. Also, a number of things were observed during experimentations, that were not found in the original model. A comparison of the agent-based implementation with an "original" version was made by Boer, Kaymak and Spiering [1]. Furthermore, Cheung and Koornneef [4] did a lot of experiments with different configurations of ABSTRACTE and illustrated the differences.

One of the major observations in the agent-based implementation is that investors can place orders simultaneously and sometimes faster the market maker can handle. As a result, a queue appears, consisting of orders waiting to be executed. In the original model, the market maker *selects* investors whenever he is ready for it, so there will never be a queue. Both [4] and [13] note that learning the order flow could be advantageous for the investors, as well as for the market maker. This is exactly what this thesis is about, a subject that is not widely discussed in scientific literature about financial markets. The main research question is:

How can a market maker in an agent-based artificial stock market use information present in the order queue to set better bid and ask prices?

1.1 Methodology

We will develop and implement a model of a stock market in which the market maker takes changes in the order flow as an input parameter for price setting, thus seeing the order book as a source of information. Our starting point is an agent-based version of the market maker model developed by Das, implemented in ABSTRACTE. Technical information on ABSTRACTE can be obtained by reading Meijer [11]. We will extend this model an implement the changes. By doing a number of experiments, we analyse whether the modifications are actual improvements or not.

1.2 Brief contents

Before dealing with the details of order book characteristics, we first discuss in chapter 2 a number of market maker models that have been developed, including the model of Das.

In chapter 3, we will pass on describing ABSTRACTE as an example of an agent-based model, we will discuss the differences between the original Das model and the agent-based implementation in ABSTRACTE, and the effects of an order queue on the market prices and the perception of the fundamental value by the market maker. This leads us to a modification in the model to account for these order queue effects. We will discuss two versions of the modified model.

In chapter 4, we will develop some experiments that must make clear how well the different versions perform. Chapter 5 shows the graphical, as well as the numerical results of these experiments. The results are then compared with results from the existing implementation to get to know whether the changes are actual improvements.

This leads to the conclusion in chapter 6, which also contains suggestions for future research.

2 Market maker models

A lot of stock market models exist, most of which concentrate on the investor as the main participant, thereby giving little attention to the market maker, if he is present at all. It should be noted though, that there are numerous types of stock markets, and depending on the system, a market maker (or specialist) may be or may not be present. One example of a market in which a market maker is present, is the New York Stock Exchange (NYSE). The rest of this chapter is devoted to those stock market models, in which the market maker plays a central role.

Garman [8] was the first one who extensively studied the microstructure of financial markets and the role of the market maker in stock markets.

Since then, most models can be put into one of two main categories that have been evolved. They are:

1. Inventory models, which focus on the order-matching problem the market maker has to deal with. When the number of buy orders is not equal to the number of sell orders, he builds up an inventory. Its value is subject to price fluctuations, so the market maker faces price risk which he tries to minimize.

2. Information models, which assume that a certain percentage of traders on the market have more information than others. The market maker, who is not well informed, expects a loss because of this information asymmetry and sets its spread accordingly to compensate for the expected loss. Or, to cite Glosten and Milgrom: "*The core idea is that the specialist faces an adverse selection problem, since a customer agreeing to trade at the specialist's ask or bid price may be trading because he knows something that the specialist does not. In effect, then, the specialist must recoup the losses suffered in trades with the well informed by gains in trades with liquidity traders. These gains are achieved by setting a spread.*" [9]

From now on, we will focus exclusively on the information models, and mention a few of them.

2.1 Information models

The first model based on the idea that information asymmetry can be a reason for the bid-ask spread, was developed by Glosten and Milgrom [9]. They state that a spread is set even when there are no transaction costs, no inventory costs, cost-free short positions, and the market maker does not want to make profit. They give a trading model with one asset and a fixed order size of one unit. Only market orders can be placed. A number of investors exist, and some of them have information about the future value of the asset that the others (including the market maker) do not have. The investors arrive one by one, although an arrival process is not explicitly defined. The market maker behaves as if he operates in a competitive environment: he does not have a profit motive. The asset has a true or *fundamental* value V, defined as the risk discounted value of all expected future cash flows. Furthermore, the bid price should be set equal to the expected value of V given that the next trader would sell, and the ask price equal to the expected value of V given that the next trader would buy. The traders have a decision function Z, which (in a simplified form) is equal to the investor's perception of V, given their own preferences and information set. When $Z > P_{Ask}$, the trader buys, and when $Z < P_{Bid}$, the trader sells. They give an example with a stock that can have two possible values, V_h and V_l. The prior probablity that the value will be V_h, is π, so the expected value is $E(V) = \pi * V_h + (1 - \pi) * V_l$. Now the arrival of a sell order is a potential signal for the market maker that the value must be V_l, because the investor might be informed and his personal perception of V might then be lower than the current bid price. He then updates his own beliefs about V by using the following rule:

$$\frac{P(V_h|Sell)}{P(V_l|Sell)} = \frac{P(V_h)}{P(V_l)} * \frac{P(Sell|V_h)}{P(Sell|V_l)} \tag{1}$$

The posterior probabilities $P(V_h|Sell)$ and $P(V_l|Sell)$ serve as the prior probabilities of the bid price for the next trader.

In another model, Chan and Shelton [3] simulated a stock market in which the market maker uses reinforcement learning in order to learn its environment. With reinforcement learning, the operating environment of the agents is divided into a number of time steps. The agent observes the current state of the world, takes an action and receives a reward, depending on the action taken and the current state of the world. The action also influences the next state of the world, and thus all future rewards. The goal of the agent is to maximize the sum of the rewards in all periods. In case of this model, the enviroment is defined as the order imbalance (measured by the number of buy orders minus the number of sell orders) and market quality (measured by the spread size). The action taken consists of adjustment of the bid and ask price. The reward is a combination of profit on a transaction and market liquidity. Although the model has some unrealistic simplifications, it can serve as an excellent starting point for a learning market maker in a more complex environment.

2.1.1 Limit order market models

A number of market maker models have been developed that simulate limit order markets. For example, Foucault, Kadan and Kandel [7] developed a dynamic stock market model in which traders can choose between placing market orders and limit orders. Buyers and sellers arrive sequentially and behave as liquidity traders. They differ in their patience with regard to the delay in the execution of their orders. The authors particularly studied "market resiliency", which is according to Harris [10] how quickly spreads are corrected to the competitive level (after a liquidity shock).

Parlour [12] developed another model of a limit order market (not necessarily a stock market) that consists of two days broken down into a number of periods. Again, buyers and sellers arrive sequentially and behave according to a personal utility function. This model focuses at non-random patterns in the order book. For example, it is observed that the probability of a limit buy order in the order book is smaller when the previous order was also a limit buy order.

Cornelli and Goldreich [5] studied the information content of the order book and its impact on price setting for *initial public offerings* (IPOs) and *seasoned equity offerings* (SEOs), and showed empirical evidence that an investment bank, responsible for distributing shares among subscribers, use information from bids to set the issue price, as a result of information asymmetry. They conclude: *"We find a strong relation between the limit prices submitted by bidders and the issue price, especially from bids from large and frequent bidders. The level of oversubscription[1] has a smaller but significant effect on the issue price for IPOs."*. This is at least one example of an order book that contains information, despite a context that is rather different from ours.

2.1.2 A simulation model

Most of the previous models are only theoretic and not suitable for computer simulation, because either the full simulation process is not well defined, or some variables cannot be calculated. For example, in the model of Glosten

[1]Oversubscription to an IPO is defined as the number of shares subscribed to, divided by the number of outstanding shares.

and Milgrom, the decision function Z for the investors is some combination of the available information, current prices and personal preferences with respect to consumption versus investment. However, it is unknown how all these parts exactly relate to Z. Because Z cannot be calculated, it is impossible to build a computer simulation of this model.

A model designed particularly with simulation in mind, is the one developed by Das [6]. This model is largely based on the ideas of Glosten and Milgrom (and others), but this model concentrates on price setting by the market maker, whereas the other models often give more attention to the behaviours and decision-making of the investors. In this model, the market consists of a trading crowd and a single competitive market maker. Trade takes place by market orders of one asset. The stock has a *fundamental* or "true" value, defined as the expected value of the sum of all risk discounted future cash flows. This fundamental value V is unknown to all participants except for a fraction of α of the trading crowd, who receive a (noisy) signal $V + \widetilde{\eta}$ of the fundamental value. $\widetilde{\eta}$ is normally distributed with mean 0 and standard deviation σ_W. When the observed signal is higher than the current ask quote P_a, the investor places a buy order. When the observed signal is lower than the current bid quote P_b, he places a sell order. Otherwise, no order is placed. The other (uninformed) investors have a fixed probabilty of trading, equal to $2 * \eta$. Given that it trades, the probability of a buy order is equal to the probability of a sell order.

The trading day is divided into a number of time intervals. The fundamental value follows a jump process in which at each time interval, there is a fixed probability (often set to 0.001) that "news" is revealed. As a result, the fundamental value changes according to the equation $V^{i+1} = V^i + \tilde{\omega}(0, \sigma)$ in which $\tilde{\omega}(0, \sigma)$ is a normally distributed random variable with mean 0 and standarddeviation σ.

