An High Frequency Trading model for NetLogo

Salvatore Vilella

May 13, 2016
# Contents

Introduction 1

1 High Frequency Trading 3
   1.1 What is High-Frequency Trading? 3
   1.2 HFT: lights and shadows 4
   1.3 Does HFT hurt the small investors? 5

2 The Code 6
   2.1 General overview 6
   2.2 The *Setup* button 6
   2.3 The *Go* button 7
      2.3.1 NAs’ decisional algorithm 7
      2.3.2 HFAs’ decisional algorithm 9
      2.3.3 The *auction* procedure 10

3 Running the model: trials and results 12
   3.1 Description of the model 12
   3.2 What we expect to see and what we are looking for 12
   3.3 Trials 13
   3.4 Conclusions 16

Bibliography 18
Introduction

This paper describes an high frequency trading model for Netlogo. Before going into the details of the work, it is important to say something about Netlogo and, in general, about Agent-Based Modeling (ABM). We can define ABM through Epstein’s words [5]: “... ABMs are thus models where individuals or agents are described as unique and autonomous entities that usually interact with each other and their environment locally”.

In an ABM model we will have intelligent agents: this intelligence lies in their simple behavioural rules, which will guide their mutual interactions and their interaction with the environment; the intelligence of the agents is what sets ABM apart from the other kind of computational models. Following the rules we tell them, agents will make choices based on external conditions, will eventually adapt to the environment and learn from their previous experience.

Netlogo (by Uri Wilensky [11]) is a tool frequently used in ABM, and it is as powerful as simple. Its programming language derives from Logo, and was initially meant to be a teaching tool for children. The agents come in the form of turtles, patches or links. It has a graphic interface, highly customizable and very intuitive, where the user can set variables by means of buttons, switches and sliders, and can visualise outputs such as plots, written text or even the actual movement of the turtles within the world. As we will see in the description of the code, the language is simple and very straight-forward.

The intention of the writer in the present paper is to create an agent-based model capable of simulating a simplified stock market, where traders are both humans and machines. We will study a particular form of algorithmic trading, i.e. high frequency trading, and we will try to see how it differs from and how it affects “normal” traders. The report is structured as follows:

- **Chapter 1: High Frequency Trading**, an introduction to the phenomenon of high frequency trading, focusing on its pros and cons;
- **Chapter 2: The Code**, a detailed description of the algorithm;
- **Chapter 3: Running the model - Trials and results**, where some runs are performed, their goals are specified and finally their results analyzed.
Chapter 1

High Frequency Trading

1.1 What is High-Frequency Trading?

As the name suggests, High Frequency Trading (hereafter HFT) generally refers to an automated, very fast trading activity. Most of the literature considers HFT as a subset of algorithmic trading (AT) but, as Vuorenmaa points out [10], an important difference in purposes should be highlighted: while AT aims at determining how to place a trade in order to minimize trading costs, HFT is about determining whether a trade should be placed at all. According to Gomber et al., a study commissioned by the Deutsche Börse Group [6], there are some characteristics which are peculiar of HFT and provide a good description of how HFT works, marking the separation from AT. Among them, we could cite the very high speed of order and order cancellation, which is necessary to make small profits per trade. The investing horizon of HFT ranges between milliseconds and hours, and all of their positions are closed at the end of the day [1]. On a huge number of trades, it will yield a significant return.

This high speed feature imposes a low latency requirement: HFT traders need to rely on high speed access to markets and high speed information exchange, i.e. sophisticated connections, great processing power, individual data feeds, use of proximity services. This basically means that only big trading and financial companies can afford HFT. There is in fact a long debate over the “ethical” aspects of this kind of business: what is the impact of HFT on markets? Does it hurt small investors? Does it need to be more regulated by central authorities?

This short essay has not the ambition of going too deep into the financial details of the matter, since it is not the scope of the writer. Nevertheless, in order to have a first approach to HFT in all of its aspects, I am going to briefly give account of the general pros and cons mostly found in literature, citing some notorious controversial episodes like that of the infamous Flash Crash as well as documented, heterogeneous opinions of a few experts.
1.2 HFT: lights and shadows

We will now have a look at the advantages and disadvantages of HFT. It is however important to underline that even though HFT is attracting more and more attention among experts and academics, especially since the flash crash of May 6, 2012, researches and literature are still quite scarce. Publications on the matter only increased in the very recent years. Moreover, empirical studies encounters difficulties in the study of actual data of stock exchanges, since it is often impossible to precisely identify orders coming from an HFT algorithm and discern them from “normal” orders.

