International Journal of Internet, Broadcasting and Communication Vol.16 No.3 363-369 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.3.363

IJIBC 24-3-40

Development of a Limit Order Book Analysis Tool for Automated Stock Trading Systems

Gyu-Sang Cho

Professor, Dept. of Computer and Software, Dongyang University
cho@dyu.ac.kr

Abstract

In this paper, we develope a LOB(Limit Order Book) analyzing tool for an automated trading system, which features real-time and offline analysis of LOB data in conjunction with execution data. The 10-tier LOB data analyzer developed in this paper, which contains ask/bid prices and the execution data, receivs transaction requests in real-time from the Kiwoom Open API+ server. In the OnReceiveTrData event, the transaction data from the server is received and processed. The real-time data, triggered by the transaction, is received and processed in the OnReceiveRealData event. These two types of data are stored in a database and replayed in the same way as if it were a real-time situation in simulation mode. The LOB data are selectively read and analyzed in a necessary time points. The tool provides various features such as bar chart analysis and pattern analysis of the total shares on the bid side and ask side, which are used to develop a tool to accurately determine the timing of stock trading.

Keywords: *LOB(Limit Order Book), Automated Stock Trading System, Real-time Data Request/Reception, Kiwoom Securities, Open API+*

1. Introduction

Automated trading systems, also referred to as algorithmic trading, automated trading, or system trading, allow traders to establish rules and programs that can be executed via a computer automatically. The development of HTS (Home Trading System) and MTS (Mobile Trading System) has revolutionized stock trading [1-3]. In addition, securities company provides API(Application Program Interface) functions that allow users to create custom trading programs that can execute trades based on predefined rules or even predictions made by artificial intelligence [4, 5].

Many studies related to automatic stock trading have been conducted. Cho's paper describes the development of an automated stock trading system using Kiwoom Securities' Open API+. The system employs a Condition Search script as its trading strategy, which is automated for systematic trading. To circumvent the restrictions on transaction frequency imposed by Kiwoom Open API+, the system employs two key techniques: time intervals between transactions and utilizes an Inter-Process Communication (IPC) technique with multiple

Manuscript Received: July. 20, 2024 / Revised: July. 26, 2024 / Accepted: August. 3, 2024
Corresponding Author: cho@dyu.ac.kr
Tel: +82-54-630-1161, Fax: +82-54-630-1179
Author's affiliation (Professor, Department of Computer & Software, Dongyang University, Korea)

login IDs [6]. Ryoo et. al., propose a deep learning-based prediction model that predicts stock price more accurately by considering both the medium and short term trends of limit order books, and the proposed model considers news headlines during the same period to reflect the qualitative status of the company in the stock price prediction [7]. H. Hultin et al. conducted a study on LOB using recurrent neural network, in which a generative model based on recurrent neural networks for the complete dynamics of a limit order book is developed. Several evaluation metrics, e.g., order type, price level, order size and time delay, the generative model is trained to fit both synthetic and real data from the Nasdaq Stockholm exchange [8].

The depth of the order book provides a detailed view of the market's liquidity by showing the number of price levels and the volume of buy and sell orders at each level. This information is crucial for traders to understand the supply and demand dynamics and to make informed trading decisions [9]. So, in this paper, we develope a LOB analyzing tool for an automated trading system, which features real-time and offline analysis of LOB data in conjunction with execution data. The system we developed is based on the Open API+ provided by Kiwoom. In Section 2, we describe the various features of the API and in Section 3, we describe development of an LOB analysis tool; transaction and real-time data request and reception, and LOB and execution data acquisition, and we conclude in Section 4.

2. Features of Kiwoom Open API+

2.1 Overview of Open API+

The Kiwoom Open API+ system have various features are as follows [5].

