Alpha Ideas in Algorithmic High-Frequency Trading

Team4.m1,[1,*] Team4.m2,[1,†] and Team4.m3[1,‡]

[1]Sofia University, St. Kliment Ohridski University of Sofia, 5 James Bourchier Blvd, 1164, Sofia, Bulgaria

(Dated: January 23, 2021)

Abstract. The paper provides novel ideas in algorithmic high frequency trading based on Reinforcement Learning (RL). Despite being a well-established tool in AI and computer science, the latter methodology as a branch of Machine Learning surprisingly proves to have diverse applications in Finance and Financial Engineering. In this context, plenty of theoretical and empirical problems arise offering a large scope for interdisciplinary research. Our paper focuses on some basic building blocks to demonstrate exactly that. Namely, we show that by suitably modifying standard RL techniques to tackle particular challenging financial problems, promising results can be achieved. This is done in two directions. First, we show how to employ RL in optimal execution of limit orders on the stock exchange. There we clearly demonstrate that high returns can be achieved by the market maker. Second, we turn attention to the different but related problem of constructing optimal algorithmic rules for stock trading that can beat the market, i.e., the so-called alpha strategies. Here we also clearly demonstrate that RL provides an edge that is well measurable. Both results motivate and pave the way for further application on which we elaborate as well.

Introduction. Algorithmic Trading is a non-universally defined term covering a broad area. It can be rather viewed as a generic concept, where the common unifying feature employed is trading of financial instruments based on some formal algorithm. An algorithm is a set of operations (mathematical, technical) to be conducted in a certain sequence to achieve a certain goal. Considering this, as well as the underlying economic implications, a well-accepted, but yet uncomprehensive, list for the sub-areas covered is: beta trading, alpha generation, static hedging, dynamic hedging, asset-liability management, market making. There is plenty of literature (see [1] for an overview as a classical reference, with [3] being an acceptable source as well) on all of them and we will just concentrate our attention on where we intend to position ourselves.

We will focus on alpha-generating strategies and market making. The former are strategies that try to generate positive returns of financial assets beyond a certain benchmark (index (SP500, NASDAQ100, etc.), special exchange traded fund (ETF), etc.). This means that we try to be independent from the market and our concentrated efforts aim at beating it. The latter deals with buying and selling securities at bid-ask spreads trying to generate profits and additionally providing liquidity to the financial system. The two areas certainly have overlaps depending on the concrete situation at hand.

When the trading is done in the high frequency space, we can speak about High-Frequency Trading (HFT), which is a separate area by its own. More generally, it is an automated trading mechanism that large investment banks, hedge funds, and institutional investors use. It employs highly sophisticated algorithms for analyzing financial data in order to perform very short-term investment trades. Usually, it doesn't take into account fundamental analysis data, like financial news or the quarterly financial statements of the traded companies. It takes into account the short-term history of the price of the securities, their volume, the bid-ask spread, and of the limit order book (LOB). A limit order book is a record of outstanding limit orders, maintained by the exchange. A limit order is a type of order to buy or sell a security at a specific price or better. A buy limit order is an order to buy at a preset price or lower while a sell limit order is an order to sell a security at a pre-specified price or higher (see for more details [3] and [4]).

The motives for HFT are diverse, but market making and alpha generation probably play a prominent role. Another one is market making/trade execution, where algorithms are deployed to optimally execute certain non-standard trades. Motives in this area might include the execution (at best possible prices) of large orders or the execution of an order with as little market and price impact as possible. A more subtle motive might be to disguise an order by executing it on a number of different exchanges.

All that provided the necessary details where we position ourselves. In short, we will deal with Algo HFT trading for alpha generation and market making. We will try to make several further steps upon carefully selected recent papers in these areas. Our contributions will be both on the empirical and the theoretical sides. We provide further details in the next paragraphs.

It is a well-known fact that generally, it is very hard to predict the market, because it depends on too many factors such as the financial statements of the traded companies, macro environment, the political situation of the countries in which they operate and, last but not least, the psychology of the traders. In the short term, the latter serves as a dominating factor. So, data driven machine learning tools happen to give an edge when employing them. With the increase of the computational power

*Electronic address: m1@uni-sofia.bg
†Electronic address: m2@uni-sofia.bg
‡Electronic address: m3@uni-sofia.bg

in the recent years, there is a boom in using exactly such by the market participants – banks, funds, trading firms, and individual investors. We are confident that (deep) reinforcement learning might be a very efficient tool for capturing and exploiting patterns in the short-term market behavior. The unique features of these models, based on the special role of the teacher within them, the presence of a feedback loop, and the exploration properties, make them particularly attractive especially in a financial context.