Most scientific papers (based on data of lit markets) do not find any negative impact of HFT on market quality. On the contrary, multiple studies (both theoretical or empirical) state that HFT, under certain conditions, has a rather good impact on it. From the model by Cvitanic and Kirilenko [4] comes to light that in the presence of an high frequency trader the transaction prices are more concentrated around the mean value, reducing volatility and improving forecastability of transaction prices.

On the other hand, evidence of the opposite is found by Frank Zhang [12], who states that HFT increases long-term stock market volatility, and that this positive correlation is stronger among the top 3000 stocks in market capitalization and in periods of market uncertainty.

The same can be said about price discovery and transparency. According to one of the most common definitions of price efficiency, there cannot be any predictability in prices to be taken advantage of because prices already reflect all relevant information; and since in theory more trading volume leads to more precise pricing, HFT should increase price efficiency and discovery. There is empirical evidence of this in Hendershott and Riordan [8], who state that HF agents achieve this goal by trading in the direction of permanent price changes and in the opposite direction of transitory pricing errors. But it is Zhang again [12] who opposes with the following argument: usually short-terms speculators, such as HF traders, may put too much emphasis on short-term information and not enough on stock fundamental information, leading to a degradation of the informational quality of the prices. So it basically is a problem of overreaction: if the mis-pricing catches momentum, this kind of trading activity becomes counterproductive [10].

The flash crash The aggressive trading style of HF traders is probably one of the main causes of the flash crash of May 6, 2010. On this day the US stock market experienced a sudden and very significant increase in volatility: a drop of more than 5% in prices was followed by a rapid rebound of almost the same magnitude, and it all happened in less than 20 minutes. It was an intraday event, but it had repercussions on confidence in markets for the following weeks. Studies of the event have been conducted, and there has been an official report commissioned by Joint Advisory Committee on Emerging Regulatory Issues in the US [3] which, along with Kirilenko et al. [7], concludes that HFT did not trigger the crash, but their overreaction to a sudden selling pressure by an institutional seller exacerbated volatility. Such an extreme event grabbed the attention of the whole financial world - both academics and executers.
1.3 Does HFT hurt the small investors?

Competition on the same market between retail investors and algorithms such as those just described, brings to surface some obvious ethical issues. Most of the papers, both academic or external reports, usually end with a dense paragraph about the current regulations in the US and European markets, as well as with a long list of proposals on the matter: this is a clear sign that the situation is yet to be well-defined, and there is a strong need in this sense, by multiple parties. In an article of the Wall Street Journal [2] four eminent academics are asked questions on the dangers a small investor could run: it is common opinion that, in absence of a clear and common regulation of the phenomenon, retail investors should be strongly recommended to “stay away from short-term trading strategies. They are going to go up against computers and lose. [...] Investors should worry about the long-term price appreciation of the stocks, not small volatilities”. But in the end it all comes to how HFT should be regulated, and how much invasive regulations should be.

Is it dangerous for small investors? As we just saw, theoretical and empirical evidence is somehow confusing on this. It seems like it is not possible yet to answer this question, because results strongly change depending on the initial assumptions of each study. The model of Zhang, for example, which comes to such critical conclusions, has been criticized of not actually proving the direction of causality in the found positive correlation between HFT and volatility [10]; in general, modeling frameworks often make quite radical assumptions that may not hold in reality.

It has already been said that this field captured the attention of experts and regulators since a relatively short time: it is still an ongoing process, and scientific opinions are still taking shape.
Chapter 2

The Code

2.1 General overview

What follows is a model of a very simplified stock market: buyers and sellers only trade one kind of stock on a one-stock-per-tick basis. The agents are divided into two main categories, or breeds: the High Frequency Agents (HFAs) and the Normal Agents (NAs). Both will operate on the same market but with different strategies and mechanisms of action, which will be explained in detail in the pertinent sections of this chapter.

2.2 The Setup button

The setup button creates the agents in a total number of nAgents, of which only a few (nHFAs) will be high frequency agents, adopting the computer workstation shape. As the market is common, both NAs and HFAs will share the same variables used in the auction procedure, i.e. stocks (the number of stocks owned by each agent) and cash (the amount of cash). The list exePriceList is owned by both HFAs and NAs, but only the former will use its mean value as a control parameter in their stop-loss strategy. As we can see in Figure 2.1(a), besides exePriceList HFAs own a tick counter and an out boolean reporter as control variables, whose use will be detailed later.