At the start of the simulation, the market maker receives the initial fundamental value and the fraction of informed traders α. It is assumed that the trading population remains the same during the whole simulation process. In each step, one investor is randomly selected from the population and using his decision rule, he chooses to place or not to place an order. The market maker receives the decision, updates his bid and ask prices, selects another trader, and so forth. In each step, there is a fixed probability p (usually set to 0.001) the fundamental value will change (jump) to another value, where $V_{i+1} = V_i + \widetilde{\omega}$, while $\widetilde{\omega}$ is normally distributed with mean 0 and variance σ_J^2. In all other steps, it remains the same. Whenever such a jump occurs, the market maker receives a signal that the fundamental value changed, but he does not know the size nor the direction of the jump.

The bid and ask prices are like those of a competitive market maker in equilibrium, which according to Glosten and Milgrom, are:

$$\begin{aligned} P_a = & \quad E[V|Buy] \\ P_b = & \quad E[V|Sell] \end{aligned} \tag{2}$$

The elegance of this model is that the spread size reflects the uncertainty with respect to the fundamental value to the market maker. By applying Bayesian learning techniques, he gradually reduces this uncertainty and makes the spread smaller.

Taking the bid price for example, and following the definition of expected value, and assuming V has a continuous range, this can be rewritten to:

$$E[V|Sell] = \int_0^\infty xP(V = x|Sell)\; dx \tag{3}$$

In order to calculate the equation above, V must be discretized, thus becoming the equation:

$$E[V|Sell] = \sum_{V_i=V_{min}}^{V_i=V_{max}} V_i P(V = V_i|Sell) \tag{4}$$

If we apply Bayes' rule to the conditional probability above and rearrange, we get:

$$E[V|Sell] = \frac{1}{P(Sell)} \sum_{V_i=V_{min}}^{V_i=V_{max}} V_i P(Sell|V = V_i)P(V = V_i) \tag{5}$$

Now let us look to each of the probabilities separately. Given a fundamental value V_i, the probability of a sell order for uninformed traders is just η and independent of V, while informed traders only place a sell order if the observed signal of V_i is smaller than the bid price, e.g. $\widetilde{\eta}(0, \sigma_W^2) < P_b - V_i$. In the form of an equation, this is:

$$P(Sell|V = V_i) = (1 - \alpha)\eta + \alpha P(\widetilde{\eta}(0, \sigma_W^2) < P_b - V_i) \tag{6}$$

The first term in this equation reflects the probability that an uninformed trader would sell given a certain fundamental value V_i, while the second term does the same thing for a noisy informed trader. We use the equation above to fill in the prior probability of sell order:

$$P(Sell) = \sum_{V_i=V_{min}}^{V_i=V_{max}} P(Sell|V = V_i)P(V = V_i)$$

$$= \sum_{V_i=V_{min}}^{V_i=V_{max}} \big[\alpha P(\widetilde{\eta}(0, \sigma_W^2) < P_b - V_i) + (1 - \alpha)\eta\big] P(V = V_i) \tag{7}$$

If we combine equation 5 with 6 and 7, we obtain the final formula for the bid price:

$$P_b = \frac{\sum_{V_i=V_{min}}^{V_i=V_{max}} V_i \left[\alpha P(\widetilde{\eta}(0,\sigma_W^2) < P_b - V_i) + (1-\alpha)\eta\right] P(V=V_i)}{\sum_{V_i=V_{min}}^{V_i=V_{max}} \left[\alpha P(\widetilde{\eta}(0,\sigma_W^2) < P_b - V_i) + (1-\alpha)\eta\right] P(V=V_i)} \tag{8}$$

The corresponding formula for the ask price is similar:

$$P_a = E[V|Buy] = \frac{1}{P(Buy)} \sum_{V_i=V_{min}}^{V_i=V_{max}} V_i P(Buy|V=V_i) P(V=V_i)$$

$$= \frac{\sum_{V_i=V_{min}}^{V_i=V_{max}} V_i \left[\alpha P(\widetilde{\eta}(0,\sigma_W^2) < V_i - P_a) + (1-\alpha)\eta\right] P(V=V_i)}{\sum_{V_i=V_{min}}^{V_i=V_{max}} \left[\alpha P(\widetilde{\eta}(0,\sigma_W^2) < V_i - P_a) + (1-\alpha)\eta\right] P(V=V_i)} \tag{9}$$

Finally, the market maker keeps track of a probability density estimate of all prior probabilities $P(V = V_i)$, stored in a discrete form in a vector from $E[V] - 4\sigma_J$ to $E[V] + 4\sigma_J$, with steps of 1 cent, or an arbitrary step size. Initially, and immediately after a jump, all values are normally distributed around the center of the vector. After each order, the algorithm uses Bayes' rule to calculate the posterior probabilities for all values in the vector, which are used as the prior probabilities for the next update. Taking market sell orders for example:

$$P(V = V_i|Sell) = \frac{P(Sell|V = V_i)P(V = V_i)}{P(Sell)} \tag{10}$$

The probabilities $P(Sell|V = V_i)$ and $P(Sell)$ are calculated using respectively equations 6 and 7. $P(V = V_i)$ is the current vector value. After each processed order, the expected value of the vector more and more converges to the real fundamental value, while uncertainty decreases.

3 Simulating in an agent-based environment

Agent-based models are models that consist of a number of entities, or agents, that interact with each other. Each agent has its own behaviour and perception of the "world". Ideally, all agents should act simultaneously rather than serially, something that is not easy to accomplish, except in a distributed computing environment. One of the advantages of agent-based simulation is that it enables someone to analyse complex situations where many objects exist parallel to each other.

A real stock market is a complex and chaotic environment where a lot of people operate simultaneously and interact with each other. Therefore, a simulation model such as the one by Das where investors arrive neatly one-by-one, is too much of a simplication. Agent-based simulation environments like ABSTRACTE try to overcome this limitation. Among the models supported by ABSTRACTE, there is an implementation of the model by Das. However, the fact that this is now agent-based not only has its consequences for the results and the performance of the participants in the market, the design of the model had to be modified on a certain points as well. A comparison of both models was made by Cheung and Koornneef [4]. The most important differences are:

1. The population size had to be explicitely defined, since all investors (and the market maker) were modelled as seperate agents. In ABSTRACTE, a population can consist of any number of investors. In the original model, only the fraction of informed traders in the population was known, but not the total number of investors.

2. Instead of waiting indefinitely until they are selected to trade, investors have a decision moment every n milliseconds. However, a lower bound on n exists, determined by the computer speed and the operating system, which distributes the available processor time among the running processes.

3. In the original model, the market maker has a special update for the bid ask spread when the selected investor chooses to place no order. In ABSTRACTE, the market maker does not know the decision moments of the investors, nor does he receive a notification if an investor decides not to place an order. Instead, the market maker keeps track of the number of milliseconds that passed since the last order was received. When this exceeds a certain limit, he applies his no-order rule.

4. The moment of jumps in the fundamental value is determined by a uniformly distributed variable that gives the number of seconds between each two jumps. Originally, between every two orders, there was a fixed probability of a jump.

5. The most interesting feature of the agent-based model is that it is possible that multiple investors place an order simultaneously, before the market maker gets the chance to look at just one of them. The result is that an order queue (or order book) appears with orders waiting to be cleared. This would be impossible in the original model, where no trade occurs unless the market maker is ready for it, and only allows one investor at a time to place an order.

An important consequence of the last difference is that a delay may exist between the placement and the execution of an order, because at the time of placement, there may be a queue of waiting orders that will be handled first. Within this time span, the bid and ask prices can change a number of times as a result of handled orders. To give an example, suppose at time t, a jump occurs in the fundamental value such that $V < P_b$. There are, say, 10 perfectly informed investors who place a sell order immediately afterwards. Suppose that between t and $t + s_1$, a queue of 10 sell orders has formed. At that moment, the information that can be extracted from the order book, is as follows: according to these investors, the fundamental value at time t is lower than the bid price at time t. At time $t + s_1$, the market maker wakes up and picks the first order from the queue. Because it is a sell order, he concludes his current bid price is too high. Then at time $t + s_2$, he looks at the next order, which is again a sell order. He interprets this order as a signal that his current bid price is still to high, but the true interpretation is that the bid price was too high at time t. Then with each order, he keeps lowering his prices and he ends up far beyond the fundamental value.

This has also negative consequences for the unlucky investors who are served in the later part of the queue. They face huge losses, because prices have dropped rapidly since they placed their orders. While selling could have been a good decision at the moment the decision was made, at the moment of execution the situation is much different. Because of the market order mechanism, the investors have no control over the execution price of their orders.

