# Trick or Treat? The Effects of High-Frequency Trading on Financial Markets

Matei Budiu     Ahmad Ghalayini     Matthew McCorkle

June 2021

## 1 Abstract

High-frequency trading (HFT) is a relatively new, quickly growing method of trading on financial exchanges involving algorithms that conduct trades on the time scale of milliseconds. Despite its substantial growth and the fact that over 50% of all orders by volume come from such algorithms, we do not know much about the effects of HFT on the market. It is generally difficult to conduct testing of high-frequency trading in a controlled environment to investigate its effects because realistic controlled environments are not easily available.

In this project, we study HFT's effects on the market by (1) implementing a financial exchange simulator and (2) running a commonly used HFT algorithm—Stale-Quote Arbitrage—on that simulator. We look at several metrics, including bid-ask spread and liquidity, to investigate the potential impact of this algorithm on markets and market dynamics, to aid lawmakers in developing regulations of these practices.

## 2 Significance of High-Frequency Trading

High-frequency trading is a relatively new technique of trading on the stock market with computer algorithms deciding and submitting orders on the time scale of milli- and microseconds. Because it can be very profitable, and because financial exchanges often encourage such trading due to it generally providing liquidity, high-frequency trading has become very prevalent lately, with over half of trades on the stock market today being done by high-frequency traders [4, 7]. Because of its sheer presence, the effects of high-frequency trading on the market can be significant and worth investigating. Hopefully, our results will contribute to the debate over the true effects of high-frequency trading and assist legislators in making more informed decisions [12, 2, 4].

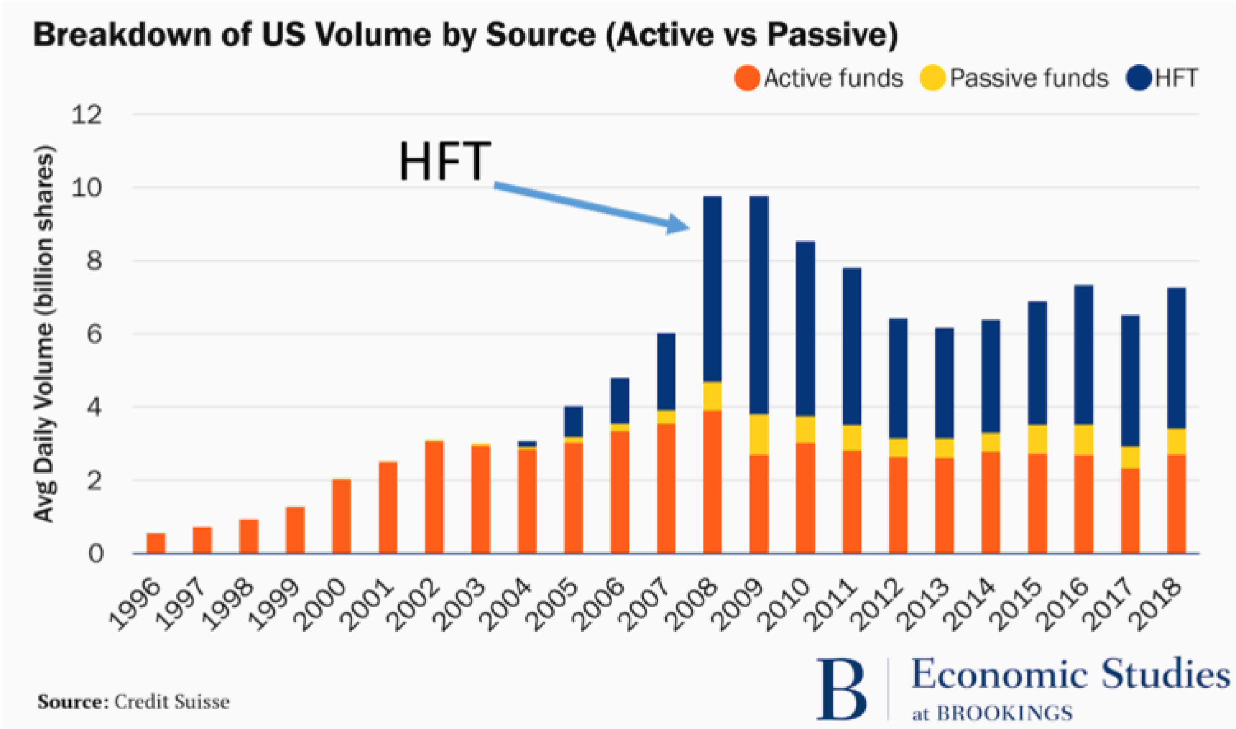


*Figure 1: HFT has increased in prevalence lately. [i1]*

High-frequency trading has led to a sort of "latency arms race" on the metric of latency—the time between two endpoints, in this case the server running the financial exchange's matching algorithms and the client running the trader's trading algorithms [4]. Traders who have low latency can receive new market data and have their orders processed quickly, allowing for opportunities to snatch good deals first, making low latency a valuable resource. As a result, many high-frequency trading firms have begun to compete over reducing latency, using techniques such as setting up microwave towers for data transmission and buying spots in the stock-exchange data centers to reduce latency [6, 11]. This competition leads to unfairness, which might harm the economy [4].

# 3 High-Frequency Trading in Other Research

Today, there is a major debate between theoretical and practical economists over whether high-frequency trading is beneficial for the market [12, 2, 4, 1]. Theoretically, it is predicted to have negative consequences for the market, benefiting only certain insiders at the expense of everyone else [4]. Algorithmic traders have also caused seemingly random market crashes for no apparent reason in the past, with algorithms entering in a feedback loop with each other due to unexpected circumstances [3]. But in a real experiment in which such trades were discouraged by legislation in Canada, a negative effect on the market was seen—with regulatory fees resulting in a 30% decrease in liquidity and a 9% increase in the bid-ask spread, suggesting a negative impact [12].