Problem formulation. Our paper consist of two main pillars (sub-problems). This approach would allow to study a practical problem in Algo HFT from different angles guaranteeing highest possible research value. As already implied, we will focus on alpha-generating strategies (pillar 1) and market making/optimal execution (pillar 2). Concretely, we intend to take a fixed security (stock price, or alternatively an exchange rate) and see how we can form an alpha HFT strategy using reinforcement learning in two related contexts.

In pillar 1, we employ only the basic characteristics of the security, bid-ask prices and volume with the less that burdensome data needs allowing to set a larger time interval. Employing tick-size data, or close to such, over a fixed trading horizon (day, week, month, year, etc.), we dig on a profitable strategy. There is plenty of literature on how this can be done with classical means such as financial, mathematical, and econometric/statistical modelling (e.g. [1]), however, under a reinforcement learning setting, it is scarce. Although there are some studies done, they are still in their infancy, or being even incomplete, in a sense that apart from the mathematical or computer science intuition, the financial one is vague. Namely, the right connection to structural financial concepts as risk, return, and arbitrage is lacking. This leads to a general lack of consistency and gives questionable practical value. We intend to face these challenges or at least part of them.

For this pillar, we will focus on the study of Zengeler and Handmann from 2020 in [8]. In our opinion, it is not only a recent one, thus considering some of the latest developments in HFT, but also a pioneering one in the application of reinforcement learning to the latter. We will extend it in two directions. First, we think that there are gaps in the empirics in a sense that the authors rely on some simulated data. We plan to apply their research to broader real market data sets. This will also be done for other market instruments over a longer time horizon. As elaborated above, we also intend to make where possible consistent referrals to finance concepts. So surgically we will intervene and modify appropriately the state space so that we can play with the risk/return characteristics of the strategy.

In pillar 2, we employ a larger data set. Namely, we take advantage of the whole LOB. The best would be to view it as an addition to the data from pillar 1 and consider its dynamics over the same time horizon. The problem is that this is a very daunting task due to: (i) just for a single day the LOB data is usually of a very large size, for liquid stocks typically ˜0.5 GB, (ii) the LOB itself as an object requires its own modelling for a single trade session. So, in pillar 2, we will restrict to the "static" case of a single trade day for one/several stocks, or alternatively exchange rates. The difference from pillar 1 is visible. Here the emphasis is not on trading dynamically over a fixed horizon, but rather on optimal trade execution in a trade session considering the state of the order book, thus on an efficient market making. Alternatively, the problem can be viewed as limited pillar 1 case - alpha generation in a day trading over a broader market information in the face of LOB. Pillar 2 is relatively more demanding and for this reason it will be our main focus.

For this pillar, we focus on the study of Lim and Gorse from 2018 in [6]. There they employ the classical Avellaneda-Stoikov model (see [2]) for market making and modify it appropriately under a reinforcement learning setting. The problem is that they work with simulated data and it is not clear how the new model will perform under real market data. Furthermore, their simulated data is not as deep as the real LOB is, which is a significant simplification. Additionally, there is an open scope for improvements in terms of better risk handling through employing different utility functions and regularizations. We will make steps in exploring this venue too. The general better handling of the structural financial terms and concepts as in pillar 1 stays too. Last but not least, considering the LOB as a day trading exercise opens additional horizons.

Investigating the dynamics of the LOB as a sort of merging of pillars 1 and 2 is an important but challenging question. The former can be viewed as relying on time series data, the latter on cross-section data (within the LOB object). So, the full dynamics is similar to panel data modeling with extreme dimensions on both sides. Comparing the results from the two pillars will pave the way for a more coherent future research in the general case. We will leave this question for a forthcoming add-up paper.

Modeling preliminaries. Here we start by some LOR heuristics which will gradually lead us to model formulation specifics of the two cases we consider. Then we will go to the technical details elaboration and their reinforcement learning projection. We will focus only on the essential modeling features and will leave the reader to refer for the precise technical aspects to the classical text of [7].

