Simulated Exchange

SRINIVASA BALAJI R

May 2020

Acknowledgement

I am deeply indebted to **Dr. R. Nadarajan**, Professor and Head, Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, for supporting my aspiration and connecting me with resourceful people.

I owe my deepest sense of gratitude and gratefulness to my mentor **Mr. Kasiviswanathan Muthaiah**, Vice President, JP Morgan Chase Co, New York, for his constant motivation, guidance, continuous support and for providing me with reading materials and ideas inspite of the busy schedule.

Abstract

A simple electronic stock exchange is implemented with an aim to obtain a profound understanding of the behind-the-scenes working of an exchange such as arrival of orders, populating the order book, prioritization of orders and matching of orders. Few statistical properties of limit order books have been observed from the real exchange data and basic models have been implemented during simulation trying to mimic the properties.

Contents

Chapter 1

Introduction

This chapter defines the goals of the project and the ways followed to achieve the goal.

1.1 Motivation and Goal

Since financial data is expensive, yet useful in many areas, simulation of realistic such data is valuable. But even if actual data is bought there are certain aspects that are interesting to investigate which cannot easily be accomplished without a simulator. Particularly market impact caused by a certain trading strategy is hard to estimate with data obtained from a real market.

In order to understand the impact of an order to the exchange, the complete knowledge of behind the scene working of exchange is essential. Chapter 2 focuses on the detailed explanation of chain of events happening after an order is submitted to the exchange. Chapter 3 presents few simple statistical properties of a limit order book observed from real exchange data. Chapter 4 introduces the implementation of the exchange. Chapter 5 defines basic models that has been used for the simulation of the orders. Chapter 6 summarizes the results obtained via the simulation.

1.2 Tools and Technologies Used

- **Database:** KDB+ 4.0 64 bit version[3]
- **Languages:** q, Python3.7
- **Tools:** Jupyter Notebooks

Chapter 2

Market Microstructure

This chapter gives a short background of modern electronic trading and a basic overview of how the order book handles and matches incoming orders. Characteristic features of modern electronic markets which are key factors in the creation of a realistic simulator are then presented.

2.1 Brief outline of Stock Exchanges

All exchanges now use electronic order books and which matches incoming orders almost instantly. The exchanges publish information about the best quotes at all times as well as make all trades information that have taken place available. Also institutes and firms acting on behalf of a customer must make sure the customer gets the best possible order execution, in terms of price, speed and execution probability.

Most exchanges are not open for trading throughout the whole day. Usually the day starts with a fixing opening auction, setting a reference price for the beginning of the day and providing liquidity. Next follows the real trading period, around six hours of continuous matching with a possible lunch break at some exchanges. The trading day ends with a fixing closing auction determining the closing price which is the reference price normally used everywhere. The closing price is also used to calculate the net asset value (NAV) of a company.

During a simple fixing auction only limit orders are accepted and the open or closing price will be the price at which most trades took place. During the auctions, information about the quotes is continuously published with increased frequency towards the last minutes of the auction. The auction time varies but is usually around half an hour up to two hours. The New York Stock Exchange opening auction for instance starts at 7:30 and lasts until 9:30.

2.2 Order Types

There are two main order types available, limit orders and market orders.

A limit order means the issuer wants to buy or sell at a specific price and the order will not be matched until a corresponding order exist on the opposite side offering the same price.

A market order is an order where the price is not specified. Instead the order is matched against the best available price. Market orders cannot be cancelled because they are instantly matched.

We say that market orders consume liquidity while limit orders provide liquidity, since market orders will remove volume at the best quotes.Apart from limit and market orders a requested (cancel order) can be made to cancel a limit order not yet fully matched.

There exist a wide range of other order types and instructions which are not as common, such as stop-loss orders, all-or-nothing orders, volume weighted orders and hidden orders which are not shown in the order book, among others. Most of these order types and instructions can be accomplished by combinations of limit, market and cancel orders and therefore only these three order types are typically of interest in studies.

2.3 Limit Order Books

It is a device that the vast majority of organized electronic markets (all equity, futures and other listed derivatives markets) use to store in their central computer the list of all the interests of market participants. It is essentially a file in a computer that contains all the orders sent to the market, with their characteristics such as the sign of the order (buy or sell), the price, the quantity, a timestamp giving the time the order was recorded by the market, and often, other market-dependent information. In other words, the limit order book contains, at any given point in time, on a given market, the list of all the transactions that one could possibly perform on this market. Its evolution over time describes the way the market moves under the influence of its participants.

2.4 Statistical properties of Limit Order Books

The data obtained from `https://www.nyse.com/market-data/historical` is used for the analysis. The step by step analysis of the limit order book data can be seen from the Python Notebook.

