○「　アメンボ式 EA 開発法；その１（入れ替え、補足版）　」

☆前回、３月３０日に投稿の「アメンボ式 EA 開発法；その１」は、正直言って、右上がり資産カーブが得られた嬉しさに、チェックをかっ飛ばして投稿したため、その直ぐ後で根本的問題に気付き、慌てて「差替予定」のコメント（赤字）をホームページに入れました。
・・まことにお恥ずかしい次第です。

☆略２ヶ月間で見直した内容を報告します。
ただし、集中して多少疲れた（息が続かず）ので、下記の手順での報告としたく。
①今回報告；　バックテスト結果と MQL コード、および基本説明のみ
②次回以降；　上記の結果に至る「考え方、手法・実施結果（含、最適化）」を３回～４回に分けて解説予定です

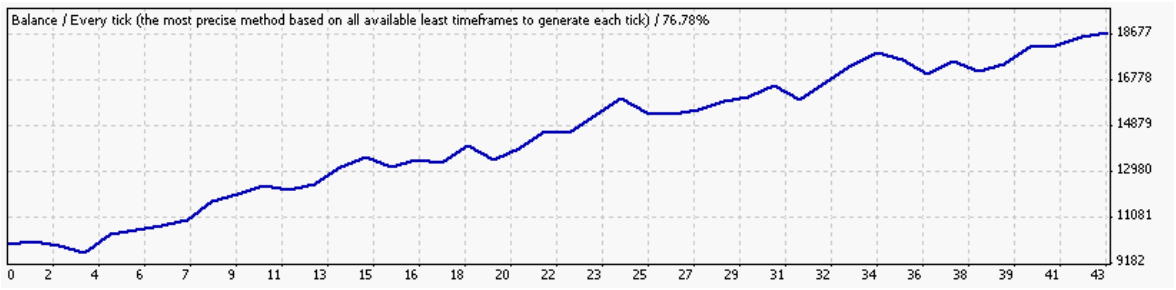
○＜ご注意＞開発品はまだまだ、不完全（以下参照）であり、アメンボが現在改良中です。
従って「実トレードでの結果」は予測できず、責任も持てません。
（**アメンボはバックテストでしか確認していないのです！、デモでのチェックも未着手**）

●アメンボからの連絡；
・約８ヶ月間四苦八苦して辿りついた内容ですが､MQL コードは全て公開する事にしました。
率直に言いますと、手法を開発するほうが、実際のトレードをするよりも、
「面白いと思う自分」に気が付いたからです。

※追記；
・この２ヶ月間の見直しとチェックで、ありとあらゆる問題に遭遇しました。
現在は、バックテストの実行ごとに「ログ（操作履歴)」を確認して、問題が発生しているか否かを確認しています。
（前回の投稿時は、投稿を急ぐあまり、ログ中のエラーに気が付きませんでした）