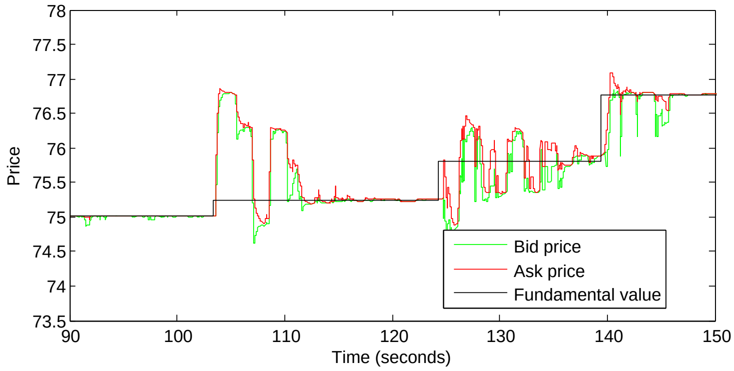
Figure 1 shows that the situation mentioned above regularly takes place during a simulation. After every jump, the bid and ask prices inially exceed the shock size, but then return to a level near the fundamental value and stabilize. As the number of (informed) investors grows, the overreaction to a fundamental value shock increases.

3.1 A modified market-maker model

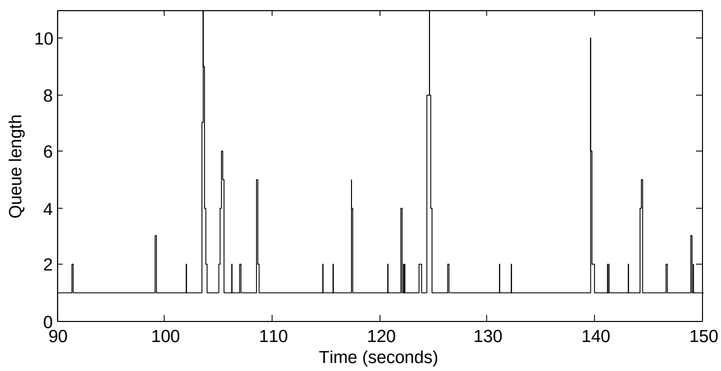
A solution to this problem is to let the market maker investigate all orders in the queue simluateously. If at a certain moment the order book contains six sell orders, it does not mean that the bid price is further away from the fundamental value than if the order book would contain three sell orders, because investors (in this model) do not place more orders when the difference between the bid price and the fundamental vlaue increases. However, it gives a stronger evidence that the fundamental value lies outside the bid-ask spread.

So let us change the market maker algorithm in order to let it investigate more than one order at a time. In order to do this, we need to change the Bayesian update rule for the vector of prior probabilities. There is no need to change the price setting equations ($P_b = E[V|Sell]$ and $P_a = E[V|Buy]$), so we let this part unmodified.

A simple solution is to count the majority of orders in the order book. If the number of sell orders exceeds the number of buy orders, we update the bid price, pretending that one sell order is received, and vice versa. If the number of sell orders is equal to the number of buy orders, we pretend that



(a) Bid and ask prices



(b) The order queue

Figure 1: Detail of a simulation with 10 perfectly informed and 10 uninformed investors

no order is received. The advantage of this method is that little changes have to be made at the implementation level, but the drawback is that this method does not make optimal use of the information present in the order book, for example: the arrival of twenty sell orders and one buy order gives a more reliable signal about the fundamental value than the arrival of two sell orders and one buy order, but in both cases this method acts the same.

Another method that uses this extra information, needs that the update rule for the prior probabilities as given in equation 10 is modified like below:

$$P(V = V_i|Orderbook) = \frac{P(Orderbook|V = V_i)P(V = V_i)}{P(Orderbook)} \tag{11}$$

Here, *Orderbook* denotes the (unordered) list of arrived orders, for example: *Sell, Sell, Buy.* Now the equation can be expanded to:

$$P(V = V_i|Orderbook) = \frac{\binom{s+b}{b} P(Sell|V = V_i)^s P(Buy|V = V_i)^b P(V = V_i)}{\binom{s+b}{b} \; P(Sell)^s P(Buy)^b} \tag{12}$$

where s denotes the number of sell orders, and b the number of buy orders placed. So the probability of a certain order book is equal to the probability that within a total of $s + b$ orders placed, this combination of buy and sell orders occurs. This method ensures that the larger a majority of orders within a total is, the more the probability mass will shift to one side. Note that $\binom{s+b}{b}$, $P(Sell)$ and $P(Buy)$ are the same for each value in the vector. Because the whole vector is normalized (the sum of all elements in the vector must be equal to 1), we do not need to calculate these values. We use this fact and at the same time rewrite $P(Buy|V = V_i)$ and $P(Sell|V = V_i)$ by using equation 6 to get the following update rule:

$$P(V = V_i|Orderbook) = \tag{13}$$

$$\left[\alpha P(\widetilde{\eta}_{0,\sigma_W^2} < P_b - V_i) + (1-\alpha)\eta\right]^s \left[\alpha P(\widetilde{\eta}_{0,\sigma_W^2} < V_i - P_a) + (1-\alpha)\eta\right]^b P(V = V_i)$$

There is, however, one underlying assumption: we earlier wrote the probability of all orders together as the product of all separate orders. This only holds if the events that generate the orders, are independent. This assumption is questionable, because the arrival of orders depends on the fundamental value: if multiple investors are operating in the same market and one sell order is received, it is likely that the next order will also be a sell order, rather than a buy order. However, we do not try to overcome this limitation and just assume independence of these events.

4 Experimental design

We run a number of simulations in order to determine whether the presented modifications lead to improved performance of the market maker.

Performance can be defined as:

1. the ability to track the fundamental value
2. the ability to operate successfully in a competitive environment (i.e. earning zero profit).

Our goal is to find a solution to the problems that arise when multiple investors place orders asynchronously.

4.1 Methodology

All experiments are done using ABSTRACTE. To make comparison easier, we run each configuration threefold: first in an agent-based Das implementation with no changes to the order book handling mechanism (this one processes one order at a time), then in an implementation of the method that looks to the majority of orders, and finally in another implementation that uses the modified update rule for the prior probabilities.

Implementation issues Initially, the ABSTRACTE framework could not cope with market makers that needed the whole order book to make their decision about the market prices. Although all received orders ended up in a queue and a limit order book would be created for limit order markets, in the case of a market-order-only model, only the last order placed was passed on to the part of the market maker that was responsible for updating the bid and ask prices. This has been changed, and now the whole order queue is available to the price setting algorithm. The market maker can choose whether or not to execute one or more of these orders and sets an execution flag for each order accordingly. The following changes have been made in order to make this possible:

1. Changed class `LOBrecord` in package `asmbasis.ontology`. It now has a field `executionTimeStamp`, with proper setter and accessor functions.
2. Added class `OrderBook` in package `marketplace.datamanagement`. This class takes care of storage and retrieval of the order book elements into and from a linked list, as well as some useful functions like counting the number of buy and sell orders in the order book.
3. Changed in package `asmbasis.agents.descriptors` class `MarketMakerDescriptor` and added a reference to `OrderBook`.
4. Changed in package `marketplace.agents.marketmakers` class `MarketMakerBehaviour`. This is the part that receives messages from investors and other agents. As a consequence, it spends most of its time waiting and peeking at the message queue. It often occurs that nothing

happens for a period of time, and suddenly a number of messages have arrived. Before the changes, it picked the first message, determined its type, invoked the right action (such as updating the market prices) and then investigated the second message, and so on. Now, it picks the first message, determines its type, and if its happens to be an order, it keeps reading messages until a message of another type is encountered, the queue is empty or a certain maximum number of orders (if specified) is achieved.

5. Changed in package `marketplace.agents.marketmakers` the classes `OrderMatchingBehaviour`, `UpdateBidAskBehaviour`, `AbstractBidAskStrategy` and all classes descending from it to work with `OrderBook`s instead of `LOBrecord`s. `OrderMatchingBehaviour` now also adds all received orders to the order book, and removes executed ones from it. The bid-ask strategy implementations are now responsible for setting the `executionTimeStamp` for each handled order. Each order is kept in the order book until the `executionTimeStamp` is set.

6. Changed in package `marketmakerstrategies.bidaskquote` class `ExtendedGlostenMilgrom`, which is responsible for updating the bid and ask prices. The price-setting function now takes a parameter of type `OrderBook`, instead of a `LOBrecord` object. Also, the part that updates the values in the probability density vector now works according to equation 13.

7. Added in package `marketmakerstrategies.bidaskquote` class `EGMmajority`, that is basically a copy of `ExtendedGlostenMilgrom`, but instead counts the majority of orders.

4.2 Base case

The experiments take place in a market environment with a number of investors and one market maker. The order-generating speed of the investors (*patience*) as well as the maximum waiting time for the market maker (*no-order time*) is adjusted to such a level that the investors do not overwhelm the market maker with orders because he is too slow. In all experiments, the no-order time is set to 120 milliseconds, which turns out well. The patience values are different for all experiments, because the best value depends on the number and type of investors. A (nearly) optimal value is chosen for all cases individually by trial and error.

The fundamental value process is exogenous and the same for all experiments. It is initialized at 75 and follows a normally distributed jump process $\widetilde{\omega}$ with mean 0 and standard deviation 0.5. Between two jumps, it is constant for a period uniformly distributed between 5 and 30 seconds. This should give enough time to the market maker to stabilize its bid and ask prices. Every experiment lasts for 300 seconds.