Bull and bear, boolean reporter as well, will set the HFAs’ strategies.

In the final part of the setup section (Figure 2.1(b)), agents take their place in the world. The world box in the graphic interface is not actually a key part of the model, since there are no moving turtles nor patches with peculiar properties; by the way it could be helpful to visualize the interactions between agents and their states through changes of colours, so it was ultimately decided to keep it in the UI.
2.3 The Go button

In Figure 2.2 we see that the go button is divided into three main procedures: the first two form the decisional part of the algorithm, which is the core of the whole code, while in auction the agents trade their stocks and write their logs. Both NAdecisionmaking and auction are based on the Continuous Double Auction model proposed by Professor Pietro Terna [9], where the former in particular has been heavily edited in order to fit in this scenario.

2.3.1 NAs’ decisional algorithm

The difference between NAs and HFAs in the frequency of their operations is implemented through a probability of action per tick. Figure 2.3 shows that a float, random number $p$ lesser than 1 is generated; the NAs will make their
Figure 2.3: The decisional process of NAs

decisions only if p is lesser than the threshold NAprobability, which is set through a slider in the user interface. We will find the same mechanism for HFAs in their corresponding part of the code. In order for the model to be effective, the difference between these two thresholds must be significant, e.g. a 20% probability for NAs and a 95% probability for HFAs, which are the standard, pre-set values in the model.

A NA will decide whether to be a buyer or a seller on random basis: the divide will be a threshold, which will move away from the middle value of 0.5 depending on the following conditions:

- **the FloorActing variable** is a boolean reporter which, if true, combined with the condition exePrice < 500, sets the threshold to 0.8, thus preventing the exePrice to drop any further and eventually to become negative;

- **the external news procedure** (Figure 2.4) has been included to see the effects on the markets of an external influence (and to test the veracity of the model), such as a rumour or official news. It is enabled by the news switch in the interface, and it is pretty basic: the variable LatestNews (shown in a monitor on the interface) is randomly set either on a good or a bad news, with equal probability. Good news will give a boost in confidence of NAs, increasing the threshold and consequently the probability for the NAs to be buyers rather than sellers. The breaking news will come out after the market starts running, precisely at the instant when the HFAs start operating.

Having set the threshold, NAs will now fix their ask/bid price and will be ready to start trading their stocks.
2.3.2 HFAs’ decisional algorithm

The decisional algorithm of the HFAs (Figure 2.5) is divided into three phases. We want the agents to begin their operations when the market is already running. We will then choose a precise instant, namely $HFA_{entranceTime}$, for the HFAs to wake up and establish their strategies. The choice of the strategies is set to take place once every 500 ticks, which is the duration of a day. Unlike NAs, HFAs will not act randomly: they will follow either a bull or a bear strategy with equal probability. The key part of the code is in this last section, where the frequency of action is taken into account (exactly in the same way as for the NAs), and the bull and bear procedures are invoked.

```plaintext
to external_news
  if news [  
    let b random-float 1  
      if b > 0.5 [set LatestNews good_news]  
      if b < 0.5 [set LatestNews bad_news]  
    ]
end
```

Figure 2.4: The external news procedure

```plaintext
to HFAdecisionmaking
  ask HFAs [  
    if ticks - HFA_{entranceTime} < 0 [set pass true set color gray]
    if remainder ticks 1000 - HFA_{entranceTime} = 0 [  
      set out false
      let f random-float 1
      ifelse f > 0.5 [set bull true set bear false]  
        [set bull false set bear true]
      if bull [set buy true set sell false set pass false]
      if bear [set buy false set sell true set pass false]
    ]
    if remainder ticks 1000 - HFA_{entranceTime} > 0 [  
      let a random-float 1
      ifelse a >= (1 - HFAprobability) [  
        set pass false  
        set exePrice=exePrice + (random-normal 0 100)
      ]
      if bull [bull-strategy]
        if bear [bear-strategy]
        [set pass true]
    ]
    if pass [set color grey]
  ]
end
```

Figure 2.5: The decisional process of HFAs

This two procedures (Figure 2.6) actually consist only of the control strategies, since the order to buy (or to sell) has already been given when the decision of being a bull (or a bear) was first taken. Let us now analyse the bull strategy in Figure 2.6(a). Since the bear one is
perfectly symmetrical, I will omit to further explain it.