1) OCX Control: OCX controls which allow for development in various programming environments such as Visual Basic, Excel, web-based applications, and Microsoft Foundation Class (MFC).
2) Data Request/Reception: Data request/reception are handled through Transaction Request services. Users can search for TR service names and call API functions via OCX to request and receive data.
3) Simulation Environment: Before deploying applications in a live environment, developers can use a simulation environment to test their programs. This helps in debugging and ensuring the reliability of the application by selecting the 'Simulation' option.
4) Supported Operating Systems and Languages: The API is compatible with Windows 32bit and supports various programming languages, including C/C++, C#, Visual Basic, Excel, and Delphi.
5) Community and Resources: There are various community resources and libraries available, which simplify the process of calling API functions and integrating them into applications.
6) Transaction Query Frequency Restrictions: a) 5 queries of a second in successive 1 times(17 seconds blocked) b) 5 queries of a second in successive 5 times(90 seconds blocked) c) 5 queries per a second in successive 10 times(180 seconds blocked) 4) 1,000 queries in an hour(undocumented restriction) [9].

2.2 System files

Kiwoom Open API+ is a service that allows customers to connect their investment strategies to various modules provided by Kiwoom Securities. These modules handle tasks such as market price inquiries, balance inquiries, and order placements. The API supports OCX control, which enhances the extensibility of program development. This means that users can create programs using various languages and platforms, including VB, Excel, web-based applications, MFC, Python PyQt5, and more. By executing the transaction service and

calling the API function through OCX, data requests and receptions are performed. Key Components of Kiwoom Open API+: 1) System DLL library files: OPCommApi.dll: Communication module, KHOpenAPI.ocx: OpenApi Execution module, etc., 2) Program data files: OPTxxxxx.enc and OPWxxxxx.enc 3) System data files: OPComms.ini: Communication attribute-related file, jongacc.dat: Account-related file.

3. Development of an LOB Analysis Tool

3.1 Overview of LOB analysis system

The essential components for configuring a stock automated trading system are account management, stock execution, and transaction management. To achieve LOB analysis, LOB data is required. To receive real-time stock price data based on the Kiwoom Open API+, a transaction request must be performed first. The real-time data manager is triggered by the transaction, and LOB data of the corresponding stock is received in real time. The ultimate purpose of this system is to implement a system that completely automatically trades stocks. In this study, it is limited to developing an analysis tool by analyzing LOB data to determine the timing of stock trading. A LOB simulator is implemented, which requires stock execution data along with LOB data. These two types of data are received in real time and stored in a database. The stored data is operated so that it can be replayed in the same way as it operates in real-time during the offline simulation mode.

3.2 Description on Transaction and Real-time Data Request and Reception

Kiwoom Open API+ offers a variety of functions to handle different events, e.g., OnReceiveTrData, OnReceiveRealData, OnReceiveMsg, OnReceiveChejanData, OnEventConnect, OnReceiveCondition, OnReceiveTrCondition, and OnReceiveConditionVer. These events allow you to interact with and process various types of data and messages from the Kiwoom trading system. Each event in Kiwoom Open API+ has its unique role, but the three key events are more crucial to implement the automated trade system of this study which are OnReceiveTrData for transaction event processing, OnReceiveRealData for real-time event processing, and OnReceiveChejanData for real-time stock execution conclusion information. Figure 1 shows the procedures of transaction and real-time request/reception of Open API+.

The Kiwoom OpenAPI+ ActiveX control provides a variety of functions to perform control and manipulation through properties and 55 methods. 55 methods including CommConnect().

To get bar chart information, for example, such as the current price, daily chart, weekly chart and monthly chart of stocks, it can be performed by requesting a transaction. There are two essential functions required for transactions: SetInputValue() is a function used to set information necessary for a transaction request, and the CommRqData() function is used to request transaction data from the server, which sends transaction code (TR code) that specifies the type of data being requested. And the function is essential for retrieving various types of data, such as stock prices, account information, and order status. In the OnReceiveTrData event in conjunction with GetCommData(), the transaction data from the server is received and processed.

To buy and sell a stock, SendOrder() is called as a transaction command to transmit stock trade orders to the server, which has nine arguments, e.g., buy/sell order type, stock code, order quantity, and order unit price etc. OnReceiveChejanData event is used to handle real-time stock transaction conclusion information. The event is triggered by the SendOrder(), and the information about the status of orders and trades received by calling the GetChejanData(), when there are updates related to orders, such as order confirmations, modifications, cancellations, and executions.