The stock market facilitates the electronic trading of securities through an instantaneous double auction. At each time instant, the market demand and the supply are represented by an electronic LOB. The latter is a cross-section of orders to execute at various price levels away from the market price. Electronic market makers quote on both sides of the market. They try to capture the bid–ask spread. Occasionally, a large market order, or a succession of smaller markets orders, can consume an entire price level. This is the reason the market price to fluctuate, especially when the market is liquid. This effect is often referred by the traders as a "price-flip." A market maker can mount losses if only one side of the order is filled as a result of an adverse price movement.

A LOB snapshot taken at time t is shown in Figure 1. The market marker places limit orders and they are denoted by the "+" symbol - red meaning a bid and blue meaning an ask. A buy market order comes up and matches the entire outstanding quantity of best ask quotes. Then at event time t+1 the limit order book is updated. The ask of the market maker has been filled (blue "-" symbol) and the bid goes away from the inside market. For avoiding adverse price selection, a preemptive strategy can be used. The ask is reconsidered and quoted again at a higher ask level. The bid is not changed. The market maker succeeds to capture a tick more than the spread when the both orders get filled.

Machine learning can be used to predict the price fluctuations. We can view queue sizes at each price level as input variables. We can additionally include properties of market orders. The latter should be done such that our machines deem most relevant to predicting the direction of price movements. In contrast to stochastic modeling, we do not impose conditional distributional assumptions on the independent variables nor we assume that price movements are Markovian. We can say also that supervised learning is ultimately not the best machine learning approach since cannot capture the effect of market impact and is too inflexible to incorporate more complex strategies. Reinforcement learning has the enough flexibility to allow us to formulate alpha strategies both on the whole LOB and on just the market price quotes (pillar 1 and pillar 2 of our paper respectively).

Model setup. We consider the problem of high-frequency market maker. We focus on a time-independent optimal policy. Let's assume that a market maker seeks to capture the bid–ask spread by placing one lot best bid and ask limit orders. The inventory shall be between -1 and 1. The problem is when to optimally bid to buy("b"), bid to sell ("s"), or hold ("h"), each time there is a limit order book update. Sometimes it may be more beneficial to quote a bid to close out a short position if it will almost surely yield an instantaneous net reward, other times it may be better to wait and capture a larger spread. In this simplified example, the agent uses the liquidity imbalance in the top of the LOB as a proxy for price movement and thus fill probabilities. The example does not use market orders, knowledge of queue positions, cancellations, and limit order placement at different levels of the ladder.

Concretely, we have at each non-uniform time update, t, the market feed provides best prices and depths: $\{p_t^a, p_t^b, q_t^a, q_t^b\}$. The state space is the product of the inventory, $X_t \in \{-1, 0, 1\}$, and gridded liquidity ratio $\widehat{R}_t = [\frac{q_t^a}{q_t^a + p_t^b} N]$, where N is the number of grid points and q_t^a and q_t^b are the depths of the best ask and bid. $\widehat{R} \to 0$ is the regime where the mid price will go up and the ask is filled. Alternatively, for $\widehat{R} \to 1$. The dimension of the state space is chosen to be 3x5=15.

A bid is filled with probability $\epsilon_t = \widehat{R}$ and an ask is filled with probability $1 - \epsilon_t$.The rewards will be the expected total profit and loss (PnL). If a bid is filled to close out a short holding, then the expected reward is $r_t = -\epsilon_t(\Delta p_t + c)$, where Δp_t is the difference between the exit and the entry price and c is the transaction cost.

In this model formulation, we apply SARSA and Q-learning for optimal market making. The exact details of the algorithms can be found in [7]. Our code implementations are given in https://github.com/RLprojectTeam4/RLProject. Both the comments there and the output figures give a complete picture of the model performance. Further details are given in Figure 2.

If all that was for pillar 2 of out research setting, for pillar 1 we just take the market quotes of a stock and apply deep Q-learning to be able to forecast the price movement and build a profitable strategy. The details are also given in the code and its comments for convenience. We employed Lobster data and focused on the google stock.

Results. We see that both leg 1 and leg 2 achieve to generate profit for the trader and market maker respectively. All that is the best proof for the model reliability. Our simulations with different stocks and day periods give similar results. Both Q-learning and SARSA provide good results with the two methods converging for many time steps.