HFAs start to check whether their strategy is or is not successful after a certain amount of ticks (\texttt{startChecking}). As we will see in the auction procedure, each agent will fill its own \texttt{exePriceList} every time it buys (or sells) a stock. The mean value of this list will be the fundamental control parameter. Since each agent trades only one stock per transaction there is no need for a weighted average, which otherwise should have been used instead of the arithmetical mean. There are two main scenarios I decided to focus on:

- **in case of loss** each agent will have a stop-loss strategy. The comparison will be

\[ m(n - 1) > p(n) + p(n) \cdot \text{stopLoss} \]  

where \( m(n-1) \) is the arithmetical mean of the prices at which the agent operated, \( p(n) \) is the current \texttt{exePrice} and \( \text{stopLoss} \in [0, 1] \). The logic behind this is plain and simple: if you are a bull, your initial bet is that prices are going to increase, therefore buying will give you a profit on the long term. If the average paid price is greater than the current price of the market, it means that things are not going as expected. \texttt{StopLoss} in some sense expresses the tolerance on the loss, and it is typically kept very low: as soon as the HFA observes a loss it quits the markets, since the chosen strategy is probably the wrong one;

- **in case of gain** the agent of course will not hold its positions forever. The exit strategy in this case is implemented by means of a counter, which is increased every time \( m(n - 1) \) is lesser than the current \texttt{exePrice}, meaning that the previsions were correct. Again the exit time is set through a slider; when \( \text{gainTimeCounter} > \text{exitTime} \), the agent quits until the next day.

In both cases when the agent quits it sets the \texttt{out} variable true, preventing the agent to be checked again at each tick even if it is no longer operative, thus making the execution faster and avoiding annoying issues with the repetition of the \texttt{go} cycle along the days. The \texttt{out} variable is then set false at the beginning of the following day, when HFAs hit the markets and make their decisions again.

### 2.3.3 The auction procedure

The trading mechanism is mostly borrowed from the Continuous Double Auction model by Professor Pietro Terna [9]. There are just a couple of edits I made:

- the \texttt{exePriceList}, which is not a global variable but a turtle-specific one, is filled by each agent every time it buys or sells a stock (Figure 2.7(a));

- the day is made of 500 ticks, so at the end of the day the log of the bids and that of the asks will be emptied, and so will be the \texttt{exePriceList}. If needed, the \texttt{exePrice} for the following day will be reset to 1000 (Figure 2.7(b)).
to bull-strategy
set color green

; controls
if remainder ticks 1000 = (IFAnrentacTime + startChecking) >= 0 and not out []
    stop loss
    if mean exPriceList > exPrice + (exPrice - stopLoss) [set buy false set pass true set out true ]
    exit strategy in case of gain
    if mean exPriceList < exPrice [set gainTimeCounter gainTimeCounter + 1] [set gainTimeCounter 0]
    if gainTimeCounter > exitTime [set buy false set pass true set out true ]
end

(a)

to bear-strategy
set color red

if remainder ticks 1000 = (IFAnrentacTime + startChecking) >= 0 and not out []
    stop loss
    if mean exPriceList < exPrice - (exPrice - stopLoss) [set sell false set pass true set out true ]
    exit strategy in case of gain
    if mean exPriceList > exPrice [set gainTimeCounter gainTimeCounter + 1] [set gainTimeCounter 0]
    if gainTimeCounter > exitTime [set sell false set pass true set out true ]
end

(b)

Figure 2.6: The bull and bear strategies

ask turtle agB [set stocks stocks + 1
    set cash cash - exPrice
    set exPriceList [] put exPrice exPriceList
    ]
ask turtle agS [set stocks stocks - 1
    set cash cash + exPrice
    set exPriceList [] put exPrice exPriceList
    ]

(a)

if remainder ticks 1000 = 0 [ set logB [] set logS []
    ask turtles [set exPriceList []]
    ; set exPrice 1000
]

(b)

Figure 2.7: Simple edits to the CDA trading model
Chapter 3

Running the model: trials and results

3.1 Description of the model

Let us briefly summarise the main features of this model. In our stock market a certain number of agents continuously trade one particular title, one stock per tick. A single day lasts 500 ticks: at the end of each day all logs are cleaned, all strategies are stopped. Only the number of agents and the exePrice variable, which is the current value of the stock, will be kept alive for the following day, while everything else will be reset.