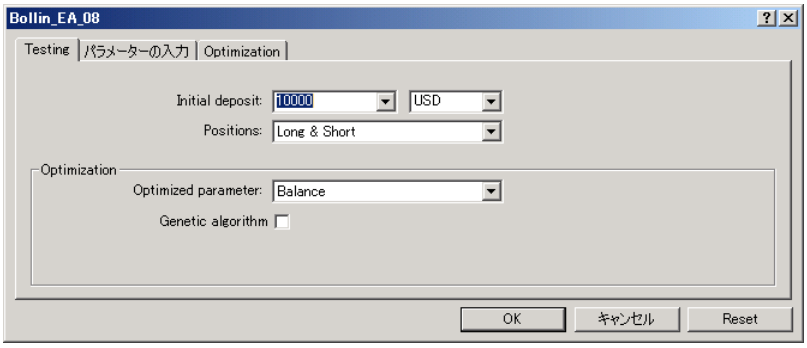
１．バックテスト結果

（１）資産カーブ

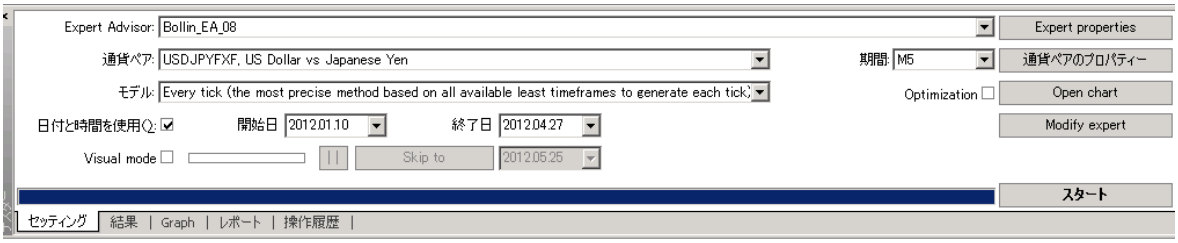


（２）バックテスト条件

①１ロットは「100,000」（10 万）通貨、「Long ＆ Short」

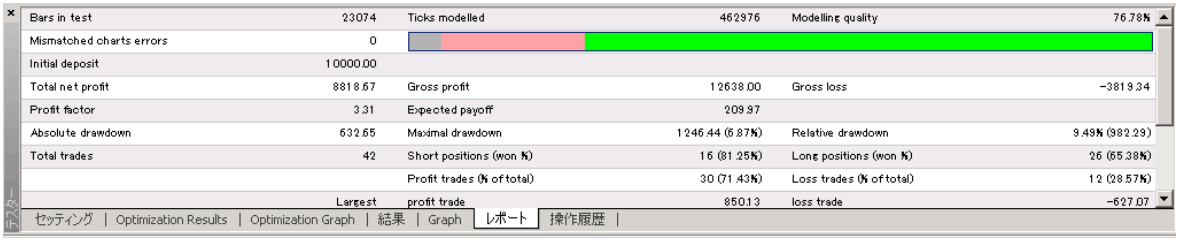


②USDJPY、５分足、Every Tick モデル、期間は「2012.01.10～2012.04.27」



（３）レポート

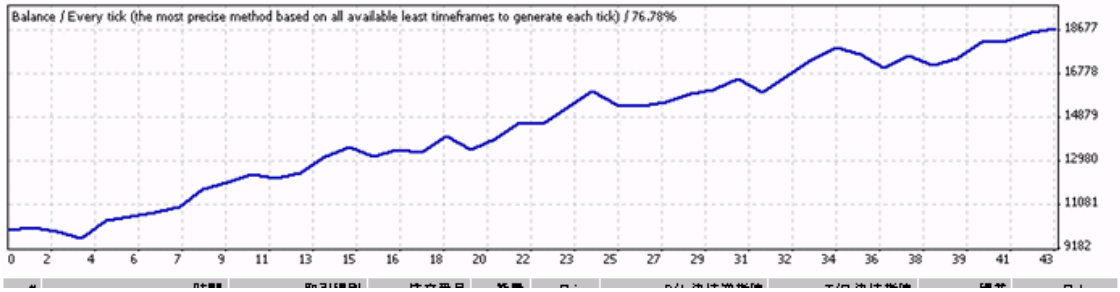
①「レポート」タブの内容



②総合レポート

Strategy Tester Report
Bollin_EA_08
FOREX.comJapan-Demo() (Build 419)

通貨ペア	USDJPYFXF (US Dollar vs Japanese Yen)				
期間	5分足(M5) 2012.01.10 00:00 - 2012.04.26 23:55 (2012.01.10 - 2012.04.27)				
モデル	Every tick (the most precise method based on all available least timeframes)				
パラメーター	Lots=1; max_position=1; profit=0.6; loss=0.7; profit_2=0.6; loss_2=0.5; period_bollin=40; IKA=0.4; IZYOU=0.2; shortPeriod_buy=50; mediumPeriod_buy=120; short_buy=-0.0044; long_buy=-0.0006; shortPeriod_sell=35; mediumPeriod_sell=180; short_sell=0.0044; long_sell=0.0006; trendPeriod=300; _up=-0.00024; _down=0.00024; div_=0.04;				
Bars in test	23074	Ticks modelled	462976	Modelling quality	76.78%
Mismatched charts errors	0				
Initial deposit	10000.00				
Total net profit	8772.68	Gross profit	13236.02	Gross loss	-4463.34
Profit factor	2.97	Expected payoff	204.02		
Absolute drawdown	632.65	Maximal drawdown	1246.44 (6.88%)	Relative drawdown	9.49% (982.29)
Total trades	43	Short positions (won %)	17 (76.47%)	Long positions (won %)	26 (65.38%)
		Profit trades (% of total)	30 (69.77%)	Loss trades (% of total)	13 (30.23%)
	Largest	profit trade	774.32	loss trade	-644.00
	Average	profit trade	441.20	loss trade	-343.33
	Maximum	consecutive wins (profit in money)	7 (2783.89)	consecutive losses (loss in money)	2 (-926.46)
	Maximal	consecutive profit (count of wins)	2783.89 (7)	consecutive loss (count of losses)	-926.46 (2)
	Average	consecutive wins	3	consecutive losses	1



