Extracting Information from the Order Book in an Artificial Stock Market

P.M. Koorneef

Faculty of Economics, Erasmus University Rotterdam

P.O. Box 1738, 3000 DR, Rotterdam, the Netherlands

Email: maarten.knf@student.eur.nl

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.

1 Introduction

With computer processing power constantly increasing, the possibilities for undertaking research by means of simulation have been greatly improved. This is especially the case for *agent-based simulation*, in which all entities that make up the system, are modelled separately and can have individual behaviours and rules for decision-making. Because this type of modelling enables one to simulate systems with many different kinds of individuals, it is perfectly suited for research on financial markets, so a number of models have been developed, commonly referred to as *Agent-based Artificial Stock Markets (ABASM)*.

Very recently, an agent-based extension [2] has been proposed to the Das market maker model [4]. In the Das model, a market maker sets bid and ask prices for a stock given the arrival of market orders placed by an investors population that is heterogeneous in terms of informedness. The market maker knows the composition of the population but does not know whether an order comes from an informed or an uninformed investor. He then uses Bayesian learning techniques to learn the fundamental value, which he tries to reflect in his prices. This simulation process is serial, in the sense that when one part of the simulation is busy (for example the market maker calculating its new bid and ask prices), all other parts are idle until they are selected to carry out their actions.

The extension to this model consists of explicit modelling of the market maker and the investors into separate computer processes that simultaneously run next to each other, which is a step towards more realistic behaviour. This has a number of consequences for both the design and the results of the model, of which the major point of interest is that orders can be placed by multiple investors at a time, while in the meanwhile the market maker is busy updating its prices. Hence, a queue of orders appears, waiting to be executed. As the number of investors grows, so does this queue. A new feature here is that a time delay exists between the placement of orders and their execution, so the market maker always lags behind with his price updates and in the meantime, new information may have been become available that leads to different decisions of investors.

This paper shows that this order queue contains useful information for price-setting decisions. By using this information, the market maker not only learns the fundamental value better and faster, he will also be albe to cope with a much bigger investor population. We will show how this information can be utilized in an intelligent way. In order to accomplish this, a number of mathematical changes are implemented on top of the extended Das model. We then show by means of experiments with both the model before and after the modifications, that these changes lead to better performance.

The rest of this paper is organized as follows: Section 2 briefly explains the original Das model, along with a number of related models. In Section 3, the differences between the original and the extended model are covered, as well as how the order flow is used as an input for price updates. In Section 4, a comparison is made between the extended model before and after the changes. Finally, the conclusion can be found in Section 5.

2 Market maker models

There are relatively few stock market models in which a market maker plays a role. Two main categories of them exist:

1. Inventory models such as [1] and [5], which focus on the order-matching problem the market maker has to deal with, and minimizing the price risk that he faces because of his inventory.

2. Information models, which use the statement made by Glosten and Milgrom [6] 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.*" [6] One interesting example of this was developed by Chan and Shelton [3], who implemented reinforcement learning, an artificial learning technique, to a market maker model.

The models we work with do not take order-matching into account and can be put into the second category. Additionally, there are a number of models that simulate limit order markets. However, we did not find any examples of these that belong in the category of information models.

2.1 Stock market simulation using the Das model

The Das market maker model is largely based on Glosten and Milgrom, but a number of things, especially regarding the calculation of the bid and ask prices, are specified at such a detailed level that a computer simulation can be made of it.

In this model, the market consists of a trading crowd and a single market maker with a zero-profit motive. Trade takes place by market orders of size one. The stock has a fundamental value V that 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, but can also be zero in case of perfectly informed investors. He places a buy order if the observed signal is higher than the current ask quote P_a and a sell order if it is lower than the current bid quote P_b, or no order if $P_b < V + \widetilde{\eta} < P_a$. All other investors are liquidity traders with a fixed probability of trading, equal to $2 * \eta$. Given that one 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 standard deviation 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 population of investors 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) that 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 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. The bid and ask prices are like those of a competitive market maker in equilibrium, which are according to Glosten and Milgrom:

$$\begin{aligned} P_a = &\quad E[V|Buy] \\ P_b = &\quad E[V|Sell] \end{aligned} \tag{1}$$

