Agent-Based Simulation of Fragmented Markets

Zehua Li

December 6, 2017

Abstract

This implementation using agent-based simulation models provides a fundamental implementation of market structure, individual markets, and agents in the markets with flexible market clearing rules and agent strategies. The model based on and used to verify the results of a study on the effect of Latency Arbitrage in continuous and discrete markets.

1 Introduction

The agent-based approach to studying financial markets allows one to replicate and generate scenario given carefully implemented market mechanism and agent strategies. With specifically chosen parameters, such models can help study the effect of certain mechanisms and actions, and further aid explaining underlying causes of certain phenomenas.

This implementation is based on the study by Wah et al.[2]. In the original study, Wah et al. Implemented a two-market model with zero-intelligence(ZI) trading agents and latency arbitrageurs. Wah et al. further demonstrated that latency arbitrage reduces trader's surplus and increases the execution time.

This implementation in Python enables a testing environment to verify the results from Wah et al.[2]. It defines the order structure, market structure, the security information processor (SIP), clearing rules, event scheduling for various processes, and agent strategies. It also expands on the two-market model, allowing an arbitrary amount of markets, each with an arbitrary amount of agents.

2 Two Market Model

2.1 Event graph

The basic environment is consisted of two market and a national best bid and offer (NBBO) statistics provided by the SIP. The SIP calculates and updates the NBBO with a certain latency δ. With δ existing, an agent with access to both markets and a better computation power will be able to obtain the best bid and offer information before those that rely on the NBBO. HFT agents are capable of such and can exploit it, which will be explained below.

Assuming that an agent is primarily associated with one market, and only routes its order to another if the NBBO is better and immediately executable, then a latency in updating the NBBO will mean that sometimes, there will be a potential matching order placed in a non-primary market, but will not be known to the agent at the time the order is placed.

When such a scenario occurs, an HFT agent can act as a latency arbitrageur (LA) and execute latency arbitrage by place an order in both markets, buying at a lower price and selling at a higher price, gaining the spread as a surplus.

2.2 Implementation

The order in this system stores the price, valuations, time of placement, and market the order is in. Valuation is used to determine if an agent will place an order, and the price it will place it at (with noise included). It will also be used to calculated the surplus generated from the order. Each Market Stores an ask queue and a bit queue. The SIP has a queue that stores the market status reports from each market. The NBBO updating events are scheduled with latency δ and are executed gradually as the simulation goes on. ZI agents are treated as the default agents in the

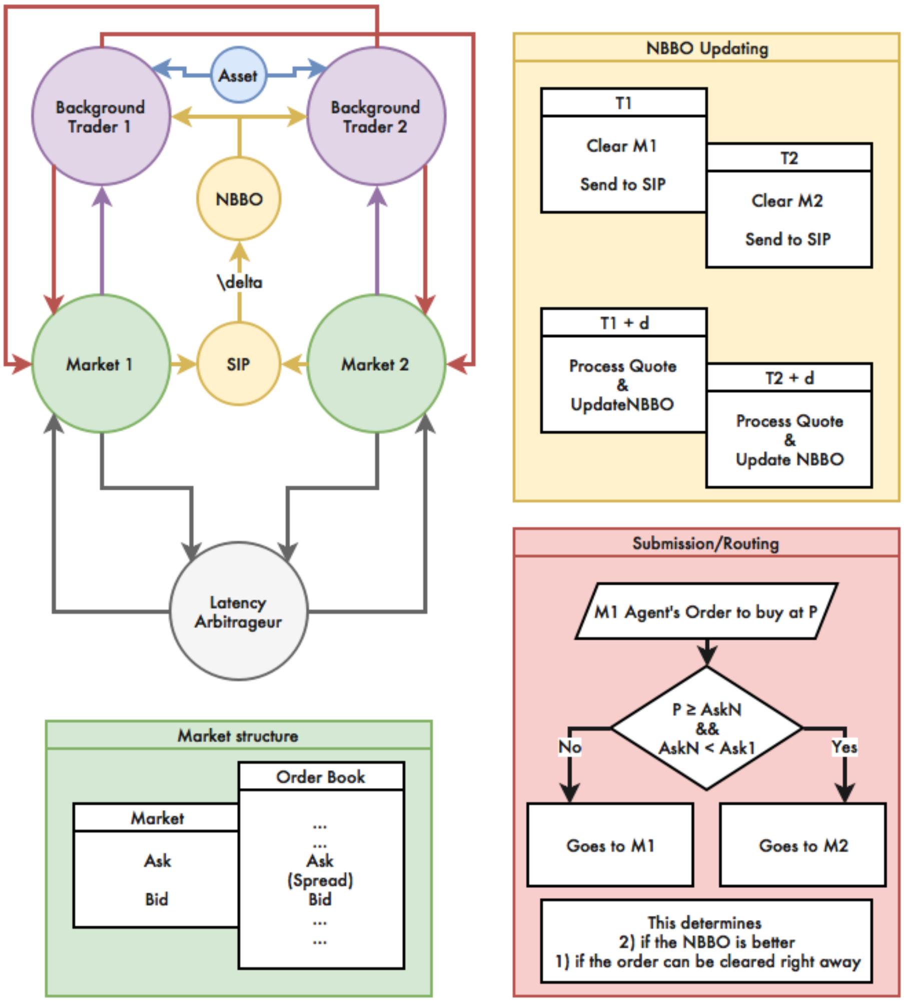


Figure 1: The event graph with explanations of the processes involved.