The SEC has outlined multiple practices used by HFT algorithms, each with potential varying effects on the market [2].

# 4 Metrics

In order to measure the effect of high-frequency trading on the market, we observe two metrics which have been used in the past as a measure of market health: bid-ask spread and liquidity. We track how these metrics change for traders over time [12].

The **bid-ask spread** describes how close to each other prices are in limit order books. We define the bid-ask spread here as the difference in limit price between the lowest sell order and the highest buy order. A tighter bid-ask spread is generally favorable, with traders being more likely to get a fair price, whereas a looser bid-ask spread results in more variation in prices, leading to trades being made farther away from the market price.

**Liquidity** is a measure of how easily stocks can convert to real money. We define liquidity here as the order-book size—the total number of buy and sell orders in an order book at one time. More trading in a market increases liquidity, as it is faster for orders to be matched. A higher liquidity is generally seen as positive.

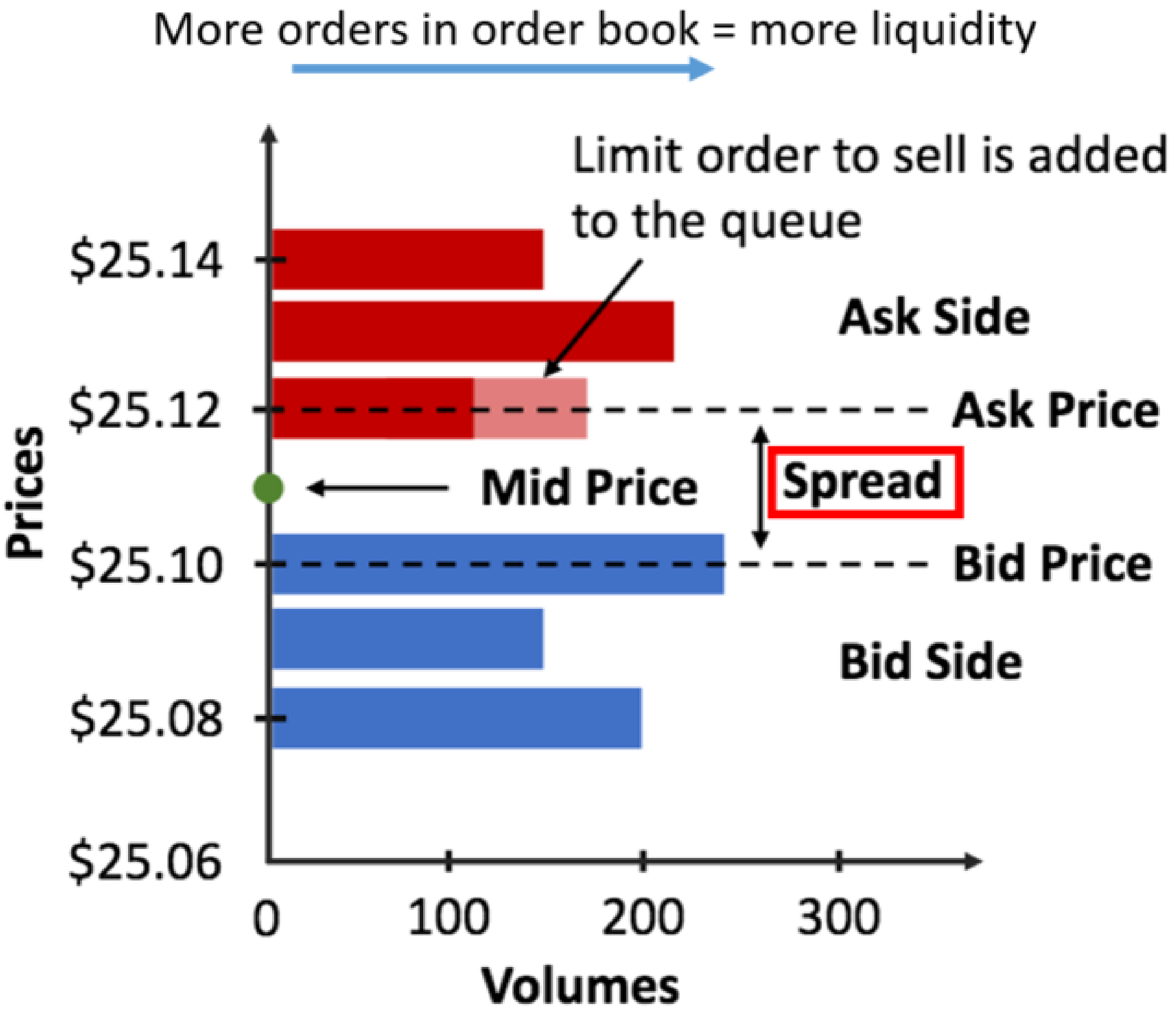


*Figure 2: Liquidity and bid-ask spread [i2]*

Additionally, we look at one other additional metric: **agent net worth**—the sum of a trader's value in cash and shares. Although this metric itself doesn't say much about the market as a whole, it can tell us how the traders are interacting with each other.

As each simulation is run, data points are collected whenever orders are added to the order book, and additional statistics are saved at the end of each simulation.

# 5 The Simulator

We ran our experiments on an open-sourced discrete-event simulator created specifically for the purpose of this project [8]. The simulator is agent-based, with each trader being an agent using strategies based on information received from the exchange. Each agent can have a different latency function, which determines the time it takes information to travel between the exchange and the agent.

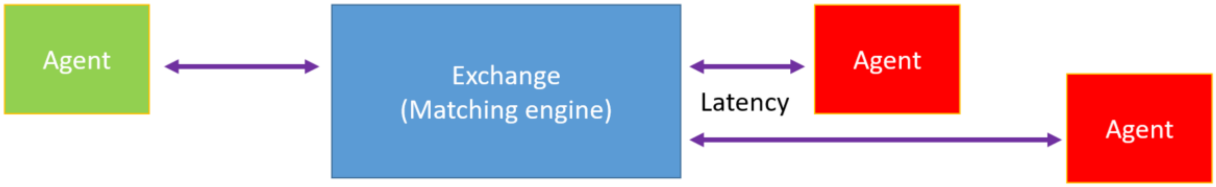


*Figure 3: It takes information a certain latency to travel between the matching engine and an agent.*

The simulator uses an event queue—each event has a time when it is scheduled to happen, and the queue is sorted by each event's scheduled time. Events are resolved in order of increasing time.