Taking the bid price for example, following the definition of expected value and assuming V is discrete, this can be written as:

$$E[V|Sell] = \sum_{V_i=V_{min}}^{V_i=V_{max}} V_i P(V = V_i|Sell) \tag{2}$$

It can be shown from Das [4] that for a market with noisy informed and uninformed traders, the bid price P_b is equal to:

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

$$= \frac{\sum_{V_i=V_{min}}^{V_i=V_{max}} V_i \left[\alpha P(\widetilde{\eta}(0, \sigma_W^2) < P_b - V_i) + (1-\alpha)\eta\right] P(V = V_i)}{\sum_{V_i=V_{min}}^{V_i=V_{max}} \left[\alpha P(\widetilde{\eta}(0, \sigma_W^2) < P_b - V_i) + (1-\alpha)\eta\right] P(V = V_i)} \tag{3}$$

The ask price P_a in turn is equal to:

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

$$= \frac{\sum_{V_i=V_{min}}^{V_i=V_{max}} V_i \left[\alpha P(\widetilde{\eta}(0, \sigma_W^2) < V_i - P_a) + (1-\alpha)\eta\right] P(V = V_i)}{\sum_{V_i=V_{min}}^{V_i=V_{max}} \left[\alpha P(\widetilde{\eta}(0, \sigma_W^2) < V_i - P_a) + (1-\alpha)\eta\right] P(V = V_i)} \tag{4}$$

where V_{min} and V_{max} are the minimum, respectively the maximum possible values of V. Usually, V_{min} is set to $E[V] - 4\sigma_J$ and V_{max} is set to $E[V] + 4\sigma_J$. 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 V_{min} to V_{max}, with steps of 1 cent, or an arbitrary step size. Initially, and immediately after a jump, V_{min} and V_{max} are recalculated using the current estimate of V, and 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:

$$P(V = V_i|Sell) = \frac{P(Sell|V = V_i)P(V = V_i)}{P(Sell)} \tag{5}$$

The probabilities $P(Sell|V = V_i)$ and $P(Sell)$ are calculated using the equations 8 to 11 from [4], and $P(V = V_i)$ is stored in the vector. After each processed order, the expected value of the vector more and more converges to the real fundamental value, while the variance decreases.

3 Simulating in an agent-based environment

The model just described, has been implemented in an agent-based environment by Boer, Kaymak and Spiering [2]. In this implementation, all investors and the market maker are separate agents that act parallel to each other and place orders independently without having to wait for other agent to finish their jobs. This is of course, analogous to a 'real' stock market. However, a number of changes had to be made to the design of the model in order to make this work. The most important differences are:

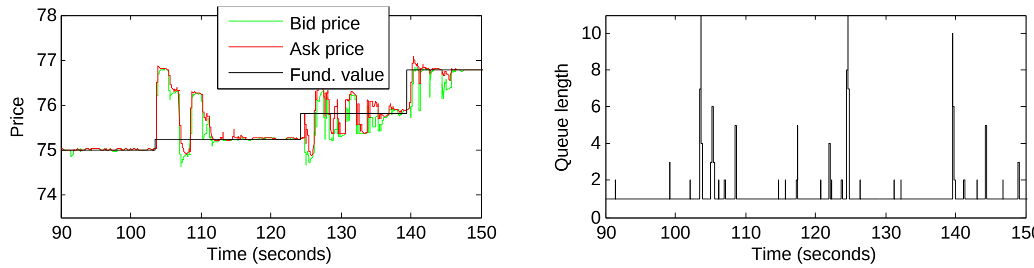
1. The population size had to be explicitly defined in addition to the fraction of informed traders, since all investors are different agents.

2. Instead of waiting indefinitely until they are selected to trade, all investors have a *minimum time between orders* parameter, which means that they are triggered to make a trade decision every n milliseconds. Because all investors live in separate computer processes, it is a task of the operating system to distribute the available processor power among the agents in the simulation. This means that when these n milliseconds have elapsed, it may take a small additional time before he is scheduled to run. Especially when there are many investors, this effect plays a role of interest. Only when an informed investor receives a signal of a changed fundamental value, he will immediately make a trade decision when it is his turn, regardless the time that passed since the last order.

3. The Das model has a special rule for its bid and ask prices in case an investor decides not to place an order. In the agent-based model, trade takes place continuously and if an investor decides not to trade, he does not come to the market maker to tell him so. The solution is that the market maker keeps track of the time that passed since the last order arrived. If this passes some predefined limit, the no-order update is applied.

