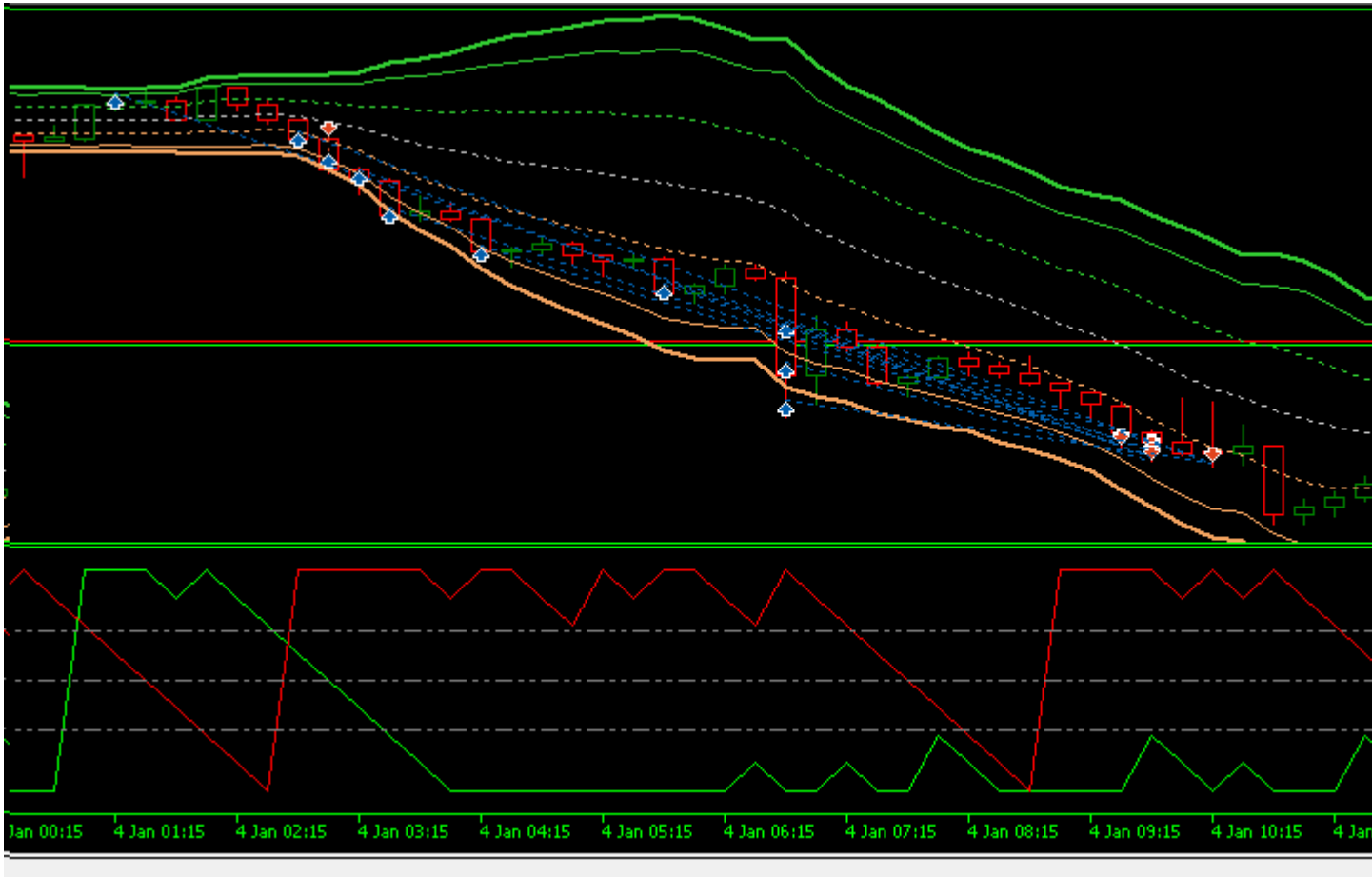


Hi, I have a problem using the strategy tester, and am searching for some ideas how to prevent this in the future.

I'm running a strategy on GBPAUD, which is consistent profitable in backtest. On real market conditions (real account running 24/7) it took a trade, that it didn't take in the backtest. (I could validate it one day later, when the newest data was provided to my Robofores Demo account). I have noticed it, hence it blew my account... And now I'd like to learn from this .

What Happened on Real account, there was a trade taken on 1:15.



The strategy tester didn't take this trade. Here the code, Bulls crosses Bears from above => sell, Bears crosses bulls from above => buy.