Kiwoom Open API+ offers 27 types of real-time data, including stock market prices, signing information, and stock preferred prices. Additionally, it provides functions to receive real-time information for ETFs, ELWs, Futures, and Options. Once a transaction request is requested, real-time information for the relevant stocks is automatically received. And it can be received or blocked real-time information for specific stocks by using the SetRealReg() and RemoveRealReg() functions manually. The OnReceiveRealData event is used to handle real-time data.

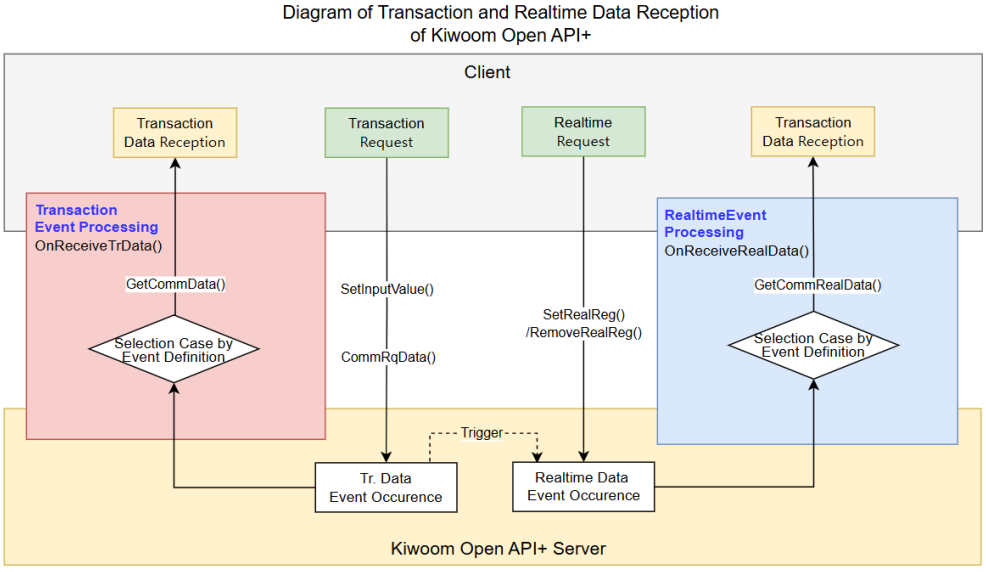


Figure 1. Procedure of transaction and real-time data request/reception of Open API+

3.3 LOB/Execution Data Acquisition

There are two types of data received in real-time from Kiwoom Open API+ server. One is 10-tier LOB data containing Ask/Bid prices and the number of stocks, and the other is the execution data containing the signing time and volume of stocks.

As described in Section 3.2, in the event occurred from the transaction request, i.e., OnReceiveTrData event in conjunction with GetCommData(), the transaction data from the server is received and processed. The real-time data manager is triggered by the transaction, and LOB data of the corresponding stock is received and processed in the OnReceiveRealData event.

On the client side, the two data are mixed together and received. This can be separated according to the type of data, but this process has the problem that the order of the data can be changed when replayed in simulation mode. Therefore, the two data are stored in the order of time generated to synchronize the data.

In Figure 2, Data starting with “[0]” represents the execution data that is formatted as [contract time, price, number of purchases (+)/sell (-) stocks]. Data starting with “[1]” represents LOB data that is formatted as [time, total pending ask orders, total pending bid orders, total execution stocks, 10-tier ask prices, 10-tier bid prices, 10-tier pending ask orders, 10-tier pending bid orders].

```
[0] 090038 9340 -147
[0] 090038 9400 +10
[0] 090038 9410 +91
[0] 090038 9410 +4
[0] 090038 9420 +28
[0] 090038 9340 -10
[0] 090038 9420 +15
[0] 090038 9340 -50
[0] 090038 9370 -100
[1] 090038 5981 20443
287080 -9590 -9570 -9560 -9550 -9520 -9500 -9480 -9460 -9440 -9420 -9340 -9320 -9310 -9300 -9290 -9280 -9270 -9260 -9250 -9240
364 100 200 1641 720 138 390 200 1386 842 2855 1034 2321 1107 4135 37 3683 20 5140 111
[0] 090038 9420 +14
[0] 090038 9420 +38
[0] 090038 9420 +477
[0] 090038 9390 -300
[0] 090038 9370 -57
[0] 090038 9340 -2881
[0] 090038 9320 -1066
[1] 090038 5352 19880
```