（４）少し「結果」を考察

①まず、結果の文字が小さいので、議論に必要な部分だけを「テキスト」で拾い出します。

Bars in test　23074
Ticks modelled　462976
Modelling quality　76.78%
Mismatched charts errors 0
Initial deposit　10000.00
Total net profit 8772.68
Gross profit　13236.02
Gross loss　-4463.34
Profit factor　2.97
Expected payoff　204.02
Absolute drawdown　632.65
Maximal drawdown　1246.44 (6.88%)
Relative drawdown　9.49% (982.29)

Total trades 43
Short positions (won %) 17 (76.47%)
Long positions (won %) 26 (65.38%)
Profit trades (% of total) 30 (69.77%)
Loss trades (% of total) 13 (30.23%)
Largest
profit trade 774.32
loss trade -644.00
Average
profit trade 441.20
loss trade -343.33
Maximum
consecutive wins (profit in money) 7 (2763.89)
consecutive losses (loss in money) 2 (-926.46)
Maximal
consecutive profit (count of wins) 2763.89 (7)
consecutive loss (count of losses) -926.46 (2)
Average
consecutive wins 3
consecutive losses 1

②考察

その１;「Modelling quality 76.78%」って何だ？・・特に気になった

未だ、正確には理解していないのですが、バックテスト自体の品質を示し、
５分足では「０．５～０．９」の範囲をとるようです。

その２;期待値はどうなった？

Total trades 43
Profit trades (% of total) 30 (69.77%)
Loss trades (% of total) 13 (30.23%)

今回は、「profit=0.6、profit_2=0.7、loss=0.6、loss_2=0.5」としたので、
大雑把に「profit=0.65、loss=0.55」として計算すると、

期待値＝0.65×0.70＋（－0.55×0.30）＝0.29

大雑把に言うと「１回のトレードで平均０．２９（約29pips）利益」を生み出す。

２．基本部分の説明（ほんの少し解説）

（１）ＥＡコードの大きな構造

```
int start()
{
//  NewBar かチェック
   if(IsNewBar() && (Bars>barsTotal))
   {
       ①「Open 値」を判断して売買にＩＮし、またはＯＵＴする。

    } else
   {
      //［５分足」内で起こる急変に対応する処理を記載する
      //  バックテストでは確認が困難な部分
       ②「擬似ティック値」を判断して、必要ならＯＵＴのみする。
    }
   return(0);
}
```

※ＥＡを上記のような構造にした理由は、理論的に考えると、
「Every Tick モード」でバックテストするとき、
「バックテスト」の結果と「実際のトレード」の対応が、一番良くとれる筈だからです。

なにせ、バックテスト用のヒストリカル・データには、「Tick data」が有りません。
このことは、５分足などの短周期でバックテストを行うときに、特に問題になります。

・・従って、１分足や５分足でバックテストするときには、Modelling quality が
低下するそうです。(まだ、すっきりしませんが)

（２）ちょっとしたノウハウ

・小生のＥＡコードでは、見通しを良くするために、

```
if(条件式) {・・・}ではなく、

bool 名前＝条件式;
if(名前) {・・・}
```

としています。

今回の様に条件式が複雑な場合に、特に威力を発揮します。
(　if(　) の中に書こうものなら、何がなんだか判らなくなりますよ！　)

