

エージェントベースの取引所理論  
～ デイトレード論 ～

平成 17 年 1 月 3 1 日

指導教官 有賀裕二 教授

氏 名 工藤寛之

学籍番号 01C21260081

## 目次

### 第1章 背景と目的

### 第2章 デイトレードとは？

- 1 U - M a r t について
- 2 一般にデイトレードとは
- 3 デイトレードのメリットとデメリット

### 第3章 U - M a r t におけるデイトレード

- 1 U - M a r t におけるデイトレード戦略
- 2 デイトレード戦略の解説
- 3 U - M a r t に存在する標準エージェントセット

### 第4章 U - M a r t によるデイトレード戦略実験

- 1 実験
- 2 結論

## 総括

## 参考文献

## 第1章 背景と目的

私はこのゼミに入って初めてU - M a r t というものがあることを知った。一昨年、我々が有賀ゼミに入り最初に取り組んだことがJ A V A というものであった。このJ A V A とはプログラミング言語で我々はU - M a r t でマシンエージェントを作成する際にJ A V A を使っている。一昨年の夏は木更津でU - M a r t 大会が開かれる予定だったのでそこに出席するマシンエージェントを作るためJ A V A を学ぶ必要があったのだ。結局その大会には私と新城君はマシンエージェントではなくヒューマンエージェントでの参加であった。この大会はU - M a r t だけでなく売り手と買い手の取引実験に参加するなど非常に有意義な大会であった。

そして、昨年に入って東工大との合同ゼミでU - M a r t の対抗戦とマシンエージェントの理論を発表することが決まり私と新城君はマシンエージェントを作ることになった。そのときにデイトレードを元にマシンエージェントを作成しようと決めた。J A V A で作ったマシンエージェントは作るのも大変だが動かすことはもっと大変である。しかし、有賀ゼミOBで現在東工大の大学院生である石山さんの力も借りながら合同ゼミの前日に何とか作成することができた。残念ながら私は内定授与式のために合同ゼミには出席することはできなかったのだが新城君の頑張りもありなんとかうまくいったようである。

有賀ゼミでは市場や進化経済学のことも学ぶのだが、1年の大半をU - M a r t に費やす。J A V A やU - M a r t を学んでいる学生はそんなに多くないと思うのだが、これからの時代は金融商品が今以上に身近に取引できるようになると思う、私自身これからは金融業会で働いていくことになるのだが2年間このゼミで学んだことは将来きっと役に立つと思うし本当に貴重な経験をしたと思う。

この論文では、有賀ゼミでの2年間で学んだことを中心にデイトレードについての解説と我々が作ったマシンエージェントをU - M a r t で実験し、どのような結果が出たということ进行分析し論じていこうと思う。

## 第2章 デイトレードとは

### 1 U - M a r t について

デイトレードを説明する前にまずはU - M a r t について触れておこう。

U - M a r t とは先物取引を元にして作られた仮想先物取引システムである。U - M a r t では実験開始時に一定の金額が与えられる。エージェント（プレーヤー）はその金額を元にいくらで（注 成行注文と注 指値注文に分けられる）何個を売りもしくは買い注文をだし利益をだしていくというものである。さらにエージェントはヒューマンエージェントとマシンエージェントに分かれる。

