

## ○「基礎の確認（その4）；フラクタル補間が見つからない！」

## ・アメンボです、

バックテストの自動化（前回投稿）を考えているうちに、以前から疑問に思っていた課題に深入りしてしまいました。（また、道草を食う癖が！）

## ・フラクタル補間って？、

小生、バックテスト時には無条件（反省もせずに）で「全ティック」を選択し続けています。『より短期のタイム・フレームのデータからフラクタル補間で補間ポイントが作られる』との資料を読むたび「フラクタル補間？」と気にはなっていましたが、調査した事は無しです。しかし、考えてみると「バックテストの信頼性」を検討する上では、「モデル」の限界を理解する必要性があり、今回道草してしまった次第です。

## ・MT 5 関連の資料では、

「3-5-3」 ウェーブ・モデル等を使って補間をしているとの資料や記事を見かけたのですが、未だ詳細に解説したことはありません。（未だ、MT5 を使ったことが無いし！）

## &lt;本稿で使用した MQL4 コード&gt;

※非常に単純なコードなので添付はしていません、本稿から「コピー、ペースト」で利用ください。（本稿は、「MT4 ; build509」にて確認済みのものです）

## 目次：

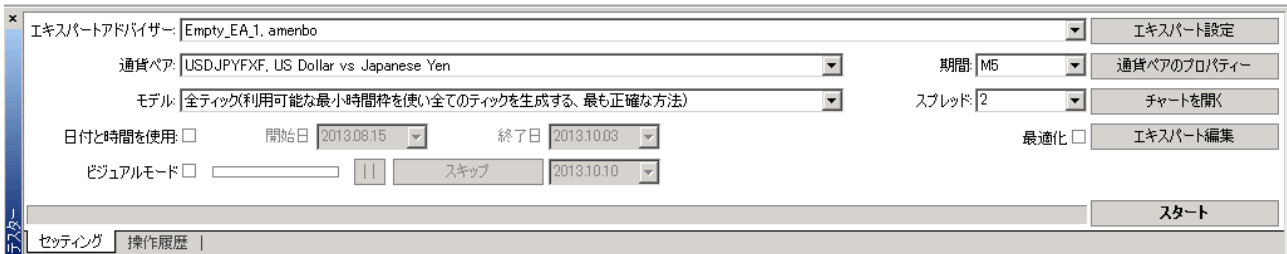
1. 「必要・最低限の知識」をおさらいする	・ ・ P 2
2. 生成されるファイル（データ）類の内容	・ ・ P 3
(1) 2回のプログレス・バー進行の理由	P 3
(2) 「. fxt ファイル」の名称	P 3
3. 各モデルでの「F X Tファイル（「. fxt」データ）とテスト・レポート」の関係	・ ・ P 3
(1) 先ず、「原データ（. hst）」と「3モデルの生成データ（. fxt）」を比較してみる	P 4
(2) バックテスト「モデル」で、何が違うのか？	P 5
(3) 幾つかの「F X Tファイル」観察結果	P 9
4. F X Tデータ（ファイル）と補間方法	・ ・ P 1 0
(1) 補間（コントロール）ポイント数の観察	P 1 0
(2) 補間点の Tick (Bid) 値は Close 値である ・ ・ Print での確認	P 1 5
(3) ビジュアルモードで確認してみる；M5全ティック. fxt	P 1 9
(4) コントロール・ポイントの生成過程イメージ（アメンボの推測）	P 2 1
(5) フラクタル補間とは？	P 2 2
(6) フラクタル補間を探したけれど！ 見つからない、そして	P 2 4
5. 補足；測定結果の追加	・ ・ P 2 8
	～ P 3 0

### 1. 「必要・最低限の知識」をおさらいする

- ・まずザッと、本稿での「解説対象に係る部分のみ」ですが、お馴染みのバックテストの各画面と実行「前後」でのおさらいと整理をします。(慣れている方は、読み飛ばしてもOKです)

バックテストを実行すると；

<テスト前>



タブ	設定項目・内容	詳細	備考
セッティング	エキスパートアドバイザー	E Aを設定	¥experts フォルダにある、コソバ 汎用 EA
	通貨ペア	通貨ペアを選択	マーケット・ウィンドウに表示中のもの
	モデル	3 種類から選択	・全ティック ・コントロールポイント ・始値のみ
操作履歴	バックテストの操作履歴	実行時に生成	¥tester¥logs フォルダに「日付.log」形式で保存される (テキスト)

