

○ 「 アメンボ式 EA 開発法 ; その 2 」

★アメンボです、

「Bollin_EA_08.mq4」は、デモ・モードでランニングする準備中ですので、
本シリーズの解説は、「最適化とバックテスト」までの範囲内で行います。

○<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！

デモでのチェックは現在準備中

目次： 1. 本EAの基本仕様

(1) 基本仕様	・・・ 2 頁
(2) 未対応項目	・・・ 2 頁
(3) 特徴、と言うか「少し変わった」ところ	・・・ 2 頁
(4) トレンド検出に使用する指標	・・・ 2 頁
(5) 誤記訂正	・・・ 2 頁
(6) 機能していないコード部分	・・・ 3 頁

2. EAコードの解説

(1) 大きな構造について	・・・ 4 頁
(2) メイン部の構造（概要）	・・・ 5 頁
(3) 関数部の構造（概要）	・・・ 5 頁
(4) 「最適化」対象	・・・ 5 頁

3. EA作成手順（概要）

(1) 手順概要	・・・ 6 頁
(2) フィルターの効果例	・・・ 6 頁
(3) 「フ形」フィルター初期値の決定法（概要）	・・・ 7 頁

※本稿では、まず「概要（概念）」を解説します。

本EAの肝とも言うべき「トレンド検出に使用する指標」の詳細については、
次回の投稿で解説予定です。

※本稿では「Bollin_EA_08.mq4」のコードは改めて掲載していません、

コードの詳細は前回の投稿資料を参照ください。

※最適化については、次回「投稿」とします。

1. 本EAの基本仕様

(1) 基本仕様

・本EAは「Bollin_EA_08.mq4」コードを既に読まれた諸兄には既にお判りの様に、下記の基本仕様となっています。

- ①USDJPYの5分足をターゲットとする
- ②トレンド・フォロー型EA
- ③「カウント・ダウン方式」対応
- ④売買の建て玉は、Open時（New Bar）に限定
- ⑤決済タイミングは限定なし
- ⑥一度に持てるポジションは「一つ」に限定
- ⑦1足（5分足）内での通常変動に対応

※現時点ではバックテストを終えたばかりで、デモ・モードでのチェックをしていません。
（したがって、上記「③」を含め、全機能はこれから確認するところです。）

(2) 未対応項目・・・以下の項目には、対応していません

- ①レンジ相場
- ②トレンド・ブレイク
- ③「Old tick」発生時の処理
- ④相場急変時（為替介入時など）の処理

(3) 特徴、と言うか「少し変わった」ところ

- ①「profitとloss値」も最適化の対象とした。

(4) トレンド検出に使用する指標・・・本バージョンで使用の範囲

- ①ボリンジャー・バンド ; 2σラインのブレイク
- ②「フ」形フィルター ; アメンボのオリジナル品
正確には指標ではないのですが、
- ③「建て玉」から「次の建て玉」まで、一定の間隔を空けた

(5) 誤記訂正・・・下記のコメントは誤記です。

```
<誤記> }//end_of_if(OrdersTotal()<=max_position)
↓
<正> }//end_of_if(OrdersTotal()==0)
```

(6) 機能していないコード部分

- ・開発途上で検討中のため、以下のコード部は現時点では有効ではありません
一応、書いてありますが Bollin_EA_08 では使っておらず、採用するか否かは未決です。

```
// 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;

-----

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_);

-----

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);
}

-----
```

2. EAコードの解説

- ・アメンボの性癖として、先ず全体から見渡してから次に詳細に入っていきます。

(1) 大きな構造について

①本EA全体に共通するグローバル変数等の記述部

```
extern . . . ;
double . . . [];
int init() { . . . . . }
int deinit() { . . . . . }
int start()
{
```

②tick 値が NewBar (Open 値) であるか否かチェックする

```
if(IsNewBar() && (Bars>barsTotal))
{
```

③NewBar (Open 値) である場合の処理を記述

```
<< 「Open 値」 で条件を判断して、売買に I N し、または O U T する >>
} else
{
```

④NewBar (Open 値) ではなく、1 足内の tick 値である場合の処理を記述
(バックテストでは確認が困難な、ウイークポイント)

```
<< 「Tick 値」 で条件を判断して、『手仕舞い』 のみ行う >>
}
return(0);
}
```

//////////////////// 以下は関数類 //////////////////////

(A) New Bar か否かのチェック用関数

```
bool IsNewBar() { . . . . . }
```

(B) 「Price_01[]」 配列を作ってから「(C)」を呼び出し、傾き (勾配値) を返す

```
double slope_kaiki(int start_s, int period_s)
{ . . . . .
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);
}
```

