

전역 변수를 이용한 유동 심볼 자동 주문 시스템의 설계

고 영 훈* · 김 윤 상**

A design of automatic trading system by dynamic symbol using global variables

Ko, Young Hoon · Kim, Yoon Sang

〈Abstract〉

This paper designs the dynamic symbol automatic trading system in Korean option market. This system is based on Multichart program which is convenient and efficient system trading tool.

But the Multichart has an important restriction which has only one constant symbol per chart. This restriction causes very useful strategies impossible.

The proposed design uses global variables, signal chart selection and position order exchange. So an automatic trading system with dynamic symbol works on Multichart program.

To verify the proposed system, BS(Buythensell)-SB(Sellthenbuy) strategies are tested which uses the change of open-interest of stock index futures within a day.

These strategies buy both call and put option in ATM at start candle and liquidate all at 12 o'clock and then sell both call and put option in ATM at 12 o'clock and also liquidate all at 14:40.

From 23 March 2009 to 31 May 2010, 301-trading days, is adopted for experiment. As a result, the average daily profit rate of this simple strategies riches 1.09%. This profit rate is up to eight times of commission price which is 0.15 % per option trade.

If the method which raises the profitable rate of wining trade or lower commission than 0.15% is found, these strategies make fascinated lossless trading system which is based on the proposed dynamic symbol automatic trading system.

Key Words : System Trading, Straddle Sell, Dynamic Symbol Order System, Open Interest, Strategy

I. 서론

한국의 주가지수 옵션시장은 1997년 7월7일에 시작되

었다. 옵션거래의 기초자산은 유가증권시장의 우량종목 200개를 사용한 KOSPI200 지수이며, 이 기초자산의 선물거래는 1996년 5월 3일에 시작되었다. 한국거래소(KRX) 통계에 의하면 주가지수 옵션의 2000년 1월의 일 평균 거래대금은 250억원에 불과했지만, 2010년 5월에는

* 협성대학교 컴퓨터공학과 부교수(제1저자)

** 한국기술교육대학교 컴퓨터공학부 교수(교신저자)

8천억원에 육박하며, 특히, 풋옵션의 일평균 거래대금은 1조원을 넘어섰다.

유가증권시장의 일평균 거래대금은 2000년 1월에 3조원에 불과했지만, 2010년 5월에는 6조원으로 2배 가까이 증가했다. 이에 반해 추가지수 선물의 경우 일평균 계약금은 2000년 1월에는 현물과 비슷한 3조원이었으나 2010년 5월에는 50조원에 육박하여 현금의 증가율보다 훨씬 큰 증가율을 보이고 있다.

이러한 옵션시장의 급속한 성장에 따라 증권사에서 옵션 매매를 HTS에서 가능하게 하였으며, 최근에는 스마트폰을 이용한 옵션 거래도 가능하게 되었다.

하지만 HTS의 주문 방식은 수동 주문 방식으로 다양한 전략을 활용한 자동 주문이 불가능하다.

일반 투자자가 사용할 수 있는 자동 주문 시스템으로 멀티차트 등의 시스템 트레이딩 툴이 있다.

멀티차트는 강력한 자동 주문 시스템으로 다양한 전략을 구현할 수 있도록 설계되었다.

하지만 하나의 차트에 주문 심볼이 하나로 고정되어 있어 프로그램에 의한 심볼 선택은 불가능하다.

본 논문에서는 이러한 제약을 극복할 있는 방법을 제시하고, 이를 적용하여 정상적으로 동작되는지를 확인하였다.