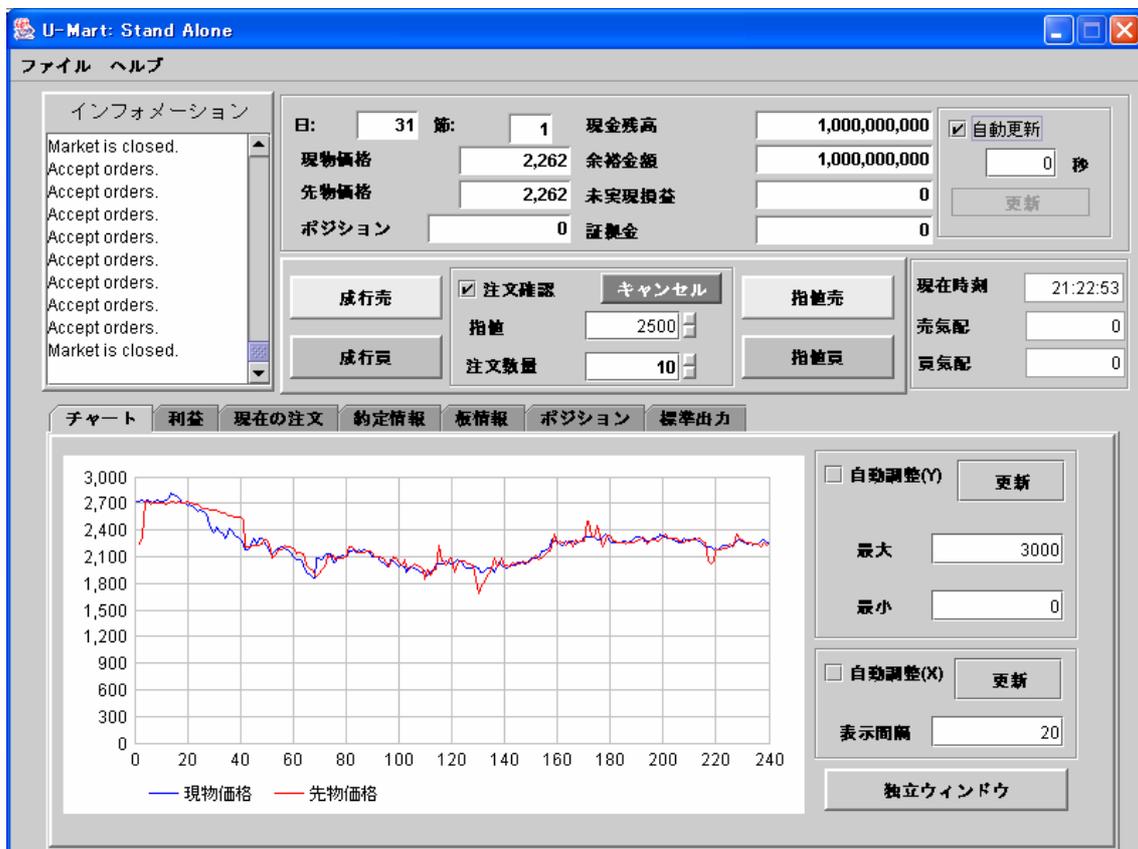
ヒューマンエージェントとは先物価格や現物価格、ポジションなどを元に人間が自ら意思決定し取引を行うということでマシンエージェントはあらかじめたてた戦略に基づき、売買判断（売り、買い、見送り）、価格決定（成行、指値、指値注文価格）、数量決定を自動で行うというものである。また、マシンエージェントは先物価格、現物価格の履歴を情報として受け取ることができ、さらに注 ポジション、現在の資産などを用いて、価格予測、ポジション管理、利益管理などを行う。

マシンエージェンとの例を挙げると我々の先輩である小林さんが作ったサイコロジカルライン、我々の同級生である阿部君が作った一目均衡表などがある。これらはテクニカル分析と呼ばれるものでありほかに移動平均線やゴールドンクロス、デッドクロスというものがある。そして我々が作ったマシンエージェントはデイトレードを用いたものである。

市場 = 安定である。これは市場に安定がないと生活をかけられないので人は市場には参加しないからである。しかし、市場というものは時に価格が高騰や急落したりと急変動をするものでもある。過去に市場の急変動を解析した人はいないといわれている。市場の急変動を回避するというを目的にU - M a r t のようなシュミレーターを使うのである。また市場を体験できるということもU - M a r t をやることの大きなメリットのひとつと言えるのではないだろうか。

- 注 成行 市場で成立した価格で注文すること
- 指値 価格を指定して注文すること
- ポジション 買いと売りの差。買いが多いときは売りポジション（ショートポジション）売りが多いときは買いポジション（ロングポジション）と呼ぶ