Immediately after each jump, all noisy informed perceive their own signal of the fundamental value. This signal $(V + \widetilde{\eta})$ is normally distributed around the new fundamental value with standard deviation 0.05. If the traders are perfectly informed, they receive the fundamental value without noise. The uninformed traders generate buy or sell orders with probability (η) 0.3, so for each decision moment of an uninformed investor, there is a 30% probability that a buy order will be placed, 30% for a sell order and 40% that no order will be placed. The value of α can be deduced from the number of investors in the market.

4.3 Experimental settings

The following populations are simulated: 10 informed traders and 10 random traders, 10 informed traders and 20 random traders, 50 informed traders and 50 random traders, and 150 informed traders and 150 random traders. These simulations are first carried out with noisy, and then with perfectly informed traders.

The reason that just these populations are chosen, is that the number of investors must be large enough to give interesting results. If there are only two investors, there would be no problem with regard to the order book, because given that the market maker responds fast enough to generated orders, the maximum order book size that would occur, is two. With 10 informed traders in the population, order book effects are large enough to be observed. The case with 10 informed and 20 random traders is chosen to make it more difficult for the market maker, because the extra random traders add more randomness to the order flow. All larger populations are simply to test up to which point the market maker gives sensible results.

Within all these settings, a comparison will be made of the original algorithm (from now on called *version 0*), the version that pretends that the whole order book is one order (*version 1*) and the version with the modified update rule for the vector of prior probabilities (*version 2*).

5 Simulation results

This section contains the results of all experiments in the form of graphs with the bid and ask prices and the fundamental value displayed over time. These plots easily show how well the market prices align with the fundamental value, as well as the spread size. We also compare the profitability of the different versions. In one case, we look at the size of the order queue to illustrate the differences between the versions with respect to order handling.

5.1 Bid and ask prices in the simulations

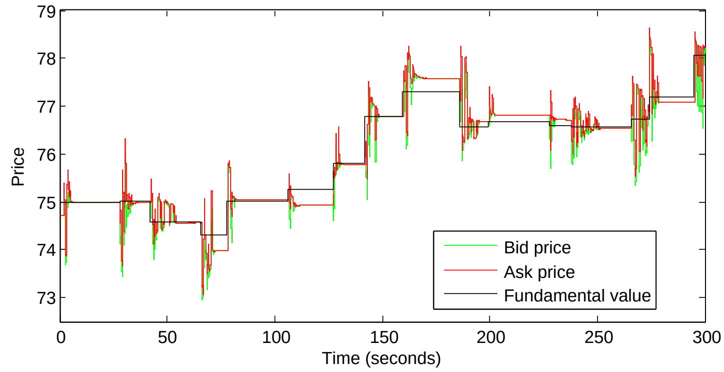
Figure 2 shows the bid and ask prices over time in the three versions with a total population size of 20 investors. Figure 2(a) and 2(b) make clear that after every shock, the market maker needs some time to properly adjust to the new fundamental value, but eventually, he finds it and stabilizes. Note that in version 1, the price movements are somewhat smaller than in version 0. Version 2 shown in figure 2(c) however, almost immediately makes the proper adjustment and once it has found the fundamental value, it never gets out of balance.

Figure 3 with perfectly informed traders shows similar behaviour, but here the bid and ask prices stabilize at the exact fundamental value, because of the perfectly informed traders.

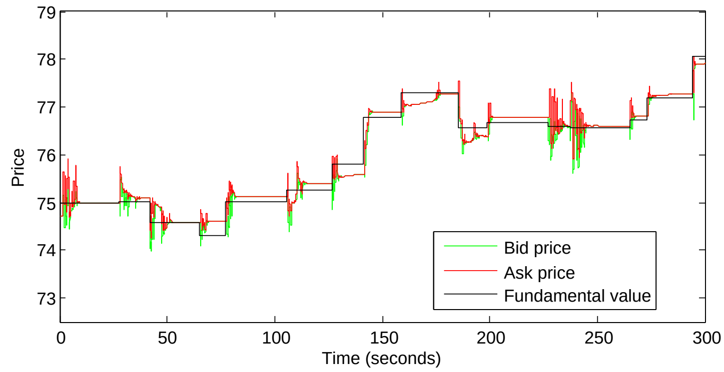
In figure 4, there are more uninformed than informed traders. This is a challenge for the market maker, because now an order more likely contains noise than information about the fundamental value. All three versions have difficulties with this situation: it takes a longer time to find the fundamental value, if it is found at all. Each jump is followed by a period of instability. Furthermore, in version 0, the amplitude of the price movements after a shock is greater than it was with 20 investors. A population size of 30 investors like in this simulation comes close to the maximum number of investors in which this version gives sensible results.

The outcomes with perfectly informed traders is shown in figure 5. The results here are similar to the previous variant, but this population seems to be less difficult to cope with. Most of the time, stability is found, or at least a movement towards stability, although it takes some time to find the proper value. However, version 2 needs notably less time to find this value than its predecessors. Furthermore, the amplitude of the price shocks are smaller in this version.

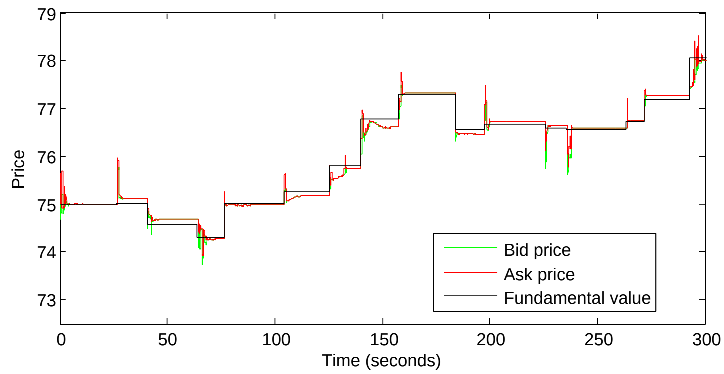
Figure 6 shows the situation where 100 investors are trading with the market maker simultaneously. This population is way too big for the market maker from version 0. Figure 6(a) shows that after every jump, the bid and ask prices move away from the starting point almost with light speed, which is certainly a result of the fact that almost all informed investors place the same order within a short time span. The reason that it does not move any further but stays the same, is that it has reached the limit of the vector of prior probabilities. The size of this vector is from $E[V]-4\sigma_W$ to $E[V]+4\sigma_W$, where σ_W is equal to the standard deviation of a jump in the fundamental value. The probabilities of all values outside the vector are assumed zero. Because σ_W is equal to 0.5, he cannot move its prices further upwards or downwards than the expected value of V before the jump plus or minus 2. Therefore, the prices move to the end of the vector and stay there until the next jump. On the contrary, the market makers that handle order books can cope well in this situation. Particularly the market maker from



(a) Version 0

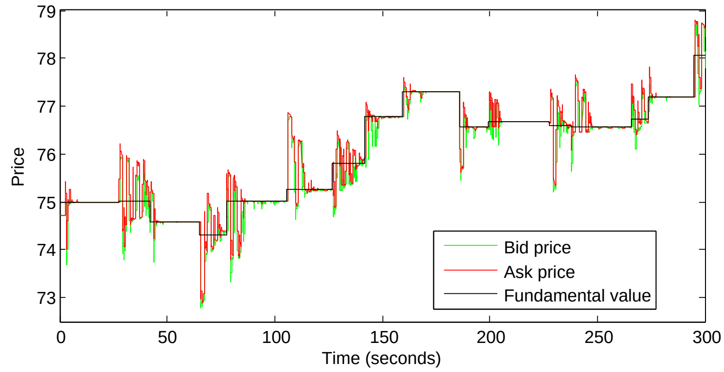


(b) Version 1

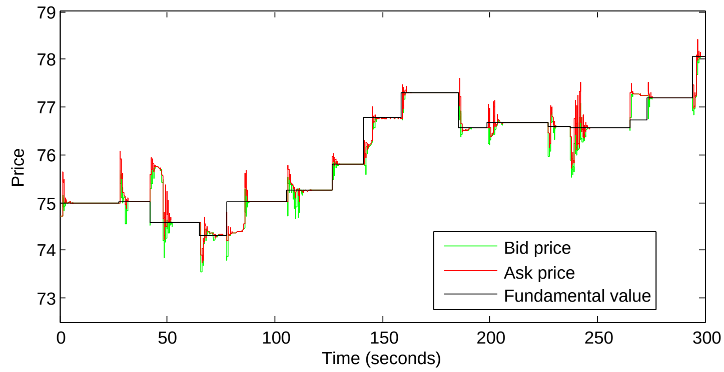


(c) Version 2

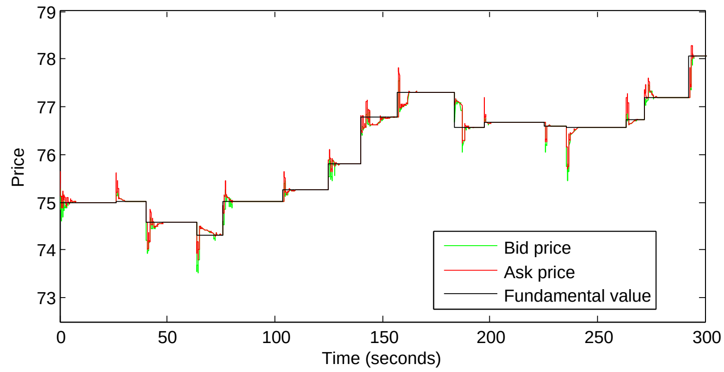
Figure 2: Bid and ask prices with 10 noisy informed and 10 uninformed investors