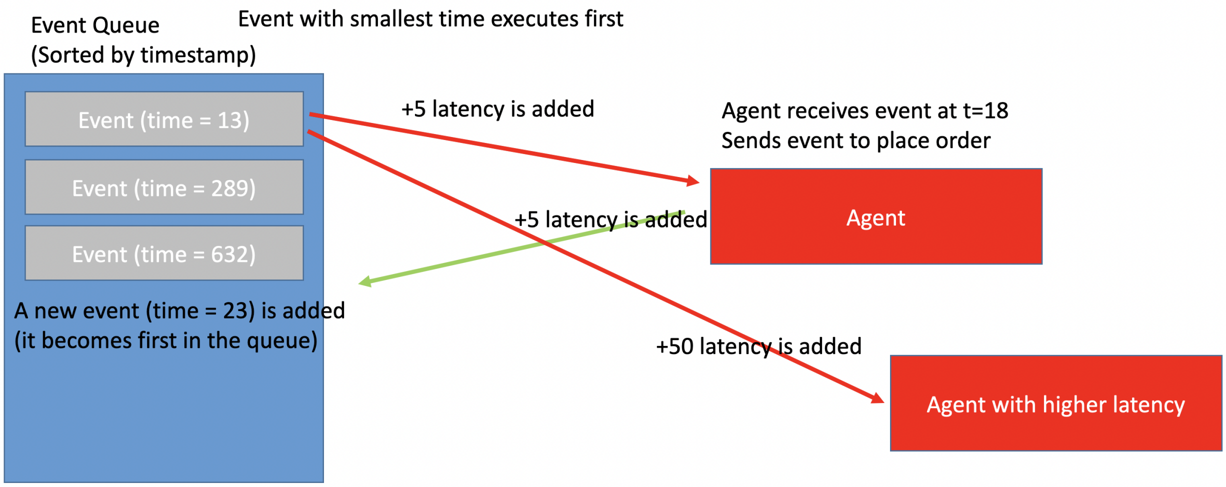

Event Queue
(Sorted by timestamp)
Event with smallest time executes first
Event (time = 13)
Event (time = 289)
Event (time = 632)
A new event (time = 23) is added
(it becomes first in the queue)
+5 latency is added
Agent receives event at t=18
Sends event to place order
+5 latency is added
Agent
+50 latency is added
Agent with higher latency

*Figure 4: The event with the smallest scheduled time is removed from the queue and run. The execution of this event may result in more events being added to the queue.*

# 6 The Agents

Our simulations have two different classes of agents—in our experiment, (1) background agents simulate background activity (which isn't necessarily high-frequency trading) in a market, like regular day traders; and (2) experimental agents which act as the high-frequency traders. The background agents are a control—they are not changed across different experiments, whereas the experimental agents are the independent variable and are different in each setup. For this project, we decided to use zero-intelligence agents for the background agents and stale-quote—arbitrage agents for the experimental agents.

# 7 Zero-Intelligence (ZI) Agents

For the background traders, we used a simplified model of ZI agents as outlined in another article [13], rather than replaying previous real market data [5]. These agents submit orders to the order book mostly randomly and cancel old orders if they have not matched within a certain amount of time after their submission. Although ZI agents may seem to act irrationally and randomly, previous research has shown that their behavior as a whole can model markets effectively in many cases [9].

Our ZI agents trade at a price determined by a few factors. Any submitted order has a 50% chance that to be a buy order, and a 50% chance that to be a sell order. ZI agents are not allowed to have more or less shares than a certain given value.

The ZI agents first determine their valuation for a share from two factors: (1) a per-simulation fundamental value shared by all the agents, which imitates the price of the share. Every tick (a time interval we defined as the time it takes for the order book to process an order), the fundamental has a chance to change its value. If its value is set to change, the current value is brought closer to a predefined mean value of the share price by a given percentage and a random "shock" number is added to that to get the new price. (2) A private share valuation. An array is built for each agent upon creation, filled with numbers pooled from a normal distribution and then sorted in descending order. The value in the array at the index corresponding to how many shares the ZI agent currently owns is added to the fundamental value to calculate the price valuation of the share. Then, a random shading factor is added, if selling, or subtracted, if buying, in order to make profit from a match.

# 8 Stale-Quote Arbitrage

We look at one of the practices outlined by the SEC as being used by high-frequency trading algorithms: stale-quote arbitrage (SQA). We test the effects of this algorithm on a market where SQA traders have higher latencies than other SQA traders.

In stale-quote arbitrage, algorithms pick off older "stale" limit orders, which do not anymore reflect the current market data, potentially leading to a bad deal for the trader who placed the now stale order. These SQA agents regularly request a few of the "best deals" in the order book. They then subtract the current fundamental value from prices of the "best deals" to get a difference. If this difference is past a certain threshold for a certain order, these algorithms send an order to attempt to match with that order.