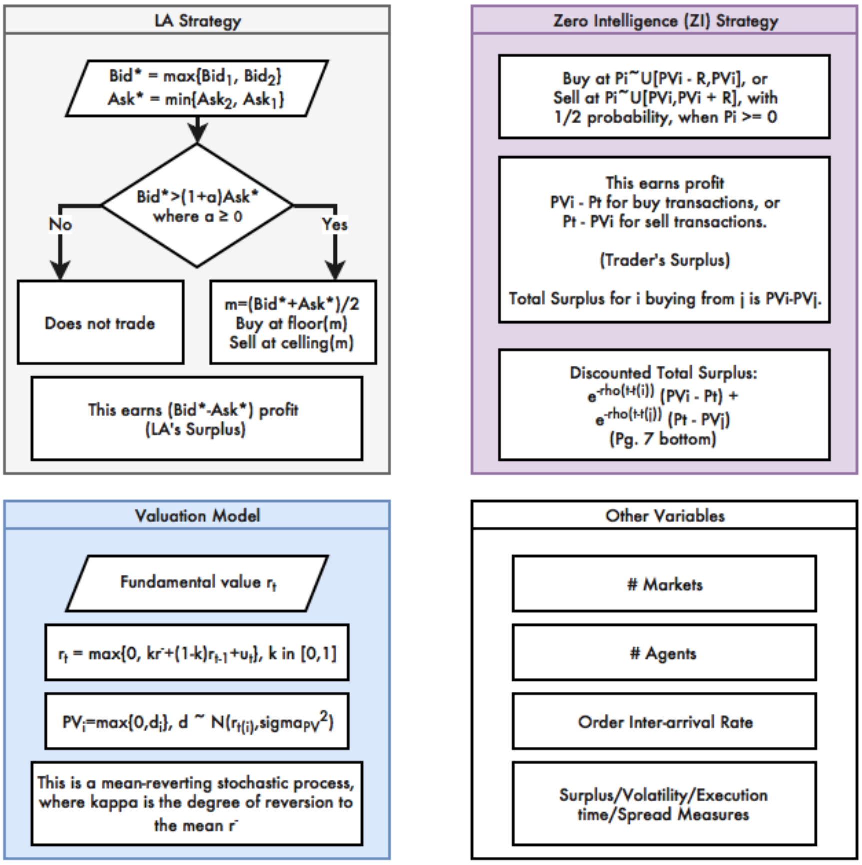


Figure 2: The event graph with explanations of the processes involved. (cont.)

system. Routing of a ZI agent's order will be based on the status of NBBO. The LA keeps track of the market's status itself and places orders based on whether a latency arbitrage opportunity has a large enough margin.

2.3 Simulation

In this Simulation, parameters are chosen to generate a long enough flow of orders as well as plenty of latency arbitrage opportunities. The volatility of the market is significantly large, where as the required margin of LA is tiny. Four different market conditions are tested: a continuous double action (CDA) market system (where a single continuous clearing market exist), a two-market system without LA, a two-market system with LA, and a central call (discrete clearing) two-market system with LA. All specifications can be found in the work of Wah et al.[2], with the exception of the clearing frequency of the central call system.

3 Results

The results from the implementation generally matches with that of Wah et al. As expected, Latency has no effect on the continuous double action market system (CDA), unless when we look at the NBBO price. In the two market system, however, latency impacts the market. As latency grows, traders can expect to get more surplus since inefficient trades are reduced. This is then taken away by latency arbitrageurs (LA) trading on both markets. Execution time increase with regard to increased latency, however LA effectively eliminates this by placing orders as soon as a pair of orders that can be cleared appears in either market. This reduction in execution time however, seems less beneficial when compared to the increase of spread and volatility LA brings to the market.

In a discrete market, We see that by executing orders only at carefully selected intervals, we can effectively eliminate the negative impacts of LA on Surplus. In the last two plots below, a market with latency around 200-300ms gains increase surplus with similar execution time compared to a two market system where LA does not exist.