参考までに下図がU - M a r t の取引画面



## 2 一般にデイトレードとは

一般にその日のうちに注文から決済までを完結させる売買と言われている。株・債券・為替等を1日の内に売買を終了させ、現金化した上で取引を終えることであり、翌日には持ち越さない。

いまや、オンライン・トレードなどがかなり一般的になり、本なども多数出版されるようになった。インターネットの発達で、個人投資家がさまざまな情報を容易に入手できるようになり、マウスをクリックするだけで世界中の市場に参加できるようになったことも大きな要因として挙げられる。また、証券の手数料自由化、商品取引での電子取引の手数料自由化されたことにより潜在的な利益の可能性を拡大し、短期的な取引の確実性を増加させた。手数料が低下するまでは機関投資家と富裕層の個人のみが手数料の恩恵を受けていたが、今日では、少しのお金しか持っていない人にも株式市場参加への道筋が開けている。

## 3 デイトレードのメリットとデメリット

### デイトレードのメリット

- ・短期売買を行うことでポジションを持ち越さない為、暴落に強い面がある。（取引終了後のニュースによる影響を受けない）
- ・ポジションを持つ時間が少ないので、中長期に比べリスクを抑えることができる。
- ・マーケットが混乱しているとき、短期のトレーダーは日ごとそれを受け入れ、その時々にあった行動をとることができる。
- ・20パーセント下落した銘柄を5パーセントの損失で防げたのなら敗北ではなく勝利である。（すなわち最良の防御は最良の攻撃である。）

U - Martでは全ての取引が終了すると持っているポジションを現物価格で消化しなければならないためにポジションを多くもっていることは利益をあげるためにはリスクを負うのである。

### デイトレードのデメリット

- ・完全なる投機であること。
- ・小さい値幅で、多額の利益を得ようとするので、大抵の場合は信用取引をする必要がある。
- ・大量に売買するので、出来高の多い銘柄を狙う必要がある。

- ・機械的に繰り返し売買するので、取引手数料が高つく。(もちろん、取引回数に関係なく定額の手数料をとる証券会社もある。)
- ・売買タイミングが非常に重要となること。(つまり、取引中は画面の前に常に居なければならない。)

マーケットで勝ち続けるための発想術、デイトレードの著者はポジションのとり方がトレーディングの8割を占めると断言しているほどポジションは大切なものである。短期のトレーディングはポジションのとり方が基本となる。適切な水準とタイミングでポジションをとれば、数分以内に(時には数時間以内に)利益が出る可能性が高い。間違ったタイミングで、そして高すぎる水準でポジションをとれば、瞬時のうちに含み損を抱えることになる。すべての取引の結果は1つの要因によって決定される。つまり、いつ、どの水準でポジションをとるかである。

デイトレードでもっとも大切なことの1つが損きりである。人はトレーディングのすべてにおいて勝ち続けるということはない。いくら世界的な名トレーダーでも、すごく運のいい人でも負けは避けられないのである。そこでその負けを最小にするために行うことが損きりである。

いくら株価が大暴落したとしても損きり水準を相対的に近くに設定することによって、大負けは避けられるのである。

注意深く選択した銘柄を損きったとき、それは負けではなく、勝ちなのである。

### 第3章 U-Martにおけるデイトレード

#### 1 U-Martにおけるデイトレード戦略

私と新城君は10月2日の東工大との合同ゼミに参加する際に有賀ゼミで2年間学んだことを生かしてマシンエージェントを作成した。そのマシンエージェントはデイトレードを戦略として作ったものであり下記のものがその戦略のソースコードである。