(C) 配列データから回帰直線を推定する

```
//回帰直線の傾き、切辺を返す
double kaiki_sen(double& price[], int start_, int period_, int what_)
{ . . . . .
if(what_==0) return(koubai_);
if(what_==1) return(seppen_);
}
```

(2) メイン部「条件判断部」の補足

- ・アメンボは、「実際のトレード (リアル・トレード)」との対応が、最もよくとれる「バックテスト」方法は、下記の組合せと考えています。

EA構成；

- ① 「Open 値」で売買条件を判断し、売買にIN (建て玉) とOUT (手仕舞い) を行う
- ② 「1足」内の変動により、条件次第でOUT (手仕舞い) を行う

バックテスト・モデル；

「Every tick」モデル

(3) 関数部の概説

- ・「A」部；各足の「Open 値」が発生したかを検出する
- ・「B、C」部；「フ形」フィルター構成に必要な直線回帰線の勾配を計算する

(4) 「最適化」対象

- ・「最適化」を実施した対象は、下記の「15個のextern部分」です。
(「最適化」前の初期値は下記の設定値とは異なっていました)

```
<Bollin_EA_08.mq4>
// 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 IZYU=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;
//-----
```

※ 「最適化」実施の詳細は、「次校」に記載予定です。

3. EA作成手順（概要）

今回、アメンボのEA作成手順は以下の通りです。

(1) 手順概要；

ステップ1；

- ・まず、「売買シグナル発生」用の指標を決める（今回はボリンジャー・バンドを採用）
⇒「Bollinの2 σ バンド」と「Open・High・Low・Closeの平均値」が、
クロスするポイントとしました。

①平均値が2 σ を上抜けしたら、買いシグナル

②平均値が2 σ を下抜けしたら、売りシグナル

ステップ2；

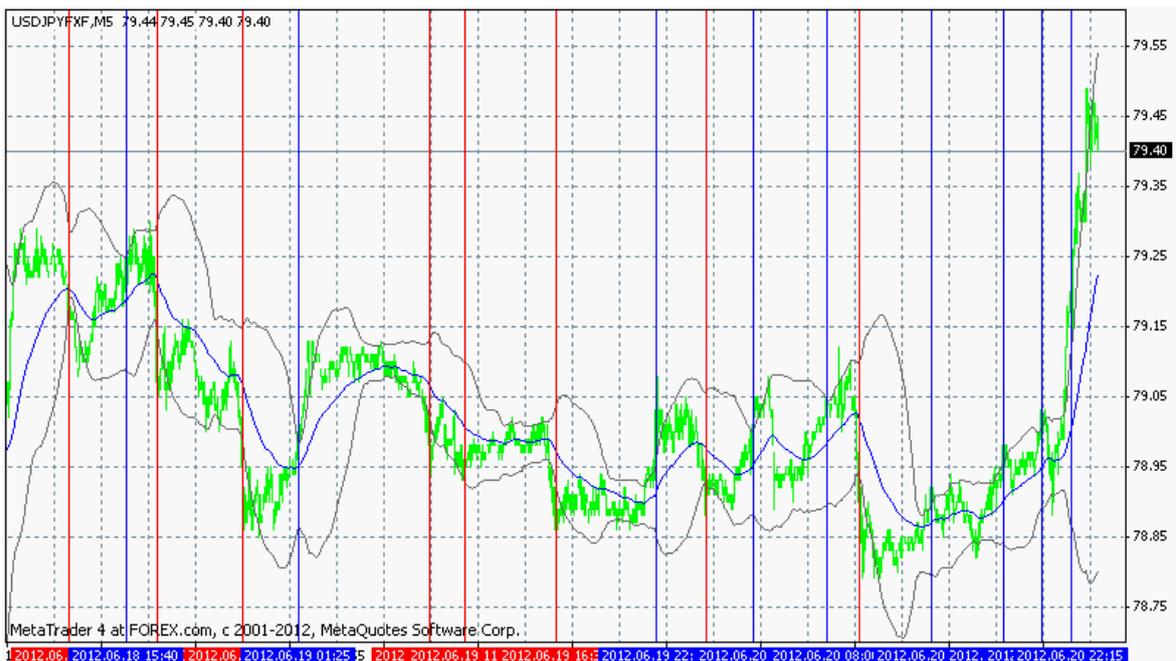
- ・次に、「フ形フィルター（アメンボのオリジナル）」で、
上記「シグナル・ポイント」が、売買ポイントとして適切か否かを個別に判断します。