II. 자동 주문 시스템

2.1 시스템 트레이딩

시스템 트레이딩은 컴퓨터를 사용한 기계적인 매매, 각종 지표를 이용한 기술적 분석을 토대로 발생한 시그널에 의한 매매로 정의할 수 있다[1]. 더 나아가 시장에서 매매를 하면서 경험하게 되는 반복적인 실수를 하지 않으려는 일관된 매매원칙이라고 해석할 수 있다. Jake Bernstein은 Strategic Futures Trading에서 다음과 같이 말했다[2]. “많은 탁월한 시스템도 과거 열번 이상의 연

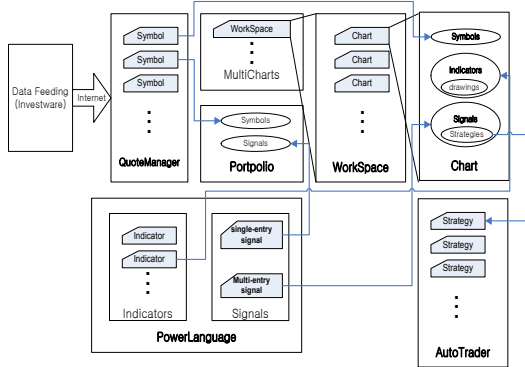
속손실 횟수를 기록한 것을 생각해 보라. 그러면 당신은 시스템 트레이딩 성패의 요소 속에 매우 중요한 끈기가 있음을 알 수 있을 것이다.”

국내에서 사용되는 시스템 트레이딩 프로그램은 Omega Research사의 TS(트레이드시스템) 8.3와 예스탁의 YesTrader(예스트레이더) 3.1 그리고 MultiCharts(멀티차트) 5.x 가 있다. 그리고 데이터를 제공하는 곳은 블라쉬넷(Blashnet)과 인베스트웨어(Investware)가 있다[3]. 특히 MultiCharts는 멀티코어 CPU 및 다수의 모니터를 사용할 수 있는 장점이 있으며, 트레이딩 전문 잡지에서 높은 평가를 받고 있다. 또한 다른 시스템트레이딩 언어 코드와의 호환성을 제공한다[4].

2.2 멀티차트 프로그램

멀티차트는 우선 데이터 피딩을 제공하는 업체에서 데이터를 네트워크 연결을 통하여 수신하여 심볼 별로 저장한다. 이 심볼을 관리하는 프로그램이 쿼트매니저(quote manager)이다. 쿼트매니저의 심볼은 실시간 자동 주문을 위한 멀티차트와 과거 데이터 분석을 위한 포트폴리오(portfolio)로 연결된다. 멀티차트 안에서는 워크스페이스로 구분되는 작업 환경 안에 다수의 차트가 존재한다. 차트 안에는 다수의 심볼(symbol), 시그널(signal), 인디케이터(indicator), 스트로티지(strategy) 등이 존재한다. 하지만 주문과 연결된 심볼은 단 하나이고 고정되어 있다. 시그널과 인디케이터는 파워에디터(power editor)를 통하여 추가 또는 수정된 후에 컴파일 되어 스테디 풀에 저장된다.

시그널은 멀티차트와 포트폴리오에서 모두 사용되며, 멀티차트에서는 자동주문을 위해서 포트폴리오에서는 수익 분석을 위해서 사용된다. 하지만 포트폴리오에서는 다중 진입 시그널을 사용할 수 없으므로 필요한 경우 싱글 진입 시그널로 변환해야 한다[5].



<그림 1> 멀티차트 프로그램의 구성연결도

2.3 시그널

시그널은 멀티차트의 차트 안에서 봉차트를 기본으로 동작하는 주문 발생 프로그램이다. 주문 및 체결 정보는 기본적으로 틱차트를 기반으로 한다. 이를 일정 시간 동안 그룹화하면 봉차트가 된다. 예를 들어 1분봉 차트는 1분 동안의 체결된 틱 데이터를 모은 것이다. 9:00에서부터 9:01까지의 1분 동안의 체결데이터를 모아서 거래량이 결정되며 그 중에서 첫 틱의 체결가가 시가, 마지막 틱의 체결가가 종가가 된다. 따라서 9:01이 되는 시점에 종가가 결정되며, 그 이후에 첫 번째 틱이 9:02봉의 시가가 되는 것이다.

시그널은 전략의 조건을 만족할 경우에 봉의 증가 또는 봉의 시가로 주문을 발생시키는 구조로 되어 있다.

예를 들어 전략의 조건이 만족되었다면 봉의 증가로 매수 주문을 발생시키는 것이다. 따라서 1분 봉이면 9:01, 9:02 에 주문이 발생되며, 30분 봉이면, 9:30, 10:00에 주문이 발생된다.