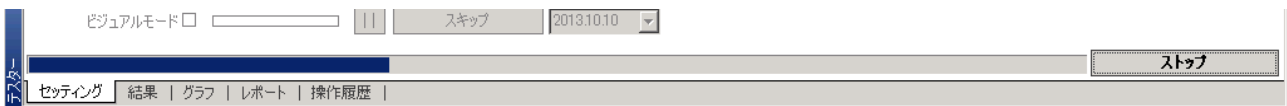
※ [エキスパート設定] は解説を省略。

<テスト中>・・・プログレス・バーは2回進行する！（モード [全ティック] の場合ですが）

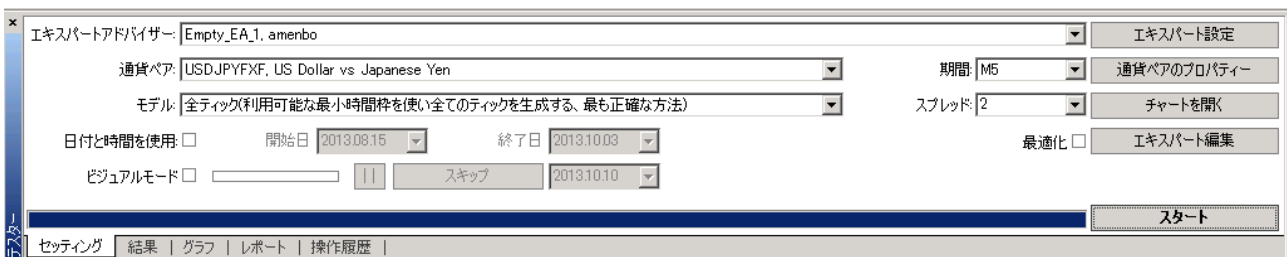
- ・ 1 回目；左上に「使用中 M1」、「使用中 M5」などと表示される



- ・ 2 回目；左上には何も表示されず



<テスト後>



タブ	設定項目・内容	詳細	備考
結果	全実行プロセスが保存される	[右クリック]-[レポートの保存]で StrategyTester.htm を保存	ファイル名は変更可能 保存先は任意フォルダ
グラフ	「損益」曲線を表示	[右クリック]-[画像として保存]で TesterGraph.gif を保存	ファイル名は変更可能 保存先は任意フォルダ
レポート	実行結果をレポート表示	[右クリック]-[レポートの保存]で StrategyTester.htm を保存	ファイル名は変更可能 保存先は任意フォルダ
(操作履歴)	(バックテストの操作履歴)	(Print 分などの出力)	

## 2. 生成されるファイル（データ）類の内容

### (1) 2回のプログレス・バー進行の理由

- ・モード「全ティック」でのバックテストの実行により、2個のプロセスが実行されます。  
 (モード「始値のみ」「コントロール・ポイント」だと、プログレス・バーの進行は1回です)

- ① 1回目で、利用可能な「最小タイムフレーム」のデータを収集し、
- ② 2回目のプログレス・バーで、ティック・データ「.fxt」作成とバックテストを実行。

### (2) 「.fxt ファイル」の名称；バックテスト時の「モデル」により3種類の名称に分かれます。

- ・バックテスト時の「モデル」とは、  
 売買トリガとなり得るティック・データの生成方法を示します。  
 つまり、限られた「Open、High、Low、Close、Volume」データから、如何にして  
 実市場におけるティック・データに近いものを、再現するかです。
- ・使用通貨ペアと周期（タイム・フレーム）；USDJPY、M15の場合

モデル（ティック・データの生成方法）	生成データ（ファイル）名称
始値のみ	USDJPYFXF15_2.fxt
コントロールポイント	USDJPYFXF15_1.fxt
全ティック	USDJPYFXF15_0.fxt

