



# 程序化交易高级教程

2014-12

上海文华财经资讯股份有限公司

Webstock Information Systems Co., Ltd

注：此教程适用于赢智 Wh8。

## 目录

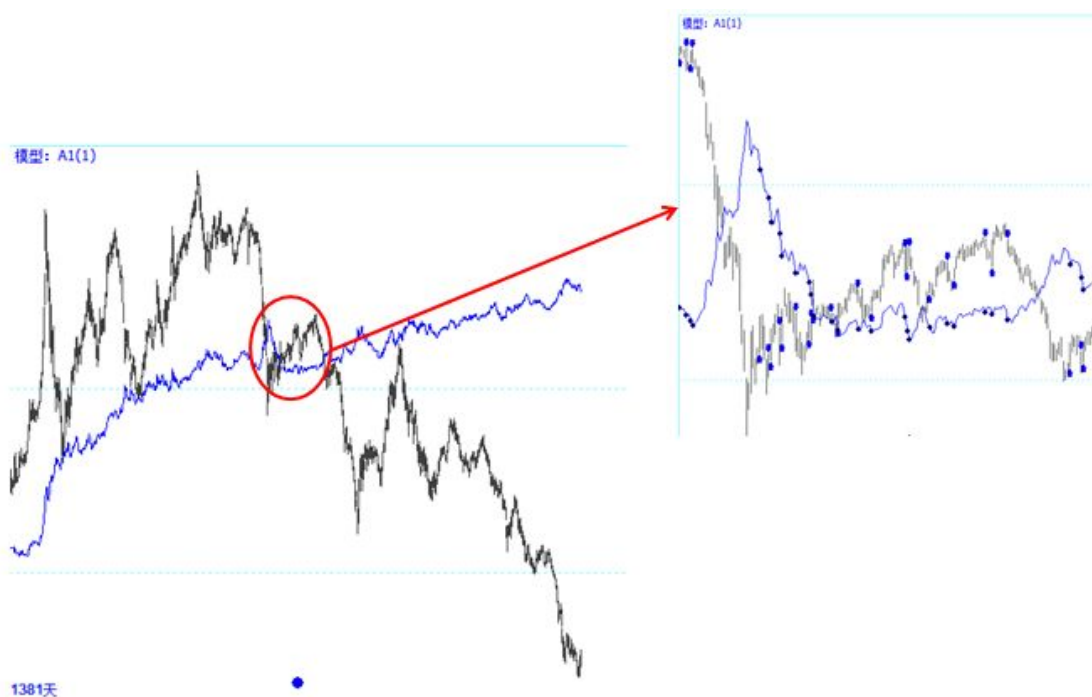
第一章 如何优化你的交易策略.....	1
1.1 PANZHENG 函数，减少盘整行情中的交易次数.....	1
1.2 CHECKSIG 函数，实现更具有优势进场价格.....	7
1.3 MULTSIG 函数，在一根 k 线上灵活进出.....	12
1.4 TRADE_OTHER 函数，在指数交易中的应用.....	17
1.5 拓展思路—结合盘口数据研发策略.....	27
第二章 多模型组合回测.....	35
2.1 回测一篮子合约.....	35
2.2 多模型组合回测.....	38
2.3 段落式交易回测.....	42
第三章 编写资金管理模型.....	45
3.1 加码模型.....	45
3.2 回撤控制模型.....	47
3.3 资金曲线跟随模型.....	48
第四章 盘口模型的基本结构与应用.....	49
4.1 盘口模型的分类.....	49
4.2 盘口模型使用的函数类型和运行机制.....	50
4.3 盘口模型的语法及编写规则.....	50
4.4 盘口模型的加载流程.....	54
第五章 盘口高频模型的编写.....	56
5.1 什么是高频交易.....	56
5.2 盘口高频模型的编写—追涨高频策略.....	57
5.3 盘口高频模型的编写—辅助判断趋势策略.....	61
5.4 盘口高频模型的编写—基差策略.....	62
第六章 盘口模型控制滑点.....	63
6.1 了解滑点的产生.....	63
6.2 盘口模型控制滑点的原理.....	64
6.3 盘口模型控制滑点策略编写.....	64
第七章 后台程序化.....	71
7.1 运行模组.....	71
7.2 盘口模型运行池.....	80
第八章 远程监控.....	86
8.1 设置运行模式.....	86
8.2 日志邮件.....	88

## 第一章 如何优化你的交易策略

### 1.1 PANZHENG 函数，减少盘整行情中的交易次数

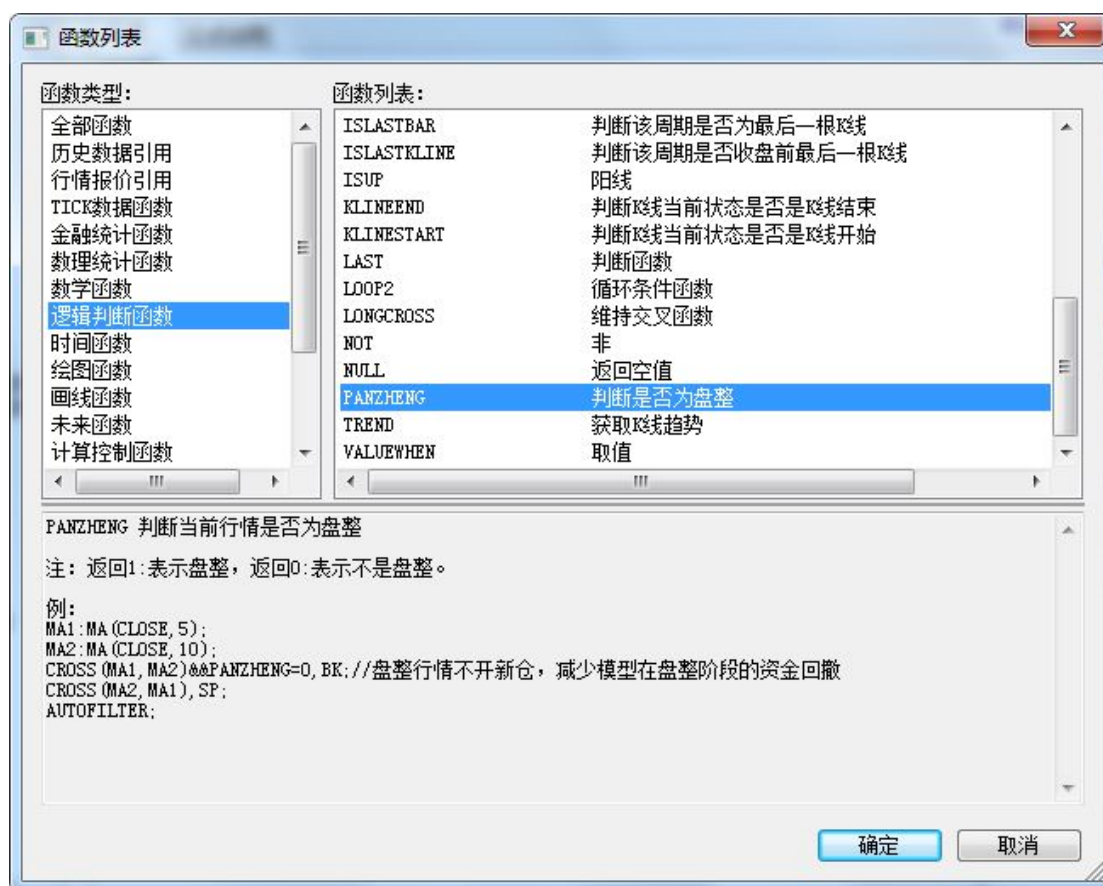
很多趋势模型，在行情出现趋势的时候，都可以很好的抓住趋势，实现盈利，但长期运行下来，最终的结果却是小赚甚至亏钱，问题出在哪里？

原因在于，盘整行情中模型在不断的反复交易，而盘整中的交易都是不盈利甚至亏损的，行情中绝大部分又都是盘整行情，长时间的连续小亏损导致之前的利润全部回吐



**关键函数：**PANZHENG，判断当前行情是否为盘整

注：返回 1:表示盘整，返回 0:表示不是盘整。



### 作用一：增加收益率

简单的均线模型

MA1:=MA(C,5);

MA2:=MA(C,10);

CROSS(MA1,MA2),BPK;

CROSS(MA2,MA1),SPK;

AUTOFILTER;

上面的模型在这段行情中实现盈利 77040 元，如下图所示



(增加阅读软件的页面放大率可查看清晰图片)

加入 PANZHENG 函数，在盘整行情中不开仓

做多代码如下：

MA1:=MA(C, 5);

MA2:=MA(C, 10);

CROSS(MA1, MA2)&&PANZHENG=0, BK;

CROSS(MA2, MA1), SP;

AUTOFILTER;

做多实现盈利 179580 元，如下图所示



(增加阅读软件的页面放大率可查看清晰图片)

做空代码如下：

MA1:=MA(C, 5);

MA2:=MA(C, 10);

CROSS(MA2, MA1)&&PANZHENG=0, SK;

CROSS(MA1, MA2), BP;

AUTOFILTER;

加入盘整函数后，做空亏损 44100 元，如下图所示



(增加阅读软件的页面放大率可查看清晰图片)

**总结:** 未加入盘整函数前, 这段行情中多空共实现盈利 77040 元, 加入盘整函数后, 做多实现盈利 179580 元, 做空亏损 44100 元, 这段行情中多空共实现盈利 135480 元, 加入盘整函数后, 盘整行情交易次数大量减少, 从而减少了亏损, 总盈利提升 76%。

## 作用二: 减小最大回撤

均线模型, PTA 指数, 2010. 1. 1 至今的测试结果

代码如下:

```
MA10:=MA(C, 10);
```

```
C>MA10, BPK; //价格大于 10 周期均线, 做多
```

```
C<MA10, SPK; //价格小于 10 周期均线, 做空
```

```
AUTOFILTER;
```

如下图所示, 模型虽然有年化, 55%的收益率, 但也有 65%的权益最大回撤, 如此大的回撤导致模型的实际可用性大大降低。



(增加阅读软件的页面放大率可查看清晰图片)

加入 **PANZHENG** 函数后，代码如下

MA10:=MA(C, 10);

C>MA10&&PANZHENG=0, BPK; //非盘整行情中，价格大于10周期均线，做多

C<MA10 &&PANZHENG=0, SPK; //非盘整行情中，价格小于10周期均线，做空

AUTOFILTER;

如下图所示

胜率提升 14%

盈利率提升 37%

最大回撤减少 45%

年化盈利率提升 21%

单次交易盈利能力提升 40%

减少盘整行情中的交易次数后，不仅仅盈利能力得到提升，模型的稳定性同时也得到大幅度提升，大大提高了模型的可执行性

模型回测报告			
回测报告 资金曲线			
名称	全部交易	多头	空头
测试天数	1794		
测试周期数	1191		
指令总数	151		
信号消失次数	0		
初始资金	10000.00		
最终权益	44830.00		
空仓周期数	1		
最长连续空仓周期数	1		
最长交易周期	35		
标准离差	1596.51		
标准离差率	6.88		
夏普比率	9.94		
盈亏总平均/亏损平均	0.37		
权益最大回撤	10185.00		
权益最大回撤时间	2013/10/16		
权益最大回撤比	36.69%		
权益最大回撤比时间	2010/11/16		
损益最大回撤	8990.00		
损益最大回撤时间	2013/10/17		
损益最大回撤比	25.23%		
损益最大回撤比时间	2010/09/21		
风险率	20.75%		
收益率/风险率	3.42		
每手最大亏损	2455.00		
每手平均盈亏	232.20		
盈利率	348.30%	98.95%	249.35%
年化单利收益率	70.86%		
月化单利收益率	5.82%		
年化复利收益率	35.70%		
月化复利收益率	2.54%		
胜率	36.87%		
风险收益评级	341.51%		
平均盈利/权益最大回撤	0.17		
平均盈利/平均亏损	2.78	2.62	2.93
净利润	34830.00	9895.00	24935.00
总盈利	94770.00	41715.00	53055.00
总亏损	59940.00	31820.00	28120.00
总盈利/总亏损	1.58	1.31	1.89
其中持仓浮盈	1710.00	0.00	1710.00
交易次数	150.00	75.00	75.00
盈利比率	0.36	0.33	0.39
盈利次数	54.00	25.00	29.00

(增加阅读软件的页面放大率可查看清晰图片)

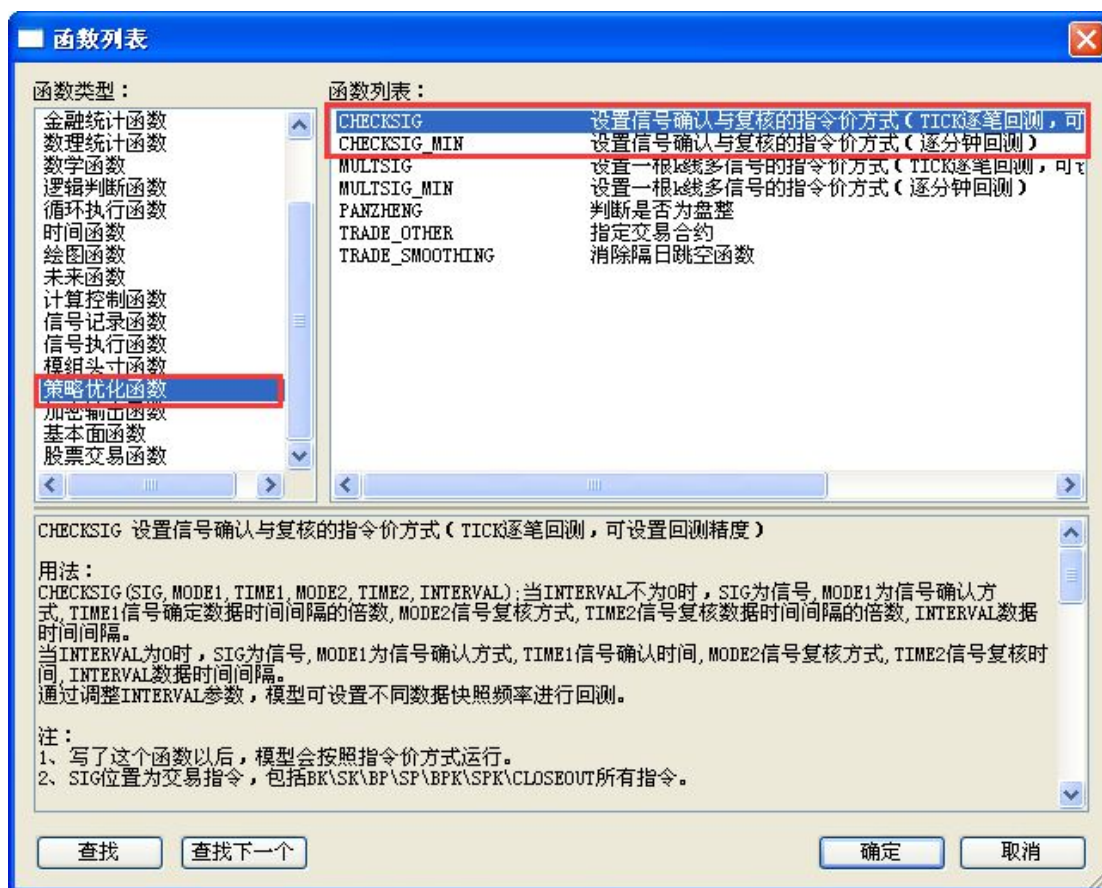
## 1.2 CHECKSIG 函数，实现更具有优势进场价格

大部分趋势模型采用趋势突破或者趋势跟踪的方法捕捉趋势，通常情况下，信号都出现在一根较长的 k 线上，如果使用收盘价模型，入场点在下根 k 线的开盘价上，因此会错过突破这根长 k 线的最佳入场时间，无形中损失掉很大一笔到手的利润；出场时也是如此，那么怎样才能以最优的价格进出场，得到更多的利润呢？

如何实现在 k 线没有走完之前进出场呢？



关键函数: [CHECKSIG](#), 实现出信号立即发出委托。



### 信号复核的意义：

指令价模型，如果策略设计思路不够严谨，可能会产生信号出现后，再次消失的情况，称之为信号忽闪，即开错仓或者平错仓，这时可以使用信号复核进行恢复持仓的操作。

**CHECKSIG** 设置信号确认与复核的指令价方式（TICK 逐笔回测，可设置回测精度）

**CHECKSIG\_MIN** 设置信号确认与复核的指令价方式（逐分钟回测）

### 函数用法：

**CHECKSIG (SIG, MODE1, TIME1, MODE2, TIME2, INTERVAL)** : 当 INTERVAL 不为 0 时, SIG 为信号, MODE1 为信号确认方式, TIME1 信号确定数据时间间隔的倍数, MODE2 信号复核方式, TIME2 信号复核数据时间间隔的倍数, INTERVAL 数据时间间隔。

当 INTERVAL 为 0 时, SIG 为信号, MODE1 为信号确认方式, TIME1 信号确认时间, MODE2 信号复核方式, TIME2 信号复核时间, INTERVAL 数据时间间隔

通过调整 INTERVAL 参数，模型可设置不同数据快照频率进行回测

**CHECKSIG\_MIN (SIG, MODE1, TIME1, MODE2, TIME2)** : SIG 为信号, MODE1 为信号确认方式, TIME1 信号确认时间, MODE2 信号复核方式, TIME2 信号复核时间。逐分钟回测

几种典型的信号复核确认方式对应的写法举例（以 CHECKSIG\_MIN 为例）：

**CHECKSIG\_MIN(SIG, 'A', 0, 'D', 0)** ; // 出信号立即下单，K 线走完复核

**CHECKSIG\_MIN(SIG, 'A', N, 'D', 0)** ; // 出信号 N 分钟确认信号下单，K 线走完复核

CHECKSIG\_MIN(SIG, 'A', N, 'C', 0); // 出信号 N 分钟确认信号下单, 不进行复核  
 CHECKSIG\_MIN(SIG, 'B', N, 'D', 0); // K 线走完前 N 分钟确认信号下单, K 线走完复核  
 CHECKSIG\_MIN(SIG, 'B', N, 'C', 0); // K 线走完前 N 分钟确认信号下单, 不复核  
 CHECKSIG\_MIN(SIG, 'B', 0, 'C', N); // K 线走完确认信号下单  
 CHECKSIG\_MIN(SIG, 'B', 0, 'D', 0); // K 线走完确认信号下单  
 CHECKSIG\_MIN(SIG, 'A', 0, 'C', 0); // 出信号立即下单, 不复核  
 CHECKSIG\_MIN(SIG, 'A', 0, 'F', 10); // 出信号立即下单, 收盘前最后一根 K 线提前 10 分钟进行复核。

模型中加入 CHECKSIG, 在实现指令价模型的同时, 可以针对不同指令实现不同的信号复核方式, 让交易策略的实现更加灵活。

如:

CHECKSIG\_MIN(BK, 'A', 0, 'D', 0); // 出信号立即下单, K 线走完复核

买开仓信号出现, 立即发出委托, K 线走完如果信号存在, 则无需进行处理。

如果 K 线走完信号消失, 则恢复原状, 即平掉多单。



源码:

MA5 $\leftarrow$ MA(C, 5);

MA10 $\leftarrow$ MA(C, 10);

RSV:=(CLOSE-LLV(LOW, 9))/(HHV(HIGH, 9)-LLV(LOW, 9))\*100;

K:=SMA(RSV, 3, 1);

D:=SMA(K, 3, 1);

```

EVERY (MA5>MA10, 5) &&CROSSUP (K, D), BK;
EVERY (MA5<MA10, 5) &&CROSSDOWN (K, D), SK;
BARSBK>2&&C<=BKHIGH-10*MINPRICE, SP;
BARSSK>2&&C>=SKLOW+10*MINPRICE, BP;
CHECKSIG_MIN(BK, 'B', 1, 'C', 0); // BK 信号的执行方式为: K 线走完前 1 秒下单, 不进行复核
CHECKSIG_MIN(SK, 'B', 1, 'C', 0); //SK 信号的执行方式为: K 线走完前 1 秒下单, 不进行复核
CHECKSIG_MIN(SP, 'A', 0, 'C', 0); //SP 信号的执行方式为: 出信号立即下单, 不进行复核
CHECKSIG_MIN(BP, 'A', 0, 'C', 0); //BP 信号的执行方式为: 出信号立即下单, 不进行复核
AUTOFILTER;

```

### 如何实现指令价模型



是不是所有的模型用指令价效果都要优于收盘价呢?