멀티차트는 매우 유연하고 강력한 자동 주문 시스템이지만 제약 조건이 있다. 바로 주문에 사용되는 심볼(종목)이 고정되어 있다는 것이다.

옵션 거래에 있어서 종목 선정이 가능할 경우 매우 유연한 전략이 가능하다. 예를 들어 전략의 계산 결과에 따

라 증가 옵션을 매수할지, 외가 옵션을 매수할지 선택할 수 있다면 매우 유용한 기능이 될 것이다.

하지만, 현실적으로 멀티차트에서 제공되는 주문 심볼은 고정되어 있다. 이는 프로그램 개발단계에서부터의 제약 조건으로 고정 심볼을 가정하고 그 위에 프로그램을 개발 발전시켜 왔다.

예를 들어 종목을 모아 종목군을 만들고 이 중에서 수익 확률이 높은 종목을 선정하여 주문하는 전략의 경우에는 고정 심볼 주문 시스템으로는 구현이 불가능하다 [6].

옵션의 경우를 들면 퍼쉬폴 전략의 경우 특정 종목을 매도하면서 다른 종목을 매수해야 하는데, 이러한 스프레드 매매 전략에 있어 유동 심볼 주문 시스템은 필수적이다[7].

그렇다면 유동 심볼 주문 시스템은 불가능한가. 꼭 그렇지만은 않다. 몇가지 우회 방법을 사용하여 유동 심볼 주문 시스템의 효과를 낼 수 있다.

이러한 방법이 신뢰성이 있고, 적절히 동작한다면, 매우 효율적인 전략의 개발이 가능해질 것이다.

III. 유동심볼 주문 시스템의 설계

3.1 전역 변수

멀티차트에서는 인디케이터나 시그널과 같은 프로그램(스터디)의 범위는 차트를 넘어갈 수 없다. 즉, 워크스페이스 안에 다수의 차트가 존재하지만 서로 다른 차트의 시그널은 독립적으로 동작한다. 하나의 차트안에 있는 변수를 다른쪽 차트에 넘겨줄 수 없다.

하지만 이러한 제약을 해결해주는 방법이 존재하는데 그것이 전역변수이다.

전역변수는 스테디의 처음에 DLL을 임포트해주고, 변수명이 아닌 변수식별자를 통하여 쓰거나 읽을 수 있다. 사용범위는 신뢰성 있는 동작을 위해서 하나의 워크스페이

스 안에 있는 차트들에 국한된다.

전역변수는 쓰고 읽는 두가지 메소드를 제공한다. 이 함수들은 기본적으로 제공되는 함수가 아니므로 DLL 파일을 정의해주어야 사용할 수 있다.

다음과 같이 스테디의 첫 번째 라인에 사용함수의 프로토타입과 DLL을 정의해준다.

```
DefineDLLFunc: "GlobalVariable.dll", int,
"GV_SetFloat", int, float ;
```

```
DefineDLLFunc: "GlobalVariable.dll", float,
"GV_GetFloat", int ;
```

전역변수를 쓸 때 함수명은 GV_SetFloat이고 읽을 때 함수명은 GV_GetFloat이다. 사용하는 변수의 자료형은 Float형이며, 변수의 구별을 위한 인덱스는 int 형이다.

GV_SetFloat 함수에는 두 개의 인수가 필요하고, GV_GetFloat 함수에는 한 개의 인수가 필요하다. 이 때 읽은 변수값은 리턴값에 담겨진다.

3.2 차트그룹과 차트선택

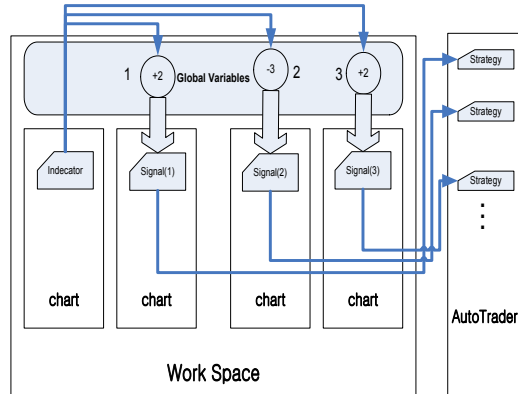
주문할 심볼을 선택할 수는 없지만, 주문할 차트를 전역변수를 사용하여 선택할 수 있다.