(a) Version 0

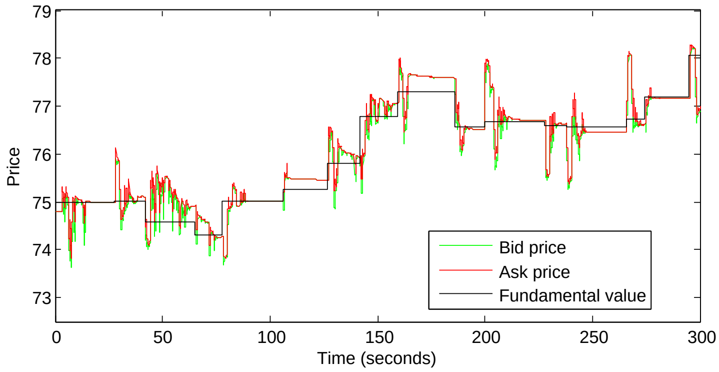


(b) Version 1

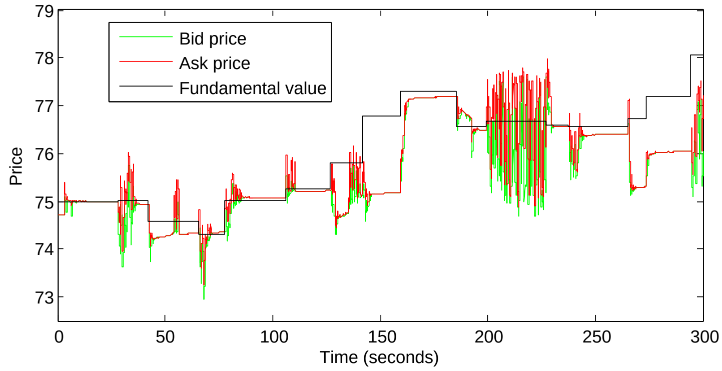


(c) Version 2

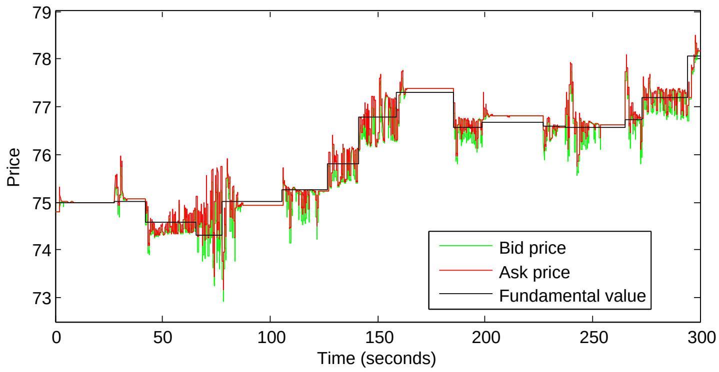
Figure 3: Bid and ask prices with 10 perfectly informed and 10 uninformed investors



(a) Version 0

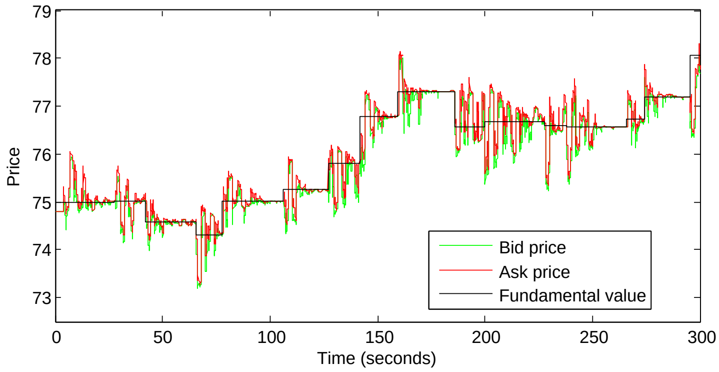


(b) Version 1

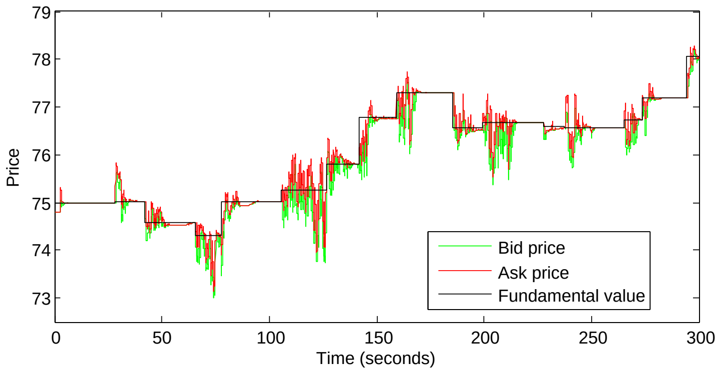


(c) Version 2

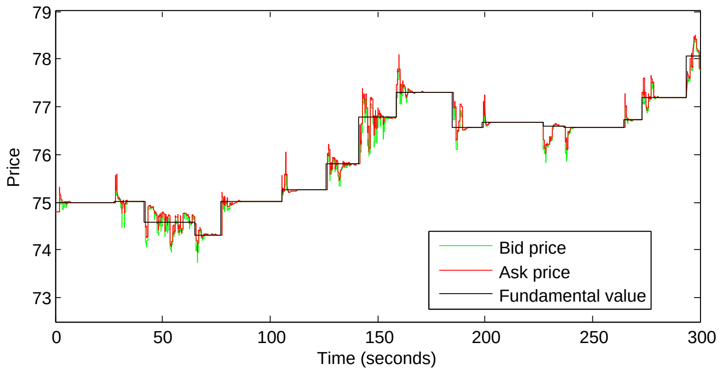
Figure 4: Bid and ask prices with 10 noisy informed and 20 uninformed investors



(a) Version 0

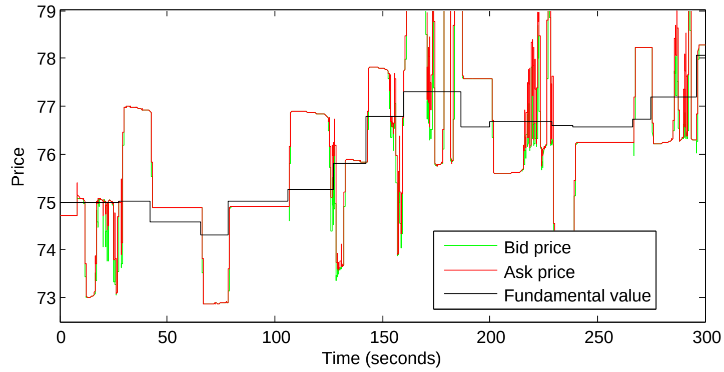


(b) Version 1

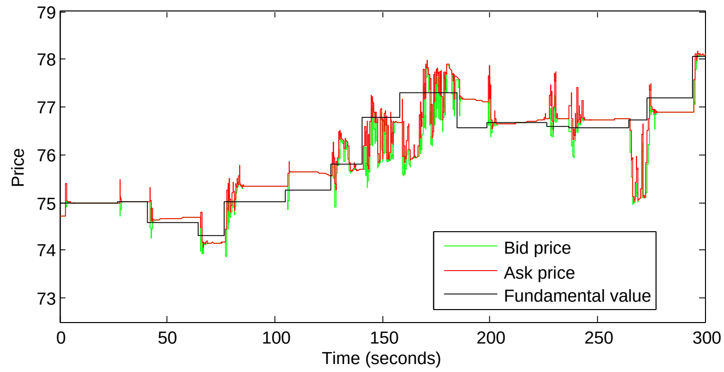


(c) Version 2

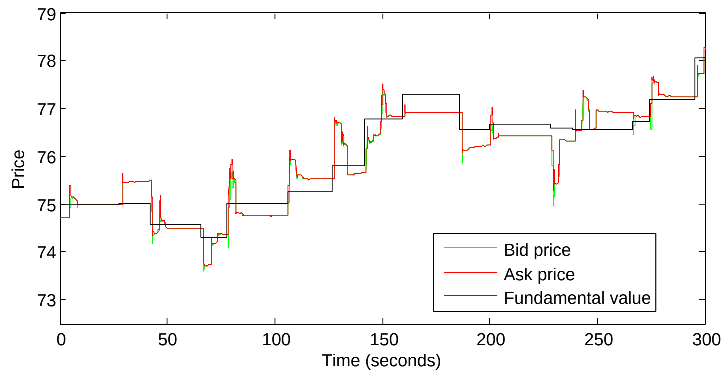
Figure 5: Bid and ask prices with 10 perfectly informed and 20 uninformed investors