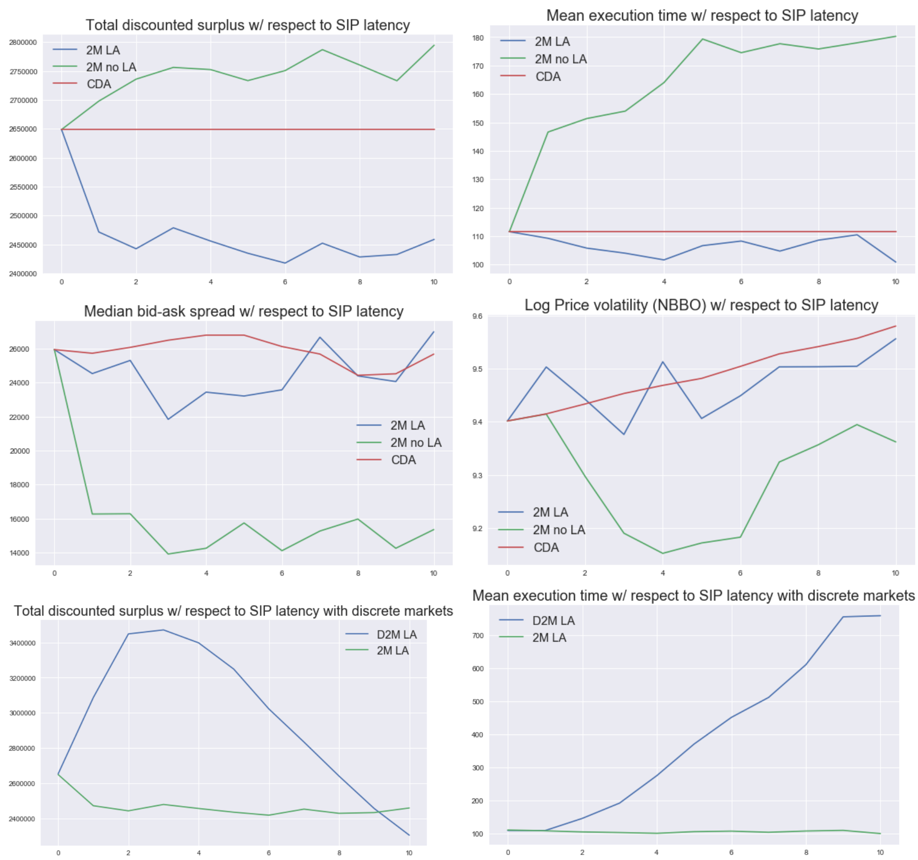
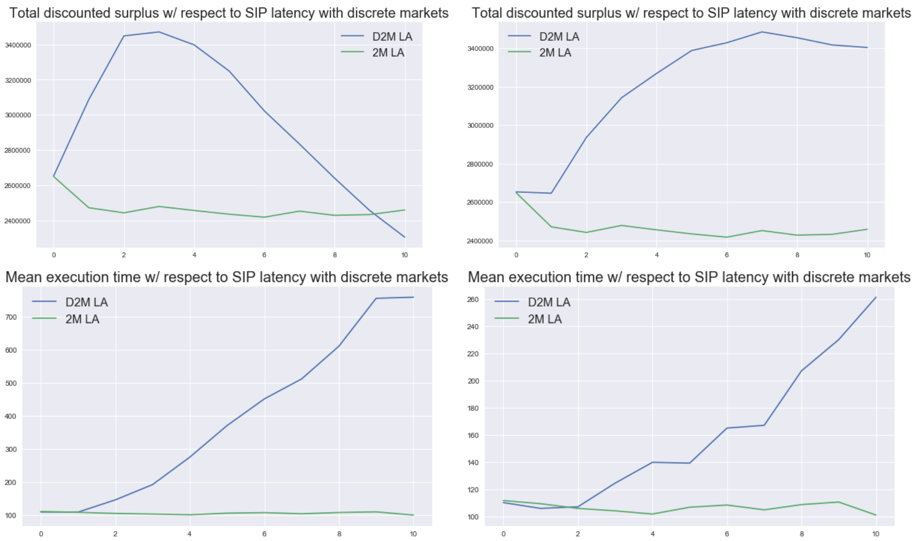


Figure 3: Simulation results. On the horizontal axis are the latencies with units of 100ms.

Due to the lack of specification of the clearing frequency in a central call system, some hypotheses where made. The first one assumes that the clearing frequency is fixed regardless of the latency δ. This one does not agree with the result of the paper. The second one assumes that the clearing frequency is adjusted as the latency δ changes. This assumption matches the result of the paper closely, indicating that the maximum surplus at a given latency is variable given the actual clearing interval.

Seen below is a comparison of surplus with clearing interval $\delta/2+1$ (on the left) and $\delta/5+1$ (on the right). it is clear that the clearing interval can be adjusted to achieve the maximum surplus given a NBBO updating latency. Comparing the two execution plots, (on the bottom) we can see that the execution times are similarly reduced. Where surplus peaked (around 700ms), the execution time is similar.



We further expand the simulation to more than two markets. With increasing markets, we find that the maximum surplus is reduced, where as the effect of latency is also less significant. We theorize this to be related to an increasing number of orders in separated market that could not be cleared. The execution times are not greatly effected with addition of markets.

In the future, more experiments with respect to agent strategies, degree of market fragmentation, and market clearing rules can be expanded into this implementation. A more sophisticatedly defined agent class will be constructed for more clearity, and the further study by Wah et al. that implements an array of strategies based on empirical game-theoretic methods [1] will be a good place to continue the work.

References

[1] Wah, E., and P. Wellman, M. Latency arbitrage in fragmented markets: A strategic agent-based analysis. 1–25.

[2] Wah, E., and Wellman, M. P. Latency arbitrage, market fragmentation, and efficiency: A two-market model. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce* (New York, NY, USA, 2013), EC '13, ACM, pp. 855–872.