Figure 2. LOB/Execution data format stored in a database

3.4 Implementation

The entire system implemented in this study, as shown in Figure 3, includes an LOB analysis system with real-time trading and an offline analyzer. The shape of the window displaying the LOB, as shown in Figure 4(a), is the same in real-time and simulation modes. The difference is that the slider bar at the bottom can be controlled to arbitrarily adjust the moment of the data you want to analyze in simulation mode. Moving the slider bar back and forth will redraw the bar graph in the right window (Figure 4(b)) and synchronize the data location. Figure 4(c) shows a screenshot of the bar min/day/week/month chart with a moving average line. It has a graphic scroll bar in the upper pane, which can move to arbitrary points in the chart.

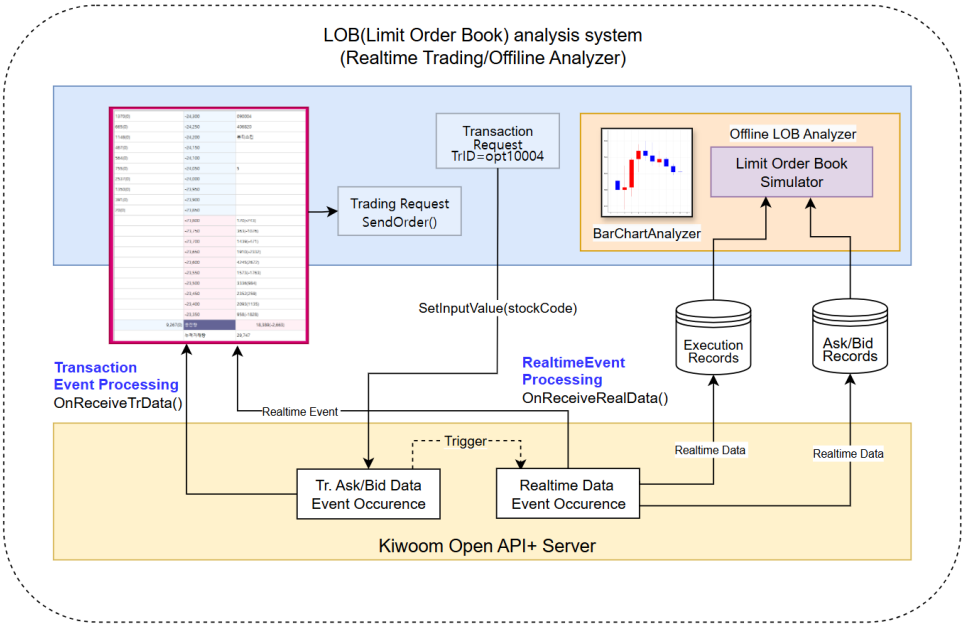
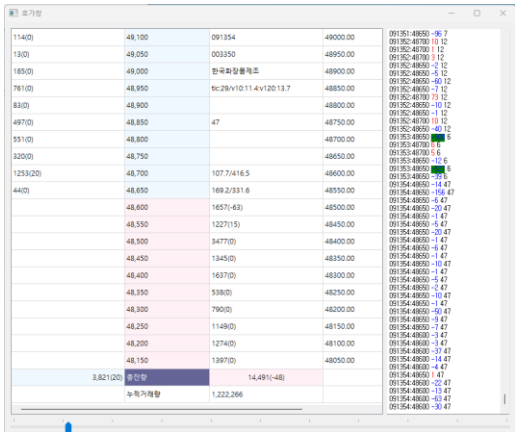


Figure 3. Configuration of LOB analysis system with real-time trading and offline analyzer