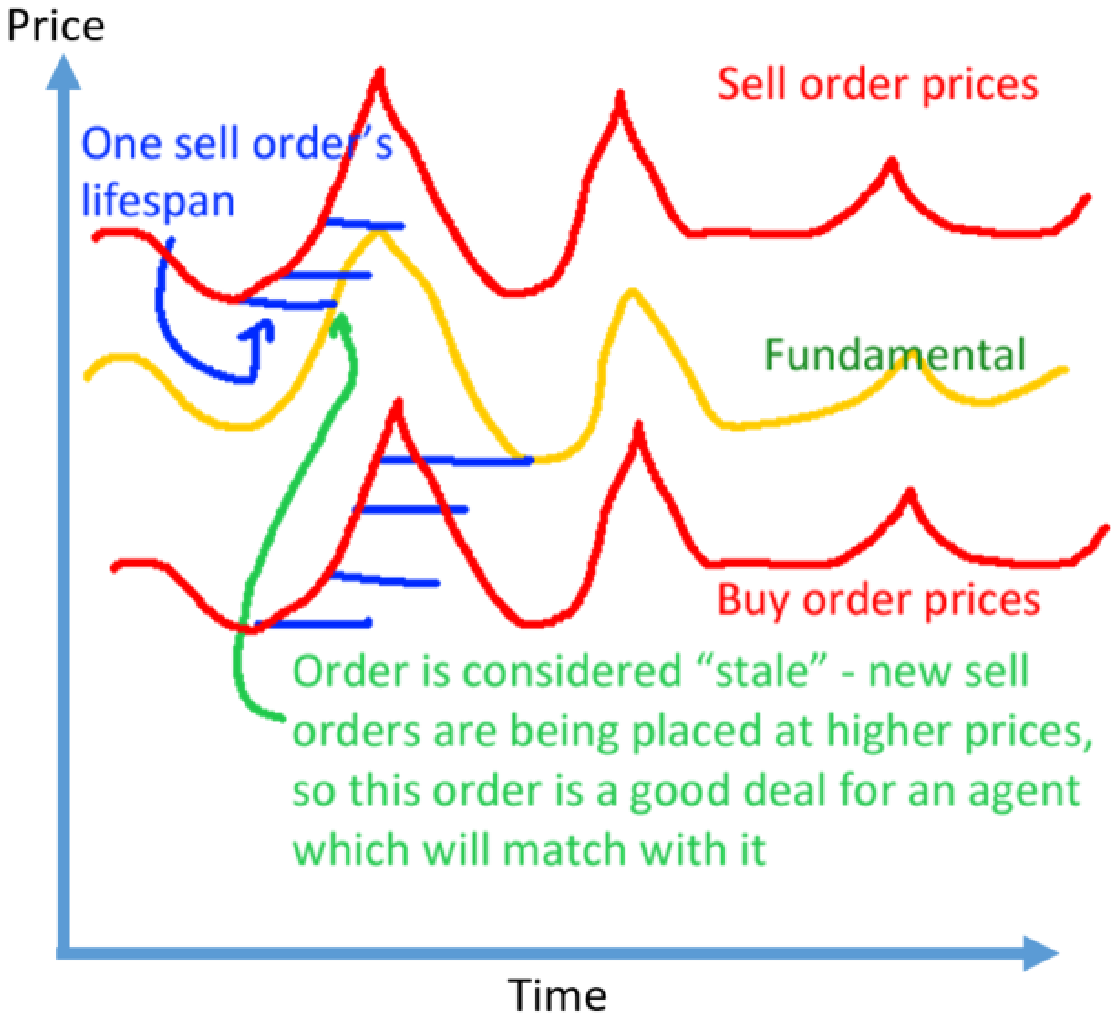

Price
One sell order's lifespan
Sell order prices
Fundamental
Buy order prices
Order is considered "stale" - new sell orders are being placed at higher prices, so this order is a good deal for an agent which will match with it
Time

*Figure 5: Our implementation of stale-quote arbitrage*

## 9 The Simulations

We ran five different configurations as experiments, with each having different setups of experimental agents. Each configuration was run in 100 separate identical simulations, to account for noise. Simulations were run for 1 million time units, with one time unit corresponding to the time it takes the matching engine to process one order.

For the background agents, we used 66 identical ZI agents for all of the simulations.

For the experimental agents, we split the stale quote arbitrage (SQA) agents into two classes—high latency and low latency. The high-latency SQA agents were given a latency 20 times greater than the low latency ones. Agents' latencies were distributed based on normal distributions - the low-latency SQA agents had a latency with a mean of 5 and a standard deviation of 1, while the high-latency ones had a mean of 100 with a deviation of 20. This difference in latency between high-latency and low-latency SQA agents simulates small businesses or garage HFTs fighting against big businesses which can afford to colocate—to put their algorithms in the same data center as the exchange itself for very low latency.

The five experiments we ran were as follows: (1) no experimental agents, (2) 1 low-latency SQA agent, (3) 1 high-latency SQA agent, (4) 2 low-latency SQA agents, and (5) 1 low-latency and 1 high-latency SQA agent together. The exact configurations are available from the GitHub repository of the simulator [8].

# 10 The Results

## 10.1 Control Group: No Stale-Quote Arbitrage

When there are no SQA agents, the zero-intelligence (ZI) agents' net worth doesn't change much on average, the liquidity (order book size) settles between 60 and 70, and the bid-ask spread settles approximately between 0 and 20.

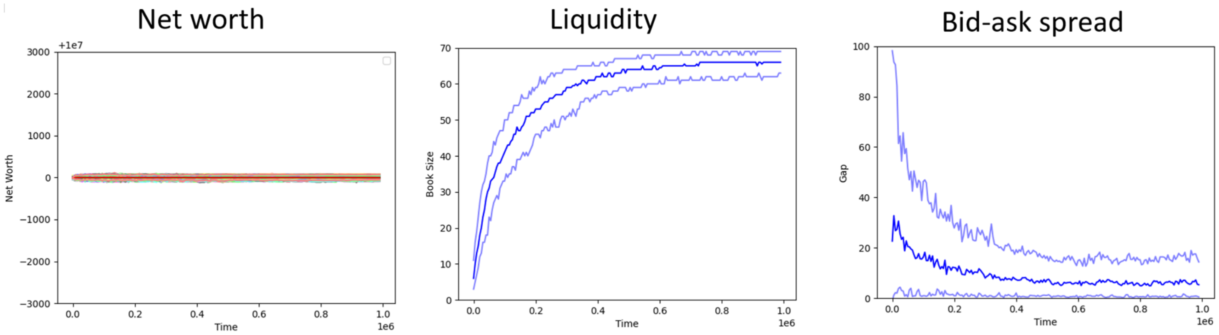


*Figure 6: The market metrics as they change over the course of a simulation as a summary of 100 simulations with only 66 identical ZI agents. The three lines for each metric show the 5th, 50th, and 95th percentiles obtained from the simulations, respectively.*

## 10.2 One Low-Latency SQA Agent

When a single low-latency SQA agent is added to the ZI agents, the liquidity decreases to about the range 50-60, while the bid-ask spread increases to around between 20 and 40. Both these changes are undesirable for a market.