따라서 전략을 사용할 인디케이터 차트와 주문에 사용할 시그널 차트그룹이 필요하다. 인디케이터 차트에서 유동 심볼에 대한 주문이 발생하면 전역변수를 사용하여 해당 심볼의 시그널 차트를 통하여 주문을 발생시킨다.

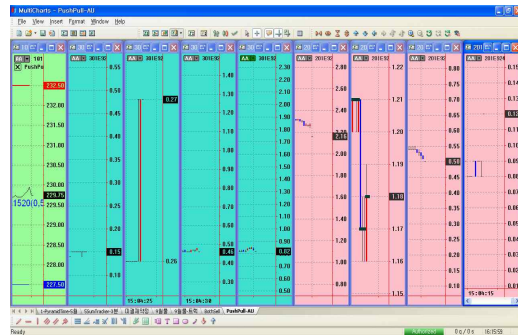
<그림 3>에서는 푸쉬풀 전략을 구현할 것을 볼 수 있다. 실제 전략의 운영을 통해서 1개의 워크스페이스 안에 10개 정도의 시그널 차트의 동작은 무리없이 동작하는 것을 확인하였다.

3.3 주문 변수

인디케이터 차트에서 다중 심볼을 사용하여 매매신호를 발생시키면 주문을 위해서 해당 심볼의 시그널 차트 쪽으로 전역변수에 주문 신호를 쓴다.



<그림 2> 전역변수에 의한 주문 차트 선택



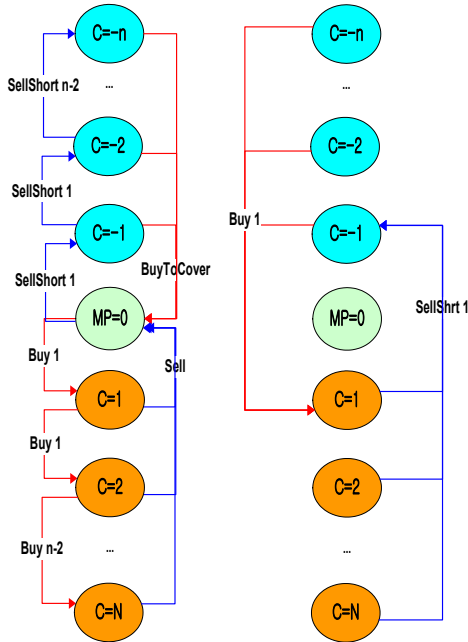
<그림 3> 푸쉬풀 전략의 멀티차트 구현

이 때 필요한 정보는 매수/매도 여부와 몇 매매 계약수이다. 주문 가격은 현재봉의 증가이므로 따로 전달할 필요가 없다.

효율적인 전역변수의 사용을 위하여 정수형을 사용한다. 부호는 매수/매도를 나타내며, 절대값은 계약수를 나타낸다. 예를들어 +1은 1계약 매수이며, -2는 2계약 매도이다.

인디케이터 차트에서 해당 심볼의 시그널 차트로 매수/매도 여부와 계약수를 전달하면 해당 시그널 차트는 전역변수를 읽어서 주문을 발생시킨 후에 전역 변수를 다시 0으로 리셋하여 다음 번 주문에 대비한다.