答案是否定的。

究竟用指令价效果好还是收盘价效果好,还要根据交易策略决定。一些交易逻辑简单的模型,指令价或者收盘价效果区别较小。但收盘价模型无法处理更加细致的交易逻辑,就需要采用指令价了。

**Wh8.2 是国内程序化平台中唯一提供指令价模型 tick 回测的程序化交易软件。是历史回测最精准的程序化交易软件。**



是否支持指令价委托并不重要,重要的是**是否支持回溯**

程序化交易平台	是否支持回溯	是否支持指令价委托	是否支持信号消失自动处理
赢智 wh8.1	价格估算	是	是
赢智 wh8.2	Tick 回溯	是	是
其他程序化交易平台	否	是	否

### 如何设置模型回溯精度

在并不需要用每笔 TICK 数据进行回溯的情况下,可以通过调整 CHECKSIG 函数的 INTERVAL 参数值来调整模型的回溯精度。

使用 CHECKSIG 函数, CHECKSIG (SIG, MODE1, TIME1, MODE2, TIME2, INTERVAL)

INTERVAL 代表数据时间间隔

- 1) 支持 0、3、5、10 四个值, 不支持变量。
- 2) 3、5、10 分别代表用每隔 3 秒、5 秒、10 秒, 计算一次模型
- 3) 参数为 3、5、10 , 回溯速度可提升 3-10 倍, 回溯精度稍差
- 4) 参数为 0 的时 为每笔 TICK 计算一次模型
- 5) 一个模型中只能写入一个 INTERVAL 值

例:

C>0, BK;

C<0, SP;

CHECKSIG(BK, 'A', 5, 'D', 0, 3); //设置 BK 信号, 出信号 5\*3=15 秒后确认下单, K 线走完复核。  
每隔 3 秒计算一次信号。

CHECKSIG(SP,'A',0,'C',10,3); //设置 SP 信号，出信号立即下单，下单后 10\*3=30 秒复核。每隔 3 秒计算一次信号。

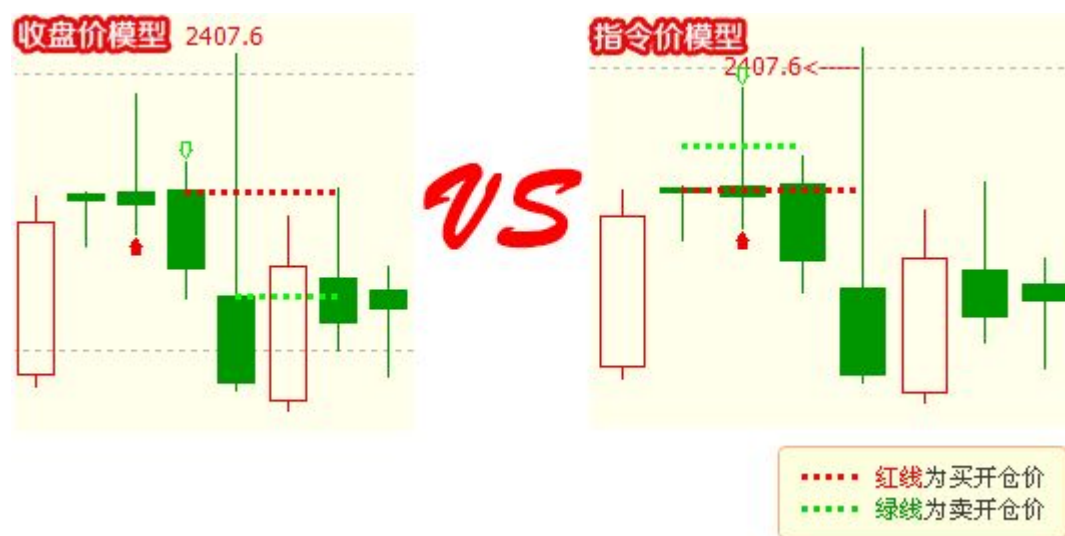
AUTOFILTER;

### 1.3 MULTSIG 函数，在一根 k 线上灵活进出

期货价格瞬息万变，经常会出现价格瞬间拉升，接着就瞬间回吐的情况。拉升时模型出现开仓信号，如果遇到秒杀行情，不能够及时平仓，往往会带来较大的亏损。能否处理好秒杀行情，已经成为重点解决的问题。

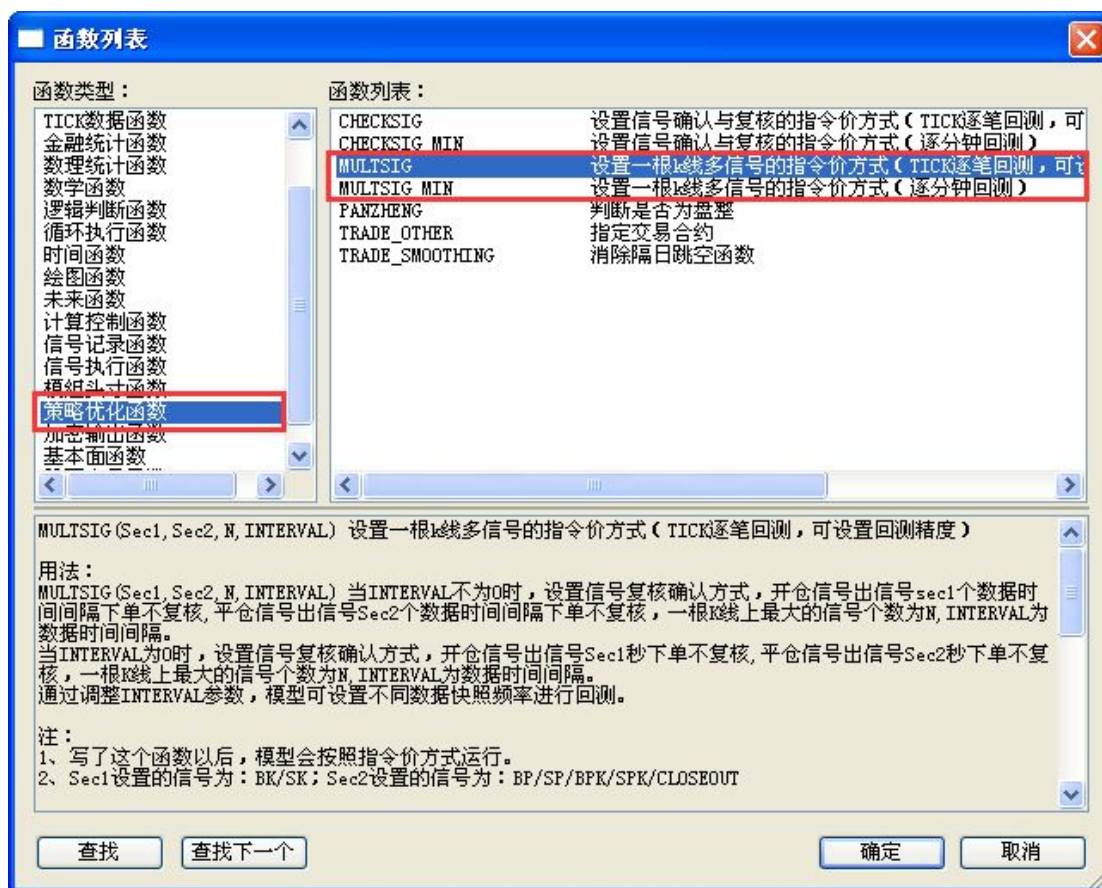
如何才能做到同一根 k 线开仓后快速止损呢？

回测止盈策略：



**总结：**在这种秒杀行情中，行情已经逆转，收盘价模型，还在执行上根 k 线的买开仓指令，显然是错误的，导致亏损。而指令价模型，则可以当根 k 线同时完成进场和止盈动作，保证既得利润。

**关键函数：**MULTSIG，可以在一根 k 线上交易多次



模型中加入 **MULTSIG**，在实现指令价模型的同时，同时可以实现在一根 k 线上反复进行交易，实现更精致的交易策略，可以很好的规避掉秒杀行情。

#### 函数用法：

**MULTSIG(Sec1, Sec2, N, INTERVAL)**：当 INTERVAL 不为 0 时，设置信号复核确认方式，开仓信号出信号 sec1 个数据时间间隔下单不复核，平仓信号出信号 Sec2 个数据时间间隔下单不复核，一根 K 线上最大的信号个数为 N，INTERVAL 为数据时间间隔。

当 INTERVAL 为 0 时，设置信号复核确认方式，开仓信号出信号 Sec1 秒下单不复核，平仓信号出信号 Sec2 秒下单不复核，一根 K 线上最大的信号个数为 N，INTERVAL 为数据时间间隔。

通过调整 INTERVAL 参数，模型可设置不同数据快照频率进行回测

INTERVAL 代表数据时间间隔

- 1) 支持 0、3、5、10 四个值，不支持变量。
- 2) 参数为 3、5、10 分别代表用每隔 3 秒、5 秒、10 秒，计算一次模型
- 3) 参数为 3、5、10，回测速度可提升 3-10 倍，回测精度稍差
- 4) 参数为 0 的时候 为每笔 TICK 计算一次模型，不限制加载周期
- 5) 一个模型中只能写入一个 INTERVAL 值

**MULTSIG\_MIN(min1, min2, N)**：设置一根 k 线多信号的指令价方式（逐分钟回测），开仓信号出信号 min1 分钟下单不复核，平仓信号出信号 min2 分钟下单不复核，一根 K 线上最大的信号个数为 N。

### 模型源码:

`C>REF(H,1),BK;`//价格大于上一根 k 线最高价, 开多仓

`C<BKPRICE-3*MINPRICE,SP;`//亏损 3 点止损

`MULTSIG_MIN(0,0,3);`//开仓信号, 出信号立即下单, 不复核; 平仓信号出信号立即下单, 不复核。一根 K 线上最大信号个数为 3。

`AUTOFILTER;`



### 案例:

收盘价模型, 只能在下一根 k 线的开盘时止损

`ST:=ABS(C-0);`

`ST>MA(ST,20)&&C>0,BK;`

`ST>MA(ST,20)&&C<0,SK;`

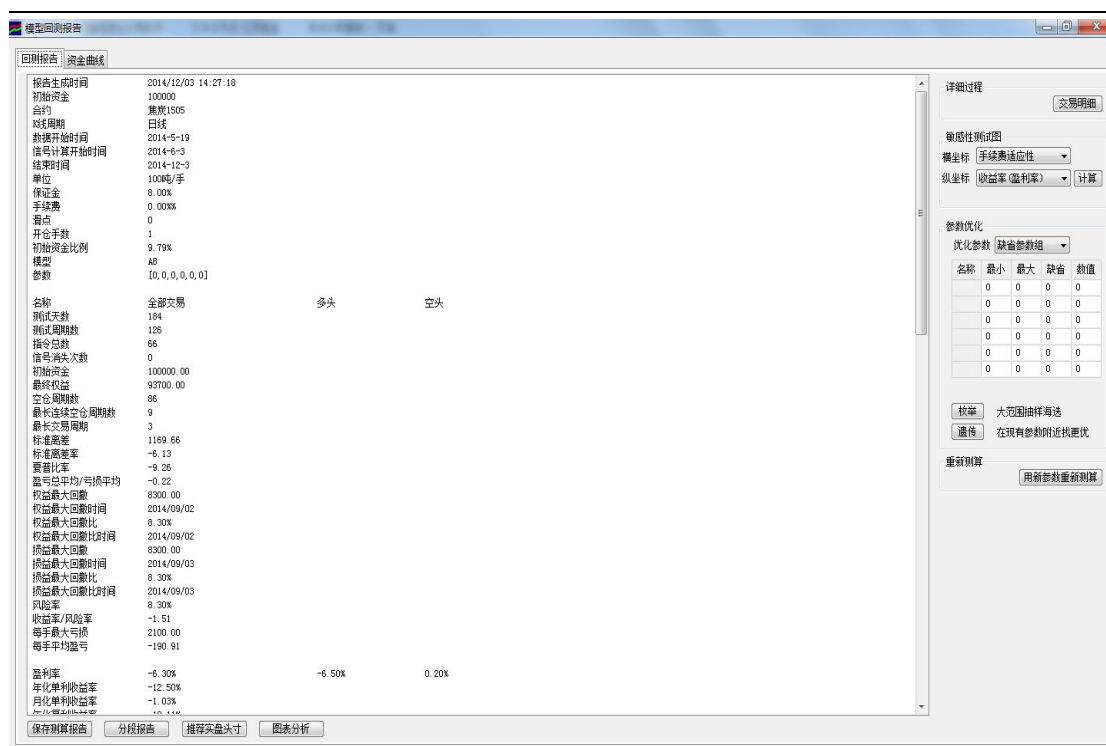
`C>HV(C,3),BP;`

`C<LV(C,3),SP;`

`(H-C)>(H-0)*0.2,SP;`

`(C-L)>(O-L)*0.2,BP;`

`AUTOFILTER;`



(增加阅读软件的页面放大率可查看清晰图片)

加入 MULTSIG 函数，实现在同一根 k 线上开仓后及时止损，亏损变为盈利

ST:=ABS(C-0) ;

ST>MA(ST, 20)&&C>0, BK;

ST>MA(ST, 20)&&C<0, SK;

C>HV(C, 3), BP;

C<LV(C, 3), SP;

(H-C)>(H-0)\*0.2, SP;

(C-L)>(O-L)\*0.2, BP;

MULTSIG\_MIN(0, 0, 2);

AUTOFILTER;



### 1.4.1 拓展思路——指数交易

我们能不能测试出指数和实际交易差别的真实情况呢？

注：

1、

回测时：信号价格取值为该函数定义的交易合约的信号价格。

模组加载时：数据合约为转载模组时选择的数据合约，交易合约为该函数指定的合约。不写入该函数时，交易和数据合约一致。

2、该函数写为 TRADE\_OTHER('AUTO') 时，可以加载到指数、主连、主指合约，实现自动换月移仓。建议加载到指数或主指合约使用，主连合约切换主力合约的接缝会有跳空，破坏合约趋势，可能导致出现不应该出现的信号

3、(1) CODE 位置写为' AUTO' 时, 如果加载在指数合约上, 主连数据开始时间在指定信号计算时间之内, 从主连数据开始时间开始计算信号;

(2) CODE 位置为具体合约时，具体合约数据开始时间在指定信号计算时间之内，从具体合约数据开始时间开始计算信号

4、数据合约和交易合约的数据不对齐时，会计算信号，交易合约的相关价格取交易合约最后一根 K 线的收盘价。

5、(1) CODE 位置写为' AUTO' 时：

如果加载在主连合约上，该函数可以和 CHECKSIG, MULTSIG、CHECKSIG\_MIN、MULTSIG\_MIN, CLOSEKLINE\_MIN 函数连用；

如果加载在指数、主指合约上，该函数可以和 CLOSEKLINE\_MIN, CHECKSIG\_MIN, MULTSIG\_MIN 函数连用；不支持和 CHECKSIG, MULTSIG 函数连用

(2) CODE 位置为具体合约时：该函数可以和 CLOSEKLINE\_MIN, CHECKSIG\_MIN, MULTSIG\_MIN 函数连用；不支持和 CHECKSIG, MULTSIG 函数连用。

6、(1) CODE 位置写为' AUTO' 时：该函数可以加载到指数、主连上，不可以加载到主指和其他具体合约上。

(2) CODE 位置为具体合约时：该函数可以加载到所有合约上。

7、该函数必须在有信号的模型中使用。

8、TRADE\_OTHER 函数不支持加载到副图中。

9、CODE 位置写为合约代码时，该函数不支持加载到 TICK 周期，量能周期，秒周期上使用；CODE 位置写为' AUTO' 时，该函数不支持加载到日线以上周期使用。

10、CODE 位置不支持写入文华码。

11、该函数不支持加载到页面盒子中使用。

**一个均线模型，在未指定交易合约的时候，测试结果如下：**

模型源码：

MA10:=MA(C, 10);

MA30:=MA(C, 30);

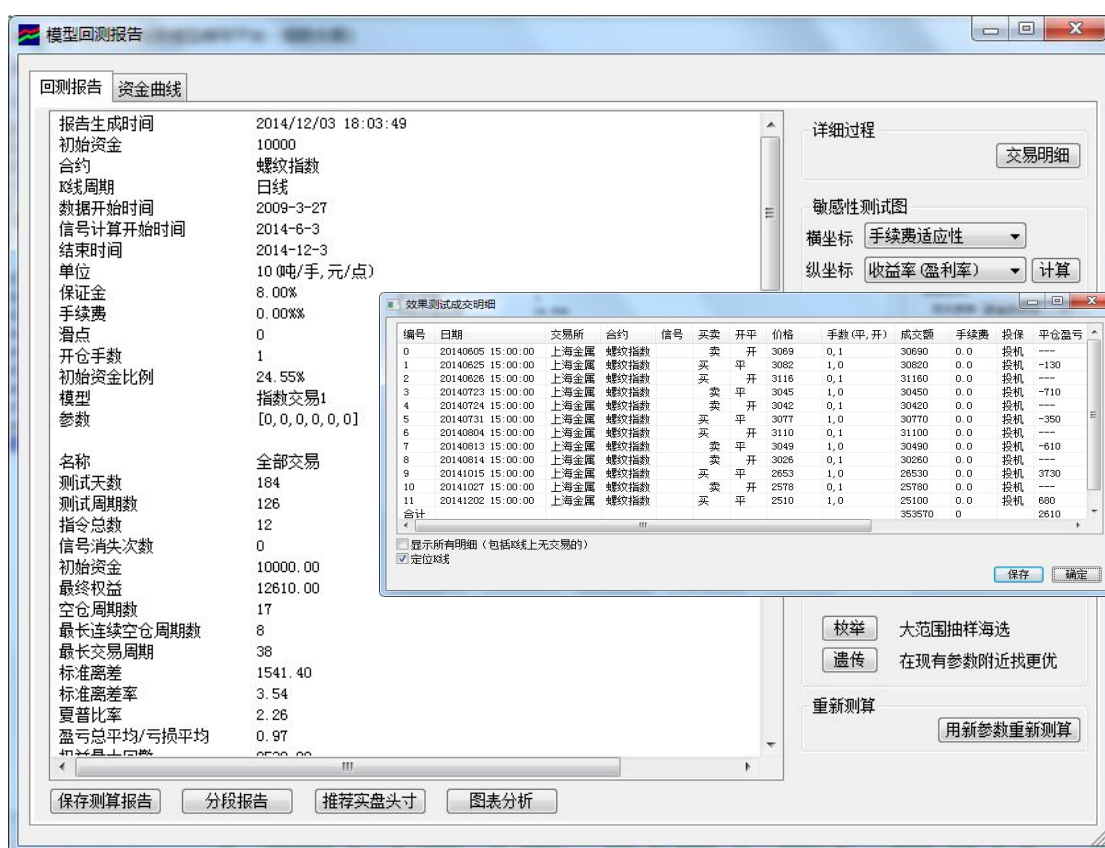
EVERY(C>MA10, 4)&&C>MA30, BK;

EVERY(C<MA10, 4)&&C<MA30, SK;

EVERY(C>MA10, 4), BP;

EVERY(C<MA10, 4), SP;

AUTOFILTER;



(增加阅读软件的页面放大率可查看清晰图片)

模型在指数上计算信号,交易也直接在指数上进行,而现实中指数是无法完成交易的,这种测试结果的真实性就无法保证。

那么真实的情况是什么样的?

加入 TRADE\_OTHER 函数,模型源码入下:

MA10:=MA(C,10);

MA30:=MA(C,30);

EVERY(C>MA10,4)&&C>MA30,BK;

EVERY(C<MA10,4)&&C<MA30,SK;

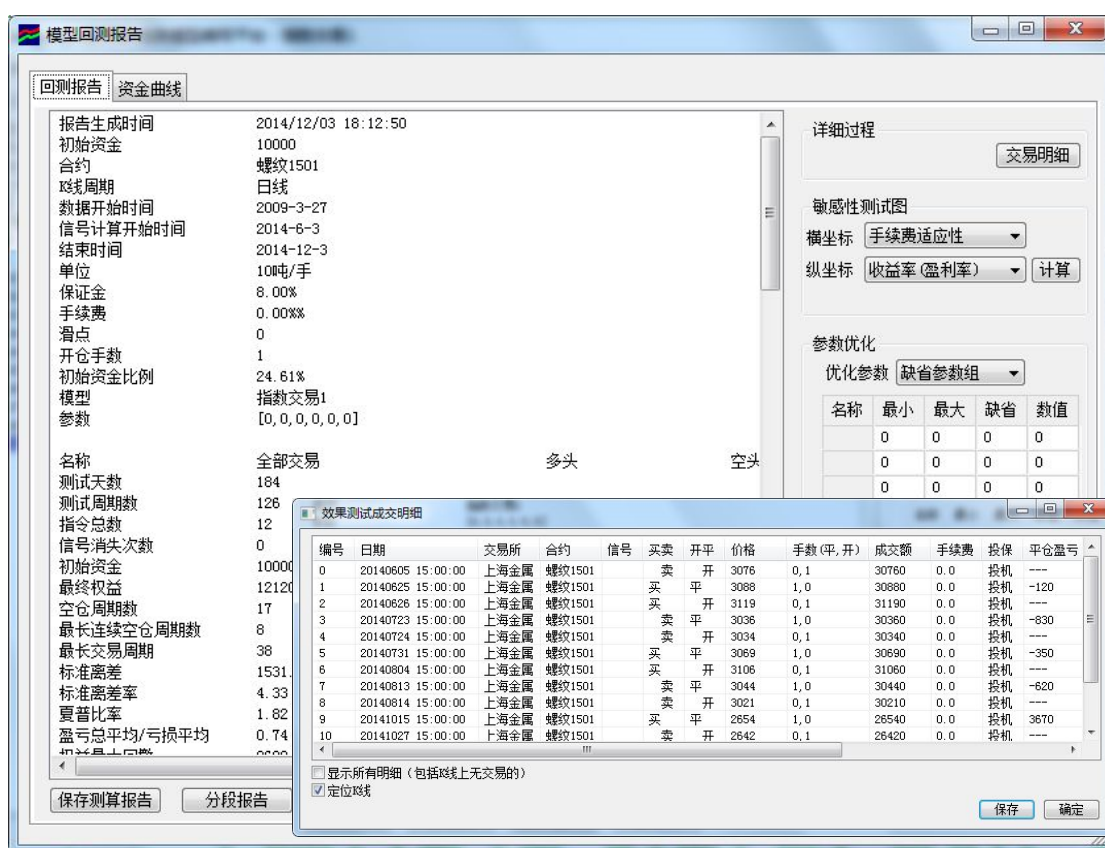
EVERY(C>MA10,4),BP;

EVERY(C<MA10,4),SP;

TRADE\_OTHER('RB1501');

AUTOFILTER;

测试结果为:



(增加阅读软件的页面放大率可查看清晰图片)

我们指定 RB1501 为交易合约后，发现盈利并没有明显变化，说明指数开平仓价格的价差，和具体合约比较接近。

但这是真实的情况吗？

下图是螺纹 1501 合约，2014 年 6 月 3 号至今的走势，红圈中交易量较少，在这个时间段中出现的信号，很难完成交易或者将付出较大的滑点成本，必定会导致盈利的减少，那么如何知道最终的真实的交易结果？



我们在模型中加入了流动性限制，即成交量。根据经验螺纹的成交量应该在 50 万手以上的时候，才能够使交易顺畅。

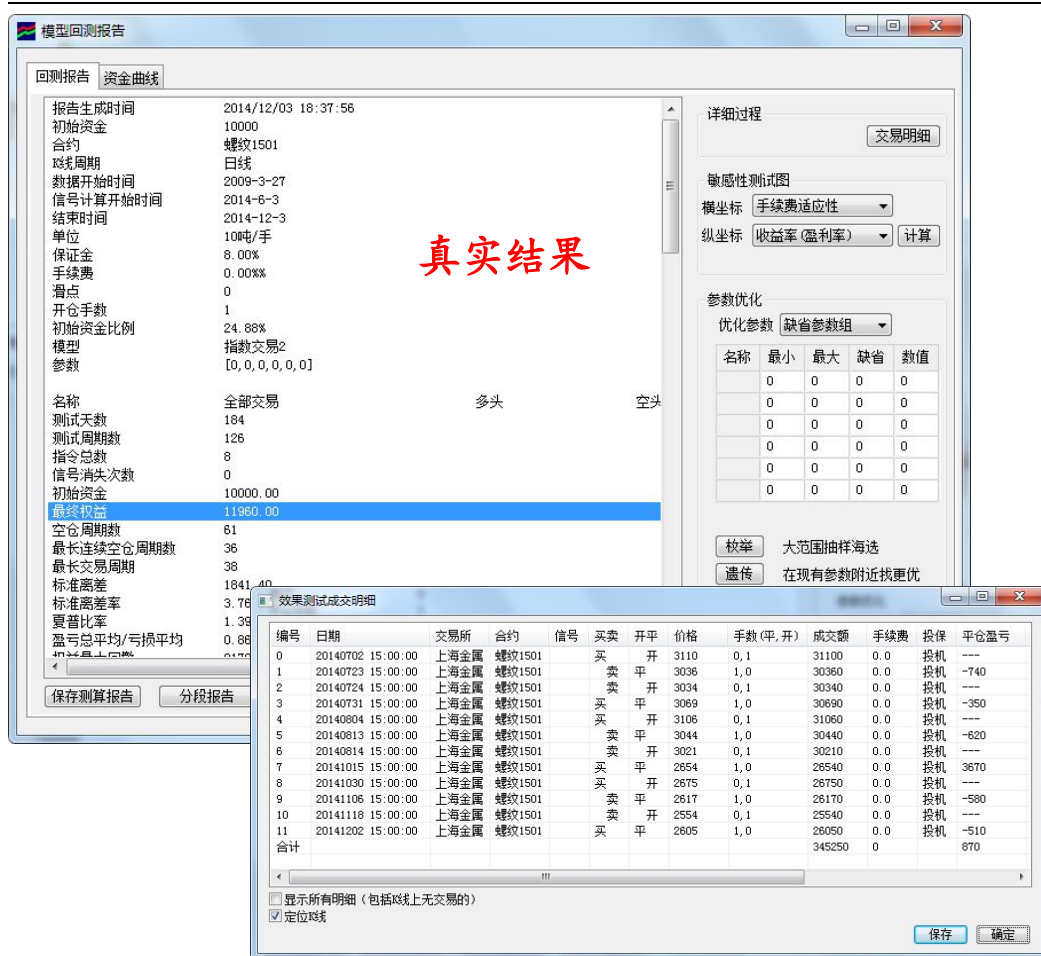
即指定交易合约又保证流动性的代码如下：

```

MA10:=MA(C, 10);
MA30:=MA(C, 30);
EVERY (C>MA10, 4)&&C>MA30&&VV>500000, BK;
EVERY (C<MA10, 4)&&C<MA30&&VV>500000, SK;
EVERY (C>MA10, 4), BP;
EVERY (C<MA10, 4), SP;
TRADE_OTHER('RB1501');
AUTOFILTER;

```

//VV 为跨合约引用 RB1501 的成交量



(增加阅读软件的页面放大率可查看清晰图片)