4. A jump in the fundamental value is no longer determined by a fixed probability between each two orders, but by a uniformly distributed time variable that indicates the number of seconds that has passed since the last jump for the next jump to occur.

5. A queue of orders can (and regularly does) appear when different investors place an order within a short time, before the market maker gets the chance to look at just one of them. The result is that a queue of pending orders (or an order book) appears. This would be impossible in the original model, where no trade occurs



(a) Bid and ask prices

(b) The order queue

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

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 delay that exists between the placement and the execution of an order, is that in the meantime, the bid and ask prices may already have changed a number of times because other orders are executed first. 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: according to 10 investors, the fundamental value at time t is lower than the bid price at time t. Later on, suppose at time $t + s_1$, the market maker picks the first order from the queue and 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, and so on. Each time, he interprets an order as a signal that his *current* bid price is too high, but the true interpretation is that the bid price was too high at time t. The fact that 10 investors place an order only gives stronger evidence that they are right, but it does not say anything about how big the difference between the P_b and V is. Because of this misinterpretation, the market maker eventually sets a price that is far below the fundamental value. The bigger the number of investors, the more this effect will take place.

Because of the market order system, the investors that are served later in the queue lose a lot of money as well because of the sudden price changes.

Figure 1 shows that after each jump, a number of oscillations occur before the market prices stabilize at the level of the fundamental value.

3.1 A modified market-maker model

A solution to this problem is to let the market maker investigate all orders in the queue simultaneously. By doing this, he interprets the queue as a whole as a signal about his current prices relatively to the fundamental value.

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, shown in equation 5, to:

$$P(V = V_i|Orderbook) = \frac{P(Orderbook|V = V_i)P(V = V_i)}{P(Orderbook)} \tag{6}$$

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.

Here, *Orderbook* denotes the (unordered) list of arrived orders, for example: *Sell, Sell, Buy*. If we fill this in, the equation is expanded to:

$$P(V = V_i|Orderbook) = \frac{\binom{s+b}{b} P(Sell|V = V_i)^s P(Buy|V = V_i)^b P(V = V_i)}{\binom{s+b}{b} \; P(Sell)^s P(Buy)^b} \tag{7}$$

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 a result from [4] to get the following update rule:

$$P(V = V_i|Orderbook) = \tag{8}$$

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

The only assumption we make here that the events that generate orders, are independent. This can be questioned, because at least the informed investors will place the same orders given that they receive the same signal about the fundamental value. However, we do not try to overcome this limitation and just assume independence of these events.

4 Experiments

By running a number of simulations, we test whether the proposed modification leads 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).

We will set up a number of configurations and analyse the outcomes from a version with and without the modification to the price vector update rule.

The experiments take place in a market environment with a number of investors and one market maker. The minimum time between two orders for the investors as well as the maximum time between two price updates for the market maker 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 maximum time between two price updates is set to 120 milliseconds, which turns out well. The time between orders is 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.

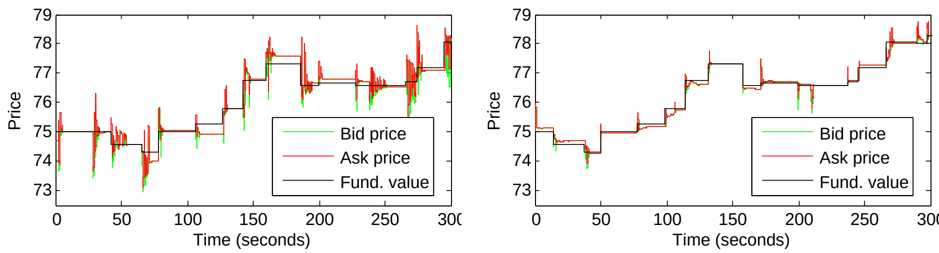
We will simulate market populations with 10 informed and 10 uninformed investors, with 10 informed and 20 uninformed investors and with 50 informed and 50 uninformed investors, everything with both noisy and perfectly informed traders. The reason for just these populations is that the number of investors must be big enough to get interesting results, because if there are only two investors, the maximum size of the order book would also be two. 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 noise to the order flow. All simulations with more investor are included simply to show up to which point the market maker gives sensible results.