3.4 포지션 주문 방식



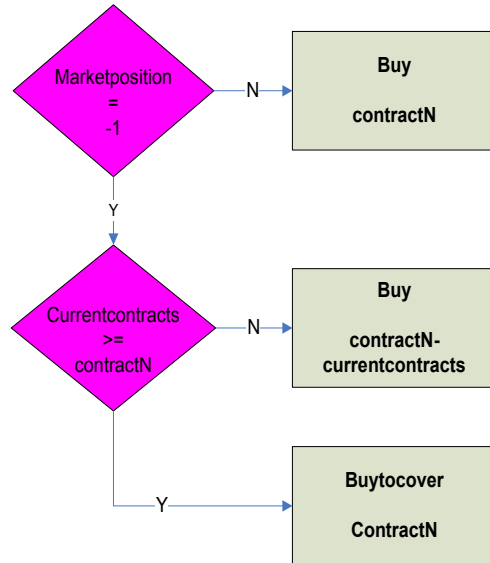
<그림 4> 주문에 따른 마켓포지션의 변화

시그널에는 주문 상태를 나타내는 market position이라는 변수가 있다. 포지션이 없으면 0이고, 매수 주문 체결 후 청산 전이면 1, 매도 주문 체결 후 청산 전이면 -1이다. 매수 주문 명령어는 buy이고 sell은 청산 주문 명령어로 현재 보유한 계약수를 모두 매도청산하고 포지션은 0으로 돌아간다. 매도 주문 명령어는 sell short이고, buytocover는 청산 주문 명령어로 계약수를 명시하지 않을 경우 보유한 매도 포지션 계약수를 모두 매수 청산하고 market position은 0으로 돌아간다. <그림 4>에 이러한 포지션 변화를 나타내었다.

반대 포지션의 신규주문의 경우 계약수는 주문 계약수가 아니라 포지션 계약수이다. 시그널에서 사용하는 매매 명령어의 계약수는 포지션 계약수이다. 매수 n계약 포지션인 상태에서 sellshort m 명령을 수행하면 매수 포지션은 청산되고 매도 포지션으로 m계약이 체결된다.

이는 기존의 계약 주문 방식에서는 n+m계약의 매도 주문과 같다.

따라서 전역변수의 계약 주문 방식을 시그널에서는 포지션 계약 방식으로 변환할 필요가 있다. 전역변수에서는 개별 차트에 연결된 심볼의 포지션을 알 수 없으므로, 시그널의 명령어는 포지션 주문 방식으로 동작하여 예상 외의 결과가 발생할 수 있다. 이를 해결하기 위해서 현재 포지션과 보유 계약수를 이용하여 계약 주문 방식의 주문이 이루어지도록 변환하는 것이다. 이를 <그림 5>에 나타내었다.



<그림 5> 포지션 주문 방식의 변환

3.5 GVTrader

유동 심볼 주문을 위해서 심볼 그룹을 차트그룹으로 만들고, 각각의 차트는 GVTrader라는 시그널을 활성화 시킨다. 각각의 차트는 고유한 GVNum값을 가지고 있어서 인디케이터 차트에서 이 고유값을 이용하여 주문 차트를 선택한다. 다음은 GVTrader 시그널의 코드이다.

<표 1> GVTrader 시그널 소스

```

DefineDLLFunc: "GlobalVariable.dll", int, "GV_SetFloat", int, float ;
DefineDLLFunc: "GlobalVariable.dll", float, "GV_GetFloat", int ;

inputs:
    GVNum(10);
variables:
    ContractN(0), BuySell(0);

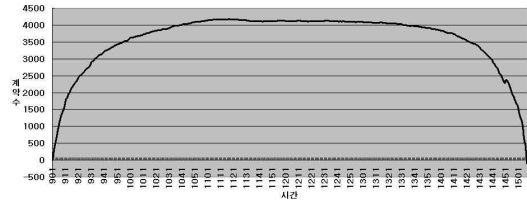
if currentbar=1 then GV_SetFloat(GVNum,0);

ContractN=GV_GetFloat(GVNum);
BuySell=Sign(ContractN);
ContractN=absvalue(ContractN);

if BuySell=1 then begin
    GV_SetFloat(GVNum,0);
    if marketposition=-1 then begin
        if currentcontracts>=ContractN then begin
            buytocover ContractN contract total this bar close
            end else begin
                buy ContractN-currentcontracts contract this bar close;
            end;
        end else begin
            buy ContractN contract this bar close;
        end;
    end;
end;

if condition1 and BuySell=-1 then begin
    GV_SetFloat(GVNum,0);
    if marketposition=1 then begin
        if currentcontracts>=ContractN then begin
            sell ContractN contract total this bar close
            end else begin
                sellshort ContractN-currentcontracts contract this bar close;
            end;
        end else begin
            sellshort ContractN contracts this bar close;
        end;
    end;
end;
end;
    
```