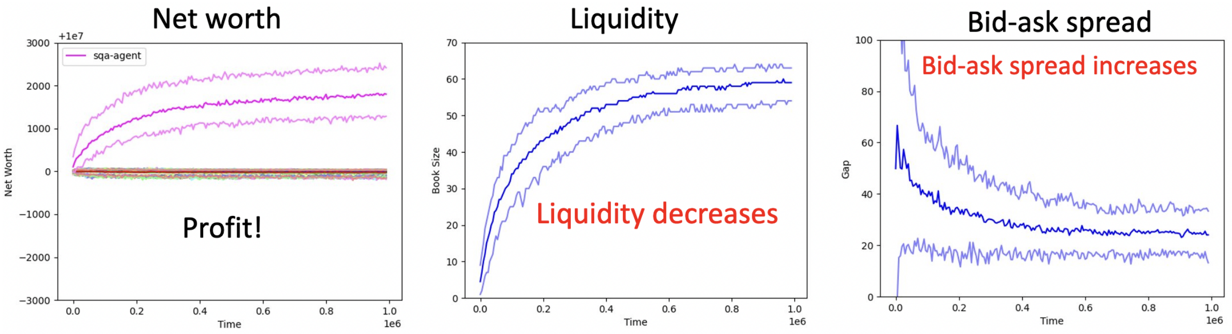
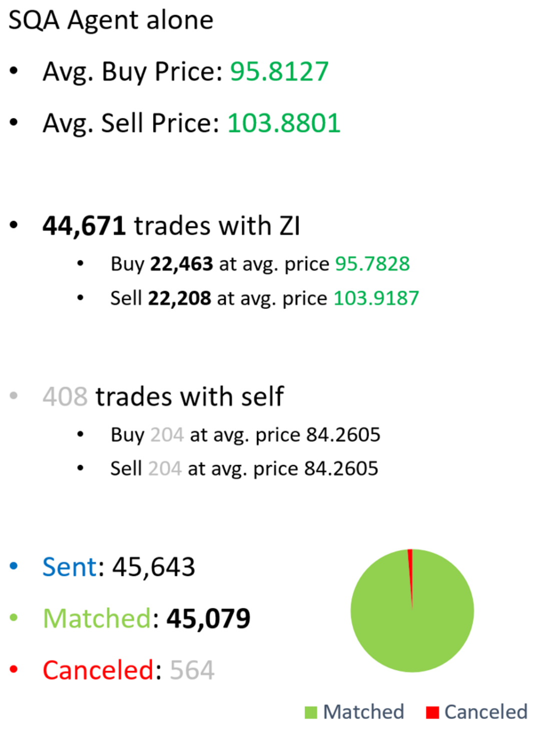


*Figure 7: Market metric summary graphs for simulations with one low-latency SQA agent (shown in pink on the net-worth graph) along with 66 ZI agents: the SQA agent makes a profit but harms market liquidity and bid-ask spread (compared to the ZI-only simulation).*



SQA Agent alone

- Avg. Buy Price: 95.8127
- Avg. Sell Price: 103.8801

- **44,671** trades with ZI
  - Buy **22,463** at avg. price 95.7828
  - Sell **22,208** at avg. price 103.9187

- 408 trades with self
  - Buy 204 at avg. price 84.2605
  - Sell 204 at avg. price 84.2605

- Sent: 45,643
- Matched: **45,079**
- Canceled: 564

🟩 Matched
🟥 Canceled

The SQA agent makes a solid profit off of the ZI agents. Overall, it does a good job buying at low prices and selling at high prices, and it only cancels a small fraction of its orders.

## 10.3 One High-Latency SQA Agent

When the previous configuration is repeated, but with the SQA's latency increased by a factor of 20, liquidity and bid-ask spread look quite similar to the previous configuration's results, except for the liquidity being slightly higher and the bid-ask spread being slightly lower (compare figures 7 and 8). Overall, this situation is still undesirable to the market as a whole when compared to the control (figure 6).