Maybe here some hints: I'm taking the last 3 values from the Bulls and Bears array and am considering value 1 and 2 against each other.

```
double BullsAroonBuffer[];
double BearsAroonBuffer[];

int res = CopyBuffer(handleAroon,0,0,3,BullsAroonBuffer);
if(res<0)
{
    Print("IndicatorClass:AROON copy buffer failed ",res);
}
res = CopyBuffer(handleAroon,1,0,3,BearsAroonBuffer);
if(res<0)
{
    Print("IndicatorClass:AROON copy buffer failed ",res);
}
```

```

20 // handling of 0 place had to be added to increase the reliability.
21 // Bulls crosses Bears from above => sell
22 // if((BullsAroonBuffer[0] > BearsAroonBuffer[0] || BullsAroonBuffer[1] > BearsAroonBuffer[1]) &&
23 if(((BullsAroonBuffer[0] > BearsAroonBuffer[0] && BullsAroonBuffer[1] < BearsAroonBuffer[1]) ||
24 (BullsAroonBuffer[1] > BearsAroonBuffer[1] && BullsAroonBuffer[2] < BearsAroonBuffer[2]))
25 // if(BearsAroonBuffer[2]>=90)
26 {
27     result.confirmation = -1;
28
29     // Bulls crosses Bears from below => buy
30 }
31 else
32     // if((BullsAroonBuffer[0] < BearsAroonBuffer[0] || BullsAroonBuffer[1] < BearsAroonBuffer[1]) &&
33 if(((BullsAroonBuffer[0] < BearsAroonBuffer[0] && BullsAroonBuffer[1] > BearsAroonBuffer[1]) ||
34 (BullsAroonBuffer[1] < BearsAroonBuffer[1] && BullsAroonBuffer[2] > BearsAroonBuffer[2]))
35 {
36     // if(BullsAroonBuffer[2]>=90) {
37     result.confirmation = 1;
38 }
39 else
40 {
41     result.confirmation = 0;
42 }
43

```

Values: 0:15

| Ausdruck | Wert |
|----------------------|----------------------|
| 1/2 BearsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 5.555555555555571 |
| 1/2 [1] | 94.44444444444442865 |
| 1/2 [2] | 83.33333333333328596 |
| 1/2 BullsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 27.77777777777771462 |
| 1/2 [1] | 16.66666666666671404 |
| 1/2 [2] | 5.555555555555571 |

Values 0:30

| | |
|----------------------|----------------------|
| 1/2 BearsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 94.44444444444442865 |
| 1/2 [1] | 83.33333333333328596 |
| 1/2 [2] | 72.22222222222228538 |
| 1/2 BullsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 16.66666666666671404 |
| 1/2 [1] | 5.555555555555571 |
| 1/2 [2] | 5.555555555555571 |

Values 0:45

| Ausdruck | Wert |
|----------------------|----------------------|
| 1/2 BearsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 83.33333333333328596 |
| 1/2 [1] | 94.4444444444442865 |
| 1/2 [2] | 83.33333333333328596 |
| 1/2 BullsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 5.555555555555571 |
| 1/2 [1] | 5.555555555555571 |
| 1/2 [2] | 5.555555555555571 |

Values 1:00

| | |
|----------------------|----------------------|
| 1/2 BearsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 94.4444444444442865 |
| 1/2 [1] | 83.33333333333328596 |
| 1/2 [2] | 72.22222222222228538 |
| 1/2 BullsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 5.555555555555571 |
| 1/2 [1] | 5.555555555555571 |
| 1/2 [2] | 5.555555555555571 |

Values 1:15

| Ausdruck | Wert |
|----------------------|----------------------|
| 1/2 BearsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 83.33333333333328596 |
| 1/2 [1] | 72.22222222222228538 |
| 1/2 [2] | 61.11111111111114269 |
| 1/2 BullsAroonBuffer | dynamic array[3] |
| 1/2 [0] | 5.555555555555571 |
| 1/2 [1] | 94.4444444444442865 |
| 1/2 [2] | 94.4444444444442865 |

As You can see here, the Bulls Buffer jumped somehow by 2 places, while te bears jumps correctly always by 1 place. (means 94.4 wasn't there at all, and a step later there were 2 values)

Aroon I downloaded and it is as follows [Free download of the 'Aroon Oscillator' indicator by 'GODZILLA' for MetaTrader 5 in the MQL5 Code Base, 2011.08.02](#)

The method is triggered by an event tick

```

2 //+-----+
3 //| ChartEvent function |
4 //+-----+
5 void OnChartEvent(const int id,
6                   const long &lparam,
7                   const double &dparam,
8                   const string &sparam)
9 {
10 //---
11
12     if(((int)lparam) == _Period)
13     {
14         bool newTrade = false;
15         bool methodSkipped = false;
16
17         mAlgorithm.OnTickNewBar(newTrade, methodSkipped, 0, inpCloseOnly) ;
18     }
19 }

```

This is triggered by the eventspy as stated [MQL5 Cookbook: Multi-Currency Expert Advisor - Simple, Neat and Quick Approach - MQL5 Articles](#)

This works fine for several algorithms. But now I'm not sure if the trades will be taken similar to the backtest.

Looking forward to a discussion. Thanks :-)