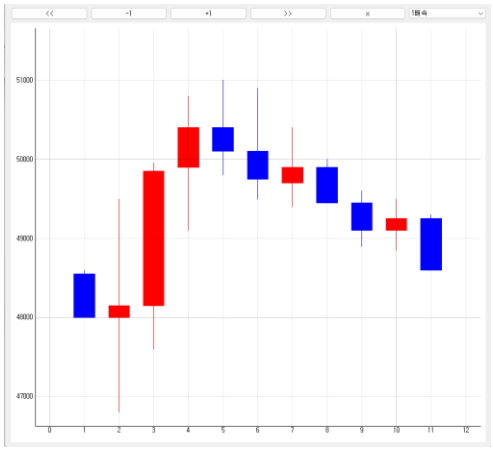
This analyzer supports various features: 1) rates comparison of bid and ask price 2) checks 10-tier stack analysis of bid/ask shares 3) rates change pattern of total shares of bid side and ask side 4) checks spread between bid and ask 5) execution rate per unit time 6) execution size of per unit 7) checks frequency of appearance of large volume execution.

5. Conclusions

In this paper, we developed a LOB analyzing tool for an automated trading system, which features real-time and offline analysis of LOB data in conjunction with execution information. The two types of data received in real-time from the Kiwoom Open API+ server are 10-tier LOB data and execution data with the volume of signed stocks, which are acquired and stored by the transaction request and the real-time data events of OnReceiveTrData and OnReceiveRealData, respectively. The stored data in a database is restored so that it can be replayed in the same way as it operates in real-time during the offline analysis process, and the LOB data are selectively read and analyzed at the necessary time points. The entire analysis process is accompanied by a visual analysis function as the bar chart analyzer works synchronously. Based on the analysis tool, it is expected to develop into a complete automated stock trading system by implementing functions to make accurate judgments about the trading point of a stock.



(a) LOB in a simulation mode



(b) Bar char synchronized with LOB

(c) Screen shot of a bar chart(min/day/week/month) with moving average lines

Figure 4. Screen shots of LOB analyzer and bar chart analyzer

Acknowledgement

"This study was supported by a grant from Dong Yang University in 2023"

References

[1] Kiwoom Open API+ Developer's Guide, https://download.kiwoom.com/web/openapi/kiwoom_openapi_plus_devguide_ver_1.1.pdf

[2] Open API Guide, eBEST INVESTMENT & SECURITIES, https://openapi.ebestsec.co.kr/intro

[3] AiDA Lab, Open API of Securities, https://aidalab.tistory.com/100

[4] B. H. Cho, "Analysis and Design of Stock Item Buy/Sell Recommend System using AI Machine Learning Technology," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 21, No. 4, pp.103-108, 2021.
DOI: https://doi.org/10.7236/JIIBC.2021.21.4.103

[5] E. Yi and W. B. Lee, "A Study on Stock Trading Method based on Volatility Breakout Strategy using a Deep Neural Network," The Journal of the Korea Contents Association, '22 Vol. 22 No. 3, 2022.
DOI: https://doi.org/10.5392/JKCA.2022.22.03.081

[6] G. S. Cho, "Development of a Stock Auto-Trading System using Condition-Search," IJIBC (International Journal of Internet, Broadcasting and Communication), Vol.15 No.3, pp. 215-222 , 2023.
DOI: http://dx.doi.org/10.7236/IJIBC.2023.15.3.215

[7] E. Ryoo, K. Y. Lee, and Y. D. Chung, "Deep Learning-based Stock Price Prediction Using Limit Order Books and News Headlines," The Journal of Society for e-Business Studies, Vol.27, No.1, February 2022, pp.63-79,
DOI: https://doi.org/10.7838/jsebs.2022.27.1.063

[8] H. Hultin, H. Hult, A. Proutiere, S. Samama, and A. Tarighati, "A generative model of a limit order book using recurrent neural networks", Quantitative Finance, Vol. 23, Issue 6, pp.931-958, 2023.
DOI: https://doi.org/10.1080/14697688.2023.2205583

[9] K. Jain, N. Firoozye, J. Kochems, and P. Treleaven, "Limit Order Book Simulations: A Review," arXiv:2402.17359 [q-fin.TR]
DOI: http://dx.doi.org/10.2139/ssrn.4745587

[10] Information guide on restrictions of transactions(rev.), Kiwoom Open API+, https://www.kiwoom.com/h/common/bbs/VBbsNoticeNWOFZView