(a) Version 0

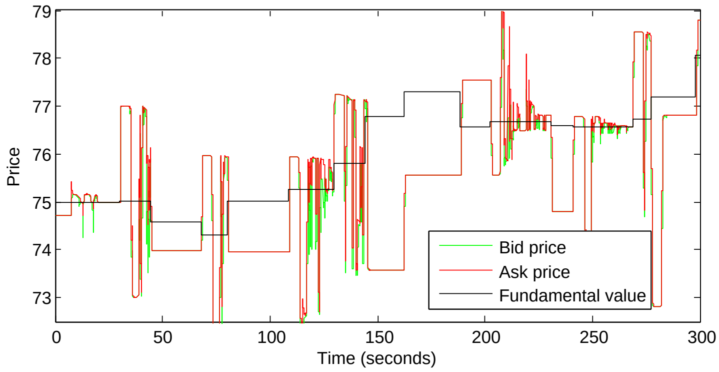


(b) Version 1

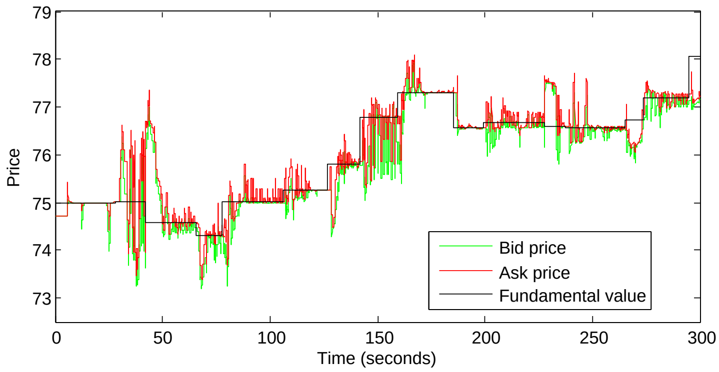


(c) Version 2

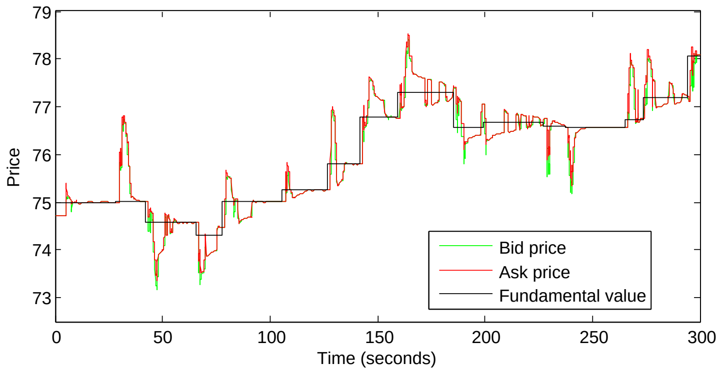
Figure 6: Bid and ask prices with 50 noisy informed and 50 uninformed investors



(a) Version 0

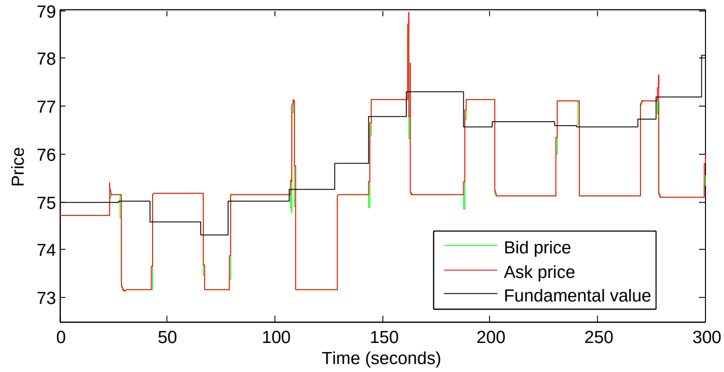


(b) Version 1

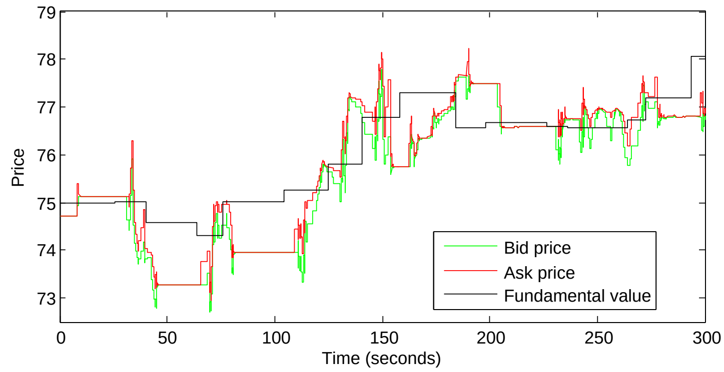


(c) Version 2

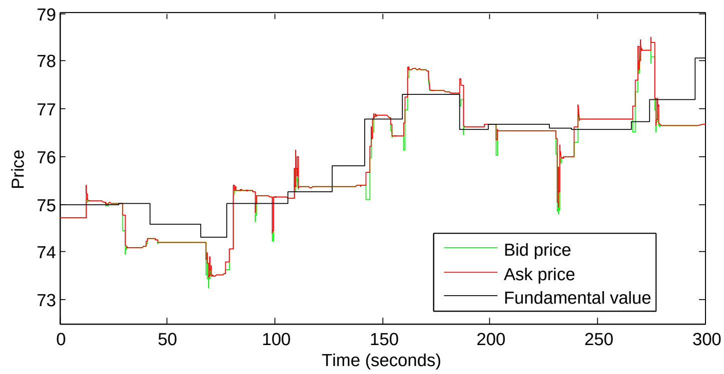
Figure 7: Bid and ask prices with 50 perfectly informed and 50 uninformed investors



(a) Version 0

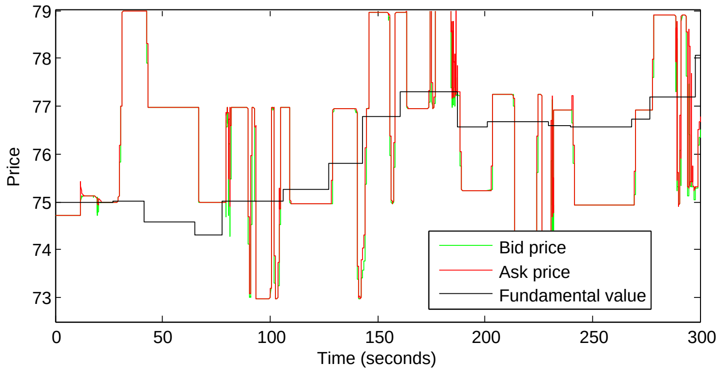


(b) Version 1

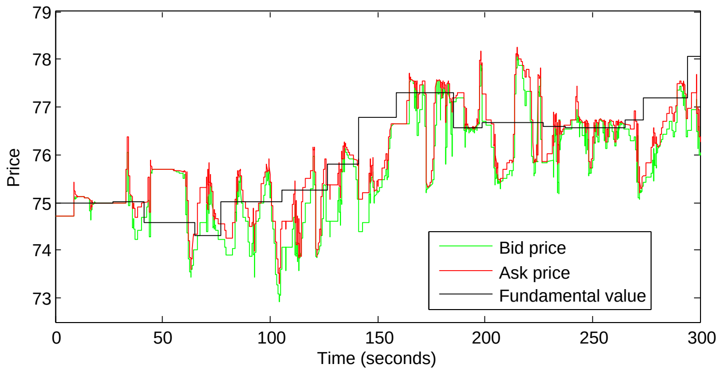


(c) Version 2

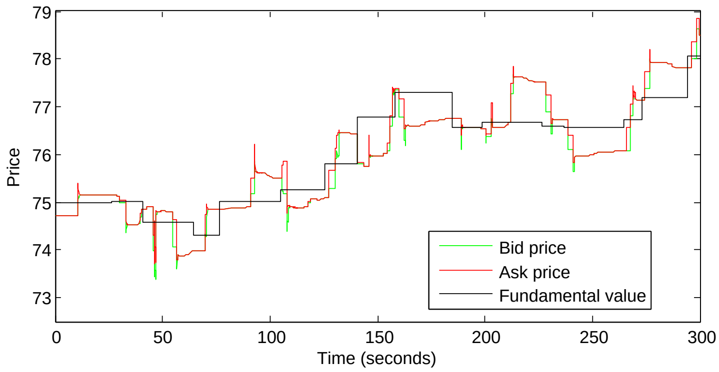
Figure 8: Bid and ask prices with 150 noisy informed and 150 uninformed investors



(a) Version 0



(b) Version 1



(c) Version 2

Figure 9: Bid and ask prices with 150 perfectly informed and 150 uninformed investors

version 2 finds the fundamental value fast with little instability. However, it sometimes stabilizes at a value that is different from the fundamental value.