TestStrategy.java

```
package strategy;

import java.util.*;
/**
 *Shinjo Strategy
 *
 * author Takuya Shinjo & Hiroyuki Kudo(Chuo University)
 */

public class TestStrategy extends Strategy {
    //初期値
    private Random random;
    private final int widthOfPrice=60;
    private final int maxQuant=50;
    private final int minQuant=10;
    private final int nominalPrice=3000;
    private boolean isOrder = false;
    //ランダムオーダー
    public TestStrategy(int seed)
        random = new Random(seed);
    }

    public Order getOrder(int[] spotPrice, int[] futurePrice, int pos,
        long money, int restDay)
```

```

    Order order = new Order();
//前回価格
    int prevPrice = getLatestPrice( futurePrice );
    if ( prevPrice == -1 )
        prevPrice = getLatestPrice( spotPrice );
    if ( prevPrice == -1 )
        prevPrice = nominalPrice;
    int latestSpotPrice = spotPrices[119];
    int latestFuturPrice = futurePrices[59];
//キャンセル
    if ((latestSpotPrice <= 0) || (latestFuturPrice <0)) {
        order.buysell = Order.NONE;
        return order;
//売買判断
    if(latestFuturePrice > latestSpotPrice) {
        order.buysell = order,SELL;
        System.out.println(1);
    } else if(latestFuturePrice < latestSpotPrice)
        order.buysell = order,BUY;
        System.out.println(2);
    } else {
        order.buysell = order.NONE;
        System.out.println(3);
    }
//注文価格
    while (true) {
        order.price = prevPrice + (int) (widthOfPrice *
random.nextGaussian() );
        if ( order.price > 0 ) break;
    }
    order.quant = minQuant + random.nextInt (maxQuant - minQuant + 1);

    if (isOrder)
        order.buysell = Order.NONE;
    } else {
        if (money >= 1200000000) {

```

```
    if (pos > 0) {
        order.buysell = Order.SELL;
        order.price = 0;
        order.quant = pos;
    } else {
        order.buysell = Order.BUY;
        order.price = integer.MAX_VALUE;
        order.quant = Math.abs(pos);
    }
    isOrder = true;
} else if(money <=800000000 {
    order.buysell = Order.SELL;
    order.price = 0;
    order.quant = 1000;
}
}
return order;
}
}
```

## 2 戦略の解説

次にこのテストストラテジーを解説していこう。

//初期値

```
private Random random;
private final int widthOfPrice=60;
private final int maxQuant=50;
private final int minQuant=10;
private final int nominalPrice=3000;
private boolean isOrder = fales;
```

ここでは初期値を設定している。

- ・ 価格幅 60
- ・ 最大注文数量 50
- ・ 最小注文数量 10
- ・ 唱え値 3000

//ランダムオーダー

```
public TestStrategy(int seed)
    random = new Random(seed);
}
```

```
public Order getOrder(int[] spotPrice, int[] futurePrice, int pos,
    long money, int restDay)
```

```
    Order order = new Order();
```

参照データ

- ・ spotPrices[] 現物価格（過去 120 期分）
- ・ futurePrices 先物価格（過去 60 期分）
- ・ pos ポジション
- ・ money 保有現金
- ・ restDay 最終日までの期間

以上のデータを参照しオーダーをする。

//前回価格

```
int prevPrice = getLatestPrice( futurePrice );
if ( prevPrice == -1 )
    prevPrice = getLatestPrice( spotPrice );
if ( prevPrice == -1 )
    prevPrice = nominalPrice;
int latestSpotPrice = spotPrice[119];
int latestFuturPrice = futurePrice[59];
```

ここでは前回価格の所得を意味している。

spotPrice ( 現物価格 ) は近いほうから spotPrices[119]から spotPrices[0]  
futurPrice ( 先物価格 ) は近いほうから futurPrices[59]から futurPrices[0]  
データがない箇所はそれぞれ-1 と表す。

//キャンセル

```
if ((latestSpotPrice <= 0) || (latestFuturPrice <0)) {
    order.buysell = Order.NONE;
    return order;
}
```

現物価格の前回価格が0 か-1 になったら注文は行わない。

先物価格が-1 になったときも注文は行わないということを表している。

//売買判断

```
if(latestFuturePrice > latestSpotPrice) {
    order.buysell = order,SELL;
    System.out.println(1);
} else if(latestFuturePrice < latestSpotPrice)
    order.buysell = order,BUY;
    System.out.println(2);
} else {
    order.buysell = order.NONE;
    System.out.println(3);
}
```