当我们加入了流动性限制后，发现真实的交易情况，比开始的测试结果又差了一些。这才是真实的结果。

所以，是否支持指数映射到主力合约交易并不重要，重要的是我们是否可以测试出这种映射交易的真实结果。

程序化交易平台	是否支持回测	是否支持映射交易
赢智 wh8.1	否	是
赢智 wh8.2	支持	是
其他程序化交易平台	否	是

#### 1.4.2 如何解决移仓换月问题

我们常常会这样做：模型加载在主力连续合约上，指定主力合约为交易合约，只要主力合约切换，就自动移仓，这种方法正确吗？正确的方法应该是怎么样的？我们可以测试出来吗？有没有更好的方法？我们能找到吗？

主连合约加载模型，自动移仓换月，会产生较大的移仓成本

MA10:=MA(C, 10);

```
MA30:=MA(C, 30);
EVERY(C>MA10, 4)&&C>MA30, BK;
EVERY(C<MA10, 4)&&C<MA30, SK;
EVERY(C>MA10, 4), BP;
EVERY(C<MA10, 4), SP;
TRADE_OTHER(' AUTO' );
AUTOFILTER;
```

效果测试成交明细																
编号	日期	交易所	合约	信号	买卖	开平	价格	手数(平,开)	成交额	手续费	投保	平仓盈亏	可用资金	权益	滑点成本	信号消失成本
0	20090511 15:00:00	上海金属	rb0909		买	开	3639	0,1	36390	0.0	投机	---	7089	10000	0	---
1	20090527 15:00:00	上海金属	rb0909		卖	平	3610	1,0	36100	0.0	投机	-290	9710	9710	0	---
2	20090604 15:00:00	上海金属	rb0909		买	开	3700	0,1	37000	0.0	投机	---	6750	9710	0	---
3	20090629 15:00:00	上海金属	rb0909		卖	平	3629	1,0	36290	0.0	投机	1290	11000	11000	0	---
4	20090714 15:00:00	上海金属	rb0910		买	开	3927	0,1	39270	0.0	投机	---	7858	11000	0	---
5	20090805 09:00:00	上海金属	rb0910	移仓	卖	平	4848	1,0	48480	0.0	投机	9210	20210	20210	0	---
6	20090805 09:00:00	上海金属	rb0911	移仓	买	开	4870	0,1	48700	0.0	投机	---	16314	20260	0	---
7	20090813 15:00:00	上海金属	rb0911		卖	平	4617	1,0	46170	0.0	投机	-2530	17680	17680	0	---
8	20090819 15:00:00	上海金属	rb0911		买	开	4219	0,1	42190	0.0	投机	---	14305	17680	0	---
9	20090901 09:00:00	上海金属	rb0911	移仓	买	平	4133	1,0	41330	0.0	投机	860	18540	18540	0	---
10	20090901 09:00:00	上海金属	rb0912	移仓	卖	开	4098	0,1	40980	0.0	投机	---	15262	18770	0	---
11	20091012 09:00:00	上海金属	rb0912	移仓	买	平	3644	1,0	36440	0.0	投机	4540	23080	23080	0	---
12	20091012 09:00:00	上海金属	rb1001	移仓	卖	开	3634	0,1	36340	0.0	投机	---	20173	22470	0	---
13	20091020 15:00:00	上海金属	rb1001		买	平	3842	1,0	38420	0.0	投机	-2080	21000	21000	0	---
14	20091021 15:00:00	上海金属	rb1001		买	开	3887	0,1	38870	0.0	投机	---	17890	21000	0	---
15	20091106 15:00:00	上海金属	rb1001		卖	平	3890	1,0	38900	0.0	投机	30	21030	21030	0	---
16	20091112 15:00:00	上海金属	rb1002		买	开	4048	0,1	40480	0.0	投机	---	17792	21030	0	---
17	20091130 09:00:00	上海金属	rb1002	移仓	卖	平	3939	1,0	39390	0.0	投机	-1090	19940	19940	0	---
18	20091130 09:00:00	上海金属	rb1005	移仓	买	开	4336	0,1	43360	0.0	投机	---	16471	20330	0	---
19	20100112 15:00:00	上海金属	rb1005		卖	平	4510	1,0	45100	0.0	投机	1740	21680	21680	0	---
20	20100113 15:00:00	上海金属	rb1005		买	开	4418	0,1	44180	0.0	投机	---	18146	21680	0	---
21	20100222 15:00:00	上海金属	rb1005		卖	平	4243	1,0	42430	0.0	投机	1750	23430	23430	0	---
22	20100302 15:00:00	上海金属	rb1005		买	开	4155	0,1	41550	0.0	投机	---	20106	23430	0	---
23	20100309 09:00:00	上海金属	rb1005	移仓	买	平	4189	1,0	41890	0.0	投机	-340	23090	23090	0	---
24	20100309 09:00:00	上海金属	rb1010	移仓	卖	开	4615	0,1	46150	0.0	投机	---	19398	22880	0	---
25	20100312 15:00:00	上海金属	rb1010		买	平	4658	1,0	46580	0.0	投机	-430	22660	22660	0	---
26	20100315 15:00:00	上海金属	rb1010		买	开	4660	0,1	46600	0.0	投机	---	18932	22660	0	---
27	20100422 15:00:00	上海金属	rb1010		卖	平	4734	1,0	47340	0.0	投机	740	23400	23400	0	---
28	20100423 15:00:00	上海金属	rb1010		买	开	4740	0,1	47400	0.0	投机	---	19608	23400	0	---
29	20100716 09:00:00	上海金属	rb1010	移仓	买	平	3909	1,0	39090	0.0	投机	8310	31710	31710	0	---
30	20100716 09:00:00	上海金属	rb1101	移仓	卖	开	4058	0,1	40580	0.0	投机	---	28464	31630	0	---
31	20100721 15:00:00	上海金属	rb1101		买	平	4177	1,0	41770	0.0	投机	-1190	30520	30520	0	---
32	20100722 15:00:00	上海金属	rb1101		买	开	4173	0,1	41730	0.0	投机	---	27182	30520	0	---
33	20100825 15:00:00	上海金属	rb1101		卖	平	4225	1,0	42250	0.0	投机	520	31040	31040	0	---
34	20100826 15:00:00	上海金属	rb1101		买	开	4257	0,1	42570	0.0	投机	---	27634	31040	0	---
35	20100902 15:00:00	上海金属	rb1101		买	平	4386	1,0	43860	0.0	投机	-1290	29750	29750	0	---
36	20100903 15:00:00	上海金属	rb1101		买	开	4385	0,1	43850	0.0	投机	---	28258	29750	0	---
37	20100917 15:00:00	上海金属	rb1101		卖	平	4359	1,0	43590	0.0	投机	-60	29690	29690	0	---
38	20100921 15:00:00	上海金属	rb1101		买	开	4317	0,1	43170	0.0	投机	---	26236	29690	0	---
39	20101014 09:00:00	上海金属	rb1101	移仓	买	平	4324	1,0	43240	0.0	投机	-70	29620	29620	0	---
40	20101014 09:00:00	上海金属	rb1105	移仓	卖	开	4515	0,1	45150	0.0	投机	---	26008	29620	0	---
41	20101014 15:00:00	上海金属	rb1105		买	平	4474	1,0	44740	0.0	投机	410	30030	30030	0	---
42	20101015 15:00:00	上海金属	rb1105		买	开	4498	0,1	44980	0.0	投机	---	26432	30030	0	---
43	20101117 15:00:00	上海金属	rb1105		卖	平	4550	1,0	45500	0.0	投机	520	30550	30550	0	---
44	20101129 15:00:00	上海金属	rb1105		买	开	4668	0,1	46680	0.0	投机	---	26816	30550	0	---
45	20101228 15:00:00	上海金属	rb1105		卖	平	4744	1,0	47440	0.0	投机	760	31310	31310	0	---

☐ 显示所有明细 (包括线上无交易的)

☒ 定位均线

(增加阅读软件的页面放大率可查看清晰图片)

怎么做才能解决指数加载模型，移仓换月的问题呢？



更好的方法是：加载两个模型，分别交易新主力合约和老主力合约。

在保证流动性的前提下，提前运行新主力合约，继续运行老主力合约，既减少移仓成本，又抓住更多的交易机会。

模型 1 源码：

```
MA10:=MA(C, 10);
MA30:=MA(C, 30);
#CALL[6881, VVA] AS AA
VV:=AA.VV; //VV 为跨合约引用 RB1501 的成交量
EVERY(C>MA10, 4)&&C>MA30&&VV>500000, BK;
EVERY(C<MA10, 4)&&C<MA30&&VV>500000, SK;
EVERY(C>MA10, 4), BP;
EVERY(C<MA10, 4), SP;
TRADE_OTHER('RB1501');
AUTOFILTER;
```

模型 2 源码：

```
MA10:=MA(C, 10);
MA30:=MA(C, 30);
#CALL[6890, VVA] AS AA
VV:=AA.VV; //VV 为跨合约引用 RB1410 的成交量
EVERY(C>MA10, 4)&&C>MA30&&VV>500000, BK;
```

EVERY (C<MA10, 4)&&C<MA30&&VV>500000, SK;

EVERY (C>MA10, 4), BP;

EVERY (C<MA10, 4), SP;

TRADE\_OTHER('RB1410');

AUTOFILTER;

对模型 1 进行回测:

MA10:=MA(C, 10);

MA30:=MA(C, 30);

#CALL[6881, VVA] AS AA

VV:=AA.VV; //VV 为跨合约引用 RB1501 的成交量

EVERY (C>MA10, 4)&&C>MA30&&VV>500000, BK;

EVERY (C<MA10, 4)&&C<MA30&&VV>500000, SK;

EVERY (C>MA10, 4), BP;

EVERY (C<MA10, 4), SP;

TRADE\_OTHER('RB1501');

AUTOFILTER;

虽然模型加载在指数上,但是交易都是在 1501 上进行,并且只在流动充足的时候交易,交易时间段为 2014 年 7 月至 11 月,其他时间 1501 因流动性差,不进行交易



(增加阅读软件的页面放大率可查看清晰图片)

对模型 2 进行回测:

MA10:=MA(C, 10);

MA30:=MA(C, 30);

#CALL[6890, VVA] AS AA

VV:=AA.VV; //VV 为跨合约引用 RB1410 的成交量

EVERY (C>MA10, 4)&&C>MA30&&VV>500000, BK;

EVERY (C<MA10, 4)&&C<MA30&&VV>500000, SK;

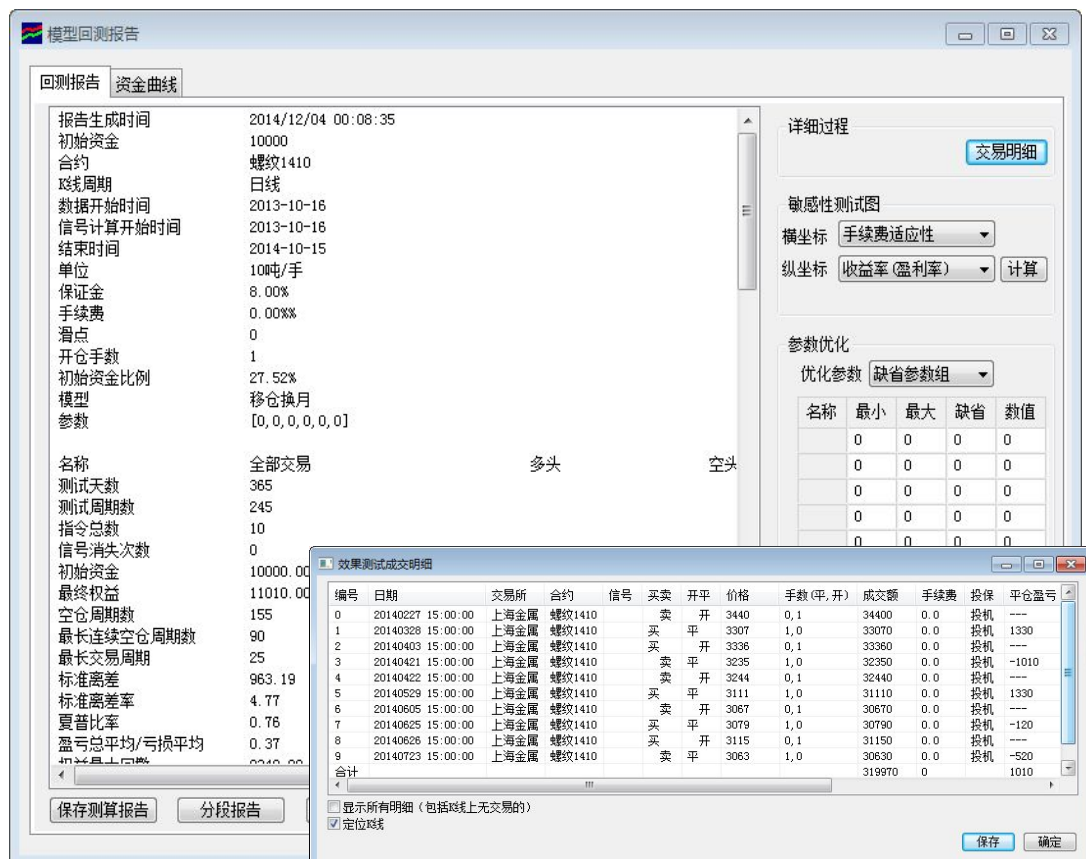
EVERY (C>MA10, 4), BP;

EVERY (C<MA10, 4), SP;

TRADE\_OTHER('RB1410');

AUTOFILTER;

虽然模型加载在指数上, 但是交易都是在 1410 上进行, 并且只在流动充足的时候交易, 交易时间段为 2014 年 2 月至 7 月初, 其他时间 1410 因流动性差, 不进行交易



(增加阅读软件的页面放大率可查看清晰图片)

总结:

这样在某段时间内, 既可以因为合约流动性低而停止交易, 减少滑点, 又在主力合约切换之前的一段时间内, 因为两个合约的流动性都较高, 获得了重叠部分的交易机会, 不但解决了移仓

换月带来的巨大成本，又因为重叠部分的交易，获得更高的收益。

### 1.5 拓展思路—结合盘口数据研发策略

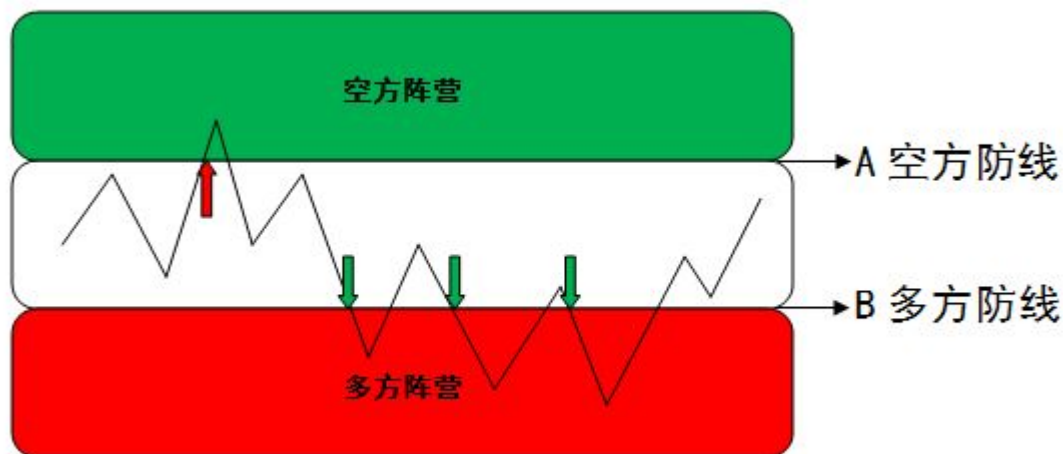
IF1412		IF1412	
卖五	2997.0	682	
卖四	2996.8	262	
卖三	2996.6	65	
卖二	2996.4	18	
卖一	2996.2	94	
买一	2995.8	35	
买二	2995.6	18	
买三	2995.4	2	
买四	2995.0	19	
买五	2994.8	17	
总卖	2996.6	1121	
总买	2995.3	91	

盘口的五档行情，可以真实反映出短时间市场多空优势。五档行情的上方是空方阵营，下方是多方阵营当一方优势明显的时候，行情便会快速上涨或者下跌益曲线

$A = (ASK1 * ASK1VOL + ASK2 * ASK2VOL + ASK3 * ASK3VOL + ASK4 * ASK4VOL + ASK5 * ASK5VOL) / (ASK1VOL + ASK2VOL + ASK3VOL + ASK4VOL + ASK5VOL)$  ; //空方挂单均价

$B = (BID1 * BID1VOL + BID2 * BID2VOL + BID3 * BID3VOL + BID4 * BID4VOL + BID5 * BID5VOL) / (BID1VOL + BID2VOL + BID3VOL + BID4VOL + BID5VOL)$  ; //多方挂单均价

我们分别用 A 和 B 代表空方和多方，当价格突破一方的时候，就会出现短暂的上涨或下跌行情，而很多时候，价格都是在多方和空方阵营之间波动



#### 1.5.1 文华通道线：

**策略思路：**分别用空头和多头的挂单均价代表多空阵营，当价格向上突破空方阵营时做多，当价格向下突破多方阵营时做空。

$MA(CLOSE, 30)$  ;

$A := (ASK1 * ASK1VOL + ASK2 * ASK2VOL + ASK3 * ASK3VOL + ASK4 * ASK4VOL + ASK5 * ASK5VOL) / (ASK1VOL + ASK2VOL + ASK3VOL + ASK4VOL + ASK5VOL);$

$B := (BID1 * BID1VOL + BID2 * BID2VOL + BID3 * BID3VOL + BID4 * BID4VOL + BID5 * BID5VOL) / (BID1VOL + BID2VOL + BID3VOL + BID4VOL + BID5VOL);$

RISING(60) && C > MA(A, 30), BK;

RISING(60) = 0 && C < MA(B, 30), SK;

RISING(60) = 0, SP;

RISING(60), BP;

AUTOFILTER;





### 1.5.2 盘口函数

**RISING(N)** 判断 TICK 图的价格趋势。

用法：参数 N 代表周期，在 N 个周期内，上涨趋势返回 1，下跌趋势返回 0，平盘时返回空值。

注：

- 1、该函数必须在 TICK 图中使用，在 K 线图上返回空值。
- 2、该函数需要有五档行情授权才能取到有效值，否则返回空值。

例：

RR:RISING(N); //加载到有五档授权的 TICK 图中，上涨趋势返回 1，反之返回 0。

#### TICK 盘口模型范例

**策略原理：**日内走势出现逐笔连续上涨或者下跌的概率较小，当短暂的上涨或者下跌形成后，价格极容易反向运行，连续 10 笔上涨趋势，并且 10 笔中有 4 笔以上主动卖，且单次手数超过 50 手，开空仓，反之开多仓。

```
DEF_TICKDATA(1,10);
SETBIGVOL(50);
SHE:=ASKBIGCOUNT;
BHE:=BIDBIGCOUNT;
SHE>=4&&RISING(10)=1,SK;
BHE>=4&&RISING(10)=0,BK;
```

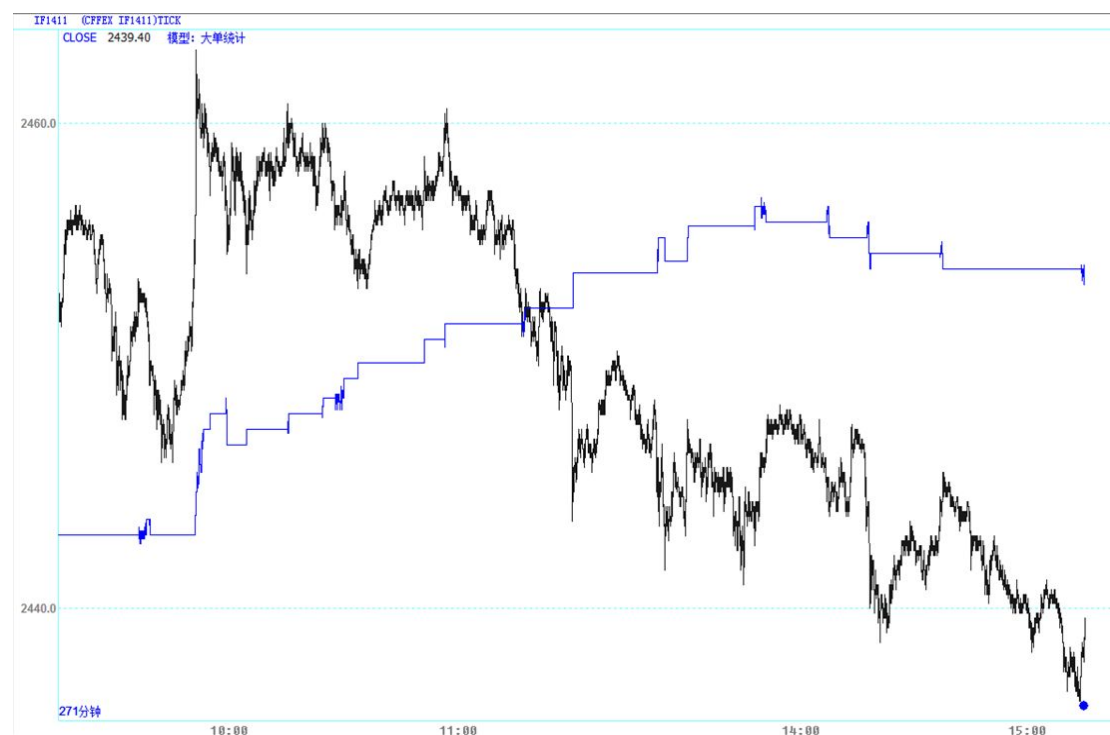
NEW<=BKPRICE-4\*MINPRICE, SP;

NEW>=SKPRICE+4\*MINPRICE, BP;

NEW>=BKPRICE+4\*MINPRICE, SP;

NEW<=SKPRICE-4\*MINPRICE, BP;

AUTOFILTER;



一些其他盘口概念的解释：

多开：多头开仓	持仓量增加
多平：多头平仓	持仓量减少
换手：多空换手	持仓量不变
双开：多头开仓 空头开仓	持仓量增加
双平：多头平仓 空头平仓	持仓量减少
空开：空头开仓	持仓量增加
空平：空头平仓	持仓量减少

时间	价位	现手	增仓	开平	IF1411	IF1411
15:03:26	2428.0	7	-1	多平	卖五 2428.8	125
15:03:27	2428.0	12	-4	空平	卖四 2428.6	167
15:03:27	2428.0	4	-4	双平	卖三 2428.4	304
15:03:28	2428.0	1	-1	双平	卖二 2428.2	230
15:03:28	2428.0	2	0	换手	卖一 2428.0	102
15:03:29	2427.8	42	11	空开	买一 2427.8	97
15:03:29	2428.2	45	-19	空平	买二 2427.6	113
15:03:30	2427.8	9	-1	多平	买三 2427.4	148
15:03:30	2427.8	26	-21	多平	买四 2427.2	93
15:03:31	2428.0	6	1	多开	买五 2427.0	164
15:03:31	2427.8	1	1	双开	总卖 2428.4	928
15:03:32	2427.8	1	1	双开	总买 2427.4	615
15:03:32	2428.0	7	-2	空平	最新 2427.8	涨跌-13.270.54%
15:03:33	2427.8	4	2	空开	现手 3	速涨 0.00%
15:03:33	2428.0	4	-2	空平	总手 677132	开盘 2442.8
15:03:34	2428.0	1	-1	双平	持仓 146199	最高 2453.2
15:03:34	2428.0	2	1	多开	日增 17624	最低 2426.8
15:03:35	2428.0	1	0	换手	外盘 335619	结算价 2431.0
15:03:35	2427.8	16	6	空开	比例 50%	昨结 2441.0
15:03:36	2427.8	1	0	换手	内盘 341513	涨停 2685.0
15:03:36	2428.0	1	-1	双平	比例 50%	跌停 2197.0
15:03:37	2427.8	2	2	双开	时间	价位
15:03:38	2428.0	2	-2	双平	14:57:46	2428.8
15:03:38	2428.0	4	-1	空平	14:58:30	2428.4
15:03:39	2427.8	26	-18	多平	15:00:00	2428.4
15:03:39	2427.8	4	-4	双平	15:00:20	2427.8
15:03:40	2427.8	3	-2	多平	15:00:23	2427.4
15:03:40	2428.0	1	-1	双平	15:00:24	2427.2
15:03:41	2427.8	4	0	空换	15:00:25	2427.4
15:03:41	2427.8	3	-1	多平	15:00:41	2427.0
					15:01:24	2427.4
					15:01:24	2427.2
					15:03:12	2428.0

让我们来思考一个问题，这些盘口概念分别反映了市场上投资者的哪种心态和行为？

主动买：买开、卖平

主动卖：卖开、买平

增仓：持仓量的增减

现手：成交量

价格和数量反映了目前多空双方达成一致的均衡：

- 1、价格不变化或者变化很小，市场正处于横盘小幅震荡的走势 交易不活跃
- 2、价格在很大范围内上下变化，市场正处于剧烈震荡中 交易不活跃
- 3、价格不断升高，上涨行情，反之下跌行情 交易活跃
- 4、价格变化缓慢，缺乏动力 交易不活跃
- 5、价格不断跳动，有走出趋势行情的动力 交易活跃
- 6、价格迅速跳动，价格均速变大，价位连续，稳健趋势行情的特征，后期可能加速
- 7、价格迅速跳动，变化不连续，呈跳跃性，表示放量突破的行情

.....