※「始値（Open）のみ」と言う、単純なモデルが残っている理由は、

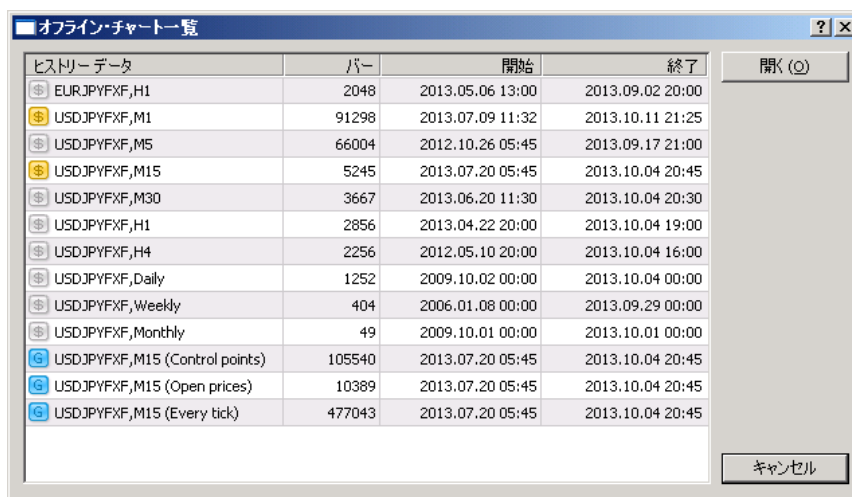
多くのEA（アルゴリズム）で、「始値（Open）」の値を売買トリガにしているため。

（アメンボは、「Openトリガ方式」が結構気に入っており、アルゴリズムには多用しています）

※「.fxt」生成例；・¥tester¥historyホルダ中



・確認法；[ファイル] - [オフラインチャート]



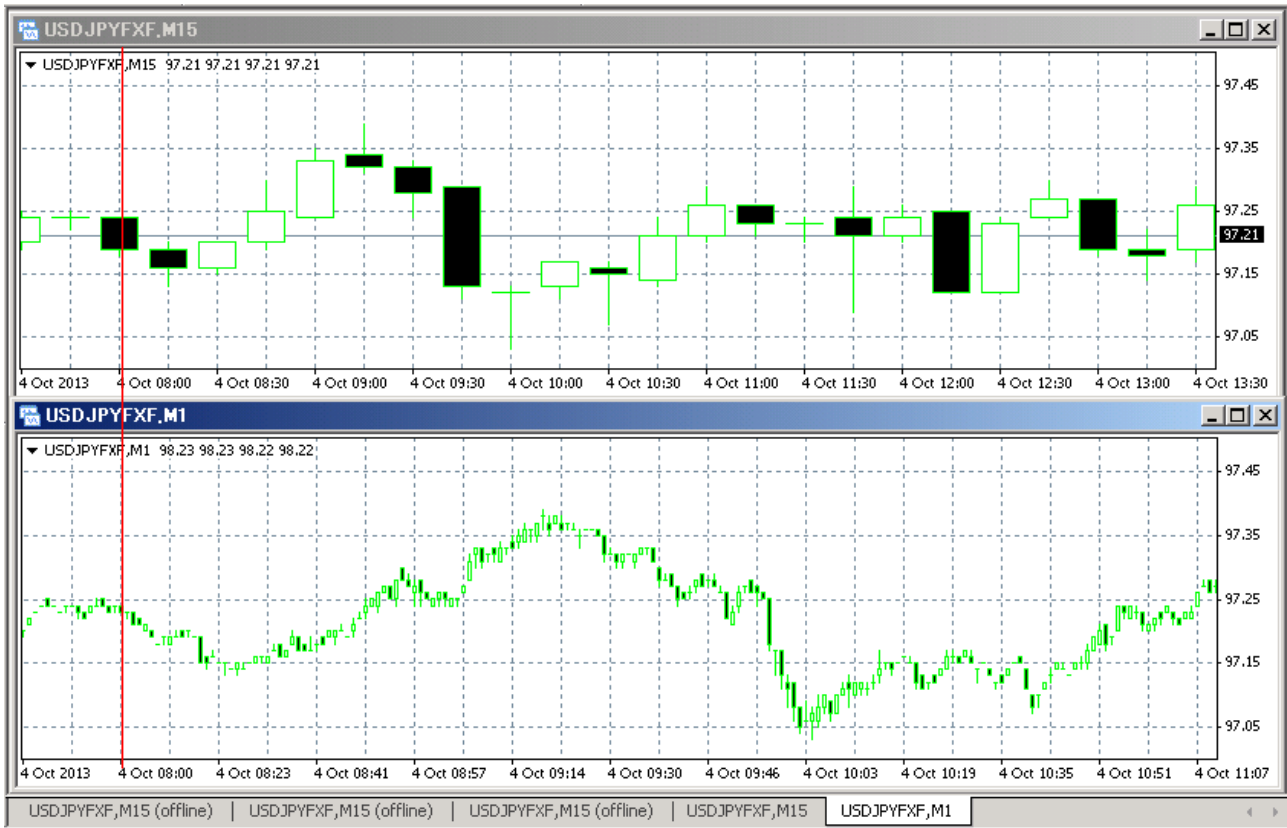
## 3. 各モデルでの「FXTファイル（.fxt）データ」とテスト・レポート」の関係