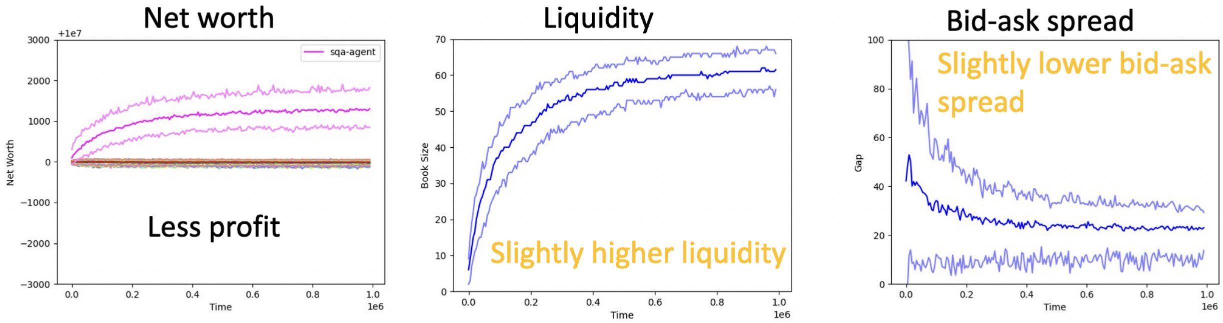
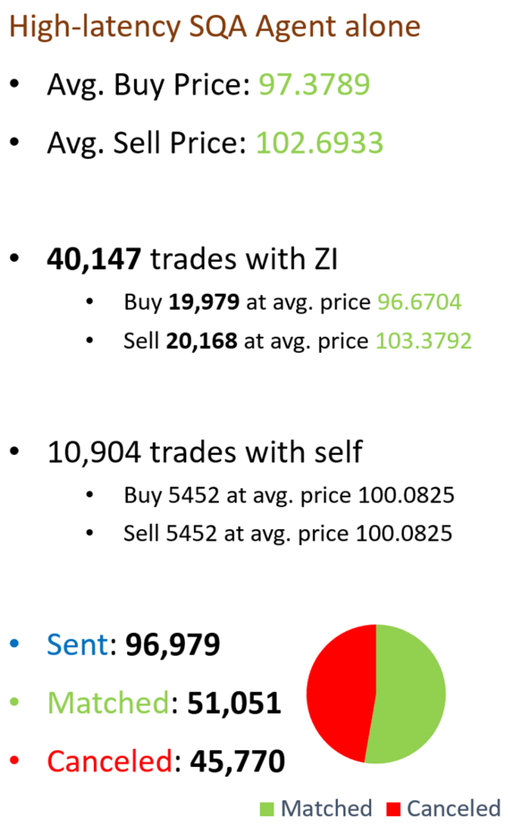


*Figure 8: Market metrics for one high-latency SQA agent (shown in pink on the net-worth graph): The SQA agent makes less profit but still harms market liquidity and bid-ask spread (compare with figure 6), although a bit less than when the SQA agent's latency was lower (compare with figure 7).*



High-latency SQA Agent alone

- Avg. Buy Price: 97.3789
- Avg. Sell Price: 102.6933

- **40,147** trades with ZI
  - Buy **19,979** at avg. price 96.6704
  - Sell **20,168** at avg. price 103.3792

- 10,904 trades with self
  - Buy 5452 at avg. price 100.0825
  - Sell 5452 at avg. price 100.0825

- Sent: **96,979**
- Matched: **51,051**
- Canceled: **45,770**

🟩 Matched 🟥 Canceled

The high-latency SQA agent still makes a profit off of the ZI agents, albeit a slightly lower one. It does a slightly worse job at buying at low prices and selling at high prices compared to a low-latency agent. This agent also sends almost twice as many orders as the low-latency one, although half of its orders are canceled, because they reach the order book after the initial order they were trying to match with was withdrawn. This agent also trades much more with itself overall.

### 10.4 Two Low-Latency SQA Agents

Having two low-latency SQA agents produces very similar curves for liquidity and bid-ask spread as just having one low-latency SQA agent (compare with figure 7).

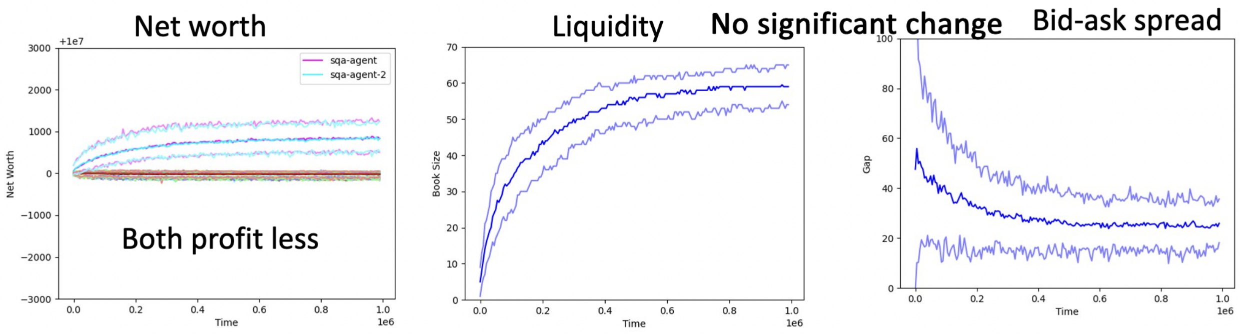
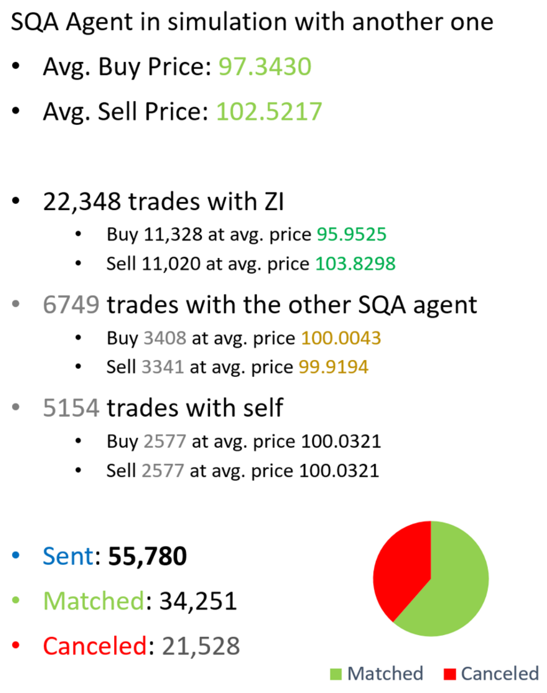


*Figure 9: When we have 2 identical SQA agents (pink and blue) in the net worth graph, they split the profit. Liquidity and bid-ask spread graphs are very similar to the simulation with only one low-latency SQA agent (figure 7).*



SQA Agent in simulation with another one

- Avg. Buy Price: 97.3430
- Avg. Sell Price: 102.5217

- 22,348 trades with ZI
  - Buy 11,328 at avg. price 95.9525
  - Sell 11,020 at avg. price 103.8298
- 6749 trades with the other SQA agent
  - Buy 3408 at avg. price 100.0043
  - Sell 3341 at avg. price 99.9194
- 5154 trades with self
  - Buy 2577 at avg. price 100.0321
  - Sell 2577 at avg. price 100.0321

- Sent: **55,780**
- Matched: 34,251
- Canceled: 21,528

When paired with another low-latency SQA agent, the agent makes about half as much profit as when it is alone in the simulation. Each paired agent sends more orders, but matches fewer, canceling more. The agents trade more with themselves when paired, and a significant number of trades happen between the two paired agents. Both traders are equally good at matching at favorable prices with the ZI agents, but get fewer opportunities to do so, as both are competing for the same ”stale” orders.