### 更多 TICK 盘口函数

ASK1 取得 TICK 图该笔 TICK 的卖一价

ASK2 取得 TICK 图该笔 TICK 的卖二价

ASK3 取得 TICK 图该笔 TICK 的卖三价

ASK4 取得 TICK 图该笔 TICK 的卖四价

ASK5 取得 TICK 图该笔 TICK 的卖五价

ASK1VOL 取得 TICK 图该笔 TICK 的卖一量

ASK2VOL 取得 TICK 图该笔 TICK 的卖二量

ASK3VOL 取得 TICK 图该笔 TICK 的卖三量

ASK4VOL 取得 TICK 图该笔 TICK 的卖四量

ASK5VOL 取得 TICK 图该笔 TICK 的卖五量

BID1 取得 TICK 图该笔 TICK 的买一价

BID2 取得 TICK 图该笔 TICK 的买二价

BID3 取得 TICK 图该笔 TICK 的买三价

BID4 取得 TICK 图该笔 TICK 的买四价

BID5 取得 TICK 图该笔 TICK 的买五价

BID1VOL 取得 TICK 图该笔 TICK 的买一量

BID2VOL 取得 TICK 图该笔 TICK 的买二量

BID3VOL 取得 TICK 图该笔 TICK 的买三量

BID4VOL 取得 TICK 图该笔 TICK 的买四量

BID5VOL 取得 TICK 图该笔 TICK 的买五量

NEW 取得 TICK 图的最新价

TICK\_OPI 取得 TICK 图的持仓量

TICK\_SCALE 取得 TICK 图该笔 TICK 是否为主动买

TICK\_VOL 取得 TICK 图的成交量

DEF\_TICKDATA(Type, N) 定义五档 TICK 数据区

BKVOLUME 取得 TICK 图所定义数据区买开成交量的和

BPVOLUME 取得 TICK 图所定义数据区买平成交量的和

BIDVOL 取得 TICK 图所定义数据区主动买成交量的和

SKVOLUME 取得 TICK 图所定义数据区卖开成交量的和

SPVOLUME 取得 TICK 图所定义数据区卖平成交量的和

ASKVOL 取得 TICK 图所定义数据区主动卖成交量的和

SETBIGVOL 设置 TICK 图所定义数据区大单阈值

ASKBIGCOUNT 取得 TICK 图所定义数据区主动卖大单次数的和

ASKBIGTOTVOL 取得 TICK 图所定义数据区主动卖大单成交量的和  
 SKBIGCOUNT 取得 TICK 图所定义数据区卖开大单成交次数的和  
 SKBIGTOTVOL 取得 TICK 图所定义数据区卖开大单成交量的和  
 SPBIGCOUNT 取得 TICK 图所定义数据区卖平大单成交次数的和  
 SPBIGTOTVOL 取得 TICK 图所定义数据区卖平大单成交量的和  
 BIDBIGCOUNT 取得 TICK 图所定义数据区主动买大单次数的和  
 BIDBIGTOTVOL 取得 TICK 图所定义数据区主动买大单成交量的和  
 KBIGCOUNT 取得 TICK 图所定义数据区买开大单成交次数的和  
 KBIGTOTVOL 取得 TICK 图所定义数据区买开大单成交量的和  
 BPBIGCOUNT 取得 TICK 图所定义数据区买平大单成交次数的和  
 BPBIGTOTVOL 取得 TICK 图所定义数据区买平大单成交量的和

### TICK 盘口策略

#### 关于成交量和持仓量

- 1、成交量代表市场活跃度
- 2、成交量/持仓量 代表投机度，投机度强一般波动性大
- 3、增仓、放量伴随大涨，代表相当强势
- 4、减仓、缩量伴随下跌，代表相当弱势
- 5、价量仓方向不统一，代表分歧和能量不均

#### 关于单量

- 1、挂单量大的一方为强
- 2、主动成交的一方为强
- 3、撤单一方为弱
- 4、反复在同一价位成交表示争夺激烈，短期价格无方向，也可能是短期趋势可能反转
- 5、不断挂单、撤单表示不坚定，也可能是吸引成交

#### 关于价格涨跌

- 1、匀速拉升或者匀速下跌容易导致大跳水或大反弹
- 2、阶梯上涨或者下跌，不易转向
- 3、不断加速易触板
- 4、不断减速需要修正走势

#### TICK 盘口模型范例

##### 范例 1:

策略原理：日内走势出现逐笔连续上涨或者下跌的概率较小，当短暂的上涨或者下跌形成后，价格极容易反向运行，连续 6 笔无主动卖，开多仓，反之开空仓。

```
DEF_TICKDATA(1,6);
```

```

SZD:=ASKVOL;
BZD:=BIDVOL;
SZD=0, SK;
BZD=0, BK;
NEW>=BKPRICE+4*MINPRICE, SP;
NEW<=SKPRICE-4*MINPRICE, BP;
NEW<=BKPRICE-4*MINPRICE, SP;
NEW>=SKPRICE+4*MINPRICE, BP;
AUTOFILTER;

```



## 范例 2:

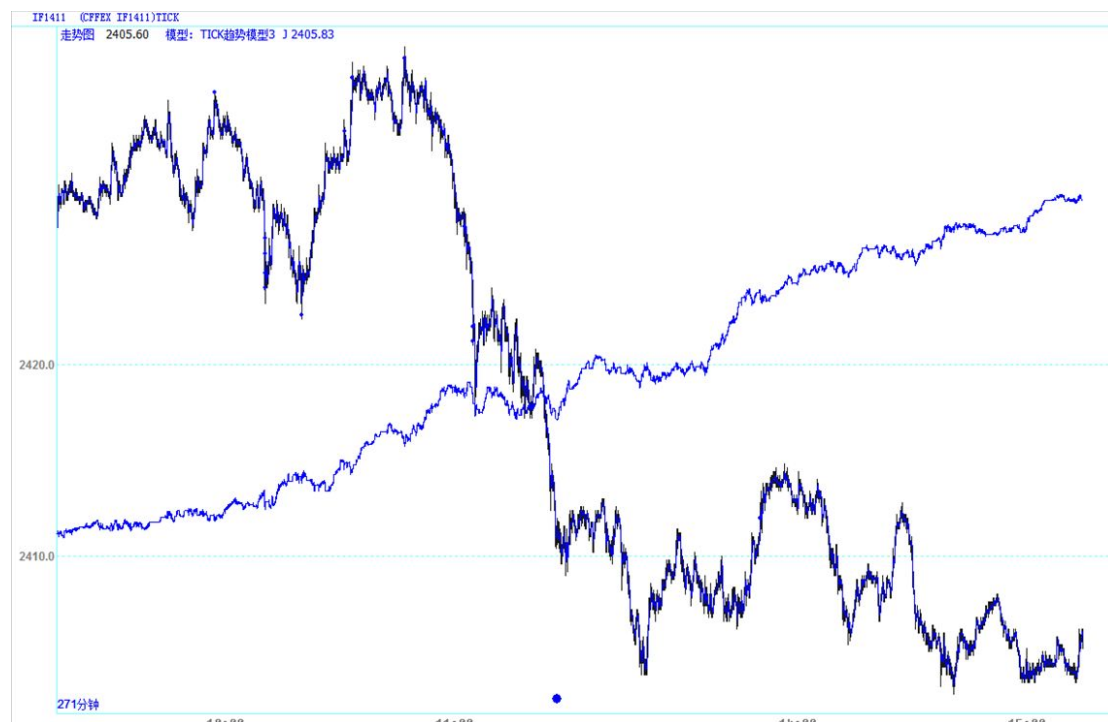
**策略原理:** 当 tick 图上, 价格突破了一定时间中的高低点后形成短暂趋势, 入场后固定止损止盈或者时间出场。

```

M:=30;
J:MA(NEW, M);
EVERY(NEW>J, 10)&&NEW>HV(NEW, 20)&&TIME<151450, SK;
EVERY(NEW<J, 10)&&NEW<LV(NEW, 20)&&TIME<151450, BK;
NEW>BKPRICE+0.8, SP;
NEW<SKPRICE-0.8, BP;
NEW<BKPRICE-0.8, SP;
NEW>SKPRICE+0.8, BP;

```

```
EVERY (NEW>=SKPRICE, 40) &&BARSSK>40, BP;
EVERY (NEW<=BKPRICE, 40) &&BARSBK>40, SP;
TIME>=151450, CLOSEOUT;
AUTOFILTER;
```



范例 3:

```
MA1:MA (NEW, 60) ;
DEF_TICKDATA (0, 10) ;
BHE:=BID1VOL+BID2VOL+BID3VOL;
SHE:=ASK1VOL+ASK2VOL+ASK3VOL;
NEW>MA1&&NEW>HV (NEW, 20) &&ASKVOL<BIDVOL&&EVERY (BHE>SHE*1.5, 3), BK;
NEW<MA1&&NEW<LV (NEW, 20) &&ASKVOL>BIDVOL&&EVERY (BHE*1.5<SHE, 3), SK;
EVERY (NEW>MA1, 5), BP;
EVERY (NEW<MA1, 5), SP;
AUTOFILTER;
```

## 第二章 多模型组合回测

### 2.1 回测一篮子合约

如果你已经有了一个模型，想看一下它在那个合约上的表现更好一点，那么回测一篮子合约

将是你的不二选择。利用这个功能，我们可以将一个模型同时加载给不同的合约，以查看它在每个合约上的表现。

如下图所示是如何回测一篮子合约：

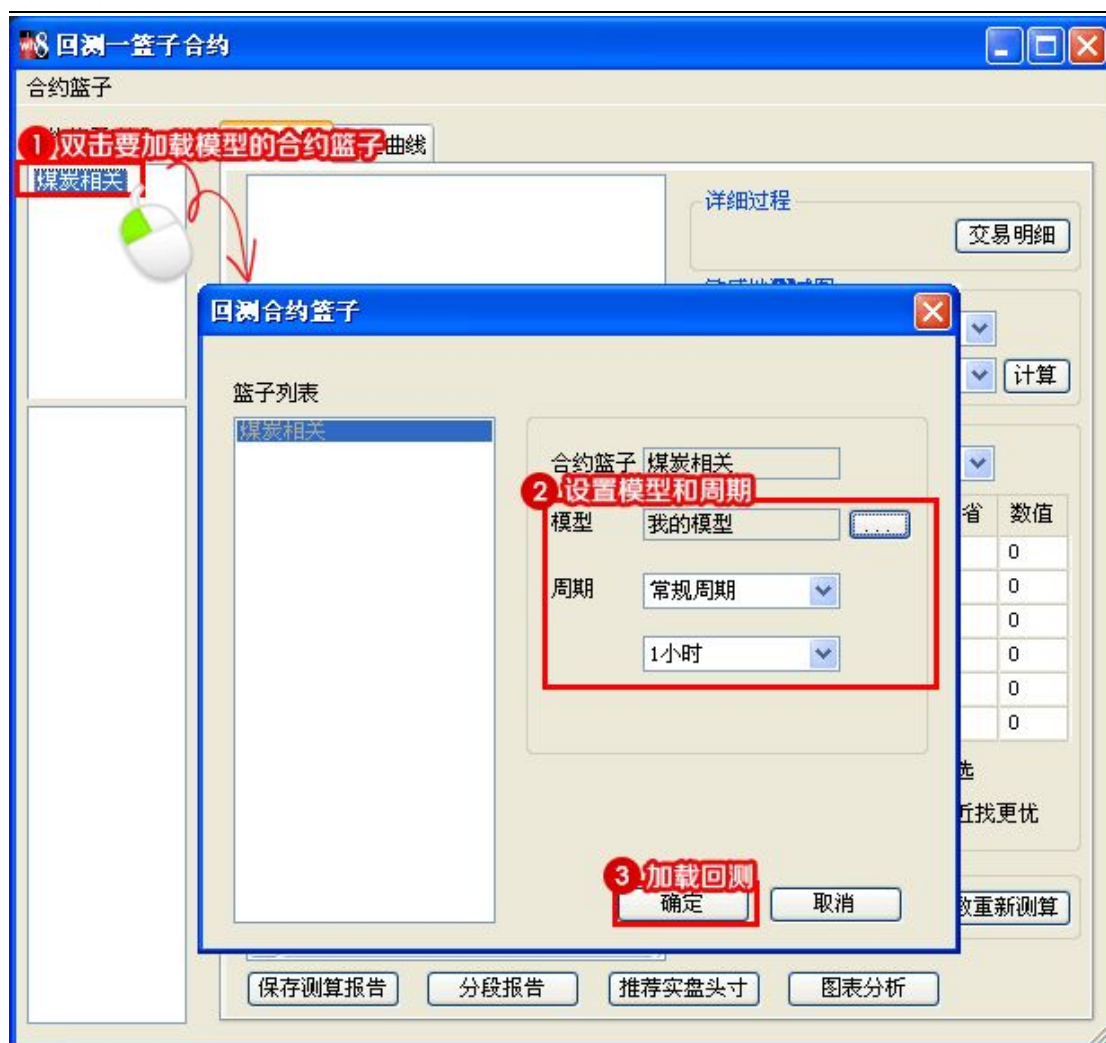
### 第一步：建立合约篮子

在这一步，我们要将打算回测的合约放在一个篮子里，统一保存起来，这样，稍后当我们加载模型的时候对该篮子加载即可同时加载到篮子中的每个合约上。



### 第二步：加载模型

在这一步，我们可以把想要测试的模型加载给合约篮子里的每一个合约。



第三步：查看测试结果

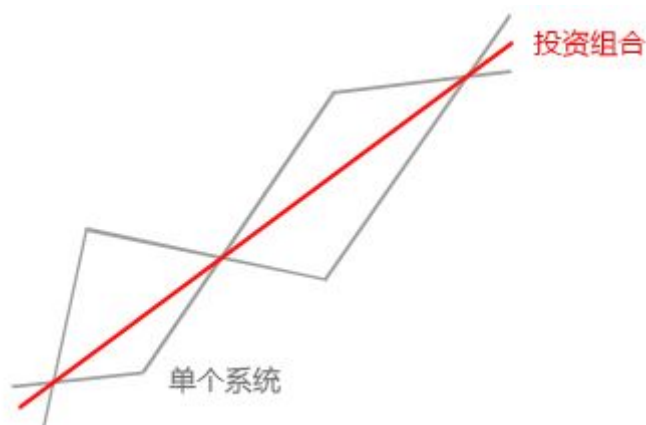


这是一个很实用的功能，特别是对于需要做模型筛选的用户，你可以很方便的找到与模型合适的合约，省去了一次一次加载的繁琐步骤。

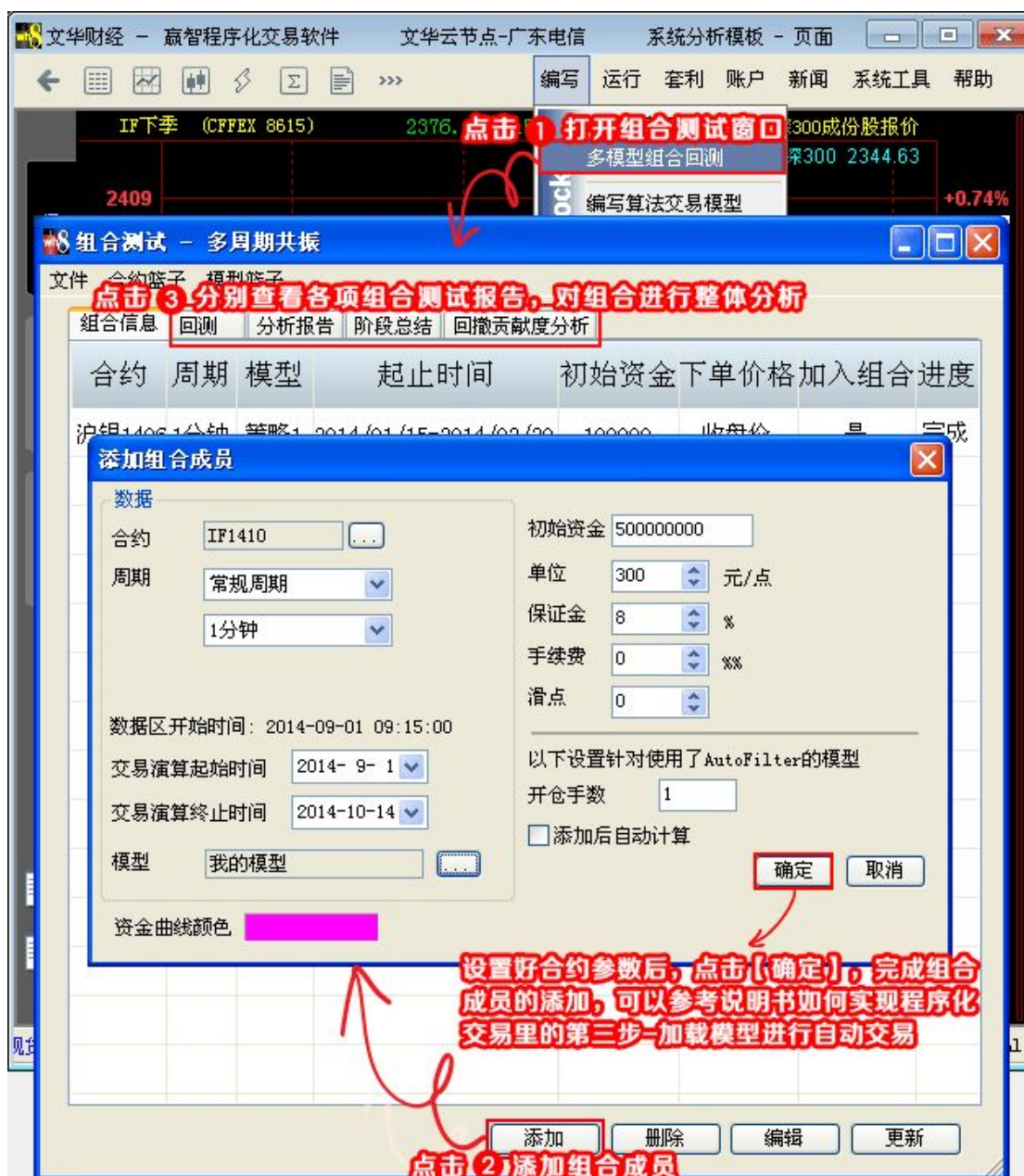
## 2.2 多模型组合回测

很多投资者在寻找一种模型可分辨趋势行情和震荡行情，因为一个在趋势行情中表现不错的模型到了震荡行情可能损失惨重，甚至反盈为亏。而使用多模型组合可将震荡模型和趋势模型同时加载到一个合约中，当行情震荡时，震荡模型的盈利可冲抵趋势模型的亏损。而当趋势行情到来的时候，这种多模型组合可共振盈利。

下图可直观的揭示投资组合的优势。组合的形式起到了削峰填谷的作用。为的是追求资金曲线平滑稳定的增长，避免资金的大幅回撤所导致的交易风险。



如下图①-③所示是如何对组合策略进行测试：



## 相关常见问题解答

### 1、已经进行过测试的组合，能否保存起来以便于下一次直接调出？

答：可以，可通过下图所示的方法对当前组合进行保存。再次打开点击【打开现有组合文件】即可。



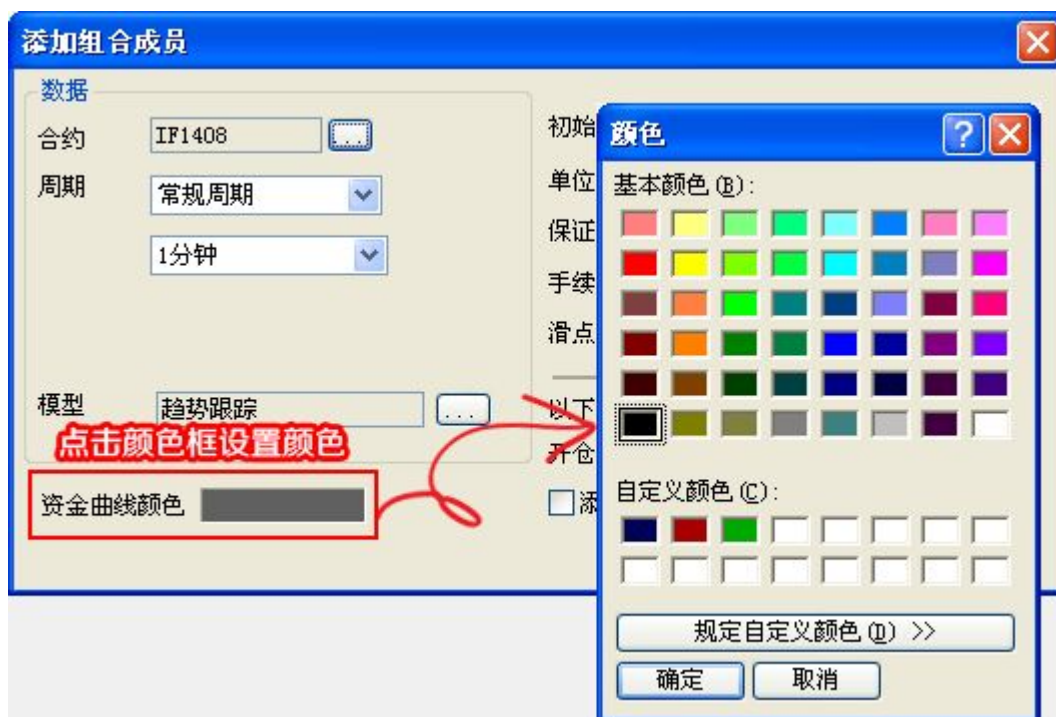
### 2、为什么添加组合成员后“进度”中显示的是未计算？

答：这是由于在添加组合成员时没有勾选【添加后自动计算】；选中未计算的组合成员，点击下方【更新】按钮即可，如下图：



### 3、组合成员的资金曲线颜色可修改么？

答：可修改，如下图所示是如何修改资金曲线颜色：



#### 4、已经添加的组合成员如何修改合约参数？

答：选中要修改的组合成员，点击组合测试界面下方的【编辑】按钮，即可对组合成员参数进行修改。

#### 5、为什么组合测试界面的【回测】中，有左右两个纵坐标，分别代表什么？

答：左侧纵坐标为各策略组合后的资金曲线坐标，右侧纵坐标为各个策略的资金曲线坐标。

#### 6、组合测试界面的【阶段总结】中，权益增长速度如何计算？

答：权益增长速度 = (本期净利润 - 上期净利润) / 本期期初权益。

#### 7、回撤贡献度得分越高越好么？

答：是的，回撤贡献度分数越高说明回撤小，贡献大。

贡献度分数算法：

- (1) 策略 1 和策略 2 每个时点上比较，回撤小的得 2 分，回撤大的得 1 分，总分就是每个时点得分之和，比如在某一个时点上，策略 1 的回撤小，策略 2 的回撤大，那么策略 1 就得 2 分，策略 2 就得 1 分，此时总分是 3 分。