このテストストラテジーの売買判断は以下の通りで、非常にシンプルなものである。

- ・先物価格が現物価格を上回ったら売りサイン
- ・先物価格が現物価格を下回ったら買いサイン

```
//注文価格
while (true) {
    order.price = prevPrice + (int) (widthOfPrice *
random.nextGaussian() );
    if ( order.price > 0 ) break;
}
```

注文価格を決定する数式である。注文価格がマイナスになってしまうときがあるのでプラスになるまでループさせている。

```
order.quant = minQuant + random.nextInt (maxQuant - minQuant + 1);
```

注文数量を決定数式である。

```
if (isOrder)
    order.buysell = Order.NONE;
```

注文価格がマイナスになると注文しない

```
} else {
    if (money >= 1200000000) {
        if (pos > 0) {
            order.buysell = Order.SELL;
            order.price = 0;
            order.quant = pos;
        }
    }
}
```

現金が 12 億円（黒字が 2 億円）以上になり買いポジションの場合、成行でポジション分を売る。

```
} else {
```

```
        order.buysell = Order.BUY;
        order.price = integer.MAX_VALUE;
        order.quant = Math.abs(pos);
    }
```

上記と同じ条件で売りポジションの場合絶対値をとってどんな価格でもポジション分を買うという判断を下す。

```
        isOrder = true;
    } else if(money <=800000000 {
        order.buysell = Order.SELL;
        order.price = 0;
        order.quant = 1000;
    }
```

もし、現金が 8 億円（赤字が 2 億円）以下になった場合は成行で 1000 個売る。

これは 1 発逆転で黒字を狙うためである。

実際この判断は製作者がギャンブ好きと負けず嫌いな性格から用いられているわけで、本当は逆転を狙わずに損きりをして損失を最小限に食い止めるべきである。

つまりこのように逆転を狙って思い切った戦略を狙えるということも U - M a r t ならではと思う。

### 3 U - M a r t に存在する標準エージェントセット

次にU - M a r t で対戦するマシンエージェントについて説明しよう。  
これらは標準エージェントセットと呼ばれるものである。

U - M a r t には以下の標準エージェントセットが存在する。

- ・トレンド戦略
- ・アンチトレンド戦略
- ・ランダム戦略
- ・RSI
- ・移動平均
- ・裁定取引

そしてこれらのほかに阿部君などが作ったエージェントも入っている。

**トレンド** 相場が安くなると買い、高くなると売るという一般的な戦略  
ヒューマンエージェントでこの戦略を用いる人は多いのではないだろうか。  
順張りとも言われる。

**アンチトレンド**トレンドと逆に相場が安くなると売り高くなると買う戦略  
逆張りとも言われる。

**ランダム**ランダムに取引をするエージェント

**RSI** オシレーター系の代表的なテクニカル指標。  
相場の上げ過ぎ、下げ過ぎを判断する他、その角度と値動きの方向とスピード  
は、値の値動きの強さを表す。  
数値は0%から100%の間を上下に動き、100%に近いほど買われ過ぎ、0%に  
近いほど売られ過ぎを示している。

**移動平均** 移動平均には大きく分けて3つの種類がある。単純移動平均、加重移動平均、  
指数平滑平均であり、一般的に用いられているのは単純移動平均。

- ・単純移動平均 いわゆる算術平均のことで採用した期間の終値の平均価格。

- ・加重移動平均 直近 n 日間における加重平均。つまり、直近の値段にウエイトを置いた移動平均。
- ・指数円滑平均 累積加重平均のことで、一定期間内の平均値ではなく、保有データの数値をすべて計算期間としている加重平均。

裁定取引 価格変動において、同一の性格を持つ 2 つの商品の間で、割安な方を買、割高な方を売ることにより、理論上リスクなしに収益を確定させる取引のことをいう。

マーケットの価格が理論価格に近づき、乖離がなくなることで、割高・割安な状態が解消された場合に、反対売買を行なうことによって収益化する。

## 第4章 U - M a r t におけるデイトレード戦略実験

### 1 実験

私はU - M a r t におけるデイトレード戦略である実験をした。  
第2章でも挙げた通りデイトレードのデメリットに手数料が高つくというものがある。そこでU - M a r t で手数料を掛けた場合と掛けなかった場合、どのようなことが結果に反映されるかというものである。