### 10.5 One Low-Latency and One High-Latency SQA Agent

When two SQA agents with differing latencies are used, and one of the agents has a latency 20 times greater than the other, the liquidity and bid-ask spread graphs look almost identical to the simulations with one or two low-latency SQA agents (figures 7, 9).

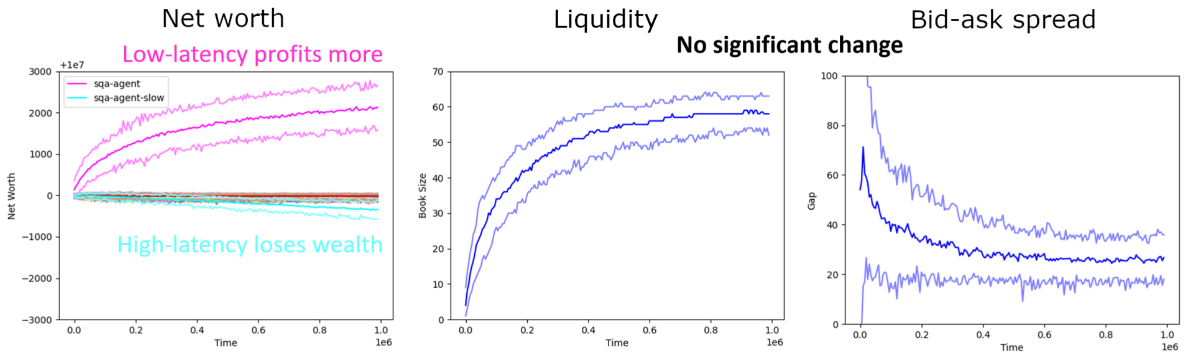
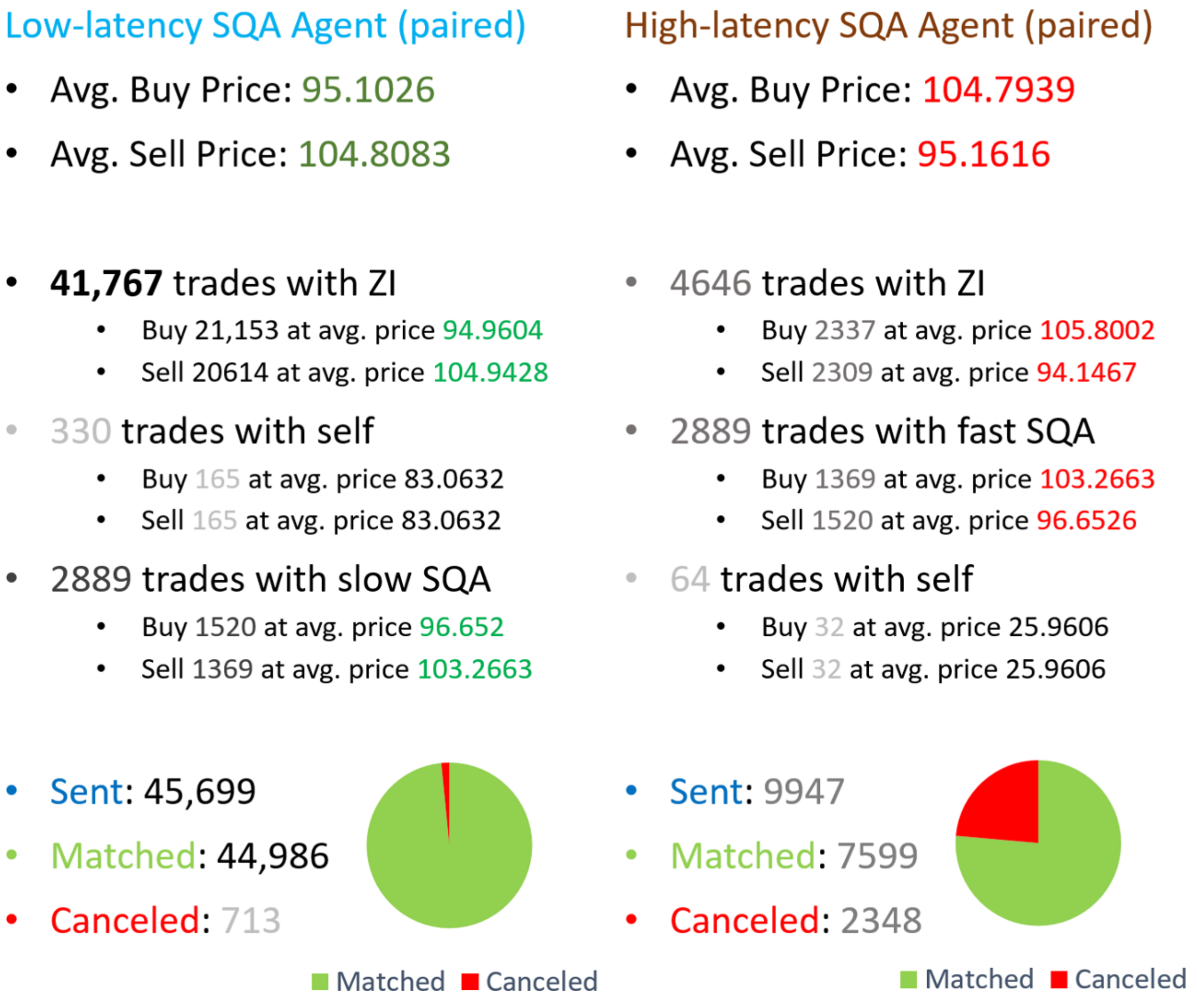