- ・そもそも、バックテスト時に「モデル」を指定することで、
  - ① どのような種類の「.fxt」（ティック）データが生成され
  - ② どのようなタイプの「バックテスト」が実行され、
  - ③ そして、「レポート」の内容は何を意味する、のでしょうか？

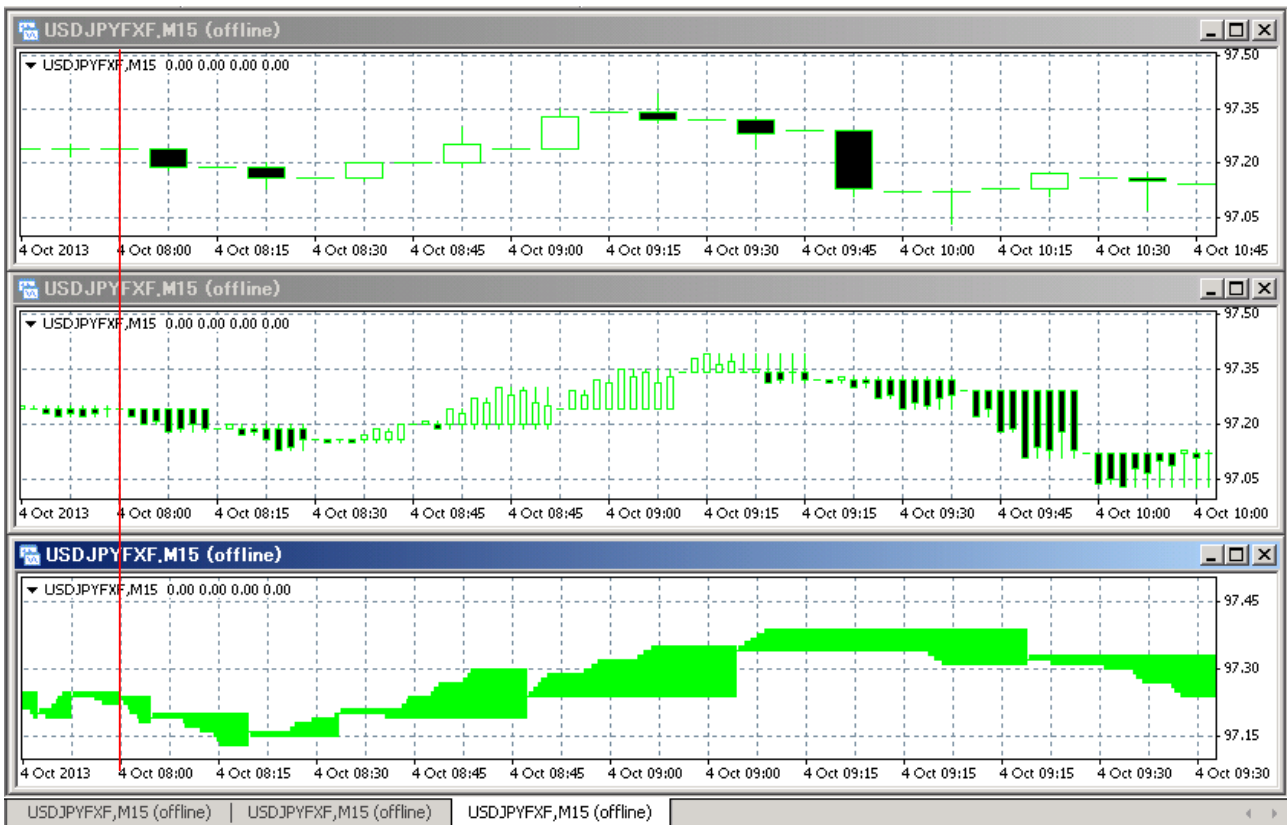
(1) 先ず、「原データ (.hst)」と「3モデルの生成データ (.fxt)」を比較してみる

※先ず、全体のイメージをチャート上で掴みます。

- ・上から、M15足「.hst」→M1足「.hst」；「赤い縦線=2013.10.04 08:00」



- ・M15 ; (初値のみ.fxt) → (コントロール.fxt) → (全ティック.fxt) ; 「赤い縦線=2013.10.04 08:00」



(2) バックテスト「モデル」で、何が違うのか？

※期間を指定して「空っぽのEA（P8；参照）による、バックテスト」を行い、イメージを観る。

エキスパートアドバイザー: Empty\_EA\_1.amenbo

通貨ペア: USDJPYFXF, US Dollar vs Japanese Yen

モデル: 始値のみ (最も早い方法。バーの始めにしか動かないEA向け)