The same population size but with perfectly informed investors in figure 7 is similar, but the tracking of the fundamental value by the versions 1 and 2 is somewhat more accurate.

Even with 150 noisy informed and 150 uninformed traders, the market maker from version 2 performs surprisingly well, as shown in figure 8. Although it not always finds the fundamental value, it is very stable. This also holds for version 1, although it sometimes stabilizes at a wrong value. Also, the spread size tends to be fairly large most of the time. Version 0 failes to give sensible results at all.

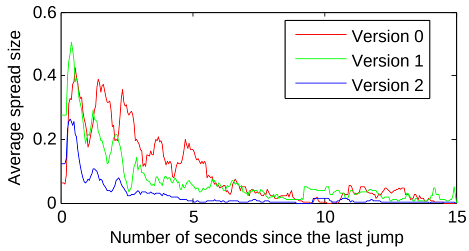
With perfectly informed investors, as shown in figure 9, the results are very similar. One important observation is that in version 2, the market maker often loses track of the fundamental value. It converges very slowly towards it, and when it is reached, it often misses its goal.

Although at least one version can cope with a population of 300 investors, the population size cannot be increased unlimitedly, because eventually, computational problems will occur. For example: if the order book would contain 400 orders, it has to calculate probabilities to the power of 400, which is so close to zero that it will output zero for almost each probability, because of the way it computes floating-point calculations. These numerical difficulties might also explain why the market maker captures the fundamental value less accurate than in the previous simulations. If it is really necessary to work with populations this big, tricks need to be applied like taking the square root or the logarithm of the number of orders.

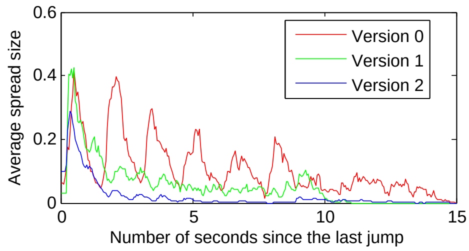
5.2 Analysing the size of the bid-ask spread

The size of the bid-ask spread should be a measure for the uncertainty of the market maker with regard to the fundamental value. Figure 10 shows for all simulations how the spread size evolves after a jump. More specifically: it shows the average spread size over all times when a jump occurred n seconds ago, where n is denoted at the horizontal (x) axis. For instance, if a jump occurs at $t = 17.4$, $t = 28.0$, $t = 30.5$ and $t = 42.7$, the value at $x = 5$ shows the average spread size from $t = 22.4$, $t = 35.5$ and $t = 47.7$. What we can learn from these plots is how the uncertainty of the market maker with regard to V decreases after a jump.

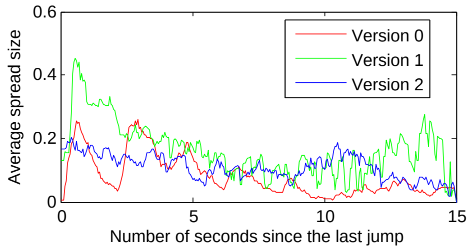
As expected, the more time has passed since a jump, the smaller the spread becomes. A number of things are noteworthy here. Firstly, version 2 has overall the smallest spreads, where version 0 seems to have the biggest spreads if the population is relatively small, and version 1 if the population is big. Smaller spreads are better, but only when the fundamental value lies between the bid and the ask price. Secondly, the average spread has peaks and dips in some simulations, most notably in the line of version 0 in figure



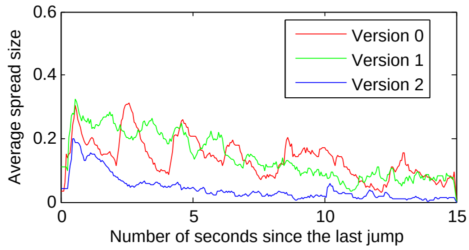
(a) 10 noisy informed and 10 uninformed investors



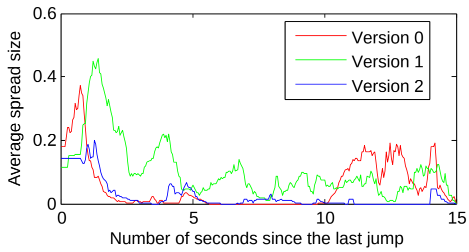
(b) 10 perfectly informed and 10 uninformed investors



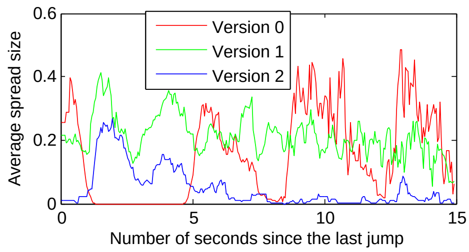
(c) 10 noisy informed and 20 uninformed investors



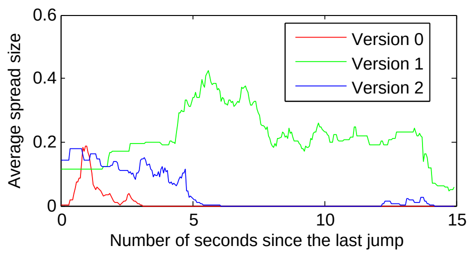
(d) 10 perfectly informed and 20 uninformed investors



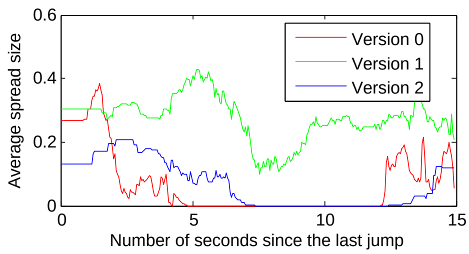
(e) 50 noisy informed and 50 uninformed investors



(f) 50 perfectly informed and 50 uninformed investors



(g) 150 noisy informed and 150 uninformed investors



(h) 150 perfectly informed and 150 uninformed investors

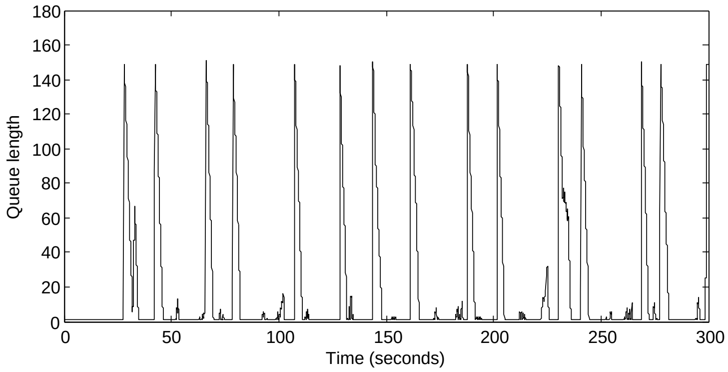
Figure 10: Average spread size as a function of the number of seconds that passed since the last jump

10(b). This is probably a result from the oscillations in the market prices that occur before they stabilize. An ideal picture would be a line that first rises quickly and then declines steadily, because the market maker experiences much uncertainty short after a jump, but this is removed gradually as more orders are received and the market maker learns the fundamental value. An example of a line that closely resembles this ideal picture is the one from version 2 in figure 10(b). Thirdly, version 2 often gives very small spreads even with many investors. As we have seen in the bid ask plots, the maker maker not always succeeds in setting the right prices. So if he fails and his spread is close to zero, he loses money to the informed investors. The last interesting observation is that in some simulations, especially those with many investors, the spread size first declines and approaches zero, but then suddenly rises again. Examples of this can be seen in figure 10(f) and 10(h). When this is the case, the prices have first reached a stable level, but then the equilibrium is disturbed. It is clear that such a situation is not desirable.

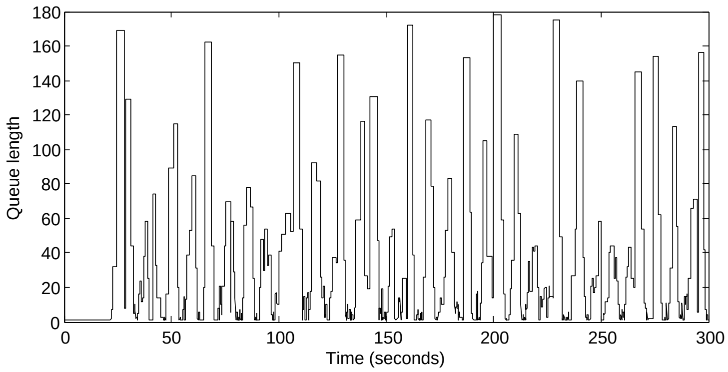
5.3 Looking at the order queue

It is interesting to see how the size of the order book fluctuates over time. Figure 11 shows this for the experiment with 150 noisy and 150 uninformed investors. Enormous peaks occur after every fundamental value jump, because all informed investors suddenly place their orders. Between two jumps, the order book size is usually much lower. The market maker from version 0 handles all orders one by one, making the order book size decrease gradually, while the market maker from version 1 and 2 handle all orders in one go. This takes some time, so the queue size remains high for a longer time than in version 0, but then drops immediately to (almost) zero. Clearly it takes more time in version 1 than version 2, because this picture is more dense. However, the shape of the picture is directly affected by the value of patience for the investors. The lower the patience, the more orders are placed in the same period of time, so the harder the market maker must work in order to clear the order queue.