4.1 Simulation results

By showing a number of graphs, the computed bid and ask prices as well as the spread are displayed. Later on, we will show some numerical results about the simulations, for example the profitability.

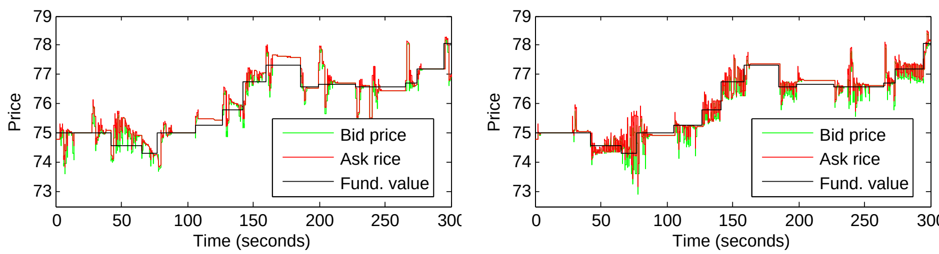
The plots 2 until 9 show that in all eight cases, the modification improves performance with respect to accuracy (finding the fundamental value as good and as quick as possible) and stability. This is true for both populations with noisy and with perfectly informed investors, although the simulation is somewhat more accurate with perfectly informed investors. Especially when the population is not too large, the modified version quickly adjusts its prices to the correct level after each shock and does not get out of balance. The only case that does not show any improvement is the one with 10 noisy informed and 20 uninformed investors. The most likely explanation of this phenomenon is that the order flow contains too much noise for the market maker in order to reach stability.

With populations of 100 investors or greater, the prices set by the original version do not make sense anymore. This is because after a jump, there are so much orders of the same type in the queue that the prices keep moving in one direction, until they cannot



(a) Original (b) Modified

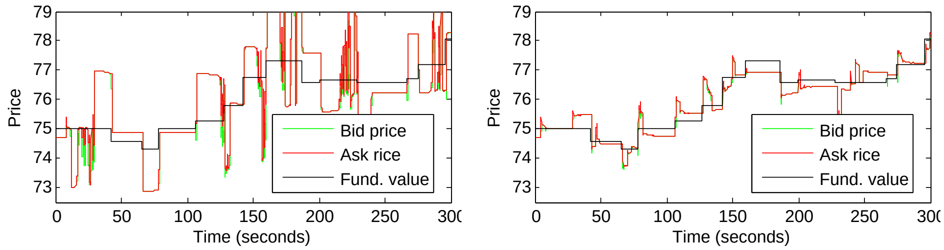
Figure 2: Bid and ask prices during a simulation with 10 noisy informed and 10 uninformed investors



(a) Original (b) Modified

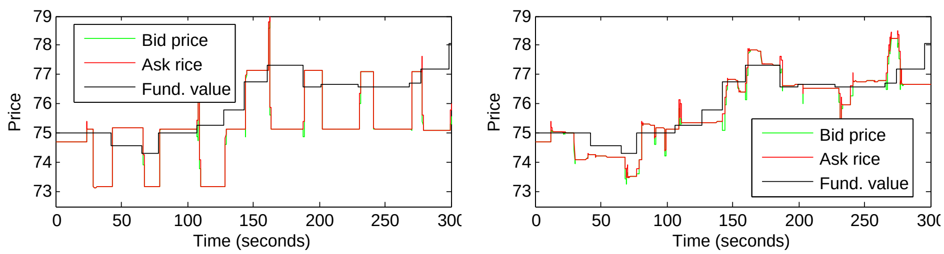
Figure 3: Bid and ask prices during a simulation with 10 noisy informed and 20 uninformed investors

move any further because the end of the vector of prior probabilities is reached. This vector runs from $E[V] - 4\sigma_W$ to $E[V] + 4\sigma_W$, where $E[V]$ is the expected value of V just before the jump and σ_W is equal to the standard deviation of the jump probability for the fundamental value. All possible values of V outside the vector are assumed to have a probability of 0, so the market maker can never set prices outside the range of the vector, so the prices move to and stabilize on one bound until the next jump. The modified version keeps giving stable results, although it sometimes stabilizes at an incorrect value. Also, it should be noted that the population size cannot be increased unlimitedly, because computational problems will occur eventually. For example: vector values (which are often very small) to the power of 400 have to be calculated, which are so close to zero that the computer will almost always give zero as output. This might also be an explanation why the market maker captures the fundamental value less accurate if the population is very big.