Chapter 3

Implementation

This chapter briefly explains how the system has been implemented.

3.1 Assumptions

As the outline of Stock Exchanges, the order types and limit order books have been briefed in the Chapter 2, we know enough information now to understand the chain of events happening in behind-the-scenes of an exchange. To keep things simple and understandable we focus on **price/time** priority for matching, which is a standard almost used universally. The implementation is based only on stocks (equities).

3.2 System Architecture

Exchange:

Receives orders from players. After validation and authentication of the order, enques the order to the appropriate queue based on the type of order.

Engine:

Keeps running in the background. If there is a chance of matching it matches. When the incoming orders alter the state of the limit order book, checks for possibility of matching and matches.

Player:

The player initially registers to the exchange and places orders (market/limit). At the time of registration each player is assigned with initial amount and initial number of stocks.

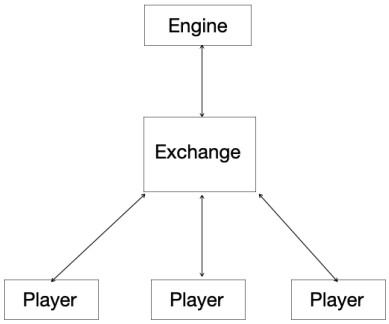

Figure 3.1: Simple Architecture of the exchange

The exchange and engine are modeled as separate entities, as the engine keeps checking for a possible match in the order book. The model is as follows,

- Engine opens a listening port and waits for the exchange to connect.
- Exchange connects to the port of the engine meanwhile opening another listening port for communications with the players.
- Once the connection between the engine and exchange is established all communication between them happen in asynchronous manner.
- The communication between the exchange and player also happens asynchronously.
- Once a player connects to the exchange, player registration event happens and the player is initialised with stocks and money.
- A player submits the order to the exchange.
- The exchange after validation and authentication, add it the order to the appropriate order queue.
- As new orders have arrived, the engine updates it's order book based on price/time priority and checks for matching possibility.
- If an order is matched, a trade takes place and the trade is reported to the exchange via an asynchronous message.
- The exchange then reports to the corresponding players participated in that particular trade via an asynchronous message.

The entire flow happens in an event-driven fashion.

Chapter 4

Simulation

This chapter describes how the simulation of orders are done. The simulation is carried out for one stock.

4.1 Arrival Times

The state of order book evolves under the action of the players operating on the market via the following events,

- arrival of market order,
- arrival of limit order,

These events are described by independent Poisson Process.[1]

- $M^+(t)$ and $M^-(t)$ are counting process of market orders with constant rates λ^+ and λ^-
- $L_i^+(t)$ and $L_i^-(t)$ are counting process of limit orders at level i with constant rates λ_i^+ and λ_i^-

The Poisson Process is simulated using the Inverse Transformation method [2].

4.2 Volume

The volume of all orders is kept constant.

4.3 Price

The price of the orders are modeled as a random walk model with drift.

Chapter 5

Results

The following figure shows the stock price due to simulated on the exchange.

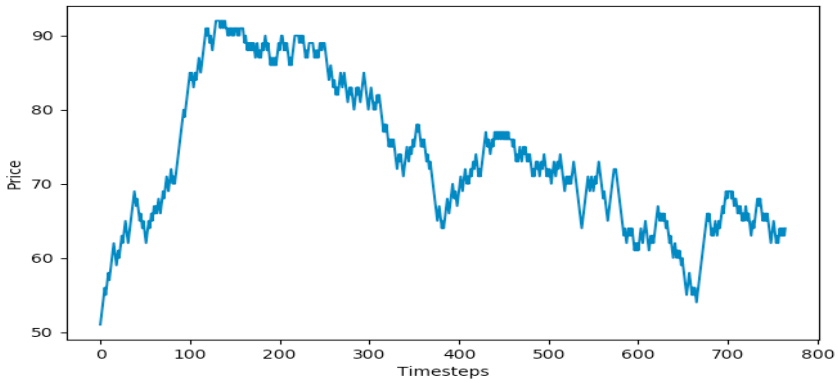

Figure 5.1: Simulated Price

Bibliography

[1] Fracdacric Abergel et al. *Limit Order Books*. Cambridge Books 9781107163980. Cambridge University Press, June 2016. URL: https://ideas.repec.org/b/cup/cbooks/9781107163980.html.

[2] Sheldon M. Ross. *Introduction to Probability Models*. Elsevier, 2010. ISBN: 9780123756862.

[3] Kx Systems. *q for Mortals*. URL: www.code.kx.com.