每个时点上都会计算得分，然后把策略 1 和策略 2 的得分分别加起来就是各自的得分。得分高说明策略的回撤小，贡献就大。

- (2) 回撤贡献度是绝对值得分，回撤比贡献度就是比例，就是每个策略各自的得分，比上所有策略加起来的总得分。

#### 8、如何实现快速将同一模型加载至不同合约进行测试？

答：如下图所示，利用合约篮子功能可实现该想法。

- (1) 在组合测试界面点击菜单栏中的【合约篮子】→【新建篮子】

- (2) 在弹出的窗口中选择一些要加载的合约。
- (3) 点击“合约篮子”中的【加载合约篮子】，在弹出窗口中设置一些选项，点击【加载】，可实现快速对同一模型加载至不同合约进行测试。



### 9、如何实现快速将同一合约同一周期加载不同模型进行测试？

答：如下图所示，利用模型篮子功能可实现该想法。

- (1) 在组合测试界面点击菜单栏中的【模型篮子】→【新建篮子】
- (2) 在弹出的窗口中选择一些要加载的模型。
- (3) 点击“模型篮子”中的【加载模型篮子】，在弹出窗口中设置一些选项，点击【加载】，可实现快速对同一合约同一周期加载不同模型进行测试。



### 10、如何将回测的组合应用到实盘程序中？

答：可以通过运行模组功能将不同的模型加载到不同合约上，这部分我们会在第七章做详细介绍。

[\(点击查看链接\)](#)

## 2.3 段落式交易回测

在实际交易中，我们通常不会从一至终的不对资金做任何调整，常用的方法是将交易这一长期工作分割成以月、度或年为单位的不同段落，每个段落开始时赋予策略相同的初始资金，当前交易段落结束时对策略平仓。

例如，每月进行一次总结，盈利则将利润拿出使账户资金恢复到月初首次入金状态；亏损则对账户资金进行补充同样使之恢复到月初状态。软件中提供了一类函数可以对这种段落交易的方式进行回测，更具有实盘参考价值。

**关键函数：**WEEKTRADE/MONTHTRADE/QUARTERTRADE/YEARTRADE

WEEKTRADE: 使用每周数据计算。

MONTHTRADE: 使用每月数据计算。

QUARTERTRADE: 使用每季度数据计算。

YEARTRADE: 使用每年数据计算。

**函数说明:** 这几个函数语法相同, 我们以 MONTHTRADE 为例进行说明

模型中写入该函数, 信号和资金每月重新初始化进行计算, 与历史割裂。

注:

- 1、该函数不支持自定义 N 日、季、年周期, 其他周期均支持。
- 2、回测报告中的出金/入金, 为每月出金/入金的和。
- 3、模型中不能同时使用 DAYTRADE1\DAYTRADE\WEEKTRADE\MONTHTRADE\QUARTERTRADE\YEARTRADE 函数。

#### **范例: 段落式交易, 每月第一个交易日资金和信号重新初始化**

模型说明: 初始资金 100000, 以月为周期进行交易, 每月第一个交易日资金和信号重新初始化  
在模型中加入 MONTHTRADE 函数来实现想法。

**模型源码:**

HH:=HHV(HIGH, N);

LL:=LLV(LOW, N);

HH1:=BARSLAST((HH > REF(HH, 1)));

LL1:=BARSLAST((LL < REF(LL, 1)));

HH1>LL1, SPK;

HH1<LL1, BPK;

CLOSEMINUTE<=1, CLOSEOUT;//收盘前一分钟, 清仓。

AUTOFILTER;//过滤模型

MONTHTRADE;//使用每月数据计算

我们可以将上面编写好的源码加载到合约上进行回测, 回测过程中软件会自动将每个交易阶段区分开来, 在每月第一个交易日时重新初始化资金和信号, 如下图所示, 是回测报告的成交明细, 通过该明细我们可以看出, 每当新一月的交易日开始, 权益都会变为我们所设定的 100000。这样一来, 及时是这种段落式交易的想法也可以实现历史数据回测了。

效果测试成交明细								
编号	日期	合约	买卖	开平	价格	手	平仓盈亏	权益
0	20140701 09:00:00	豆粕主连	卖	开	3634	0,5	---	100000
1	20140716 09:00:00	豆粕主连	买	平开	3249	5,5	19250	119250
2	20140721 09:00:00	豆粕主连	卖	平开	3212	5,5	-1850	117400
3	20140724 09:00:00	豆粕主连	买	平开	3221	5,5	-450	116950
4	20140731 14:00:00	豆粕主连	全清	平	3281	5,0	3000	119950
5	20140801 09:00:00	豆粕主连	卖	开	3256	0,5	---	100000
6	20140805 09:00:00	豆粕主连	买	平开	3285	5,5	-1450	98550
7	20140811 14:00:00	豆粕主连	卖	平开	3314	5,5	1450	100000
8	20140818 11:00:00	豆粕主连	买	平开	3257	5,5	2850	102850
9	20140821 09:00:00	豆粕主连	卖	平开	3244	5,5	-650	102200
10	20140825 09:00:00	豆粕主连	买	平开	3274	5,5	-1500	100700
11	20140826 09:00:00	豆粕主连	卖	平开	3240	5,5	-1700	99000
12	20140829 14:00:00	豆粕主连	全清	平	3253	5,0	-650	98350
13	20140901 09:00:00	豆粕主连	卖	开	3227	0,5	---	100000
14	20140902 14:00:00	豆粕主连	买	平开	3261	5,5	-1700	98300
15	20140904 09:00:00	豆粕主连	卖	平开	3221	5,5	-2000	96300
16	20140912 11:00:00	豆粕主连	买	平开	3125	5,5	4800	101100
17	20140919 09:00:00	豆粕主连	卖	平开	3115	5,5	-500	100600
18	20140930 14:00:00	豆粕主连	全清	平	2933	5,0	9100	109700
19	20141008 09:00:00	豆粕主连	卖	开	2965	0,5	---	100000
20	20141008 11:00:00	豆粕主连	买	平开	2995	5,5	-1500	98500
21	20141013 09:00:00	豆粕主连	卖	平开	2990	5,5	-250	98250
22	20141014 09:00:00	豆粕主连	买	平开	3042	5,5	-2600	95650
23	20141020 09:00:00	豆粕主连	卖	平开	2826	5,5	-10800	84850
24	20141024 09:00:00	豆粕主连	买	平开	2922	5,5	-4800	80050
25	20141031 14:00:00	豆粕主连	全清	平	3024	5,0	5100	85150
26	20141103 09:00:00	豆粕主连	买	开	3004	0,5	---	100000
27	20141103 14:00:00	豆粕主连	卖	平开	2965	5,5	-1950	98050
28	20141107 10:00:00	豆粕主连	买	平开	2955	5,5	500	98550
29	20141111 14:00:00	豆粕主连	卖	平开	2930	5,5	-1250	97300
30	20141112 09:00:00	豆粕主连	买	平开	2998	5,5	-3400	93900
31	20141114 11:00:00	豆粕主连	卖	平开	2944	5,5	-2700	91200
32	20141121 14:00:00	豆粕主连	买	平开	2875	5,5	3450	94650
33	20141128 14:00:00	豆粕主连	全清	平	2903	5,0	1400	96050
34	20141201 09:00:00	豆粕主连	卖	开	2851	0,5	---	100000
35	20141208 09:00:00	豆粕主连	买	平开	2872	5,5	-1050	98950
36	20141216 09:00:00	豆粕主连	卖	平开	2921	5,5	2450	101400
37	20141226 14:59:59	豆粕主连	全	平	2862	5,0	2950	104350
合计							13550	

相应的，在回测报告中，我们也可以通过查看调整后的权益，来了解模型在历史回测中的变现。

如下图所示：



**最终权益**=最终权益 = 初始资金 + 入金 - 出金 + 净利润。也就是初始资金+最后阶段的盈亏（例如：使用 DAYTRAD 函数，是初始资金+最后一天的盈亏值）之后的指标均由“调整后的权益”计算得来的。

**调整后的权益**=初始资金+净利润（即交易明细中各平仓盈亏之和）

**出金**=当上一阶段交易结束，且获利时，开始下一阶段交易前所出金额总和

**入金**=当上一阶段交易结束，且亏损时，开始下一阶段交易前所入金额总和

## 第三章 编写资金管理模型

### 3.1 加码模型

加仓模型有一套固定的编写套路，在初接触加码模型编写的时候，我们可以按照下面的模板来进行编写，熟练后，可以更自由的进行发挥。

**加减仓模型编写模板：**

A:=多头开仓条件；

A1:=多头加仓条件 1；

```
A2:=多头加仓条件 2;
B:=空头开仓条件;
B1:=空头加仓条件 1;
B2:=空头加仓条件 2;
E:=多头减仓条件;
E1:=多头全平条件;
F:=空头减仓条件;
F1:=空头全平条件;
A && NOT(ISLASTBK) && NOT(ISLASTSK), BK(2);
B && NOT(ISLASTSK) && NOT(ISLASTBK), SK(2);
(A1 | A2) && BKVOL>=2 && BARSBK>1&&BKVOL<N, BK(1);
(B1 | B2) && SKVOL>=2 && BARSSK>1&&SKVOL<N, SK(1);
E&&(ISLASTBK | ISLASTSP), SP(1);
F&&(ISLASTSK | ISLASTBP), BP(1);
E1&&(ISLASTBK | ISLASTSP), SP(BKVOL);
F1&&(ISLASTSK | ISLASTBP), BP(SKVOL);
```

**需要注意的是：**由于加仓模型必须保证加仓时所开仓位和当前已经持有仓位方向相同，因此，在模型中需要判断当前信号、当前持仓、前一个信号等状态，进行过滤信号的条件

#### 相关几种资金管理编写方法：

- 1、SETDEALPERCENT(20); //每次下单使用当时资金的 20%
- 2、控制交易资金使用率不超过 50% 资金使用率=持仓保证金/权益  

```
BKVOL>2 && A && BARSBK>1&&MONEYRATIO<0.5, BK(1);
```
- 3、设置每次下单手数按照可用资金的 20%  

```
K:=MONEY*0.2/(C*MARGIN*UNIT+FEE);
BKVOL>2 && A && BARSBK>1&&MONEYRATIO<0.5, BK(K);
```

#### 范例：海龟法则模型编写

**关键函数：**BKPRICE, ISLASTBK, MONEYTOT, SKVOL

**建议使用周期：**所有K线周期

#### 模型说明：

入市：以 20 日突破为基础的短期系统

加仓：以上次开仓价格的 0.5 倍 ATR 逐步加仓

止损：止损标准是 2 倍的 ATR

退出：采用 10 日突破退出法则

//中间变量

```

TR : MAX(MAX((HIGH-LOW), ABS(REF(CLOSE, 1)-HIGH)), ABS(REF(CLOSE, 1)-LOW));
ATR : MA(TR, N); //平均真实波幅 ATR 定义
TC:INTPART((MONEYTOT*0.01/(UNIT*ATR))); //权益的 1%和平均波动浮动来确定交易头寸
MTC:4*TC; //最大头寸上限计算
//交易条件
TMK1: CROSSUP(C, HV(H, 20))&&ISLASTBK=0&&ISLASTSK=0;
TMK2:CROSSDOWN(C, LV(L, 20))&&ISLASTBK=0&&ISLASTSK=0; //入市条件
JMK1:C>=BKPRICE+0.5*ATR&&BKVOL<MTC&&ISLASTBK;
JMK2: C<=SKPRICE-0.5*ATR&&SKVOL<MTC&&ISLASTSK; //加仓条件
JMP1:C<=(BKPRICE-2*ATR)&&BKVOL>0;
JMP2: C>=(SKPRICE+2*ATR)&&SKVOL>0; //止损条件
TMP1:CROSSUP(H, HV(H, 10))&&SKVOL>0;
TMP2: CROSSDOWN(L, LV(L, 10))&&BKVOL>0; //出场条件
//交易系统
TMK1, BK(TC);
TMK2, SK(TC); //入市
JMK1, BK(TC);
JMK2, SK(TC); //加仓
JMP1, SP(BKVOL);
JMP2, BP(SKVOL); //止损
TMP1, BP(SKVOL);
TMP2, SP(BKVOL); //退出
//控制信号连续发出次数（保证了加仓信号可以执行）

```

## 3.2 回撤控制模型

运作一只基金，回撤的控制至关重要，而程序化交易是如何实现对基金回撤的控制呢？

我们用一个具体的案例来探讨这个问题，假如现在有一只规模为 3000 万的基金，经理人将其分成 30 份，以每份 100 万的金额分配到 30 个不同的模组中。那么，这只基金的总体表现就由这 30 个模组的综合表现来决定了。我们都知道，每个模组都盈利只是一种理想状态，实际上在基金运作过程中必然会有表现不佳的模组，而我们要做的，就是控制好每一个模组的回撤，这样一来，整个基金的资金曲线才有可能保持稳定上升的状态。

**关键函数：**INITMONEY 模组初次加载时的起始资金

MONEYREAL 模组子账户实际权益

**范例：**多模组运行，各子账户做回撤控制

**策略思路：**还是上面的案例，在将 3000 万资金分成 30 等份后，我们将用 INITMONEY 记录每一个子账户的初始资金，模型运行过程中，监控资金相对于初始资金的变化情况，确保每个子账户的回撤不超过 20%，例如子账户初始资金 100 万，当权益回撤至 80 万时平掉所有持仓，模型终止出新信号。

**模型说明：**INITMONEY 函数可以取得模组初次加载时的起始资金，MONEYREAL 可以取得模组子账户实际权益。我们可以用  $(INITMONEY - MONEYREAL) / INITMONEY$  计算出各子账户资金回撤幅度，当达到我们设定的上限时，平掉持仓。

//中间变量

HC:=(INITMONEY-MONEYREAL)/INITMONEY //子账户资金回撤比例

MA1:=MA(C,100);

MA2:=MA(C,350); //定义双重均线

//交易条件

TMK1:=CROSS(MA1,MA2);

TMK2:=CROSS(MA2,MA1);

TMP1:=CROSS(MA2,MA1) || (C>BKPRICE&&C<BKHIGH-M);

TMP2:=CROSS(MA1,MA2) || (C<SKPRICE&&C>SKLOW+M); //回撤止损

//交易系统

TMK1&& HC<0.2,BK; //在可容忍回撤范围内开仓

TMK2&& HC<0.2,SK; //在可容忍回撤范围内开仓

TMP1 || TMP3 || HC>0.2,SP; //回撤超过可容忍上限，平仓出场

TMP2 || TMP3 || HC>0.2,BP; //回撤超过可容忍上限，平仓出场

//过滤函数

AUTOFILTER;

### 3.3 资金曲线跟随模型

在软件的运行模组中会绘制两条资金曲线，实际资金曲线和理论资金曲线。实际资金曲线由实际成交价计算产生，理论资金曲线由信号指令价计算产生。提起与资金曲线有关的交易思路我们通常会想到控制实际资金的回撤幅度、控制实际资金的使用率等。那么理论资金曲线是没有价值了么？当然不是。

理论资金曲线，根据模组信号绘制，也就是说，它并不包含实盘运行中信号忽闪、滑点成本等因素的干预，可以认为是模型在顺利交易的状态下应该有的表现。恰恰因为它不受前面说道的几个因素干预，所以理论资金曲线更能反映出一个模型对当前行情的根本适应度。如果连理论资金曲线都下滑了，这个模型的实际资金曲线不会更好，当模型表现不好时，我们暂时不交易。

**关键函数：**MONEYTOT 模组子账户权益

### 范例：资金曲线跟随模型

**策略思路：**我们可以将理论资金曲线作为监控模型好坏的一个变量，计算出理论资金曲线五日均值，同时结合当前值进行比较，如果当前模型表现不好（即当前值低于五日平均值），即使开仓条件满足，也不进行开仓。

**模型说明：**利用 MONEYTOT 函数取得理论资金曲线值，计算出近日来 MONEYTOT 的均值。将策略思路中的判断条件作为开仓条件之一。

//中间变量

MONMA5:=MA(MONEYTOT, 5); //理论资金曲线五日平均线

MA1:=MA(C, 100);

MA2:=MA(C, 350); //定义双重均线

//交易条件

TMK1:=CROSS(MA1, MA2);

TMK2:=CROSS(MA2, MA1);

TMP1:=CROSS(MA2, MA1) || (C>BKPRICE&&C<BKHIGH-M);

TMP2:=CROSS(MA1, MA2) || (C<SKPRICE&&C>SKLOW+M); //回撤止损

//交易系统

TMK1&& MONEYTOT>=MONMA5, BK; //当前 MONEYTOT 值大于五日均值才开仓

TMK2&& MONEYTOT>=MONMA5, SK; //当前 MONEYTOT 值大于五日均值才开仓

TMP1 || TMP3, SP;

TMP2 || TMP3, BP;

//过滤函数

AUTOFILTER;

## 第四章 盘口模型的基本结构与应用

### 4.1 盘口模型的分类

#### a. 盘口高频模型

这类模型是从那些极为短暂的市场变化中寻求获利的程序化交易，通过挖掘盘口报价和逐笔成交数据，把交易策略应用到秒级周期，实现更高频率的日内交易。

适用群体：依据盘口报价和逐笔数据决定策略的跟随市场趋势者

#### b. 盘口结合趋势模型

这类模型以策略模型中趋势判断为主要进出场依据，结合盘口行情报价决定进场时机，从而提高入场效率的高频交易策略。

适用群体：依据策略模型及盘口报价决定交易思路的趋势投资者

#### c. 基差策略模型

这类模型类似套利交易，以两个合约价格差作为进场依据，通过基差判断两个合约间强弱关系并结合盘口报价，从而寻找入市机会的策略。

适用群体：能够准确分析合约间关系的高频投资者

#### d. 下单控制模型

这类模型以策略模型信号为进出场依据，对下单过程进行精细控制（减少滑点）的策略

适用群体：对下单精细控制、风险控制有严格要求的投资者

## 4.2 盘口模型使用的函数类型和运行机制

基于不同的表现形式，使用的函数大致分为 2 类

引用数据类，如：

`Def_TickData(Code, Type, Num)` 定义 tick 数据区

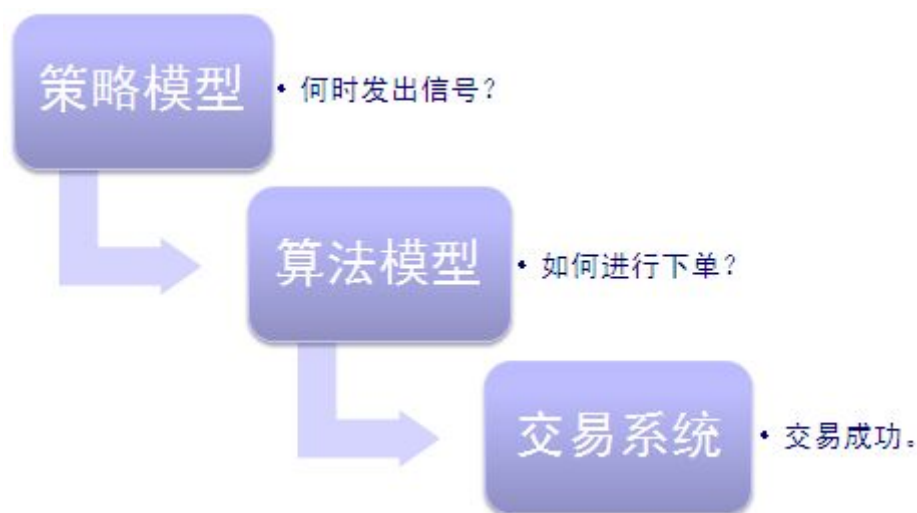
`Offers(Code, strContent)` 某合约 5 档数据

指令状态类，如：

`F_Sig()` 当前模型信号

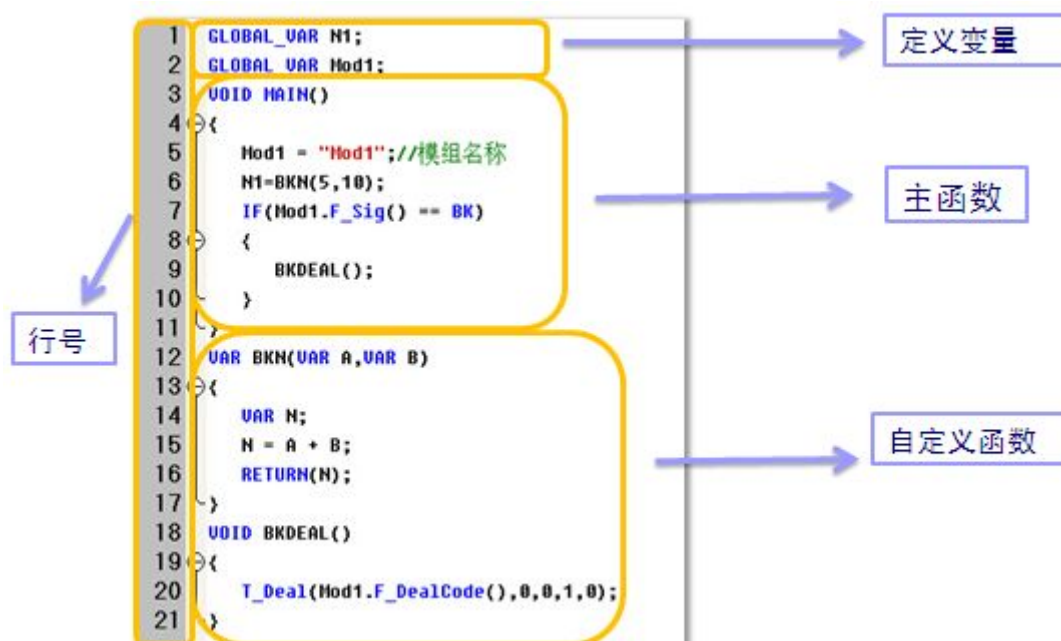
`F_BuyPosition()` 当前模型多单仓位

下单控制模型的运行机制



## 4.3 盘口模型的语法及编写规则

### 4.3.1、基本结构：



注：主函数中所取的模组名称（即上图的 Mod1）为运行模组的名称，而不是模型的名称。

#### 4.3.2、基本语法：

```

VAR N;                //定义变量 N
VAR N1;               //定义变量 N1
VAR N2;               //定义变量 N2
N=3;                  //整型赋值（下单手数）
N1=Price(“CU1409”);  //浮点型赋值(沪铜现价)
N2=“IF1409”；        //字符串型赋值（合约代码）

```

```

VOID MAIN()           //定义主函数
{...}

```

```

VAR BKDEAL()          //带返回值的函数
{
    RETURN(10);        //返回值
}

```

```

VOID BKDEAL()         //不带返回值函数
{...}

```

主函数：

```
VAR M; //定义变量 M
VOID MAIN() //定义主函数
{
    M= T_BuyPosition("m1409"); //对 M 赋值
    MessageOut(M); //输出 M
}
```

#### 带返回值的函数:

```
VAR BDL(A) //带返回值的函数
{
    VAR C; //定义变量 C
    C= A*(Mod1.F_High(0)- Mod1.F_Low(0));
    RETURN(C); //返回值
}
```

.....

```
D=BDL(0.5); //使用函数
```

.....