The setting paves the way for further applications. As elaborated, we plan to make such in considering the LOB dynamics. Additionally, an interesting application would be to consider deep reinforcement learning for leg 2 as we did for leg 1. A real hindrance for this is the extreme computational time. Further application could be to consider inverse reinforcement learning for both the cases. This would be the highest possible generalization.

References.

[1] Aldridge, I., (2013), "High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems," Hoboken: JohnWiley & Sons, Volume 604.

[2] Avellaneda, M, Stoikov, S., (2008), "High-Frequency Trading in a Limit Order Book," Quantitative Finance, 8(3):217-224.

[3] https://en.wikipedia.org/wiki/Algorithmic_trading.

[4] https://en.wikipedia.org/wiki/High-frequency_trading.

[5] https://en.wikipedia.org/wiki/Order_(exchange) #Limit_order.

[6] Lim, Y. S., Gorse, D., (2018), "Reinforcement Learning for High-Frequency Market Making," In ESANN.

[7] Sutton, R., Barto, A., (2018), "Reinforcement Learning: An Introduction," 2nd ed. Bradford Books.

[8] Zengeler, N., Handmann, U., (2020), "Contracts for Difference: A Reinforcement Learning Approach. Journal of Risk and Financial Management," 13. 78.

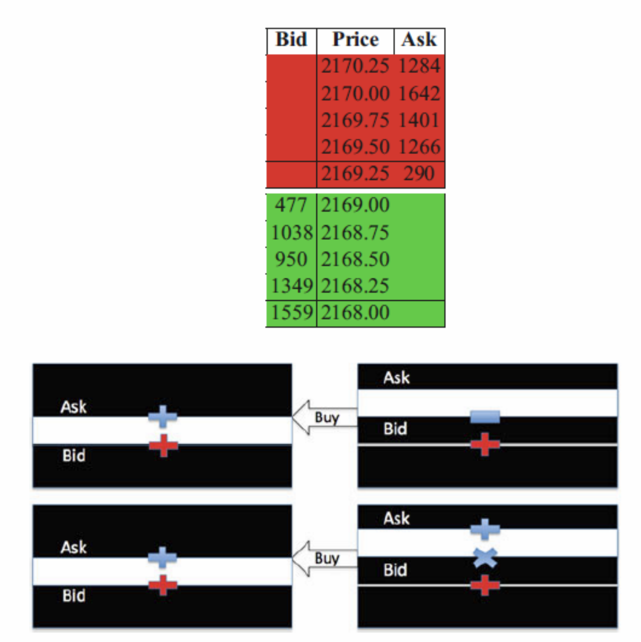

Figure 1: LOB

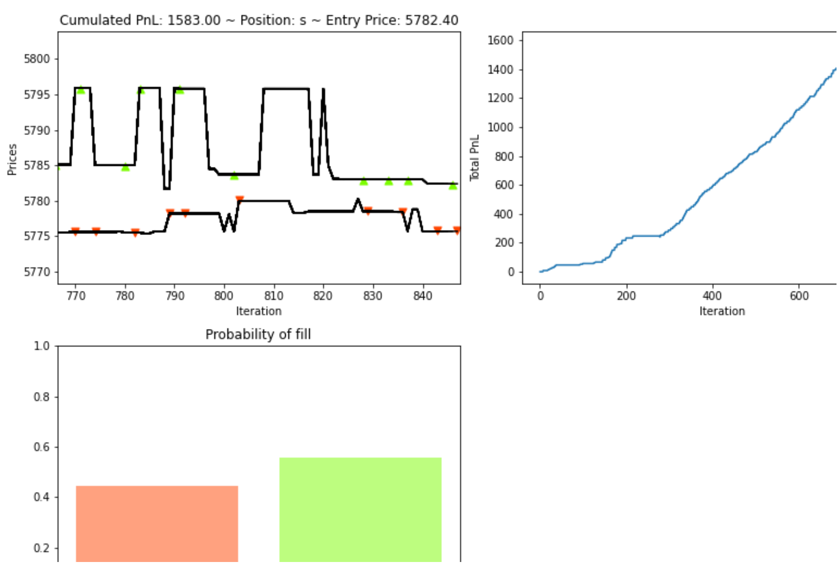

Figure 2: Leg 1 results