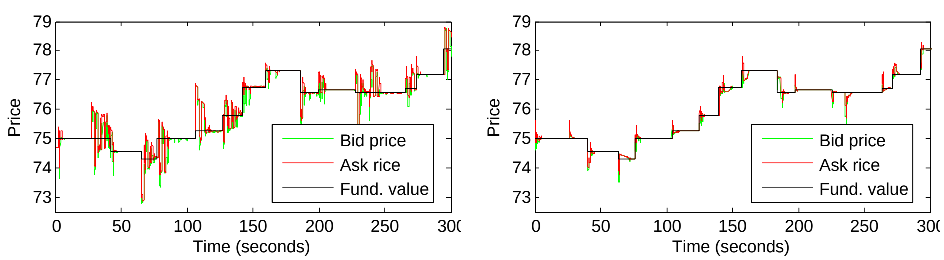
(a) Original (b) Modified

Figure 4: Bid and ask prices during a simulation with 50 noisy informed and 50 uninformed investors



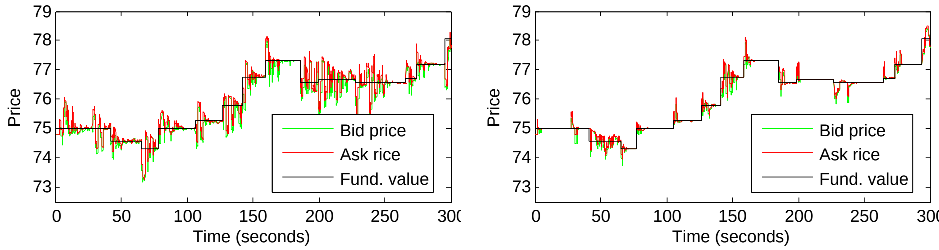
(a) Original (b) Modified

Figure 5: Bid and ask prices during a simulation with 150 noisy informed and 150 uninformed investors



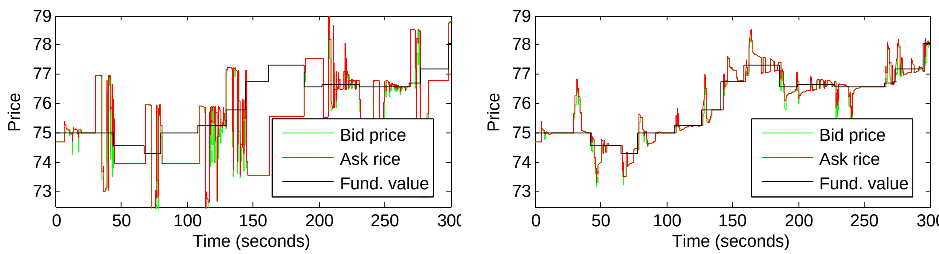
(a) Original (b) Modified

Figure 6: Bid and ask prices during a simulation with 10 perfectly informed and 10 uninformed investors



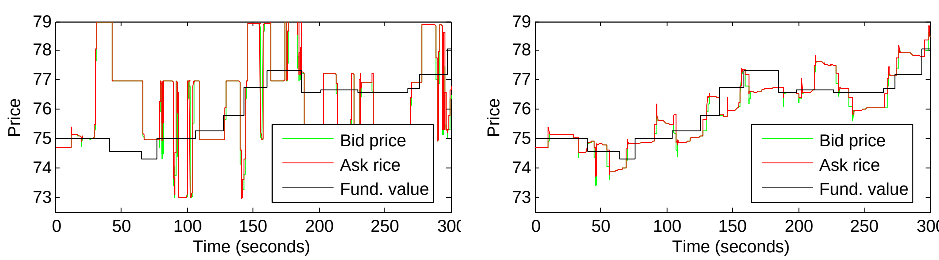
(a) Original (b) Modified

Figure 7: Bid and ask prices during a simulation with 10 perfectly informed and 20 uninformed investors