#### 不带返回值的函数:

```
VOID BKDEAL() //不带返回值函数
{
    T_Deal("IF1409", 0, 0, 1, Offers("IF1409", "ask1"));
}
```

.....

```
IF(...) //当条件成立
```

```
{
    BKDEAL() //运行函数
}
```

### 4.3.3、常用函数

#### 判断函数:

```
IF(Mod1.F_Sig()==BK) //如果当前是 BK 信号
{
    BKDeal(); //运行开多仓函数
}
ELSE IF(Mod1.F_Sig()==SK) //如果当前是 SK 信号
{
```

```
SKDeal(); //运行开空仓函数
}
```

信号函数:

```
IF (Mod1.F_FreshSig()==1)
{
    IF (Mod1.F_SigValid()==1)
        //如果当前信号是没有处理过的新信号
        {
            IF (Mod1.F_Sig()==BK) //如果当前是 BK 信号
            {
                ...
            }
        }
}
```

委托函数:

```
Code=Mod1.F_DealCode()
T_Deal(Code,bs,kp,vol,price)
T_AddBuyOpiTo(Code, Price, Vol)
T_AddSellOpiTo(Code, Price, Vol)
T_ReduceBuyOpiTo(Code, Price, Vol)
T_ReduceSellOpiTo(Code, Price, Vol)
```

返回函数:

```
(1)
VAR result;
VAR ADDTEST(VAR a,VAR b)
{
    VAR x;
    VAR y;
    x = a + b;
    y = a - b;
    MessageOut(x);
    MessageOut(y);
    RETURN (x*y);
}

(2)
VOID MAIN()
```

```
{  
    result = ADDTEST(50,100);  
    MessageOut(result);  
}
```

循环函数：

while(条件)

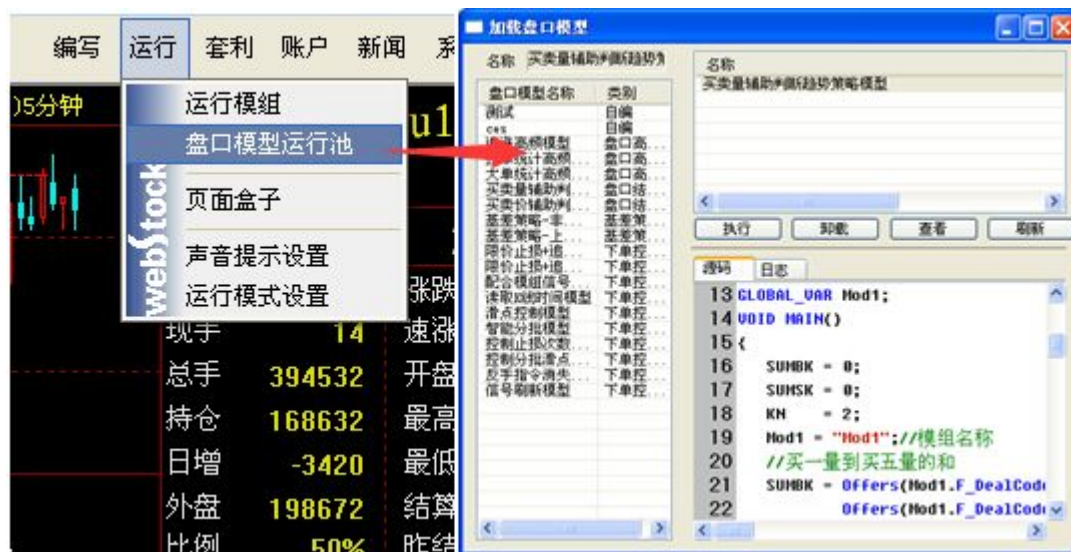
```
{  
    执行语句 1;  
}  
  
    执行语句 2;
```

当满足条件时循环执行语句 1

直到不满足条件在执行语句 2。

#### 4.4 盘口模型的加载流程

1、盘口高频及基差策略模型加载方法：

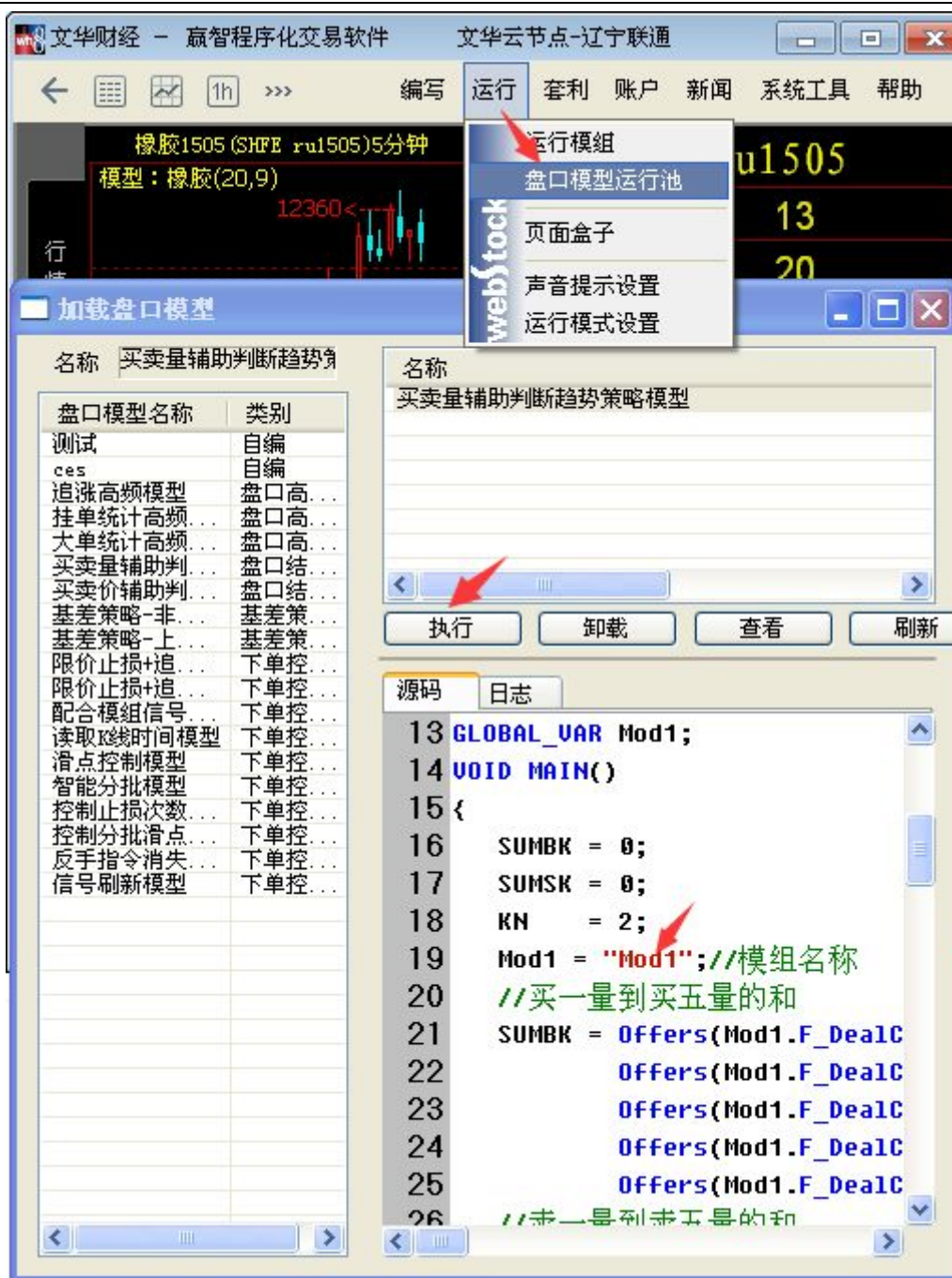


2、盘口趋势模型及下单控制模型加载方法：

A、运行——运行模组——选择策略模型加载运行。



B、运行---算法交易运行池---选择对应盘口模型执行。



## 第五章 盘口高频模型的编写

### 5.1 什么是高频交易

#### 什么是高频交易

高频交易是指从那些极为短暂的市场变化中寻求获利的程序化交易,通过挖掘盘口报价和逐笔成交数据,把交易策略应用到秒级周期,实现更高频率的日内交易。

这种由强大的计算机系统和复杂的运算所主导的交易能在毫秒之内自动完成大量买、卖以及取消指令；而为了争取这千分之一秒的优势，有些机构将服务器安置在交易所同一座建筑，以缩短交易指令通过光缆以光速旅行的距离

### 高频交易的特征

- A、由计算机自动完成的程序化交易；
- B、越成熟市场高频交易量的比重越大；
- C、日内交易次数多、持仓周期短；
- D、有严格的风险控制和执行纪律性；
- E、每笔平均利润小，以胜率取胜，利润积少成多；
- F、超低的网络信息延迟。

## 5.2 盘口高频模型的编写—追涨高频策略

### 实现思路：

1. 连续 3 笔 tick 数据逐次递增且当前成交量超过前一笔 tick 成交量时顺势做多。
2. 当合约价格上涨 N1 个最小变动价位时，追加 1 手。
3. 限价止盈+限价止损策略。

### 适用范围：

在市场急涨（急跌）时依据逐笔数据瞬间把握行情，迅速离场

### 分段量化：

- (1) 连续3笔tick数据逐次递增且当前成交量超过前一笔tick成交量时顺势做多。

```
CodeName = "IF1409";

data = Def_TickData(CodeName, 1, 4); // 保存最近四笔的 tick 数据
IF( data.State == 1 ) // 数据保存完成
{
    IF( data[0].Bid1 < data[1].Bid1 && data[1].Bid1 < data[2].Bid1 && data[2].Bid1
    < data[3].Bid1 ) //连续 3 次当笔 TICK 的买一价都大于上一笔 TICK 的买一价
    {
        IF( data[2].TickVolum < data[3].TickVolum )
            //并且最新一笔的成交量大于上一笔 TICK 的成交量，对价做多
        {
            BKID=T_Deal (CodeName, 0, 0, N, Offers (CodeName, "ask1"));
        }
    }
}
```

(2) 当合约价格上涨 N1 个最小变动价位时, 追加 1 手。

```
IF( T_OrderState(BKID) == 1 ) //BKID 标识的委托单已经成交
{
    MessageOut("当合约价格上涨 2 个最小变动价位, 加仓 1 手");
    IF( New - T_OrderMatchAvPrice(BKID) > N1 * MinPrice(CodeName) )
        //最新价-成交均价>N1 个最小变动价位
        {
            BKID = T_Deal(CodeName, 0, 0, 1, Offers(CodeName, "ask1"));
        }
}
```

(3) 限价止盈+限价止损策略

```
BPRICE=T_BuyAvgPrice(CodeName); //取得持仓栏中该合约多头持仓均价
IF( BPRICE - New >= Lost*MinPrice(CodeName) || New - BPRICE >= Win*MinPrice(CodeName))
{
    IF( T_BuyRemainPosition( CodeName ) > 0 )
    {
        MessageOut("止盈止损");

        T_Deal(CodeName, 1, 1, T_BuyRemainPosition( CodeName ), Offers(CodeName, "bid1"));
    }
}
```

完整源码:

```
VAR_TICKDATA data;
VAR N, N1, BPRICE, Lost, Win, CurTime;
GLOBAL_VAR High, BKID, New, Type, Typp, CONQC;
VAR CodeName;
VOID MAIN()
{
    CodeName = "IF1409";
    N = 2; // 下单手数
    N1 = 2; //当合约价格上涨 2 个最小变动价位
    Lost = 3; //止损 3 个最小变动价位
    Win = 10; //止盈 10 个最小变动价位
    New = Price(CodeName, "New"); //获取当前合约最新价
```

```

BPRICE=T_BuyAvgPrice(CodeName); //取得持仓栏中该合约多头持仓均价
data = Def_TickData(CodeName, 1, 4); // 保存最近四笔的 tick 数据
CurTime = CurrentTime(); // 获取当前时间
// 收盘前 30 秒清仓
CONQC = (Hour( CurTime ) == 14 && Minute( CurTime ) == 59) || ( Hour( CurTime )
    == 10 && Minute( CurTime ) == 14 ) || ( Hour( CurTime ) == 11 && Minute( CurTime )
    == 29 ) && Second( CurTime ) > 30 ;
IF( Typp == 0&&(CONQC == 1))
{
    MessageOut("收盘前 30 秒清仓");
    T_Deal(CodeName, 1, 1, T_BuyRemainPosition( CodeName ),
    Offers(CodeName, "bid1"));
    Type = 0;
    Typp = 1;
}
IF( data.State == 1 ) // 数据保存完成
{
    //连续 3 次当笔 TICK 的买一价都大于上一笔 TICK 的买一价，并且最新一笔的成交
    量大于上一笔 TICK 的成交量，对价做多
    IF( Type == 0 && data[0].Bid1 < data[1].Bid1 && data[1].Bid1 < data[2].Bid1 &&
        data[2].Bid1 < data[3].Bid1 && CONQC != 1)
    {
        IF( data[2].TickVolum - data[1].TickVolum < data[3].TickVolum -
            data[2].TickVolum )
        {
            MessageOut("连续 3 次当笔 TICK 的买一价都大于上一笔 TICK 的买一价");
            BKID = T_Deal(CodeName, 0, 0, N, Offers(CodeName, "ask1"));
            Type = 1;
            Typp = 0;
        }
    }
    //当合约价格上涨 2 个最小变动价位，加仓 1 手
    IF(Type == 1 && T_OrderState(BKID) == 1 && CONQC != 1 )
    {

```

```

        MessageOut("当合约价格上涨 2 个最小变动价位，加仓 1 手");
        IF( New - T_OrderMatchAvPrice(BKID) > N1 * MinPrice(CodeName) )
        {
            BKID = T_Deal(CodeName, 0, 0, 1, Offers(CodeName, "ask1"));
            Type = 2;
            Typp = 0;
        }
    }
    IF( Type == 2 && T_OrderState(BKID) == 1 )
    {
        Type = 1;
    }
    PingCang();
    SPDeal();
    High = Price(CodeName, "High");
}
}
//止盈 10 个最小变动价位
//止损 3 个最小变动价位
}

VOID SPDeal()
{
    IF( BPRICE - New >= Lost*MinPrice(CodeName) || New - BPRICE >=
Win*MinPrice(CodeName))
    {
        IF( Type != 0 && T_BuyRemainPosition( CodeName ) > 0 )
        {
            MessageOut("止盈止损");
            T_Deal(CodeName, 1, 1, T_BuyRemainPosition( CodeName ), Offers(CodeName, "bid1"));
            Type = 0;
        }
    }
}

//当持仓手数超过了 5 手，并且最新价格超过了今天的最高价，对价平仓

```

```

VOID PingCang()
{
    IF( Type != 0 && T_BuyRemainPosition( CodeName ) > 5 && New > High ) // 该组件多头持仓超过 5 手
    {
        MessageOut("当持仓手数超过了 5 手，并且最新价格超过了今天的最高价，对价平仓");
        T_Deal( CodeName, 1, 1, T_BuyRemainPosition( CodeName ), Offers( CodeName, "bid1" ) );
        Type = 0;
    }
}

```

### 5.3 盘口高频模型的编写—辅助判断趋势策略

#### 实现思路：

1. 盘口买 1 到买 5 档总量大于卖 1 到卖 5 档总量，且模型出现多头信号时，买开
2. 盘口买 1 到买 5 档总量小于卖 1 到卖 5 档总量，且模型出现空头信号时，卖开
3. 当策略模型发出 SP 信号时，卖平
4. 当策略模型发出 BP 信号时，买平

#### 适用范围：

以策略模型判断市场趋势，结合盘口数据决定进场时机。

#### 分段量化：

(1) 读取盘口买1到买5行情数据，计算总买量。

```

Mod1 = "Mod1"; //模组名称
//买一量到买五量的和
SUMBK = Offers( Mod1.F_DealCode(), "bidvol1" ) +
        Offers( Mod1.F_DealCode(), "bidvol2" ) +
        Offers( Mod1.F_DealCode(), "bidvol3" ) +
        Offers( Mod1.F_DealCode(), "bidvol4" ) +
        Offers( Mod1.F_DealCode(), "bidvol5" );

```

(2) 读取盘口买 1 到买 5 行情数据，计算总买量。

```

Mod1 = "Mod1"; //模组名称
//买一量到买五量的和
SUMBK = Offers( Mod1.F_DealCode(), "bidvol1" ) +
        Offers( Mod1.F_DealCode(), "bidvol2" ) +

```

```
Offers (Mod1.F_DealCode(), "bidvol3") +
Offers (Mod1.F_DealCode(), "bidvol4") +
Offers (Mod1.F_DealCode(), "bidvol5");
```

### (3) 依据策略模型信号结合盘口 5 档数据进行下单处理

```
IF (Mod1.F_Sig() == BK && SUMBK > SUMSK) //如果信号是买开信号
{
    MessageOut("BK 信号-----");
    BKID = T_Deal (Mod1.F_DealCode(), 0, 0, KN, Offers (Mod1.F_DealCode(), "ask1"));
    //以对价发出 KN 手的买开委托
}
IF (Mod1.F_Sig() == SK && SUMSK > SUMBK) //如果信号是卖开信号
{
    MessageOut("SK 信号-----");
    SKID = T_Deal (Mod1.F_DealCode(), 1, 0, KN, Offers (Mod1.F_DealCode(), "bid1"));
    //以对价发出 KN 手的卖开委托
}
```

## 5.4 盘口高频模型的编写—基差策略

### 实现思路:

1. 股指合约与沪深 300 指数基差小于 5 时，平掉之前空单，买开股指合约
2. 股指合约与沪深 300 指数基差大于 15 时，平掉之前多单，卖开股指合约
3. 当连续 5 分钟基差绝对值没有减少，平多单。
4. 当连续 5 分钟基差绝对值没有增加，平空单。

### 适用范围:

通过两个合约基差判断上方强弱，寻求入场时机。

### 分段量化:

#### (1) 读取两个合约价格并做基差计算。

```
CODE = "IF1411";
PRICE_IF = Price (CODE, "new");
// 读取沪深 300 最新价
PRICE_HS = Price ("999300", "new");
//基差计算
DIFF = ABS (PRICE_IF - PRICE_HS);
```

#### (2) 股指合约与沪深 300 指数基差小于 5 时，平掉之前空单，买开股指合约

```

IF(AL_BuyPosition(CODE) == 0 && COIN == 0 && T_IsNoOrder() == 1)
{
    BKID = T_Deal(CODE, 0, 0, 1, PRICE_IF);
    BKDIFF = DIFF;
}
ELSE IF(AL_BuyPosition(CODE) != 0) //多头持仓
{
    IF (DIFF < BKDIFF)
    {
        BKTIME = CurrentTime();
    }
}

```

(3) 当连续 5 分钟基差绝对值没有减少，平多单

```

IF (AL_BuyPosition(CODE) != 0 && CurrentTime()-BKTIME > 300 )
// 连续 5 分钟基差的绝对值都没有减少，平仓
{
    PRICE_IF = Price(CODE, "new");
    IF (AL_BuyPosition(CODE) != 0 && COIN == 0) // 平多头
    {
        SPID = T_Deal(CODE, 1, 1, AL_BuyPosition(CODE), PRICE_IF);
        MessageOut("平多头");
    }
}

```

## 第六章 盘口模型控制滑点

### 6.1 了解滑点的产生

在讲解这类盘口模型编写前，我们需要先来了解一下滑点是如何产生的。在交易的过程中，会有行情急速拉升或者回落的时候，如果模型在这种极速行情中委托可能需要不断的撤单追价，就会导致滑点增大。除了这种行情外，震荡行情也是产生滑点的原因之一，因为在震荡行情中会出现信号忽闪的现象，这样滑点就在无形中增加了。

那么滑点会产生影响呢？它可能会导致一个本可以盈利的模型转盈为亏。所以我们要控制滑点。

模型能够准确把握行情结果却亏损，您是否也有这样的困惑？

## 为什么要控制交易滑点？ 因为它是变盈为亏的蛀虫！

**原因一** 交易策略在快速行情中不能顺利进场，反复追价导致滑点增大，增加交易成本。

**原因二** 交易策略在震荡行情中信号忽闪，频繁开平仓，增加交易成本。



## 6.2 盘口模型控制滑点的原理

通常我们从两个方面来控制盘口模型的滑点，一是控制下单过程，二是对下单后没有成交的委托做适当的节约成本的处理。

### 1、控制下单时间：

比如我们如果担心在震荡行情中信号容易出现消失，那么就可以控制信号出现后 N 秒，待其稳定了，再发出委托。

### 2、控制下单的过程：

比如我们可以控制读取交易合约的盘口价格和委托量来判断现在委托是否有成交的可能，如果我们自己的委托量大，还可以做分批下单处理。

### 3、控制未成交委托：

比如同样是追价，我们可以利用盘口模型结合当前的盘口价格进行追价，而不是每一只以一样的价格死板的追价。对于那些现价偏离信号出现价过大的委托，我们可以对其进行撤单，成本过高的生意，不如不做。

## 6.3 盘口模型控制滑点策略编写

### 策略一：自适应追价

#### 适用范围：

1. 趋势、震荡均可使用。
2. 盘整行情能够以最优价格成交；趋势行情能够避免因上涨/下跌过快而导致实际滑点过大问题。

#### 主要作用：

1. 自适应追价：根据现价偏离首次下单价格的幅度来调整委托价格。
2. 盘尾特定时间全平。

### 分段量化：

#### (1) 判断挂单时间

```

IF(T_IsNoOrder() != 1) //有挂单的情况
{
    IF(CurrentTime()-LastOrderTime() >= 5 && Price <= AA.F_SigPrice() + 5 * MinPrice &&
LastWithdrawFinsh == 1)
//当挂单超过 5 秒并且最新价 <= 指令价 + 5 个最小变动价位时
{
    IF(AA.F_Sig() == BK) //如果取出的信号为 BK
    {
        IF(M == 0)
        {
            T_DeleteOrder(BKID); //撤掉多仓挂单
            M = 1;
        }
    }
    IF(AA.F_Sig() == SK) //如果取出的信号为 SK
    {
        M = ReadGlobal("M");
        IF(M == 0)
        {
            T_DeleteOrder(SKID); //撤掉空仓挂单
            M = 1;
        }
    }
    IF(AA.F_Sig() == BP) //如果取出的信号为 BP
    {
        M = ReadGlobal("M");
        IF(M == 0)
        {
            T_DeleteOrder(BPID); //撤掉买平挂单
            M = 1;
        }
    }
}

```

```
}  
  
}  
  
IF (AA.F_Sig()==SP) //如果取出的信号为 SP  
{  
    M=ReadGlobal("M");  
    IF (M==0)  
    {  
        T_DeleteOrder (SPID); //撤掉卖平挂单  
        M=1;  
    }  
}  
  
IF (AA.F_Sig()==BPK) //如果取出的信号为 BPK  
{  
    M=ReadGlobal("M");  
    IF (M==0&&AA.F_SellPosition()==0&&AA.F_BuyPosition()==0)  
        //当模组多空持仓都为 0 时  
    {  
        T_DeleteOrder (BKID); //撤掉多仓挂单  
        M=1;  
    }  
    IF (M==0&&AA.F_SellPosition()==KN&&AA.F_BuyPosition()==0)  
        //当模组空仓为 KN 手，多仓为 0 时  
    {  
        T_DeleteOrder (BPID); //撤掉买平挂单  
        M=1;  
    }  
}  
  
IF (AA.F_Sig()==SPK) //如果取出的信号为 SPK  
{  
    M=ReadGlobal("M");  
    IF (M==0&&AA.F_BuyPosition()==0&&AA.F_SellPosition()==0)  
        //当模组多空持仓都为 0 时  
    {  
        T_DeleteOrder (SKID); //撤掉空仓挂单
```

```

        M=1;
    }
    IF(M==0&&AA.F_BuyPosition()==KN&&AA.F_SellPosition()==0)
    {
        //当模组多仓为 KN 手，空仓为 0 时
        T_DeleteOrder(SPID); //撤掉卖平挂单
        M=1;
    }
}
}
WriteGlobal("LASTWITHDRAW", LastWithdrawFinsh);
(2) 撤单追价 (以开仓部分为例)
IF(T_OrderState(BKID)==2&&M==1) //买开撤单成功
{
    IF(Price<=AA.F_SigPrice()+5*MinPrice)
    {

        BKID=T_Deal(AA.F_DealCode(), 0, 0, KN, Offers(AA.F_DealCode(), "ask1")+N*MinPrice);
        //超 N 价位买开 KN 手
    }
    IF(Price>AA.F_SigPrice()+5*MinPrice)
    {

        BKID=T_Deal(AA.F_DealCode(), 0, 0, KN, Offers(AA.F_DealCode(), "ask1")+2*N*MinPrice)
;

        //超 2*N 价位买开 KN 手
    }
    LastWithdrawFinsh=0;
    M=0;
}
IF(T_OrderState(SKID)==2&&M==1) //卖开撤单成功
{
    IF(Price>=AA.F_SigPrice()-5*MinPrice)
    {