을 확인 할 수 있다. 이는 장중 거래는 장중에 청산하는 것으로 볼 수 있으며, 그 이유는 포지션 유지로 인한 리스크 회피로 판단된다. 거래가 불가능한 장종료 후에 예상치 못한 악재가 발생할 경우에 매우 큰 손실이 발생할 수 있는데, 투자자는 이를 회피해야 한다.



<그림 6> 당일 미결제약정 평균 증감차트

이러한 불연속 거래의 특성상 나타나는 장중 주가지수 선물의 미결제약정의 변동은 필연적으로 옵션 시장의 변동성과 밀접한 관련이 있다. 미결제약정이 증가하는 오전에는 변동성이 증가하여 등가 콜과 풋의 가격의 합인 양합이 올라가는 경향이 있으며, 미결제약정이 감소하는 오후에는 변동성이 감소하여 양합이 내려가는 경향이 있다[8].

주가지수의 방향을 예측하는 것은 매우 어려운 일이므로, 이러한 변동성에 근거한 투자전략을 세울 수 있으며, 이를 두가지 전략으로 구현하였다[9].

오전에 미결제 약정의 증가를 이용한 BS(Buythensell) 전략은 콜옵션과 풋옵션의 두개의 심볼을 선택하여 스트래들 매수 후에 스트래들 매도 청산하는 것으로 첫봉에 매수하여 12시에 매도하는 간단한 전략이다.

그리고 오후에 미결제 약정의 감소를 이용한 SB(Sellthenbuy) 전략은 마찬가지로 콜옵션과 풋옵션의 두개의 심볼을 선택하여 스트래들 매도 후에 스트래들 매수 청산하는 것으로 12시에 매도하여 오후 2시40분에 매수청산한다. 2시 40분을 선택한 이유는 동시호가 2시50분에 시작되므로, 이보다 10분 전에 청산하여 증가 변동성을 피하기 위해서이다.

미결제약정의 특성을 이용한 BS, SB 전략의 수익성이

IV. 실험 및 결과

4.1 Buythensell, Sellthenbuy 전략

선물거래의 미결제약정의 살펴보면 <그림 6>과 같은 특성을 나타낸다. 미결제약정 특성의 일반성을 입증하기 위하여 2008년부터 2년 이상의 기간동안 미결제 약정 데이터를 평균한 것이다.

즉, 장 시작 순간의 미결제 약정을 기준값으로 하여 미결제 약정의 증감을 차트로 나타내었다. 장 시작시 0으로 출발하여 미결제 약정 증감이 증가하다가 장 마감 때가 되면 미결제약정이 증감이 원점으로 돌아가는 특성

있는지를 확인하기 위해서 다음과 같이 실험하였다. 실험 환경은 멀티차트를 사용하였으며, 실데이터와 유동 심볼 자동주문 시스템을 활용하였다. 심볼은 매일 선물 의 시가를 기준으로하는 등가 콜옵션과 풋옵션을 선택하였다. 실험 기간은 2009년 3월23일부터 2010년 5월 31일 까지 301 거래일을 사용하였다.

장이 시작하면 선물 봉차트를 통하여 선물 시가를 알아낸 후에 등가 행사가를 알아낼 수 있다. 이 행사가에 해당하는 시그널 차트에 전역 변수로 +1값을 써주면 1계약 매수 주문한다. 12시가 되면 -2값을 써주어 9시 매수분을 청산하고 추가로 1계약을 더 매도하여 매도 포지션으로 넘어간다. 2시 40분에 +1값을 써주면 매도 포지션을 청산하고 marketposition이 0으로 돌아간다.

4.2 실험 결과

등가옵션의 BS, SB 전략은 선물의 시가에 따라 등가 행사가가 달라지므로 유동심볼 주문 방식으로만 구현이 가능하다.