ステップ3；

- ・フィルターされた後の「シグナル」で売買を実施する。

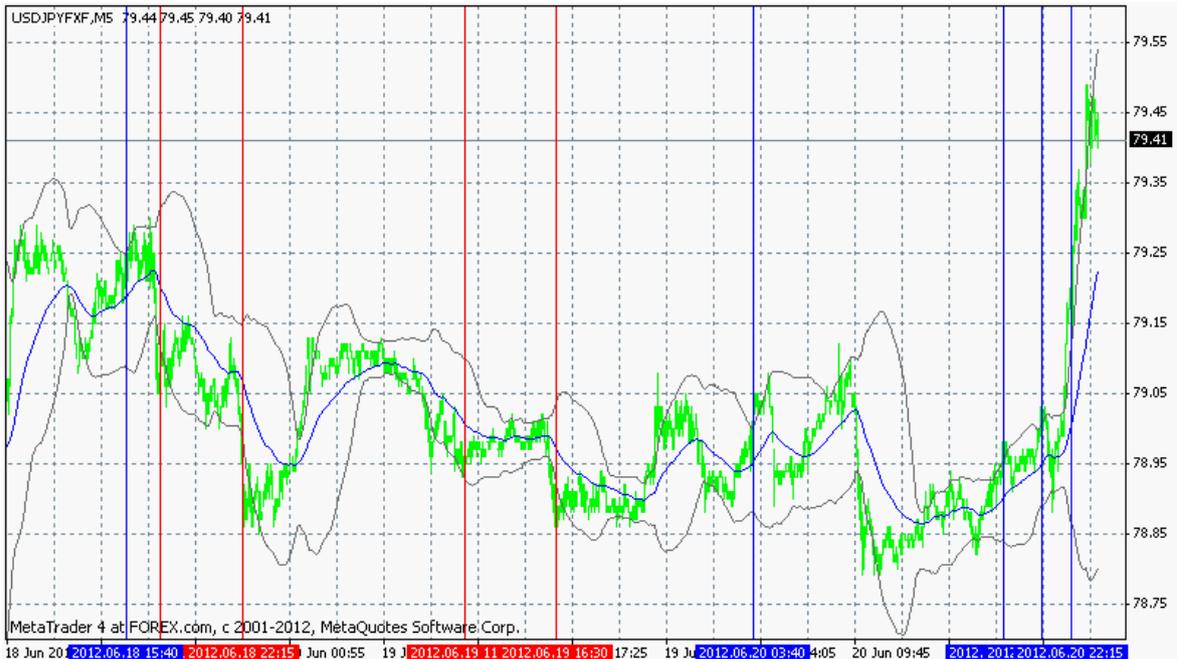
(2) フィルターの効果例

「ステップ1での売買シグナル」（青縦線＝買いシグナル、赤縦線＝売りシグナル）



「フ形」フィルターを挿入する

「ステップ3での売買シグナル」 フィルター挿入後；



※ 「フ形」フィルターにより、売買シグナルとしては不適切と思われるところは、排除されます。

(3) 「フ形」フィルター初期値の決定法 (概要)

< 「フ形」フィルター関連部コード >

```
// 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;
//-----
. . . . .
//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));
. . . . .
//Filter ; 本当に売るか
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);
. . . . .
```

目視判断による基準値決定・・・<以下の画像を覗てください>

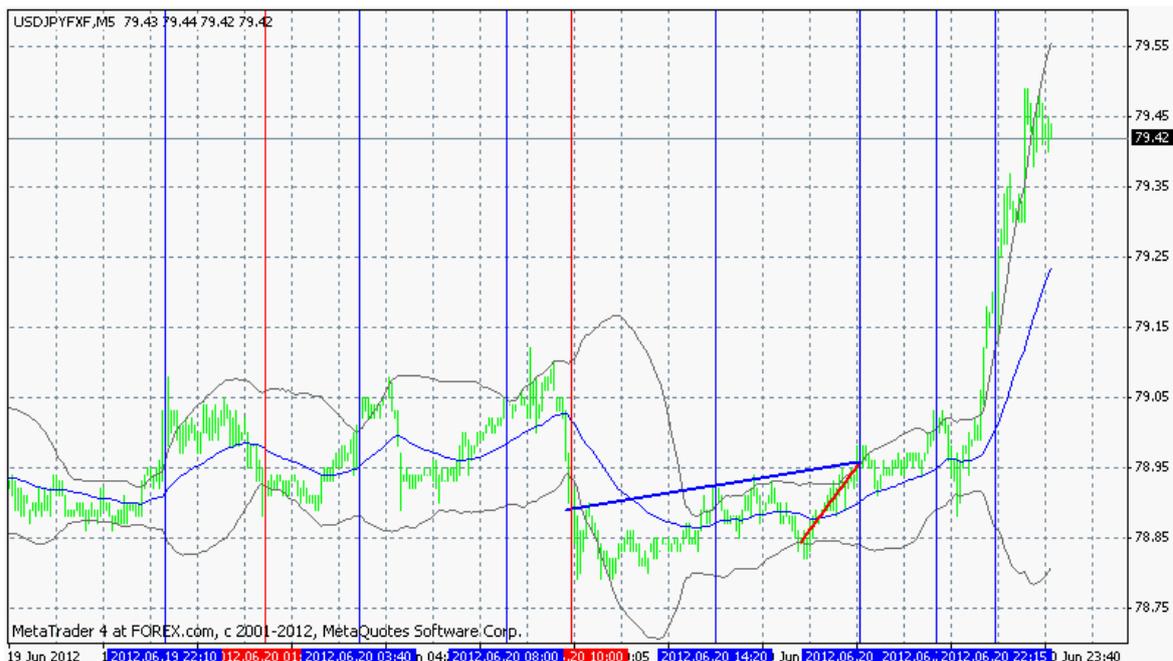
「slope_s (赤)」と「slope_m (青)」を回帰直線として表示したものです