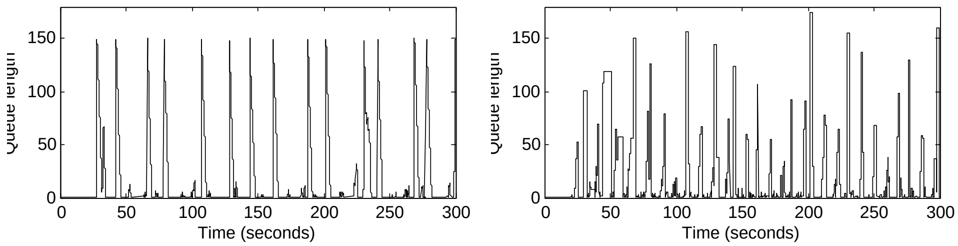
(a) Original (b) Modified

Figure 8: Bid and ask prices during a simulation with 50 perfectly informed and 50 uninformed investors



(a) Original (b) Modified

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



(a) Original (b) Modified

Figure 10: The size of the order book during the simulations with 150 noisy informed and 150 uninformed investors

population	original MTBO	modified MTBO
10 noisy and 10 uninformed investors	1000	800
10 perfectly and 10 uninformed investors	1500	800
10 noisy and 20 uninformed investors	2000	800
10 perfectly and 20 uninformed investors	2000	1200
50 noisy and 50 uninformed investors	4000	3000
50 perfectly and 50 uninformed investors	4000	2400
150 noisy and 150 uninformed investors	20000	9000
150 perfectly and 150 uninformed investors	10000	9000

Figure 11: The values of the Minimum Time Between Orders for all simulations

population	version	#buy	#sell	money	#shares	wealth
10 noisy informed	orig.	2263	2272	536.01	-9	-165.99
and 10 uninformed	mod.	2687	2915	17198.75	-228	-598.93
10 perfectly informed	orig.	1407	1496	6696.88	-89	-245.12
and 10 uninformed	mod.	2664	2847	14154.56	-183	-130.42
10 noisy informed	orig.	1657	1518	-10762.93	139	79.07
and 20 uninformed	mod.	3509	3446	-4674.90	63	239.10
10 perfectly informed	orig.	1428	1477	3767.06	-49	-54.94
and 20 uninformed	mod.	2501	2571	5319.53	-70	-140.47
50 noisy informed	orig.	2728	2852	8348.92	-124	-1323.08
and 50 uninformed	mod.	3747	4004	19233.96	-257	-812.04
50 perfectly informed	orig.	2135	3299	87257.52	-1164	-3534.48
and 50 uninformed	mod.	4196	4605	30160.22	-409	-1741.78
150 noisy informed	orig.	1723	2350	48488.26	-627	-417.74
and 150 uninformed	mod.	4015	3649	-28422.92	366	125.08
150 perfectly informed	orig.	2957	3428	36505.79	-471	-232.21
and 150 uninformed	mod.	3304	3793	36709.35	-489	-1432.65

Figure 12: Market maker statistics at the end of the simulations

4.2 Looking at the order queue

It is interesting to see how the size of the order book fluctuates over time. Figure 10 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 original market maker handles all orders one by one, making the order book size decrease gradually. The modified one handles all orders in one go, which takes some time, so the queue size remains high for a longer time, but then then drops immediately to (almost) zero.

The minimum time between orders for the investors is shown in figure 11. 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.

4.3 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 12 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 per order. 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, but overall, it complies to its zero-profit motive quite well.

4.4 Pros and contras of the modification

It is clear that taking the order book as an input for determining the bid and ask prices leads to better results. The only drawback we see in the modified version is that the spread can be 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. However, we have seen that this does not result in a bigger loss at the end of the simulation. We cannot incur from the simulation results that the modification to the system has any consequences for the profitability of the market maker.

5 Conclusion

We have shown 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 an agent-based simulation environment. 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 proper way. The modified market maker shows better performance in terms of tracking the fundamental value of the stock, with regard to accurateness and stability.

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. If he remembered this, the fraction of informed traders would not have to be given as a parameter on initialization, but could be dynamically adjusted as investors entered and exited from the market during the simulation. 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, although it is a significant aspect of all stock markets.

References

[1] Y. Amihud and H. Mendelson. Dealership market: Market-making with inventory. *Journal of Financial Economics*, 8:31–53, 1980.

[2] 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.

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

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

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

[6] 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.