Among all our agents, only a few of them will be high frequency traders. At each tick they will operate with a probability which is much higher than the one normal agents will have: this translates into the difference in frequency that we are seeking. HFAs will not operate randomly: they will hit the markets with a precise strategy, which will be either bull or bear. They will adopt a stop loss strategy, as well as they will close their position once they are happy with their gain. This is where the intelligence of the agent, the crucial feature in Agent-Based simulation, comes into play.

Normal agents, on the other hand, will operate more seldomly, on a totally random basis. They could be however influenced by a breaking news, which comes out at a certain point during the day and can bias confidence in the market.

3.2 What we expect to see and what we are looking for

The logic onto which the model is based is the following. HFAs operates utterly faster than common agents, thus making the most of the volume: in literature, proportion is said to be at least 70-30. It follows that their action will probably be crucial in deciding the trend of the price. Therefore what we intend to observe is:

- if the strategies defined for HFAs are actually working. We will compare
them to the exePrice trend to check if the HFAs are actually “understanding” the trend of the price;

• how the market is influenced by a large number of agents acting in the same way (or, better, by a large number of identical choices coming from a relatively small amount of traders);

• the last point naturally leads to the concept of self-fulfilling forecast: if such a huge number of operations happen in the belief that - supposing it is a bull strategy - prices will rise... prices will probably rise due to the huge demand, thus making the prophecy correct. It is reasonable, and it actually happens in reality, and we want to check if the model depicts well this phenomenon;

• last but not least, we will focus our attention on how the various variables (strategies, time settings, news, number of different agents) impact the markets, trying to find some points of balance between them.

3.3 Trials

We will first run the code step by step. We begin by seeing how the market behaves in absence of HFAs, and without the breaking news feature polarizing the normal agents. In Figure 3.1 (a) we can see the exePrice plotted as a function of ticks, as well as all the sliders placed in the user interface, which hereafter will be omitted unless there is a change in the settings. The trend here is quite regular: without HFAs or any option turned on, this model is basically the one it derives from [9]. Prices are almost regular, without any sudden drop or peak, regularly distributed around a mean value of about 750. The vertical lines mark the separation between days (500 ticks), so these plots cover a period of time of 5 days.

If we now turn on the news (Figure 3.1(b)), we notice that this procedure seems to work as we want it to: here we have a sequence of good - good - bad - bad - good news in the five days, and we can see that the traders were actually influenced by these news, with higher prices in the first two days, followed by a couple of bad days with prices dropping and then slowly recovering on the last day, when news were good again.

This very light effect on prices is due to the fact that thresholds are altered by just ±0.02. This is exactly what I wanted: as we will see, this slight modification is enough to cause sudden and relevant changes in prices, but only when a great number of NAs is active. Since their probability of acting is quite low, the model somehow “survives” this feature without becoming too unstable.

The model is working, now we can go on and wake up the HFAs. We set the following options: we have 100 agents, of which only 20 will be HFAs, which will begin their operations after 30 ticks. At each tick HFAs will operate with a probability of 95%, and they will start checking their strategies 50 ticks after their entrance. The stopLoss variable is set at 0.05: as described in section 2.3.2 it means that, by comparing the mean value of all the exePrice they listed, they will not tolerate a loss beyond ±0.05% of the current exePrice. In terms of time,
it means that almost as soon as they observe such a small loss, they will retire: if
the algorithm they have chosen is wrong, they will probably retire immediately
after they start checking it.

The experimental evidence (Figure 3.2) is consistent with these premises. We
observe one single day and we can see a downward trend of the prices: HFAs start
their checks and the bull algorithm is almost immediately shut down because of
the slight fall of the prices, proving the checking procedure is sensitive enough.
Meanwhile bear HFAs continue to operate until they are happy with their gains,
according to the temporal limit set by the variable exitTime. The bad news came
out at the same tick as HFAs did, while the big drop in prices happens quite
a long time after. It probably means that under these circumstances, NAs do
not have much weight on the overall trend of the prices. In the last part of the
day only bear HFAs are operating, selling all of their stocks, thus the big drop
in prices. When they stop, and only NAs continue their trades, prices stabilise
again.