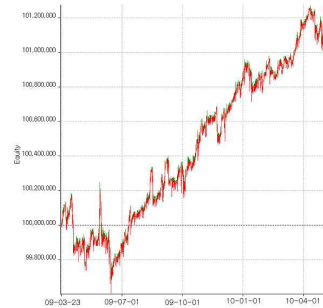
본 논문에서 제안한 방법으로 유동심볼 주문 시스템을 구현하여 성능을 분석하였다.

선물가격의 시가를 기준으로 등가 행사가의 콜옵션과 풋옵션을 동시에 매수하는 스트래들 매수를 9:01에 주문하고 이의 청산을 12:01에 하였을 경우의 순자산 곡선을 보면 <그림 7>과 같다. 수익나는 날과 손실나는 날이 있지만 대체적으로 순자산이 증가하는 모습을 보여준다. <그림 8>은 Sellthenbuy 전략의 순자산 곡선을 보여준다.

BS전략은 602 거래중 273번이 수익 거래이며, SB전략은 602 거래중에 336번이 수익 거래다. BS전략은 수익이 난 날의 평균 수익금이 71,506원이고, 손실이 난 날의 평균 손실금은 57,655이다. SB전략은 수익이 난 날의 평균 수익금이 44,467원이고, 손실이 난 날의 평균 손실금은 55,566원이다. 두 전략을 모두 사용하면 301거래일 동안 하루 평균 7,691원의 수익을 기대할 수 있다. 이를 원금



<그림 7> 등가 BuythenSell 전략의 순자산곡선



<그림 8> 등가 Sellthenbuy 전략의 순자산곡선

대비 수익률로 계산하기위해 해당 기간의 옵션 평균값을 계산하면 다음과 같다. 등가 옵션의 시가 콜평균이 3.41이고 풋평균이 3.63이며, 12시의 등가 옵션 콜평균이 3.39이고 풋 평균이 3.63이다.

<표 2> 전략별 거래수와 수익금

	BS	SB	합계
거래수	602	602	1204
수익거래	273	335	608
수익거래율	45%	55%	50.4%
수익거래시 수익금	71,506	44,467	7,691
손실거래시 손실금	-57,655	-55,566	

따라서 BS 전략의 스트래들 매수 초기값은 7.04이고, SB 전략의 스트래들 매도 초기값은 7.02이다. SB 전략은 매도 우선이므로 매도 증거금이 필요하지만, 이는 매수 청산하면 돌려받으며, 당일 청산이므로 마틴콜의 위험이 없다고 가정하면 무시할 수 있다. 따라서 BS전략의 일평균 수익률은 0.505 % 이며, SB 전략의 일평균 수익률은 0.588%이다. 두 개의 전략을 모두 사용하면 일평균 1.09 %의 엄청난 수익을 기대할 수 있다.

하지만 순수익을 계산하기 위해서는 슬리피지와 수수료료를 계산해야 하는데, 우리선물의 경우 옵션 거래당 0.15%의 수수료를 부과한다. 등가 슬리피지는 약 1틱 정도면 98 % 이상 주문이 체결되므로 무시할 수 있다. 수수료는 거래당 평균 1,055원이고, 두개의 전략에서 하루 8번의 거래가 이루어지므로 8,440원의 수수료가 발생한다. 따라서 BS-SB 전략의 수익은 수수료를 넘지 못해 수익성은 없지만, 수수료가 낮은 증권사나 승률은 높이는 다른 조건을 찾아낸다면 충분히 수익성이 있는 구조이다. 최근에는 증권사의 경쟁이 심화되면서 인터넷 매매 수수료가 점점 낮아지는 추세이므로, 단순한 전략의 운영을 통하여 안정적인 수익률을 실현할 수 있다. 이와같은 전략을 구현하기 위해서는 유동 심볼 자동주문이 필수 요소이며, 무위험 수익의 가능성을 높였다고 볼 수 있다.

실험 결과 제안된 시스템이 안정적으로 동작하는 것을 확인 할 수 있으며, 이는 유동 심볼 자동 주문 시스템이 가능하다는 것을 의미한다. 이는 보다 효율적인 전략의 구현에 매우 중요한 역할을 할 수 있다.