- ・「目視」で判断すると、Aの場合は「買いシグナル」として妥当とされますが、Bは不適切と判断されるでしょう。

このように、「目視」で「妥当な勾配値」の組合せを探しました。

- ・同様に売りの場合は、CはNG、DはOKと「目視」判断できると思いますので、このときの「勾配値」を記録し、フィルターの判断基準値決定に使用しました。

画像A；買いシグナル・・・OKと判断



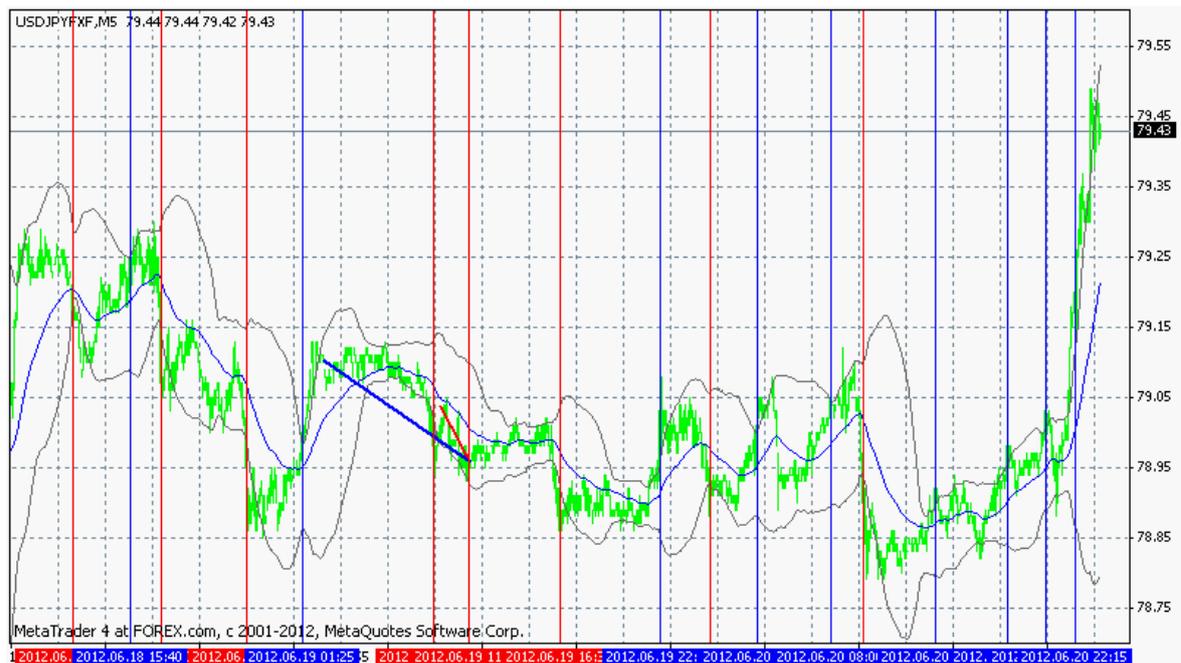
画像B；買いシグナル・・・NGと判断



画像C ; 売りシグナル・・・NGと判断



画像D ; 売りシグナル・・・OKと判断



※「目視」判断時のポイント；

- ・判断するシグナル部「以降のチャート」は存在しないものとして判断します。
(シグナル部から観れば、未来の部分になる訳ですか)

※「フ形」フィルターの「回帰直線」表示に若干の問題があるのですが、

試作段階のものでありますので、ご容赦ください。(定性的な把握と、お考えください)

以 上