*Figure 10: When a low-latency SQA agent (pink in the net-worth graph) is combined with a high-latency one (blue in the net-worth graph), there is a leeching effect, where the low-latency agent steals the high-latency agent's profit. There is, however, no change in the market health compared to the simulations with one or two low-latency SQA agents (figures 7 and 9)*



Low-latency SQA Agent (paired)

- Avg. Buy Price: 95.1026
- Avg. Sell Price: 104.8083

- **41,767** trades with ZI
  - Buy 21,153 at avg. price 94.9604
  - Sell 20614 at avg. price 104.9428
- 330 trades with self
  - Buy 165 at avg. price 83.0632
  - Sell 165 at avg. price 83.0632
- 2889 trades with slow SQA
  - Buy 1520 at avg. price 96.652
  - Sell 1369 at avg. price 103.2663

- Sent: 45,699
- Matched: 44,986
- Canceled: 713

High-latency SQA Agent (paired)

- Avg. Buy Price: 104.7939
- Avg. Sell Price: 95.1616

- 4646 trades with ZI
  - Buy 2337 at avg. price 105.8002
  - Sell 2309 at avg. price 94.1467
- 2889 trades with fast SQA
  - Buy 1369 at avg. price 103.2663
  - Sell 1520 at avg. price 96.6526
- 64 trades with self
  - Buy 32 at avg. price 25.9606
  - Sell 32 at avg. price 25.9606

- Sent: 9947
- Matched: 7599
- Canceled: 2348

The low-latency SQA agent ends up making a greater profit when paired with the high-latency SQA agent, which loses money, compared to the simulation with one low-latency SQA agent. The low-latency SQA sends, matches, and cancels about the same number of trades in both scenarios. However, when the high-latency SQA agent is present, some of those matched trades are with the high-latency SQA agent. The low-latency SQA agent gets even better deals on average when paired than when alone. However, the high-latency SQA greatly suffers when being paired. In both its trades with ZI agents and with the lower-latency SQA agent, it makes horrible deals. It also sends way fewer orders in general, and very rarely trades with itself. There definitely is a leeching effect here, with the low-latency SQA agent eating up the profits of the high-latency SQA agent by both matching with all the ZI agents' good deals and taking advantage of the high-latency SQA agent's orders.

# 11 Conclusions

Based on our simulations, stale quote arbitrage agents seem to be harmful to the market, regardless of their latency—their presence results in an increased bid-ask spread and in decreased liquidity, both of which are detrimental. Additionally, we find that the profit of an SQA agent is closely related to its latency. More specifically, there seems to be a leeching effect, where lower-latency agents leech the profit of the higher-latency ones. Absolute latency seems to be less significant for profit than relative latency. This finding holds when we have 3 different SQA agents with different latencies as well.

Overall, our results support the idea that regulating stale-quote arbitrage in the stock market could have beneficial effects.

# 12 Next Steps

Future steps to take could include refining our model for SQA and the market to be a more accurate representation of the real market. Modeling regulations to SQA in order to see the their effects on market health would be helpful in order to determine which would work best. Or, research could be done on how to solve the problem of latency unfairness. These algorithms could also be run in an environment such as CloudEx [10], a research stock exchange platform which attempts to compensate for latency, to see if latency still gives an advantage.

# References

[1] Fia epta comments on the fca occasional paper "quantifying the high-frequency trading 'arms race'", 2020.

[2] Staff report on algorithmic trading in u.s. capital markets. Technical report, 2020.

[3] Eric M. Aldrich, Joseph Grundfest, and Gregory Laughlin. The flash crash: A new deconstruction. *SSRN Electronic Journal*, 2016.

[4] Matteo Aquilina, Eric Budish, and Peter O'Neill. Quantifying the high-frequency trading "arms race", 2021.

[5] Tucker Hybinette Balch, Mahmoud Mahfouz, Joshua Lockhart, Maria Hybinette, and David Byrd. How to evaluate trading strategies: Single agent market replay or multiple agent interactive simulation?, 2019.

[6] Jonathan Brogaard, Bjjrn Hagstrrmer, Lars L. Norden, and Ryan Riordan. Trading fast and slow: Colocation and market quality. *SSRN Electronic Journal*, 2013.

[7] Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621, Jul 2015.

[8] Matei Budiu. High-frequency trading financial exchange simulator on github (https://github.com/aehmttw/hftsimulator), 2021.

[9] J. Doyne Farmer, Paolo Patelli, and Ilija I. Zovko. The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences*, 102(6):2254–2259, 2005.

[10] Ahmad Ghalayini, Jinkun Geng, Vighnesh Sachidananda, Vinay Sriram, Yilong Geng, Balaji Prabhakar, Mendel Rosenblum, and Anirudh Sivaraman. Cloudex. In *Proceedings of the Workshop on Hot Topics in Operating Systems*. ACM, Jun 2021.

[11] Eun Jung Lee, Kyong Shik Eom, and Kyung Suh Park. Microstructure-based manipulation: Strategic behavior and performance of spoofing traders. *Journal of Financial Markets*, 16(2):227–252, May 2013.

[12] Katya Malinova, Andreas Park, and Ryan Riordan. Do retail traders suffer from high frequency traders? *SSRN Electronic Journal*, 2012.

[13] Elaine Wah, Mason Wright, and Michael P. Wellman. Welfare effects of market making in continuous double auctions. *Journal of Artificial Intelligence Research*, 59:613–650, Aug 2017.

Image sources:
i1. https://www.brookings.edu/opinions/congress-wants-to-tax-stock-trades-investors-shouldnt-fret/
i2. https://arxiv.org/pdf/2006.08682.pdf