３．使用ＥＡのコード・・詳細な解説は、次回以降に順次行います。

```
//+------------------------------------------------------------------+
//|                                                  Bollin_EA_08.mq4  |
//|                                                  2012.05.25 amenbo |
//+------------------------------------------------------------------+
#define Magic_ID 1930

extern double Lots=1;
extern int max_position=1;//最大保有ポジション数
// 損益設定
//「①②」は最適化の実施後での設定値
extern double profit=0.60;
extern double loss=0.70;
extern double profit_2=0.60;
extern double loss_2=0.50;
// ①Bollinの幅
extern int period_bollin=40;
extern double IKA=0.4;
extern double IZYOU=0.2;
// ②フィルター設定
extern int shortPeriod_buy=50;
extern int mediumPeriod_buy=120;
extern double short_buy=-0.0044;
extern double long_buy=-0.0006;
//     ・・sell専用
extern int shortPeriod_sell=35;
extern int mediumPeriod_sell=180;
extern double short_sell=0.0044;
extern double long_sell=0.0006;
//-----------------------------
//  ③トレンド判断
extern int trendPeriod=300;//70
extern double _up=-0.00024;//-0.0004
extern double _down=0.00024;
extern double div_=0.04;
//==================================
static datetime lastbar;
datetime in_time,out_time;
int barsTotal;
//売買シグナル判定用の配列データ
double EMA_[10];
double Bol_hi[10];
double Bol_lo[10];
double Price_01[10000];
double Trend_[10000];
```

```
//////////////////////////
int init()
{
   //
   lastbar=Time[1];
   barsTotal = Bars;
   in_time=Time[20];
   //
   return(0);
}
int deinit()
{
   return(0);
}
//////////////////////////////////////
int start()
{
   if(Bars<100 || IsTradeAllowed()==false) return;
//  NewBar かチェック
   if(IsNewBar() && (Bars>barsTotal))
   {

      barsTotal=Bars;

      //＜ポジションが在る場合の処理＞
      if(OrdersTotal()>=1)
      {
         for(int i=0;i<OrdersTotal();i++)
         {
            if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==false) break;
            if(OrderMagicNumber()!=Magic_ID || OrderSymbol()!=Symbol()) continue;

            if(OrderMagicNumber()==Magic_ID && OrderSymbol()==Symbol())
            {
               bool ikisugi=(Time[0]-in_time)>=(300*Period()*60);

               if(OrderType()==OP_BUY)
               {
                  double kachi_buy_price=OrderOpenPrice()+profit;
                  double make_buy_price=OrderOpenPrice()-loss;
                  //
                  bool kachi_buy=(Bid>=kachi_buy_price);
                  bool make_buy=(Bid<=make_buy_price);
                  //
                  if(kachi_buy || make_buy || ikisugi)
                  {
```

```
                OrderClose(OrderTicket(),Lots,Bid,0,Blue);
                continue;
             }
          }

          if(OrderType()==OP_SELL)
          {
             double kachi_sell_price=OrderOpenPrice()-profit;
             double make_sell_price=OrderOpenPrice()+loss;
             //
             bool kachi_sell=(Ask<=kachi_sell_price);
             bool make_sell=(Ask>=make_sell_price);
             //
             if(kachi_sell || make_sell || ikisugi)
             {
                OrderClose(OrderTicket(),Lots,Ask,0,Blue);
                continue;
             }

          }
       }
    }
}//end_of_if(OrdersTotal()>=1)

//＜新規ポジションの設定と方針＞
//アップトレンドでは⇒buyINのみ実施（順張り）
//ダウントレンドでは⇒sellINのみ実施（順張り）
//レンジ相場では　　⇒buyとsell（逆張り？―
//if(OrdersTotal()<=max_position)
if(OrdersTotal()==0)
{

   //判断用の配列を準備
   ArraySetAsSeries(Bol_hi,true);
   ArraySetAsSeries(Bol_lo,true);
   //
   for(int j=0;j<=6;j++)
   {