```

```

SKID=T_Deal(AA.F_DealCode(), 1, 0, KN, Offers(AA.F_DealCode(), "bid1")-N*MinPrice);
    //超 N 价位卖开 KN 手
}
IF(Price<F_SigPrice()-5*MinPrice)
{
SKID=T_Deal(AA.F_DealCode(), 1, 0, KN, Offers(AA.F_DealCode(), "bid1")-2*N*MinPrice)
;

    //超 2*N 价位卖开 KN 手
}
LastWithdrawFinsh=0;
M=0;
}

```

## 策略二：自适应行情

### 适用范围：

1. 趋势、震荡均可使用。
2. 根据不同行情自适应下单。盘整行情以收盘价委托避免信号频繁忽闪；趋势行情以指令价委托快速进场抓住后市行情。

### 主要作用：

1. 自适应行情进行下单判断。
2. 趋势行情中的信号忽闪处理。

### 分段量化：

#### (1) 自适应算法交易模型策略部分

```

1 CMI:=ABS(C-REF(C,29))/(HHV(H,30)-LLV(L,30))*100;
2 TRUERANGE:=MAX(REF(C,1),H)-MIN(REF(C,1),L);
3 ATR10:=MA(TRUERANGE,10);
4 KOD:=(H+L+C)/3;
5 BE:=IFELSE(C>KOD,1,0);
6 SE:=IFELSE(C<=KOD,1,0);
7 AVG3HI:=MA(H,3);
8 AVG3LO:=MA(L,3);
9 LEP:=IFELSE(C>KOD,0+ATR10*0.5,0+ATR10*0.75);
10 SEP:=IFELSE(C>KOD,0-ATR10*0.75,0-ATR10*0.5);
11 LEP1:=MAX(LEP,AVG3LO);
12 SEP1:=MIN(SEP,AVG3HI);
13 UPBAND:=MA(C,50)+STD(C,50)*2;
14 DNBAND:=MA(C,50)-STD(C,50)*2;
15 MA50:=MA(C,50);
16 ST:=ABS(C-0);
17 CMI<20&&C>=LEP1&&ST<3*REF(MA(ST,5),1),BK;
18 CMI>=20&&C>=UPBAND&&ST>3*REF(MA(ST,5),1),BK;
19 CMI<20&&C<=SEP1&&ST<3*REF(MA(ST,5),1),SK;
20 CMI>=20&&C<=DNBAND&&ST>3*REF(MA(ST,5),1),SK;
21 CMI<20&&C>=BKPRICE+3*ATR10,SP;
22 CMI>=20&&C<=MA50,SP;
23 CMI<20&&C<=SKPRICE-3*ATR10,BP;
24 CMI>=20&&C>=MA50,BP;
25 AUTOFILTER;

```

(2)

```

IF(AA.F_FreshSig()==0&&AA.F_SigValid()==1) //如果是处理过的旧信号
{
    IF(AA.F_Sig()==BK&&AA.F_SigPos()==1&&T_IsNoOrder()==1&&SFF==2&&SS==55&&A1==1)
    //如果取出的信号为 BK 并且时间为当根 k 线 2:55 秒
    {
        BKID=T_Deal(F_DealCode(),0,0,KN,Price(F_DealCode())); //买开 KN 手。
    }
    IF(AA.F_Sig()==SK&&AA.F_SigPos()==3&&T_IsNoOrder()==1&&SFF==2&&SS==55&&A1==1)
    //如果取出的信号为 SK 并且时间为当根 k 线 2:55 秒
    {
        BKID=T_Deal(F_DealCode(),0,0,KN,Price(F_DealCode())); //卖开 KN 手。
    }
}
IF(AA.F_FreshSig()==0&&AA.F_SigValid()==0)
{
    IF(SFF==2&&SS==55&&A2==1) //当前时间为当根 k 线 2:55 秒
    {
        XHXS(); //进入信号消失处理函数
    }
}

```

```

    }
}
IF (SFF==2&&SS==59) //K 线走完 A1、A2 清零
{
    A1=0;
    A2=0;
    WriteGlobal("A1", A1);
    WriteGlobal("A2", A2);
}

```

## (2) 信号消失处理

```

VOID XHXS() //信号消失处理函数
{
    IF (AA.F_Sig()==BK) //如果 BK 信号消失
    {
        T_Deal(AA.F_DealCode(), 1, 1, AA.F_BuyPosition(), Offers(AA.F_DealCode(), "bid1")
        -MinPrice(AA.F_DealCode())*N); //平掉所有多单
    }
    ELSE IF (AA.F_Sig()==SK) //如果 SK 信号消失
    {
        T_Deal(AA.F_DealCode(), 0, 1, AA.F_SellPosition(), Offers(AA.F_DealCode(), "ask1")+
        MinPrice(AA.F_DealCode())*N); //平掉所有空单
    }
    ELSE IF (AA.F_Sig()==BP) //如果 BP 信号消失
    {
        T_Deal(AA.F_DealCode(), 1, 0, KN, Offers(AA.F_DealCode(), "ask1")+MinPrice(AA.F_Dea
        lCode())*N); //反向开 KN 手空单
    }
    ELSE IF (AA.F_Sig()==SP) //如果 SP 信号消失
    {
        T_Deal(AA.F_DealCode(), 0, 0, KN, Offers(AA.F_DealCode(), "bid1")-MinPrice(AA.F_Dea
        lCode())*N);
        //反向开 KN 手多单
    }
    ELSE IF (AA.F_Sig()==BPK) //如果 BPK 信号消失

```

```

{
T_Deal(AA.F_DealCode(), 1, 1, AA.F_BuyPosition(), Offers(AA.F_DealCode(), "bid1")-MinPrice(AA.F_DealCode())*N);
T_Deal(AA.F_DealCode(), 1, 0, KN, Offers(AA.F_DealCode(), "ask1")+MinPrice(AA.F_DealCode())*N); //平掉所有多单并反向开 KN 手空单
}
ELSE IF (AA.F_Sig()==SPK) //如果 SPK 信号消失
{
T_Deal(AA.F_DealCode(), 0, 1, AA.F_SellPosition(), Offers(AA.F_DealCode(), "ask1")+MinPrice(AA.F_DealCode())*N);
T_Deal(AA.F_DealCode(), 0, 0, KN, Offers(AA.F_DealCode(), "bid1")-MinPrice(AA.F_DealCode())*N); //平掉所有空单并反向开 KN 手多单
}
}
}

```

## 第七章 后台程序化

### 7.1 运行模组

经过一篮子合约回测和多模型组合测试后，我们已经确定了自己的整套程序化交易，这时要做的就是将研究结果付诸于实战中。运行模组正是完成这一任务的功能载体，多模组可同时运行，各模组间资金独立、信号独立互不干扰。

#### 7.1.1 实现多模型组合从回测到实盘运行

当整套策略所交易的合约、加载的模型、运行的模组不止一个时，能否井然有序的运行，准确无误的计算每一个模组的数据是至关重要的，接下来我们要看一下为什么运行模组可以将多模型组合测试策略应用于实盘。

##### 原因一：可多个模组同时运行

运行模组可以同时加载多个模型，您可以自由组合，将不同的模型加载到不同的合约及不同的周期上。

##### 原因二：各模组独立运行互不干扰

每一个加载的模组都有属于自己的子账户，各自独立计算信号、持仓、资金等，这样一来就可对不同模组分别做资金管理。

### 运行了多个模组担心程序化乱成一团麻？您多虑了！他们各行其道。

多模型组合可将震荡模型和趋势模型同时加载到一个合约中，当行情震荡时，震荡模型的盈利可冲抵趋势模型的亏损。而当趋势行情到来的时候，这种多模型组合可共振盈利。但如何确保多个模组的运行互不干扰呢？运行模组为程序化有序运行保驾护航：

#### 模组之间互不干扰

每个模组有自己独立的资金池，运行过程中自行记录持仓、挂单、信号和信号执行情况，各自计算平仓盈亏和浮动盈亏与其他模组互不干扰。



#### 原因三：实时监控各模组运行状态

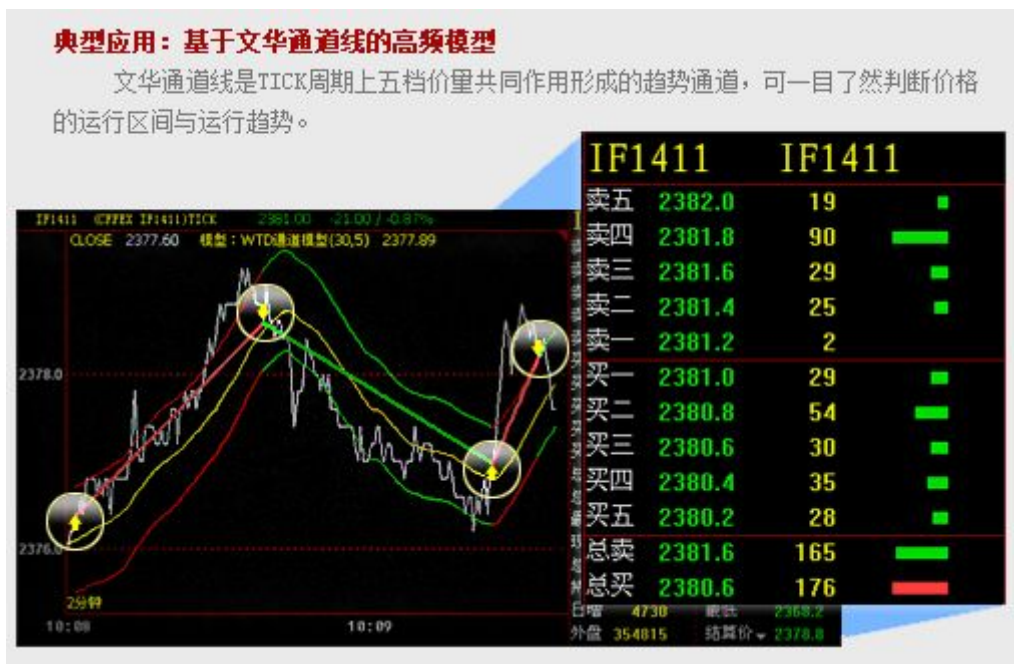
运行模组可实时显示几十个模型的信号执行、资金、持仓、挂单等状态。还可后台运行，不影响其他操作。这样一来，即使我们要运行多个策略，它们的运行状态也一目了然。



### 7.1.2 运行模组其他功能特点介绍

#### TICK 周期运行高频率程序化模型

利用五档行情数据分析价量关系及走势，模型加载在 TICK 周期上，根据判断结果向盒子发出自动交易的指令。



## 运行资金管理策略模型

资金管理策略可以有效统筹账户资金，控制交易风险，科学安排您的交易。

### 还在以固定手数开仓？是时候让你的交易提高一个档次了！

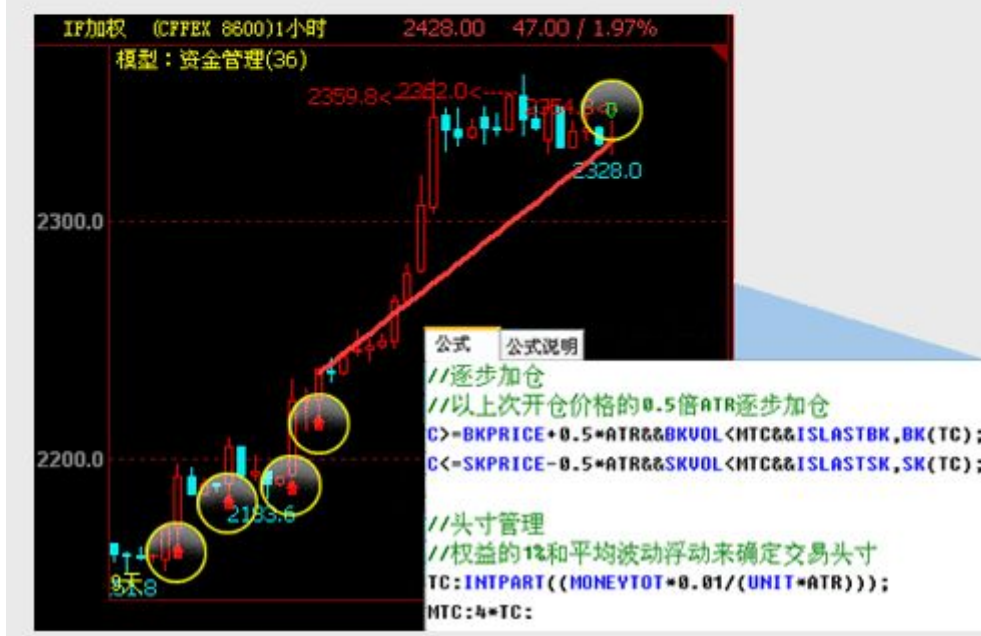
资金管理在交易中是一个非常重要的思想，我们常用控制资金使用率、按资金的固定百分比下单等方式来控制交易的风险。将这种思想加入策略模型，就有了程序化全自动交易最重要的特点之一-风险可控。

#### 加减仓可控：

交易思路举例：按照一定条件实行加仓策略并将仓位控制在一定范围内。

#### 按资金的固定百分比开仓：

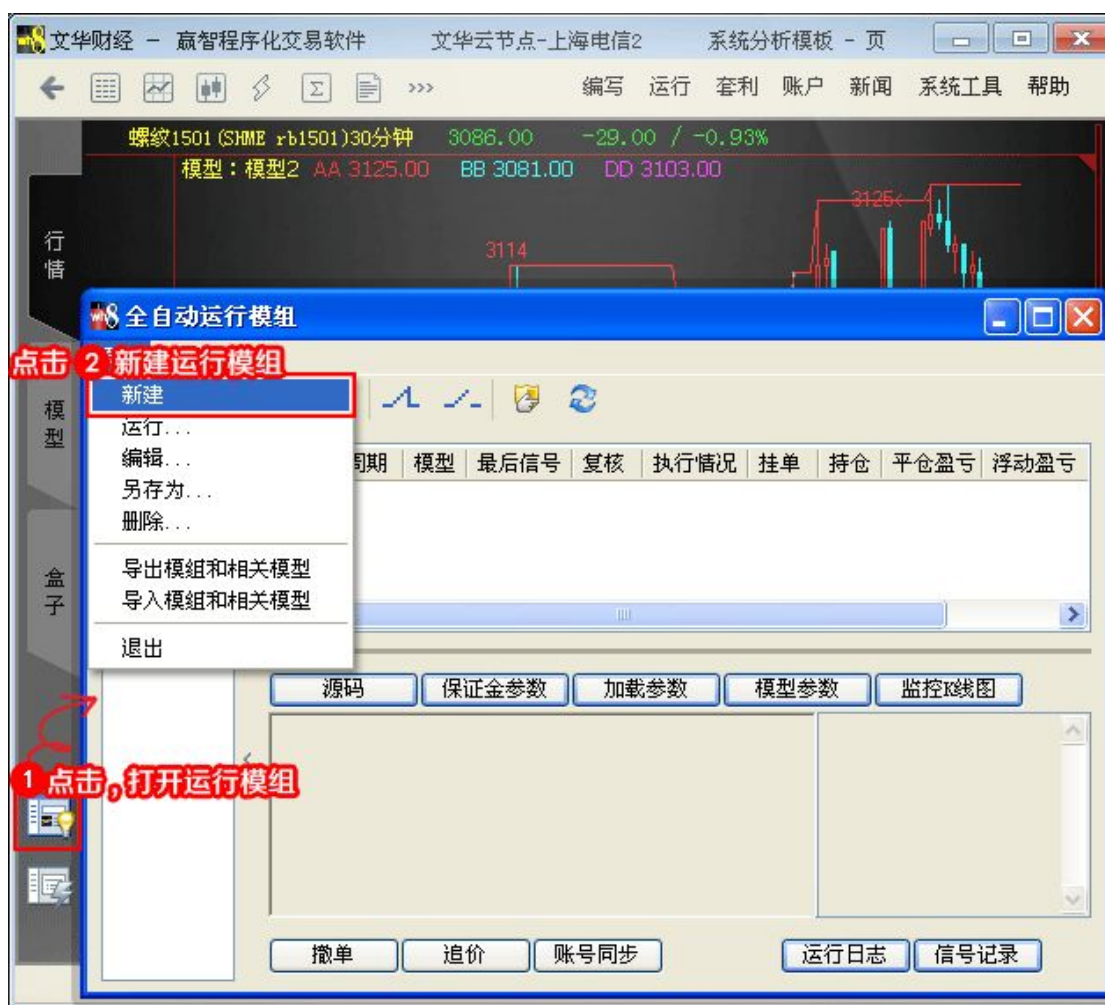
交易思路举例：每次开仓手数按照权益的1%计算。



### 7.1.3 操作方法

#### 新建模组项目

如下图所示，是如何加载运行模组



第一步 定义数据区



周期：程序化交易软件的模组支持常规周期、自定义周期和秒周期的加载。

## 第二步 选择模型



## 第三步 模型参数



对不同合约，加载同一模型使用的参数值可能是不同的。如在使用模型参数做止损时，不同合约使用的止损参数值是不一样的，难道仅仅因为止损参数值的不同就要新建几个模型？当然不需要，加载模型中的第四步就可解决这个问题，对同一模型每次加载时“缺省值”位置设置不同的参数值（每次加载的模组都是独立保存，参数值互不影响）就可以了，如上图所示。

#### 第四步 交易合约

当模型中写有 TRADE\_OTHER 函数时，该处会自动入去模型中所写模型，模型中未写 TRADE\_OTHER，可以在该处手动选择交易合约。

模型中写入 TRADE\_OTHER 函数规则如下：

**(1) 数据合约为主连合约：**

按照模型中 TRADE\_OTHER 函数指定的合约作为交易合约。

**(2) 数据合约为主连合约（如 IF1501）：**

按照 TRADE\_OTHER 函数指定的合约作为交易合约。

**(3) 数据合约为主连合约（实现自动换月）：**

TRADE\_OTHER 函数写为 TRADE\_OTHER('AUTO') 时，可加载到主连合约，实现自动换月移仓。

**例子：**用“螺纹指数”合约数据加载模型，计算模型信号，对螺纹 1501 合约下单是否可以实现？

可以通过 TRADE\_OTHER 函数实现指定螺纹 1501 合约交易，在加载时数据合约选择螺纹指数。

TRADE\_OTHER('CODE') 指定 CODE 合约作为交易合约，CODE 为合约代码。

**注：**

①回测时：信号价格取值为该函数定义的交易合约的信号价格。

模组加载时：数据合约为主连合约时选择的数据合约，交易合约为该函数指定的合约。

不写入该函数时，交易和数据合约一致。

②该函数写为 TRADE\_OTHER('AUTO') 时，可以加载到主连合约，实现自动换月移仓。

③从数据合约的数据和指定交易合约的数据对齐的位置开始计算信号。

④CODE 位置为具体合约时：该函数可以和 CLOSEKLINE\_MIN, CHECKSIG\_MIN, MULTSIG\_MIN 函数连用；不支持和 CHECKSIG\_SEC, MULTSIG\_SEC 函数连用。

CODE 位置写为 'AUTO' 时：该函数可以和 CHECKSIG\_SEC, CHECKSIG\_MIN, MULTSIG\_SEC、MULTSIG\_MIN, CLOSEKLINE\_MIN 函数连用。

⑤该函数必须在有信号的模型中使用。

⑥TRADE\_OTHER 函数不支持加载到副图中。

⑦CODE 写为合约代码时，该函数不支持加载到 TICK 周期，量能周期，秒周期上使用。

⑧CODE 位置不支持写入文华码。

⑨CODE 写为 'AUTO' 时，不支持加载到页面盒子中。

例：

CROSS(C, MA(C, 5)), BK; //最新价上穿 5 周期均线做多

CROSS(MA(C, 5), C), SP; //最新价下穿 5 周期均线平多

TRADE\_OTHER('IF1411'); //指定交易合约 of 股指 1411

AUTOFILTER;



第五步 子账户参数

**新建模組，第五步——子账户参数**

模組子账户起始资金  元

保证金比例  %

手续费  %%

wh8的每一个模組在自己的资金池里独立运行，与其他模組运行的资金、持仓是互不干扰的，可以用资金函数编写，实现头寸控制目的

未使用资金管理函数的基本模型，不具有头寸控制功能，请忽略本页设置

< 上一步(B) 下一步(N) > 完成 取消

有的用户模型源码中使用资金管理函数实现头寸控制，头寸大小可能是信号的判断条件。模型的历史 k 线图上，也需要计算信号，就需要给历史 k 线赋予一定的资金、合约的保证金比例、手续费，否则历史 k 线图可能会因为头寸计算错误不显示信号，或者没按照我们的思路计算信号。在加载模型的第四步中“模组起始资金”设置的值会赋予历史 k 线中的第一根 k 线，参与之后历史 k 线图上的信号计算。带有资金管理函数的模型在历史 k 线图上就能正确计算信号了。