V. 결론

본 논문에서는 옵션거래에서 유동 심볼 자동 주문 시스템을 구현하였다. 옵션 거래에서 다양한 전략을 구현하기 위해서는 유동 심볼 자동 주문 시스템이 필수적이다. 하지만 현재 제공되는 트레이딩 시스템인 멀티차트

에서는 고정 심볼 자동 주문만 제공한다. 이의 한계를 극복하기 위해서 전역변수와 시그널 차트 그룹 선택, 그리고 포지션 주문 변환의 기법을 사용하여 유동 심볼 자동 주문 시스템을 디자인하였다.

제안된 시스템을 평가하기 위해서 선물 미결제약정변화를 이용한 BS-SB 전략을 적용하여 실험하였다.

이 전략은 9시에 스트래들 매수하고 12시에 청산하며, 12시에 스트래들 매도하여 2시 40분에 청산하는 간단한 전략으로 실험 결과 수수료에 육박하는 일평균 1.09%의 수익률을 낼 수 있었다.

따라서 전략의 보완으로 승률을 조금만 높일 수 있다면 무위험 수익이 가능한 매우 매력적인 시스템이 될 수 있다.

참고문헌

- [1] 강수철 · 김희철, 시스템트레이딩 전략 모음집 인베스트라, 서울, 범한서적, 2004, p. 126.
- [2] 김중근, 국제금융시장의 기술적 분석, 법문사, 서울, 1994, p. 155.
- [3] 정영근, 시스템 트레이딩 가이드, 한국경제신문, 서울, 2001, p. 243.
- [4] 장재건, 기술적 분석지표를 이용한 선물투자기법, 진리탐구, 2000, pp. 86-138.
- [5] 고영훈, "MultiCharts의 포트폴리오를 위한 다중 진입 전략의 시그널 변환 시스템 설계," 소프트웨어공학소사이어티 논문지, 제22권, 제1호, 2009, pp. 44-52.
- [6] 고영훈, 김윤상, "멀티차트를 사용한 종목군 계단식 매매 전략에 대한 성능 분석," 디지털산업정보학회 논문지, 제6권, 제2호, 2010, pp. 225-231.
- [7] 고영훈, "옵션시장에서 푸쉬풀 전략의 성능분석," 한국정보처리학회 춘계학술발표대회 논문집, 제 17권, 제 1호, 2010, pp. 1051-1054.

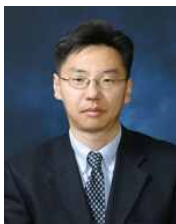
- [8] 고영훈, 김윤상, “시스템 트레이딩에서 진입시점과 델타에 따른 스트래들매도의 성능 분석,” 디지털산업정보학회 논문지, 제6권, 제1호, 2010, pp. 151-157.
- [9] Young Hoon Ko, “Analysis of Straddle trades using open interest of stock index futures in Korea option market,” PPBRI 2010 Summer International Conference at CSUSB, 29 July 2010, p. 18.

■ 저자소개 ■



고 영 훈
Ko, Young Hoon

1999년~현재
협성대학교 컴퓨터공학과 교수
1997년 연세대학교 전자공학과 졸업
(공학박사)
1993년 연세대학교 전자공학과 졸업
(공학석사)
관심분야 : 데이터통신, 데이터베이스
E-mail : tigerko@uhs.ac.kr



김 윤 상
Kim, Yoon Sang

12005년 3월~현재
한국기술교육대학교 컴퓨터공학부
부교수.
1999년 성균관대학교 전기공학과 졸업
(공학박사).
1995년 성균관대학교 전기공학과 졸업
(공학석사)
1993년 성균관대학교 전기공학과 졸업
(공학사)
관심분야 : 로봇 제어, 가상 현실, 인공 지능
E-mail : yoonsang@kut.ac.kr

논문접수일 : 2010년 7월 12일
수 정 일 : 2010년 9월 2일
게재확정일 : 2010년 9월 7일