まず、U - M a r t における手数料について説明しよう。  
一般の取引市場では板寄せするたびに手数料がかかっていると思うのだがU - M a r t での手数料の定義は以下の通りになっている。

手数料 前日までの買いポジションと売りポジションの差の絶対値が決済を要したポジションである、対象が取引所でなければこれに対して手数料を徴収する。

このようにU - M a r t では翌日に持ち越すポジションに手数料がかかるという仕組みになっているのである。

#### 基本設定

|              |       |
|--------------|-------|
| ・初期金額        | 10億円  |
| ・証拠金         | 30万円  |
| ・取引単位        | 1000  |
| ・取引日数        | 30日   |
| ・一日の板寄せ回数    | 8回    |
| ・ヒューマンエージェント | 参加しない |

U - M a r t のすべてのマシンエージェントに1万円の手数料をかけてみた。  
その結果手数料の合計は次の通りになった。

取引手数料はさまざまな証券会社の手数料を調べた結果約定代金が500から1000万円のとときは約1万円だったので1万円とした。

| エージェント名                 | 手数料総額     | 手数料 1 万<br>円時の損益 | 手数料 0 円<br>時の損益 |
|-------------------------|-----------|------------------|-----------------|
| MyTrendStrategy         | 4760000   | -41188000        | -39420000       |
| AntiTrendStrategy1      | 45860000  | -97326000        | -100046000      |
| AntiTrendStrategy2      | 42200000  | -92611000        | -96511000       |
| RandomStrategy          | 44380000  | -83339000        | -89176000       |
| SRandomStrategy1        | 45220000  | 61601000         | 55704000        |
| SRandomStrategy2        | 46280000  | 103110000        | 98619000        |
| SRandomStrategy3        | 50560000  | 104445000        | 103966000       |
| RsiStrategy             | 15400000  | -77305000        | -79443000       |
| SRsiStrategy1           | 13840000  | 70172000         | 70172000        |
| SRsiStrategy2           | 12640000  | 51640000         | 51055000        |
| SRsiStrategy3           | 15320000  | 48039000         | 47773000        |
| MovingAverageStrategy   | 4060000   | -3022000         | -6370000        |
| SMovingAverageStrategy1 | 2500000   | -20701000        | -19909000       |
| SMovingAverageStrategy2 | 2900000   | 4045000          | 4507000         |
| SMovingAverageStrategy3 | 3340000   | 3224000          | 3576000         |
| SFSspreadStrategy1      | 28060000  | 168434000        | 168154000       |
| SFSspreadStrategy2      | 27480000  | 166942000        | 164624000       |
| DayTradeStrategy1       | 48700000  | -196082000       | -200343000      |
| DayTradeStrategy2       | 47540000  | -164031000       | -171607000      |
| T01_Ayaki               | 60240000  | 71250000         | 50503000        |
| T01_Yamanaka            | 8000000   | 180040000        | 180040000       |
| T02_Abe                 | 27420000  | -197595000       | -204595000      |
| T03_Irie                | 17000000  | 141005000        | 141370000       |
| T04_Kitakubo0           | 74920000  | -118242000       | -91960000       |
| T04_Kitakubo1           | 165680000 | 235814000        | 242036000       |
| T04_Kitakubo2           | 75340000  | -111422000       | -86058000       |
| T05_Aiba                | 10140000  | -100145000       | -100145000      |
| Shinjo_                 | 28080000  | -106752000       | -96516000       |