**注：**

- 1、“模组子账户起始资金”只会影响模型历史 k 线图的信号计算，模型加载后参与信号计算的资金是模组初始化时剩余的“模组可用资金”。

**小技巧：**

**秒周期直接加载到模组计算量太大，设置信号计算起始时间提高效率。**

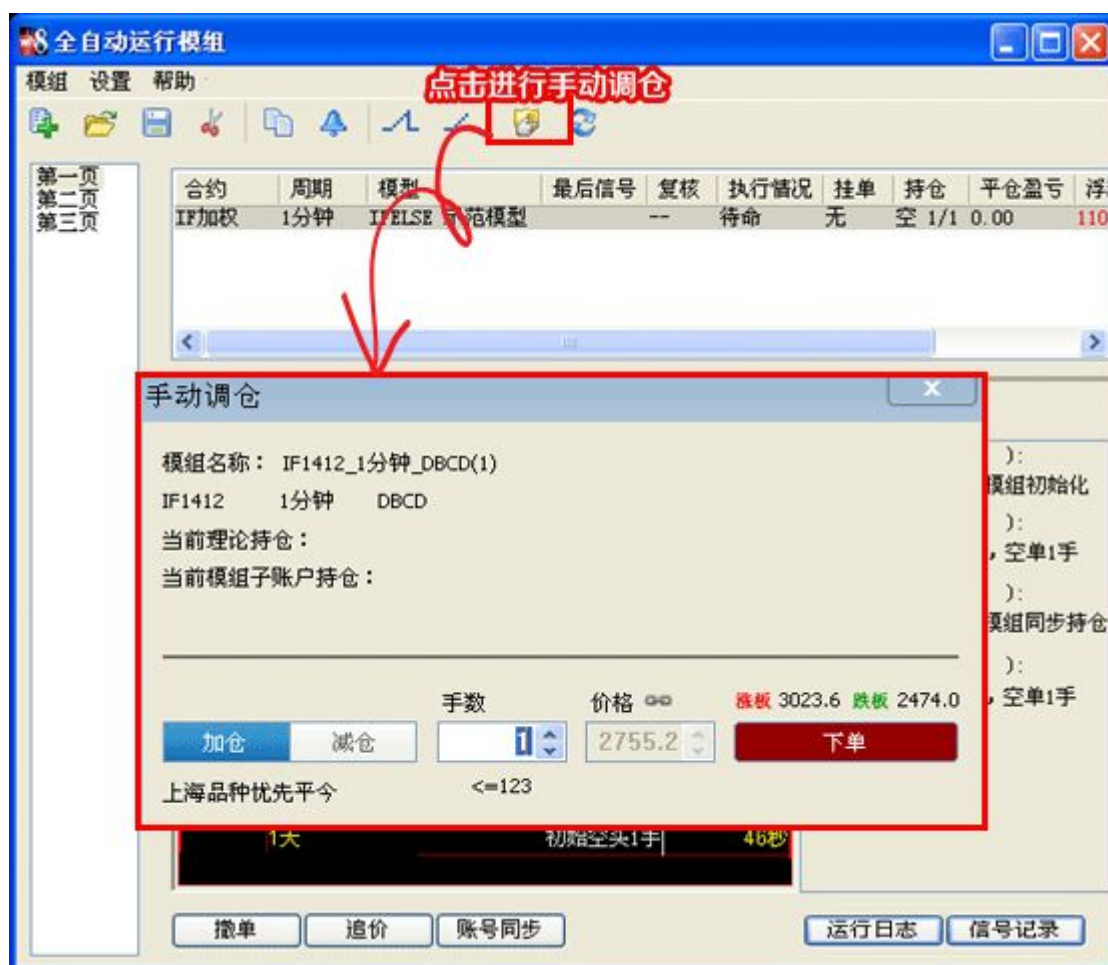
由于秒周期加载模型计算量大，为保证模组运行稳定需要先在主窗口 K 线图完成计算，再通过右键菜单中的【加入运行模组，后台运行】加载到运行模组中。

本地秒周期数据量过多会严重影响模型计算的速度，在主窗口 K 线图上单击鼠标右键【重设信号计算起始时间】可以灵活选择加载所用的数据量，避免这一问题。

步骤：主窗口 K 线图单击鼠标右键【重设信号计算起始时间】——设置时间——加载模组，这时软件会从刚刚设置的时间对模组进行计算。

**手动调仓**

对于一些用户来说，在做程序化交易的同时还需要手动进行调仓，通过手动调仓的功能，可以进行加减仓，BKVOL 和 SKVOL 这些读取模组持仓的函数可以根据你加减仓的手术获取当前模组的理论持仓。如下图所示是如何进行手动调仓（该功能适用于运行模组）：



## 7.2 盘口模型运行池

很多成功的程序化交易投资者每年算一笔账，在滑点上的损失比交的手续费还要多。随着投资者对程序化交易的认识不断提高，投资者之间将逐渐从交易策略的较量转变为下单精细化控制的博弈上来。盘口模型可对程序化委托的每个环节进行精细化控制，达到降低交易成本的目的，如何减小滑点成本，减小交易的摩擦成本是他们的着力点。

### 7.2.1 盘口模型功能特点

#### 实现基于盘口策略的算法交易

档行情数据分析价量关系及走势，盘口模型根据判断结果向算法交易运行池发出自动交易的指令。

### 看见别人看不见的，才有机会得到别人得不到的-显微镜下的盘口高频交易！

五档行情数据如显微镜般让我们看到了市场更深层的数据，而盘口模型则是我们在这个市场中为自己安置的精密探测仪，可及时发现盘口行情异动并以极快的速度执行我们想做的操作。

```
Offers(Mod1.F_DealCode(),"bidvol5");
//买一量到买五量的和
SUMSK = Offers(Mod1.F_DealCode(),"askvol1") +
Offers(Mod1.F_DealCode(),"askvol2") +
Offers(Mod1.F_DealCode(),"askvol3") +
Offers(Mod1.F_DealCode(),"askvol4") +
Offers(Mod1.F_DealCode(),"askvol5");
//卖一量到卖五量的和
IF( data[data.Num-1].Askvol1 > Big )
//盘口卖1挂单大于设置的大单量.
{
    IF( QuShi() == 2) //最近5笔主动买量大于主动卖量
    {
        SKID = T_Deal(CodeName,1,0,N,Offers(CodeName,"bid1"));
    }
}
IF( data[data.Num-1].Bidvol1 > Big )
//盘口买1挂单大于设置的大单量
{
    IF( QuShi() == 2) //最近5笔主动买量大于主动卖量
    {
        BKID = T_Deal(CodeName,0,0,N,Offers(CodeName,"ask1"));
    }
}
```

#### 有效控制交易滑点

在接收到交易指令后，通过盘口模型对下单过程进行精细控制，包括控制下单时间、监控当前市场价格偏离指令发出价格的程度等。

如何通过盘口模型控制滑点：

方法一、自适应追价：根据现价偏离首次下单价格的幅度来调整委托价格。

```
IF(T_IsNoOrder()!=1) //有挂单的情况
{
    IF(CurrentTime()-LastOrderTime())>=7 &&
    Price>AA.F_SigPrice()+5*MinPrice)
//开多挂单超7秒
//并最新价>指令价+5个最小变动价位，撤单不再委托。
    .....
}
```

方法二、控制信号时间，减少信号忽闪， 信号出现后 N 秒确认信号下单，信号消失 M 秒后做信号消失处理。

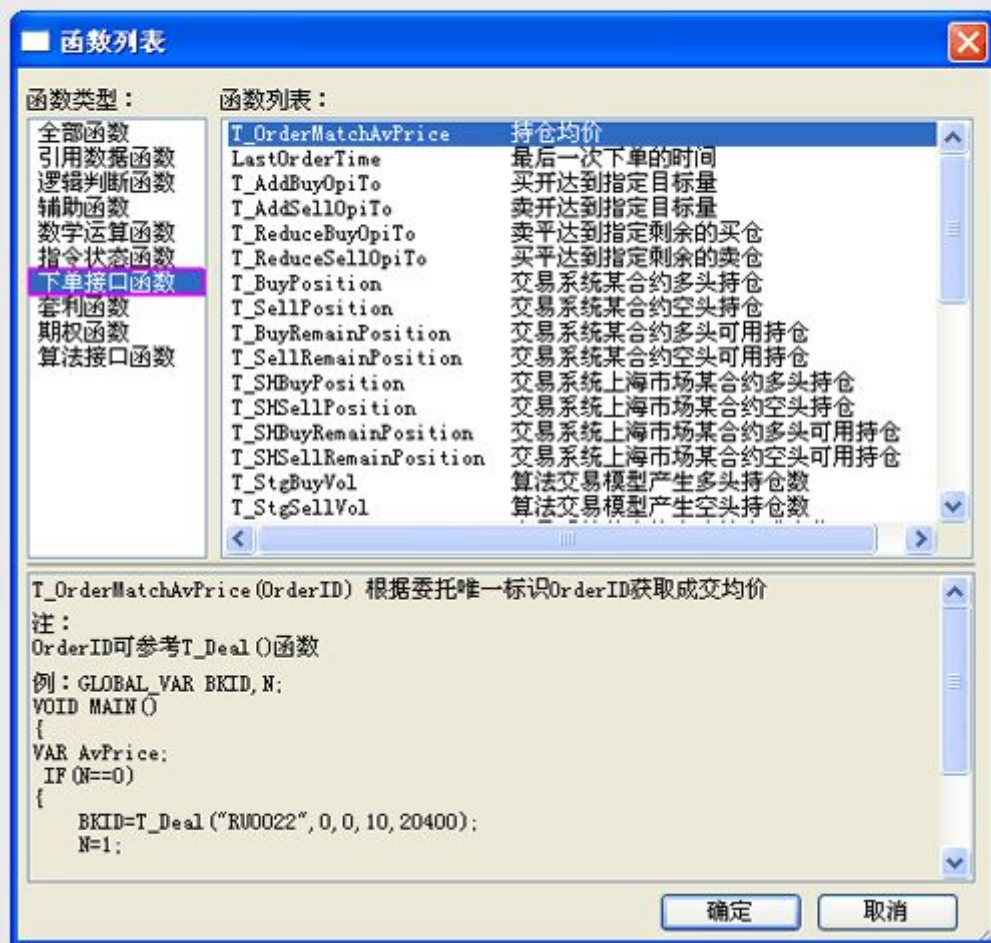
```
7 VOID MAIN()
8 {
9     OrderVol = 1;
10    IF (F_FreshSig()==1) //如果当前信号是没有处理过的新信号
11    {
12        CompareTime = CurrentTime();
13    }
14    IF(F_FreshSig() == 0 && F_SigValid() == 1 &&
15       CurrentTime() - CompareTime> N) //判断信号出现时间
16    {
17        TDEAL(); //信号出现N秒, 执行开平仓函数
18    }
19    IF(F_FreshSig() == 0 && F_SigValid() == 0 &&
20       CurrentTime() - CompareTime> M) //判断信号消失时间
21    {
22        BTDEAL(); //信号消失N秒, 执行信号消失处理函数
23    }
24 }
```

取交易账户中合约持仓价格、浮动盈亏等信息

盘口模型可取得交易账户的信息, 包括权益、可用资金、平仓盈亏、实际挂单情况等。

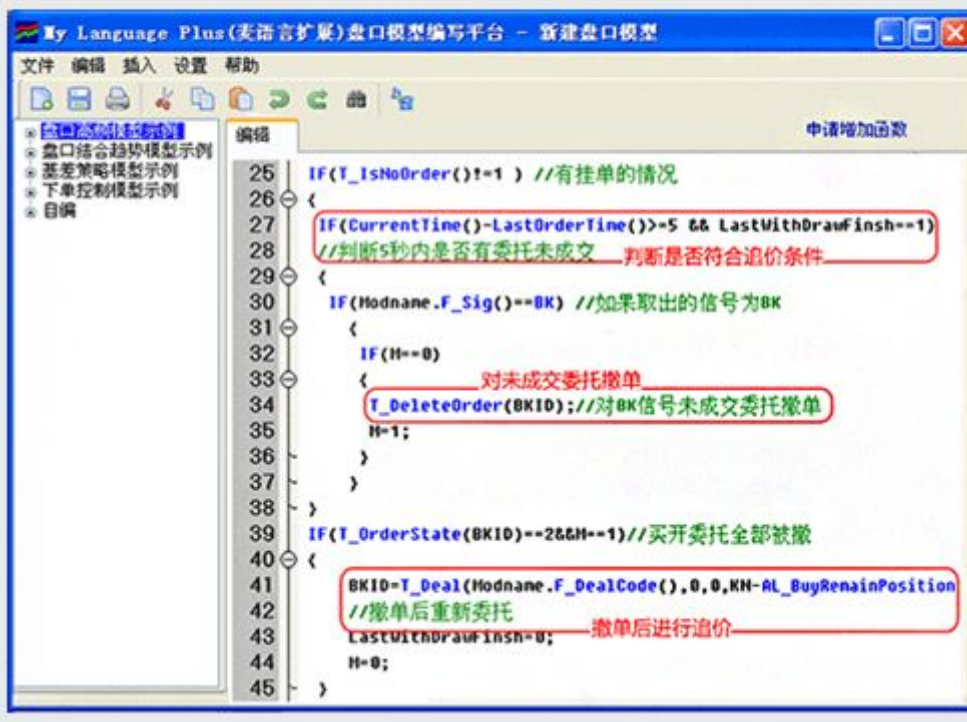
## 读取实际账户信息的意义是什么？实现手动开仓盘口模型平仓

盘口模型可取到实际账户的持仓均价、权益以及浮盈等信息，手动开仓后可以通过盘口模型读取这些信息来实现止损止盈、风险控制等平仓操作，这是运行模组和盒子全自动无法做到的。



## 跟踪交易进度

盘口模型可以读取交易进度，如某合约是否有挂单，委托是否全部成交等，这样就可以实现“挂单5秒没有成交撤单重新委托”这类更灵活的交易思路。



## 取运行模组的指标、信号等信息

运行模组的信号或指标后，利用算法做进一步数据处理，以实现更灵活的交易思路。

例：取模型出信号时最新价，判断当前价格是否已经构成过大滑点

```

1  VAR Modname;
2  VAR H;
3  VAR Code1; //定义合约名称
4  VAR Price1; //定义最新价
5  VOID MAIN()
6  {
7      Modname="模组名"; //取读模组信息
8      Code1=Modname.F_DealCode();
9      Price1=Price(Code1,"New");
10     IF(ABS(Price1-Modname.F_SigPrice())<=H&&T_IsNoOrder()==1)
11     //判断滑点是否过大
12     {
13         T_Deal(Code1,0,1,1,Offers(Code1,"ask1"));
14     }
15     ELSE IF(T_IsNoOrder()!=1&&ABS(Price1-Modname.F_SigPrice())>H)
16     //滑点过大对当前委托撤单
17     {
18         MessageOut("价差过大");
19         T_DeleteOrderAll();
20     }
21     //注：H为自定义滑点。
22 }
    
```

## 7.2.2 操作方法

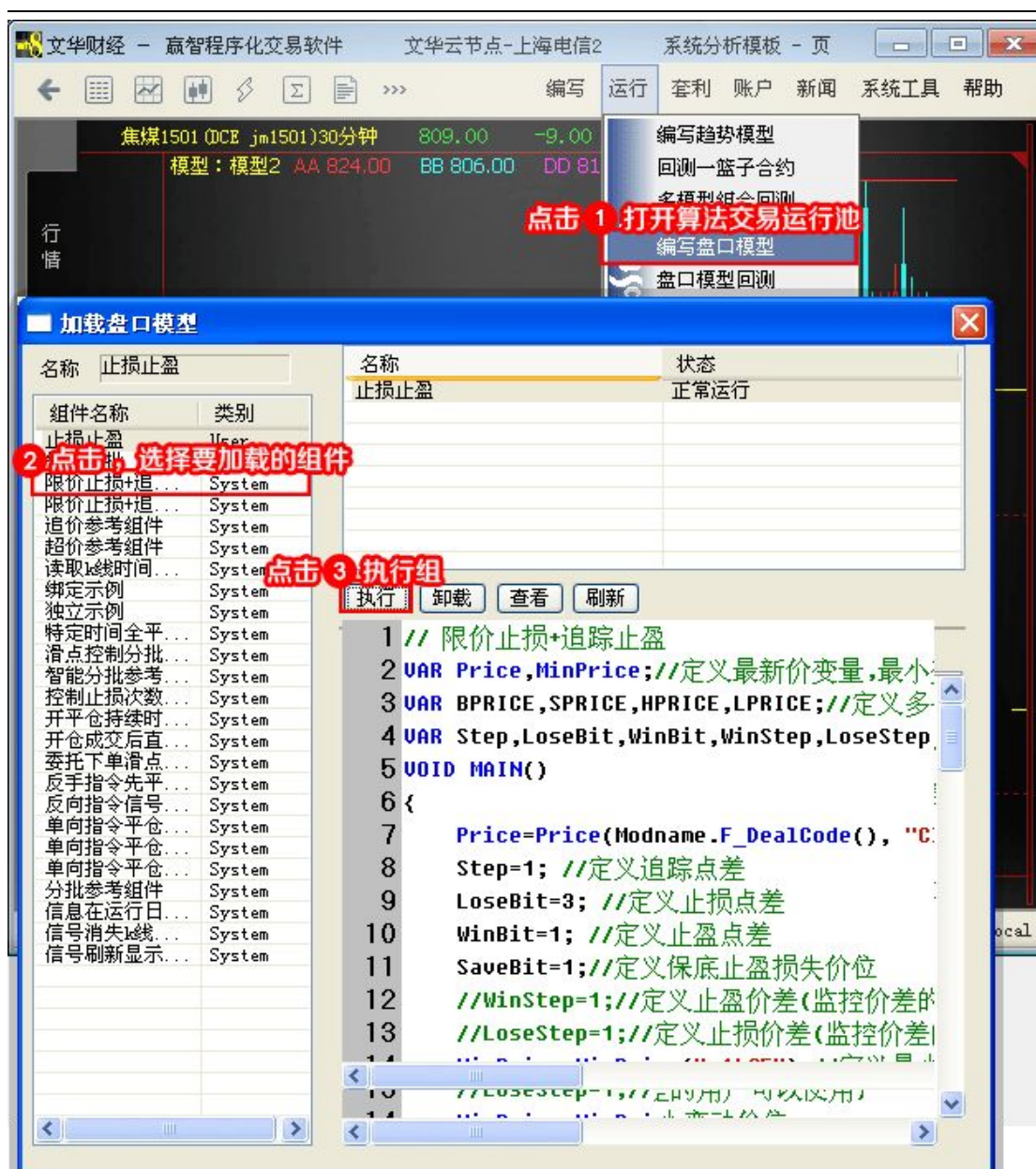
### 1、建立下单组件：

如下图所示是如何建立下单组件:



## 2、运行下单组件:

如下图所示, 是如何运行下单组件。



更多关于盘口模型使用的方法可以查看此链接 ([点击查看](#))

## 第八章 远程监控

### 8.1 设置运行模式

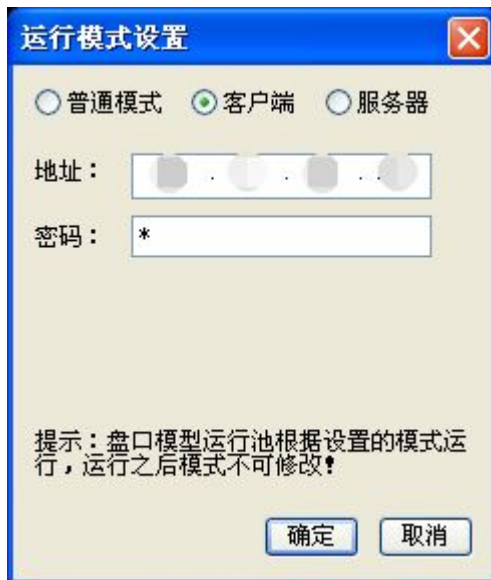
当我们需要对程序进行远程监控或者一个团队间需要通过远程来互通信号有无时,就可以利用远程监控功能。

在软件中,模型运行的方式有三种:普通模式、客户端和服务端。

**普通模式:**这是我们最常用的一种模式,自己做程序化交易,不需要和其他人做远程信号沟通的

可以使用这种模式。

**客户端：**这是相对于服务器端而言的，客户端可以接收到服务器端模型发来信号指令，在客户端设置好 IP 地址后就可以连接到相应服务器上，与之同步交易。



运行模式设置

☐ 普通模式 ☒ 客户端 ☐ 服务器

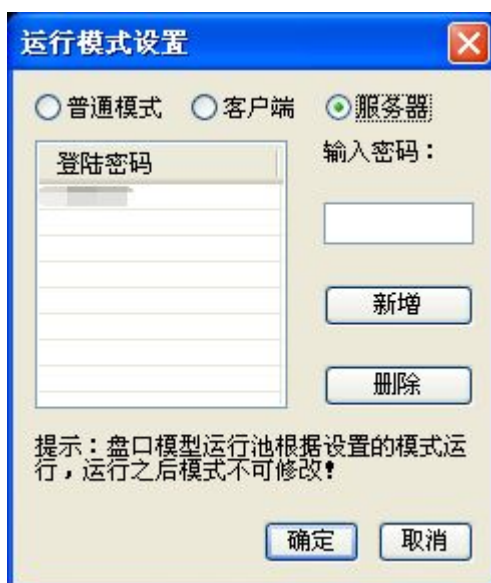
地址：

密码：

提示：盘口模型运行池根据设置的模式运行，运行之后模式不可修改！

确定 取消

**服务器：**服务器是发出指令的一方，客户端可以实时接受到服务器端模型出现的指令，服务器端需要设置的是查看密码，这个密码是客户端连接到该服务器端的唯一口令。



运行模式设置

☐ 普通模式 ☐ 客户端 ☒ 服务器

登陆密码

输入密码：

新增 删除

提示：盘口模型运行池根据设置的模式运行，运行之后模式不可修改！

确定 取消

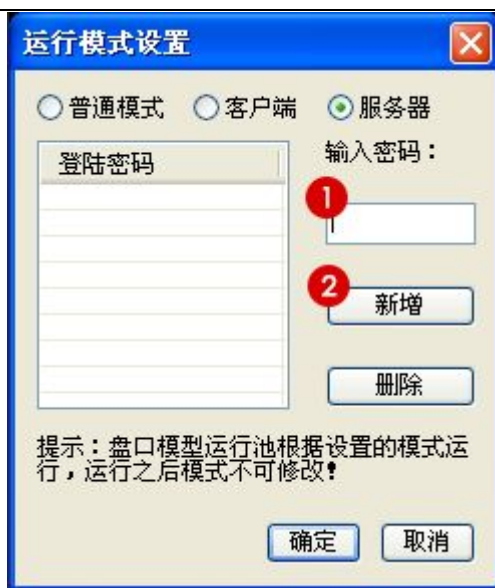
#### 远程监控设置：

假如交易员想把自己 A 电脑上的模组运行状态同步到 B 电脑上监控

第一步：A 电脑设置服务器端

在上面的例子中，由于要把 A 电脑的模组运行状态同步到 B 电脑上监控，因此 A 电脑为服务器。

打开软件后，点击菜单栏的【运行】—>【运行模式设置】，在弹出的设置框中按照下图的标号顺序设置服务器密码，设置后点击确定。

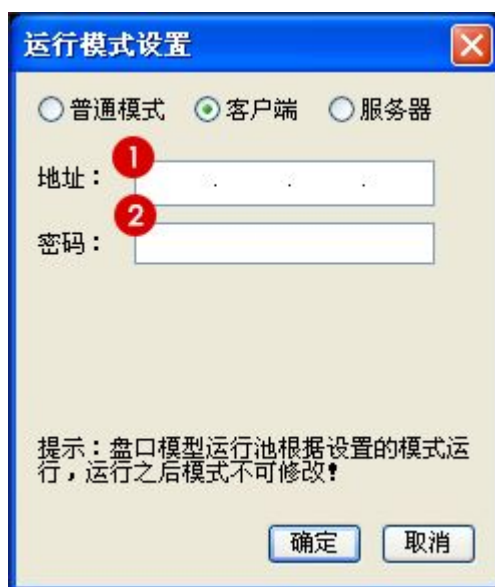


注意：打开软件后，第一件事就是设置运行模式，不要打开盘口模型运行池和运行模组。盘口模型运行池及运行模组根据设置的模式运行，运行之后模式不可修改！

第二步：B 电脑设置客户端

在上面的例子中，由于要用 B 电脑监控 A 电脑的模组运行状态，因此 B 电脑为客户端。

打开软件后，点击菜单栏的【运行】→【运行模式设置】，在弹出的设置框中输入 A 电脑的 IP 地址及 A 电脑刚刚设置的密码，设置后点击确定，这时 B 电脑就已经连接到了 A 电脑上，可以接受信息了。



第三步：A 电脑打开运行模组或盘口模型运行池，这时当 B 电脑也打开运行模组或盘口模型运行池时就能在自己的客户端上看见 A 电脑上运行的情况了。

## 8.2 日志邮件

我们经常回遇到这样的情况，因为有急事需要外出，又没有办法把家里的电脑随身携带，程

序化运行情况没法随时掌握。

有没有办法可以让我们随时掌握模型运行动态呢？【日志邮件】功能可以满足这一需求。该功能可以帮助您接收到模型运行动态的邮件，人不在电脑前一样可以监控自己的程序。操作方法：在全自动运行模组界面点击菜单栏上的【设置】→【日志邮件】进行设置。