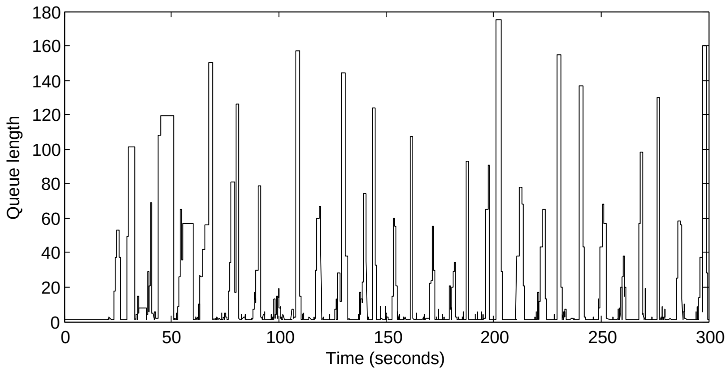
The values of patience for all simulations is shown in figure 12. These values have shown to give good results for the simulations, but an 'optimal' value is hard to find, if it exists at all, because a higher value lets the market maker work slower, but not necessarily more accurate. Furthermore, a value that works well on one computer, may give bad results on another (slower or faster) machine. So we must not draw too many conclusions about these particular values.



(a) Version 0



(b) Version 1



(c) Version 2

Figure 11: The size of the order book during the simulations with 150 noisy informed and 150 uninformed investors

population	version	patience
10 noisy	0	1000
informed and	1	800
10 uninformed	2	800
10 perfectly	0	1500
informed and	1	800
10 uninformed	2	800
10 noisy	0	2000
informed and	1	1200
20 uninformed	2	800
10 perfectly	0	2000
informed and	1	1200
20 uninformed	2	1200
50 noisy	0	4000
informed and	1	2400
50 uninformed	2	3000
50 perfectly	0	4000
informed and	1	2400
50 uninformed	2	2400
150 noisy	0	20000
informed and	1	7000
150 uninformed	2	9000
150 perfectly	0	10000
informed and	1	7000
150 uninformed	2	9000

Figure 12: Values of *patience* for the different simulations

population	version	#buy	#sell	wealth	#shares	profit
10 noisy	0	2263	2272	536.01	-9	-165.99
informed and	1	2354	3034	52094.58	-680	-945.42
10 uninformed	2	2687	2915	17198.75	-228	-598.93
10 perfectly	0	1407	1496	6696.88	-89	-245.12
informed and	1	2532	2854	24397.60	-322	-718.40
10 uninformed	2	2664	2847	14154.56	-183	-130.42
10 noisy	0	1657	1518	-10762.93	139	79.07
informed and	1	1912	3123	91484.21	-1211	-2973.79
20 uninformed	2	3509	3446	-4674.90	63	239.10
10 perfectly	0	1428	1477	3767.06	-49	-54.94
informed and	1	2113	2722	46285.69	-609	-1216.31
20 uninformed	2	2501	2571	5319.53	-70	-140.47
50 noisy	0	2728	2852	8348.92	-124	-1323.08
informed and	1	3875	5236	102835.62	-1361	-3322.38
50 uninformed	2	3747	4004	19233.96	-257	-812.04
50 perfectly	0	2135	3299	87257.52	-1164	-3534.48
informed and	1	3214	4208	75697.39	-994	-1834.61
50 uninformed	2	4196	4605	30160.22	-409	-1741.78
150 noisy	0	1723	2350	48488.26	-627	-417.74
informed and	1	3972	5356	104069.49	-1384	-3882.51
150 uninformed	2	4015	3649	-28422.92	366	125.08
150 perfectly	0	2957	3428	36505.79	-471	-232.21
informed and	1	3589	5244	125704.71	-1655	-3385.29
150 uninformed	2	3304	3793	36709.35	-489	-1432.65

Figure 13: Various statistics

5.4 Wealth of the market maker

One of the primary goals of the market maker is that it must perform in a competitive environment, so we want it to make no profit or loss. However, when choosing his actions, the market maker does not take its own wealth or inventory into account. Its decisions are only based on its perception of the fundamental value. Figure 13 shows in the rightmost column that it makes a loss in all but three simulations. But fortunately, within a total of around 7000 orders, it is a very small loss. The other columns give the number of buy and sell orders placed during the simulation, and the amount of money and the number of shares possessed by the market maker at the end of the simulation. In almost all cases, it ends up with a lot of money and a short position in his stock. Note that it almost always makes the biggest loss in version 1. But overall, it does its job quite well in terms of profitability in all versions.

5.5 Determining the best solution

It is clear that taking the order book as an input for determining the bid and ask prices leads to better results. In all simulations, both versions 1 and 2 track the fundamental value better than version 0.

Version 2 gives even more accurate results than version 1, because it makes use of the order book in a more intelligent way. It is able to make the right adjustment to its prices, particularly when the number of investors in the market is not too high. It often gives the most results, too.

The only drawback to version 2 is that the spread is often very small or zero. If the fundamental value does not lie between the bid and the ask price, which frequently occurs with many traders, this would mean a loss to the market maker. The other versions usually give a bigger spread. However, we have seen that at the end of the simulation, version 2 does not make bigger losses than the other versions.

With regard to profitability, the differences between the three versions are less significant. The measured profits are taken from individual simulations, so the height may be affected by factors such as the movements of the fundamental value. However, version 1 has by far the biggest loss, the largest short position and the highest amount of money at the end of all simulations except for one. In other words, this version complies least with the zero-profit motive of the market maker.

With everything taken into account, we choose version 2 as the best solution because of its accurateness and stability.

6 Conclusion

We showed how a market maker in an artificial stock market could use the order book to its advantage. We suggested a modification to the market maker algorithm based on the model by Sanmay Das in ABSTRACTE. If there are many investors in the market, the model before the change had trouble setting the right prices because the size of the order book was big and it did not use it in an intelligent way. The modified market maker comes in two versions, both of which show better performance than the previous one in terms of tracking the fundamental value of the stock. Particularly the second version of the modified algorithm turned out to be more accurate and very stable.

6.1 Suggestions for future research

There are a number of subjects that may be suitable for future research. Firstly, there may be more information present in the order book that could be used by the market maker. For example, the size of the order book closely reflects the number of informed traders in the market. By remembering the maximum size of the order book after each jump for example, it may be possible to learn the fraction of informed traders in the population. Currently, the fraction is given as a parameter on initialization. Furthermore, a sudden increase of the order book size is often a signal that the fundamental value has changed. Currently, the market maker receives a notification if a jump has occurred, but by using the order book, he could also obtain this information.

Secondly, order book handling could be applied to other market maker models. There are very few models in which an order book is present at all: in most models the investors are selected to trade at random or by round robin scheduling. The market maker then immediately executes the order, so there is no queue or order book. An order book handling mechanism could lead to dramatically different results in those models.

References

[1] K. Boer, U. Kaymak, and J. Spiering. From discrete-time models to continuous-time, asynchronous modeling of financial markets. Technical report, Erasmus School of Economics - Econometric Institute, 2006.

[2] K. Boer, M. Polman, A. de Bruin, and U. Kaymak. An agent-based framework for artificial stock markets. *Proceedings of 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'04)*, pages 83–90, 21-22 October 2004, Groningen, The Netherlands.

[3] N. Chan and C. Shelton. An electronic market-maker. Technical Report 5, Massachussets Institute of Technology - Artificial Intelligence Division, 2001.

[4] M. Cheung and M. Koornneef. Een artificiële market maker in een continue en discrete omgeving. Workshop report, Informatica & Economie, Erasmus Universiteit Rotterdam, 2006.

[5] F. Cornelli and D. Goldreich. Bookbuilding: How informative is the order book? *The Journal of Finance*, 58(4):1415–1443, 2003.

[6] S. Das. A learning market-maker in the Glosten-Milgrom model. *Quantitative Finance*, 5(2):169–180, 2005.

[7] T. Foucault, O. Kadan, and E. Kandel. Limit order book as a market for liquidity. *Review of Financial Studies*, 18(4):1171–1217, 2005.

[8] M. Garman. Market microstructure. *Journal of Financial Economics*, 3:257–275, 1976.

[9] L. Glosten and P. Milgrom. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, 14:71–100, 1985.

[10] L. Harris. Liquidity, trading rules and electronic trading systems. Papers 91-8, Southern California - School of Business Administration, 1990.

[11] M. Meijer. Analysis of an ASM simulation program. Bachelor's thesis, Informatica & Economie, Erasmus Universiteit Rotterdam, 2005.

[12] C. Parlour. Price dynamics in limit order markets. *Review of Financial Studies*, 11(4):789–816, 1998.

[13] J. Spiering. Informational asymmetry and market prices in a continuous asynchronous agent-based artificial stock market. Master's thesis, Informatica & Economie, Erasmus Universiteit Rotterdam, 2005.