手数料を約定回数から求めると次のようになる。

| エージェント名                 | 約定回数 | 約定回数 × 手数料 |
|-------------------------|------|------------|
| MyTrendStrategy         | 147  | 1470000    |
| AntiTrendStrategy1      | 152  | 1520000    |
| AntiTrendStrategy2      | 135  | 1350000    |
| RandomStrategy          | 158  | 1580000    |
| SRandomStrategy1        | 158  | 1580000    |
| SRandomStrategy2        | 151  | 1510000    |
| SRandomStrategy3        | 172  | 1720000    |
| RsiStrategy             | 56   | 560000     |
| SRsiStrategy1           | 42   | 420000     |
| SRsiStrategy2           | 39   | 390000     |
| SRsiStrategy3           | 41   | 410000     |
| MovingAverageStrategy   | 15   | 150000     |
| SMovingAverageStrategy1 | 9    | 90000      |
| SMovingAverageStrategy2 | 9    | 90000      |
| SMovingAverageStrategy3 | 10   | 100000     |
| SFSpreadStrategy1       | 79   | 790000     |
| SFSpreadStrategy2       | 86   | 860000     |
| DayTradeStrategy1       | 167  | 1670000    |
| DayTradeStrategy2       | 176  | 1760000    |
| T01_Ayaki               | 122  | 1220000    |
| T01_Yamanaka            | 7    | 70000      |
| T02_Abe                 | 79   | 790000     |
| T03_Irie                | 116  | 1160000    |
| T04_Kitakubo0           | 137  | 1370000    |
| T04_Kitakubo1           | 213  | 2130000    |
| T04_Kitakubo2           | 135  | 1350000    |
| T05_Aiba                | 28   | 280000     |
| Shinjo_                 | 82   | 820000     |

## 2 結論

手数料の総額は上記のようになる、U - M a r t では板寄せ回数に比例して手数料がかかるわけではないので手数料が多いからといって板寄せが頻繁に行われているわけではない。この手数料の金額で言えることはどれだけポジション管理が頻繁に行われているかである。U - M a r t ではポジションの絶対値に対して手数料がかかるため翌日までなるべく売りポジションと買いポジションの差が0に近ければ手数料は安くなる。つまり上記の図で手数料が低いマシンエージェントはポジション管理がうまいエージェントだといえるのではないだろうか。

さらに手数料が0円のとくと手数料が1万円のとくでは損益が違うマシンエージェントと変わらないマシンエージェントがいるということがわかった。

これはもともと乱数というものがあり乱数が一緒だと実験の結果は同じになる、しかし乱数を変えることにより時系列等が変わり、結果が違ってくる。また、今回のように乱数を変えなくても手数料などを変えることによって結果が変わってくるのであるが、エージェントによって手数料をかけてもかけなくても差がでないエージェントもあり、手数料がかかることにより戦略に何らかの影響のする戦略と影響が出ない戦略があるということではないだろうか。

最後にU - M a r t でわれわれの作ったエージェントの手数は28080000だが、手数料を約定回数から求めると、手数料は820000となりかなりの差がでていることがわかる。

このことはU - M a r t では手数料が損益に与える影響が大きいといえるのではないだろうか。

## 総括

U - M a r t という仮想取引市場においてマシンエージェントを作成し実験するということは大変難しいことで、この卒論を書くにあたって協力してくれた有賀ゼミOBの石山さんにまずは感謝したい。

10月の東工大との合同ゼミのように今回のデイトレード戦略を作成し、今回卒論を書くことによってU - M a r t とデイトレードのことを1からまた学びなおすことができた。この論文ではU - M a r t とわれわれの作ったエージェント、さらにそのエージェントの元になったデイトレードについて自分なりにわかりやすく説明してきたつもりである。ゼミの後輩やこれからU - M a r t を始めようと思う人に参考になるかわからないが参考にしていただけるとありがたい。

最後に2年間ご指導をしてくださった有賀先生に感謝の意を示しこの論文を締めたいと思う。

## 参考文献

デイトレード マーケットで勝ち続けるための発想術

オリバー・ベレス / 著 グレグ・カブラ / 著  
林康史 / 監訳 藤野隆太 / 訳 日経B P社

U - M a r t 概論

小山祐介 / 著

エージェントベースの取引所理論 ~ 理論編 ~

Kim Kyoung You / 著

<http://www.nomura.co.jp/terms/index.html>