      Bol_hi[j]=iBands(NULL,0,period_bollin,2,0,PRICE_OPEN,MODE_UPPER,j);  //2_σ
      Bol_lo[j]=iBands(NULL,0,period_bollin,2,0,PRICE_OPEN,MODE_LOWER,j);  //2_σ
   }

   //判断用の共通指標を準備
   bool Subtract=((IZYOU<(Bol_hi[0]-Bol_lo[0])) && ((Bol_hi[0]-Bol_lo[0])<IKA));
   bool hanareta=((Time[0]-in_time)>=(19*Period()*60));
```

```
         bool TREND_UP=((slope_kaiki(0,trendPeriod))<=(_up)) &&
(call_hensa(0,trendPeriod)<=div_);
         bool TREND_DOWN=((slope_kaiki(0,trendPeriod))>=(_down)) &&
(call_hensa(0,trendPeriod)<=div_);

         ////if(TREND_UP)
         ////{
            //USDJPY の上昇相場（円安）
            //買い条件は整ったか
            bool cross_up_1=((High[1]>Bol_hi[1]) && (Open[0]>Bol_hi[0]));
            bool cross_up_2=((Open[3]<Bol_hi[3]) && (Open[2]<Bol_hi[2]));
            bool cross_up_3=((Open[4]<Bol_hi[4]) && (Open[3]<Bol_hi[3]));
            bool cross_up_4=((Open[5]<Bol_hi[5]) && (Open[4]<Bol_hi[4]));
            bool cross_up_5=((Open[6]<Bol_hi[6]) && (Open[5]<Bol_hi[5]));
            bool cross_up=(cross_up_1 && (cross_up_2 || cross_up_3 || cross_up_4 ||
cross_up_5));
            //条件の補足
            bool cross_up_rapid=((High[2]>Bol_hi[2]) && (High[1]>Bol_hi[1]) &&
(Open[0]>Bol_hi[0]));
            //Filter；本当に買いか
            double slope_s_buy=slope_kaiki(0,shortPeriod_buy);
            double slope_m_buy=slope_kaiki(0,mediumPeriod_buy);
            bool SLOPE_BUY=((slope_s_buy<=short_buy) && (slope_m_buy<=long_buy));
            //
            if(((cross_up || cross_up_rapid) && SLOPE_BUY) && Subtract && hanareta)
            {
               OrderSend(Symbol(),OP_BUY,Lots,Ask,0,0,0,"my_Buy_order",Magic_ID,Green);
               in_time=Time[0];
            }

         ////} else if(TREND_DOWN)
         ////{
         ///+
            //USDJPY の下降相場（円高）
            //売り条件は整ったか
            bool cross_down_1=((Low[1]<Bol_lo[1]) && (Open[0]<Bol_lo[0]));
            bool cross_down_2=((Open[3]>Bol_lo[3]) && (Open[2]>Bol_lo[2]));
            bool cross_down_3=((Open[4]>Bol_lo[4]) && (Open[3]>Bol_lo[3]));
            bool cross_down_4=((Open[5]>Bol_lo[5]) && (Open[4]>Bol_lo[4]));
            bool cross_down_5=((Open[6]>Bol_lo[6]) && (Open[5]>Bol_lo[5]));
            bool cross_down=(cross_down_1 && (cross_down_2 || cross_down_3 || cross_down_4
|| cross_down_5));
            //条件の補足
            bool cross_down_rapid=((Low[2]<Bol_lo[2]) && (Low[1]<Bol_lo[1]) &&
(Open[0]<Bol_lo[0]));
            //Filter；本当に売りか
```

```
         double slope_s_sell=slope_kaiki(0,shortPeriod_sell);
         double slope_m_sell=slope_kaiki(0,mediumPeriod_sell);
         bool SLOPE_SELL=(slope_s_sell>=short_sell) && (slope_m_sell>=long_sell);
         //
         if(((cross_down || cross_down_rapid) && SLOPE_SELL) && Subtract && hanareta)
         {
            OrderSend(Symbol(),OP_SELL,Lots,Bid,0,0,0,"my_Sell_order",Magic_ID,Red);
            in_time=Time[0];
         }
      //*/
      ////} else if(TREND_UP==false && TREND_DOWN==false)
      ////{
         //レンジ相場
         //内容は検討中です
      ////}

   }//end_of_if(OrdersTotal()<=max_position)


} else
{
   //［５分足］内で起こる急変に対応する処理を記載する
   //  バックテストでは確認が困難な部分

   if(OrdersTotal()>=1)
   {
      for(int ii=0;ii<OrdersTotal();ii++)
      {
         if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==false) break;
         if(OrderMagicNumber()!=Magic_ID || OrderSymbol()!=Symbol()) continue;

         if(OrderMagicNumber()==Magic_ID && OrderSymbol()==Symbol())