The following two runs come in support of this argument. First, we increase
the exitTime to check how it affects our model. We can see from Figure 3.3(a)
that the algorithm still works in the same way, and - again - that in spite of
a bad news we have a positive trend. Since the beginning, there are more bull
than bear HFAs, and this causes the prices to rise and the bear HFAs to retire
as soon as the check their losses. A lot of agents are now buying, a very few of
them are selling, and price fastly increases until bull agents stop buying, then
it stabilises around its maximum value of about 10200. The exact opposite rea-
soning is valid for the case depicted in Figure 3.3 (b), where we have good news
but a negative trend of the prices.

Figure 3.1: First run: (a) without news and (b) with the news feature enabled
Let us do a different kind of experiment: we will now reduce the number of HFAs and see how it goes over a longer period - let us say a week. We observe in Figure 3.4 that this time we encounter a very different situation. We have 15 HFAs, 5 less than the previous runs, and we see that their dominance is becoming less evident: except for the first day, there is no major change in the price, which remains stable around its maximum value of 7860 for the whole week. We are probably closer to that point of balance we were referring to in Section 3.2.

As much as concerns the algorithm, there is something very important to notice. When prices are stable it seems like the HFAs cannot decide which of the two strategies is wrong. We observe drops in the number of agents adopting one of the two strategies, but it almost never goes to zero. It actually happens only on the 6th day, when the fall in prices is more evident. This is attributable to how the algorithm is structured: this is probably below its minimum sensitivity. The only way to try reaching a different result could be to have a similar run, setting stopLoss at 0 and a very short exitTime.

We make one last attempt, pursuing the way of reducing HFAs to see how they affect the market (Figure 3.5). All the behaviours already observed are found again. We see in fact that we do not have an explosion or a crash in prices anymore, but also that news are still not decisive in determining the price, proving again that there is more equilibrium between the agents. HFAs again are finding some difficulties in the choice of the right strategy and, in general, are experiencing some issues with the sensitivity of the algorithm: e.g. on the 6th day, when prices are stable and just slightly increasing, the bear algorithm is immediately shut down but the bull agents never close their positions, since they do not judge their gains as sufficient.
3.4 Conclusions

In this study an agent-based model for high frequency trading was introduced. A simplified stock market is the playground for two different kind of agents: normal and high frequency traders. They operate following different strategies and external influences. The goal is to see if the model can reproduce actual features of the markets: if it is well constructed, it would respond in a certain way to the operations of the agents, and the variable we refer to in order to monitor this response is the exePrice. In the meantime, it is interesting to see how the presence of high frequency traders affects the markets (and the normal investors).

An handful of runs were performed to test the veracity of the model and to try to observe the features mentioned in section 3.2. Following the same scheme used in that section, it has been found that:

- strategies actually work. By plotting the number of agents with bull/bear strategies and comparing it to the trend of exePrice we see that they respond correctly, choosing the right algorithm to preserve and shutting down the unsuccessful one. Under particular circumstances, i.e. when prices are particularly stable, it is harder for them to determine the right action to perform:
Figure 3.4: Fifth run: less HFAs, longer period. In the highlighted day the algorithm is finally sensitive enough to establish a winning strategy.

Figure 3.5: Sixth run: again less HFAs. We observe the same behaviour of the model.

- the influence of HFAs on the trend of exePrice is clear. When HFAs are placing a great amount of the same orders they dominate over NAs, as it is evident from the plots; as soon as they quit, prices stop their growth (or their decrease) and go back stable again;

- for this reason we can infer that self-fulfillness of the forecasts is observed. Once HFAs sense a small trend in prices they choose an algorithm; then, with their action, they actually feed the trend their decision was based upon, automatically making it a winning choice;

- finally, the point of balance is - quite obviously - found in the reduction of the number of HFAs. This way the impact on markets is more balanced between HFAs and NAs: both HFAs (with their high frequency) and NAs (with their positions polarised by the news) are not decisive enough to establish a clear trend in prices.

The model can be improved or refined in several ways. Surely the more interesting part to work on would be the refining of the decisional procedures of HFAs, thus making them more sensitive and responsive. Also an improved and more various system of breaking news could be implemented, e.g. including a variable impact of the news on the NAs. A different point of view in the experimental work could also be adopted: instead of searching for a point of balance, an extensive study of extreme cases would be interesting as well.
Bibliography