         {

            if(OrderType()==OP_BUY)
            {
               double i_kachi_buy_price=OrderOpenPrice()+profit_2;
               double i_make_buy_price=OrderOpenPrice()-loss_2;
               //
               bool i_kachi_buy=(Bid>=i_kachi_buy_price);
               bool i_make_buy=(Bid<=i_make_buy_price);
               //
               if(i_kachi_buy || i_make_buy)
               {
                  OrderClose(OrderTicket(),Lots,Bid,0,Blue);
                  continue;
```

```
                  }
               }

               if(OrderType()==OP_SELL)
               {
                  double i_kachi_sell_price=OrderOpenPrice()-profit_2;
                  double i_make_sell_price=OrderOpenPrice()+loss_2;
                  //
                  bool i_kachi_sell=(Ask<=i_kachi_sell_price);
                  bool i_make_sell=(Ask>=i_make_sell_price);
                  //
                  if(i_kachi_sell || i_make_sell)
                  {
                     OrderClose(OrderTicket(),Lots,Ask,0,Blue);
                     continue;
                  }

               }
            }
         }
      }//end_of_if(OrdersTotal()>=1)

   }

   return(0);
}
//////////////////////  以下は関数類  ////////////////////////////////////
bool IsNewBar()
{
datetime curbar = Time[0]; // Open time of current bar
if (lastbar!=curbar)
   {
    lastbar=curbar;
    return (true);
   }
  return(false);
}
//----------------------------------------
double slope_kaiki(int start_s,int period_s)
{
   ArraySetAsSeries(Price_01,true);
   Price_01[0]=Open[0];
   for(int j_=(start_s+1);j_<=((start_s+1)+(2*period_s));j_++)
   {
         Price_01[j_]=(1.0/4.0)*(Open[j_]+High[j_]+Low[j_]+Close[j_]);
   }
```

```
   double slope=kaiki_sen(Price_01,start_s,period_s,0);
   //
   return(slope);
}
//
double call_hensa(int start_h,int period_h)
{
   double koubai_M=kaiki_sen(Price_01,start_h,period_h,0);
   double seppen_M=kaiki_sen(Price_01,start_h,period_h,1);
   //
   double ooM=0;
   double sigma=0;
   for(int pM=start_h;pM<=(start_h+period_h);pM++)
   {
      Trend_[pM]=koubai_M*ooM+seppen_M;
      ooM=ooM+1;
      //
      double div_=(Price_01[pM]-Trend_[pM])*(Price_01[pM]-Trend_[pM]);
      sigma=sigma+div_;
   }
   double div=sigma/100;//100足あたりの値
   //Print内容は、logファイルにも記録されるので、バックテスト時に異常解析の邪魔になる
  // Print("勾配は＝　",koubai_M);
  // Print("回帰線からの偏差＝　",div);
  // Comment("勾配は＝ ",koubai_M,":回帰線からの偏差＝　",div);
//---
   return(div);
}
//----------------------------------------
//回帰直線の傾き、切辺を返す
double kaiki_sen(double& price_[],int start_,int period_,int what_)
{
   double X_av=0.0,X_i=0.0;
   double Y_av=0.0,Y_i=0.0;
   double i_D;
   //
   ArraySetAsSeries(price_,true);
   //まず、平均値を求める
   for(int i=start_;i<=(start_+period_);i++)
   {
      i_D=i*1.0;
      X_i=X_i+i_D;
      Y_i=Y_i+price_[i];
   }
      X_av=(X_i/(period_+1));
      //X_av=(start_+(start_+period_))/period_;
```

```
      Y_av=(Y_i/(period_+1));
   //加算
   double bunshi=0.0,bunbo=0.0;
   double koubai_=0.0,seppen_=0.0;
   double j_D;
   //
   for(int j=start_;j<=(start_+period_);j++)
   {
      j_D=j*1.0;
      bunshi=bunshi+(j_D-X_av)*(price_[j]-Y_av);
      bunbo=bunbo+(j_D-X_av)*(j_D-X_av);
   }
      koubai_=bunshi/bunbo;
      //////seppen_=Y_av-(koubai_*X_av);//良い値が出ない
      seppen_=price_[start_];
   //
   if(what_==0) return(koubai_);
   if(what_==1) return(seppen_);
   //
}
//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

以　上