期間: M15

スプレッド: 2

日付と時間を使用  開始日: 2013.08.15 終了日: 2013.10.03

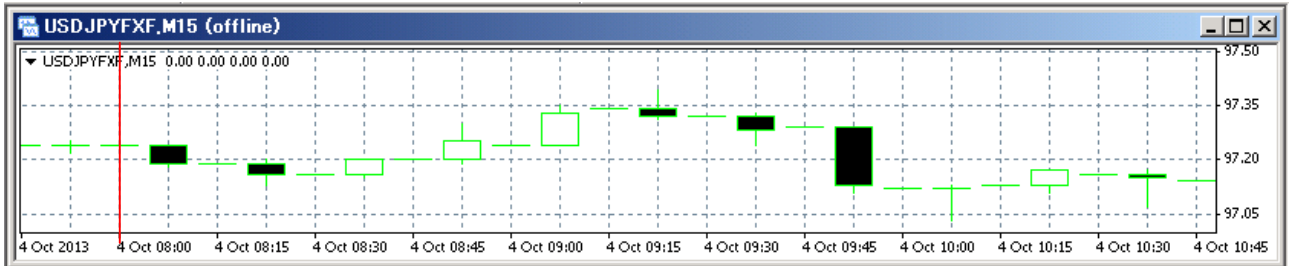
最適化

エキスパート設定  
通貨ペアのプロパティ  
チャートを開く  
エキスパート編集  
スタート

設定 | 結果 | グラフ | レポート | 操作履歴

F1キーでヘルプが表示されます Default 回線不通!

－ 1. 始値のみ (最も早い方法。バーの初めにしか動かない EA 向け) ;

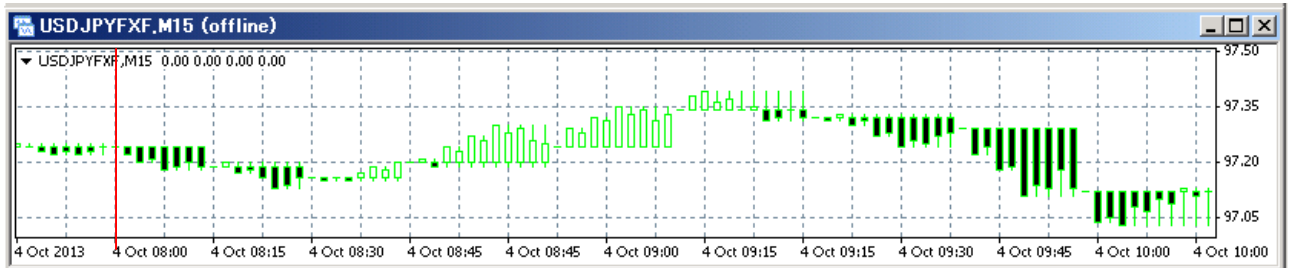


テストバー数	4361	モデルティック数	7720	モデリング品質	n/a
不整合チャートエラー	0				
初期証拠金	10000.00				
総損益	0.00	総利益	0.00	総損失	-0.00
プロフィットファクター		期待利得	0.00		
絶対ドローダウン	0.00	最大ドローダウン	0.00 (0.00%)	相対ドローダウン	0.00% (0.00)
総取引数	0	ショートポジション(勝率%)	0 (0.00%)	ロングポジション(勝率%)	0 (0.00%)
		勝率(%)	0 (0.00%)	負率(%)	0 (0.00%)
		最大 勝トレード	0.00	負トレード	-0.00
		平均 勝トレード	0.00	負トレード	-0.00
		最大 連勝(金額)	0 (0.00)	連敗(金額)	0 (-0.00)
		最大化 連勝(トレード数)	0.00 (0)	連敗(トレード数)	-0.00 (0)
		平均 連勝	0	連敗	0

設定 | 結果 | グラフ | レポート | 操作履歴

F1キーでヘルプが表示されます Default 回線不通!

－ 2. コントロールポイント (ひとつ下の時間枠を使ったおおまかな方法。結果はあまり信頼性は無い) ;



テストバー数	4361	モデルティック数	58792	モデリング品質	n/a
不整合チャートエラー	0				
初期証拠金	10000.00				
総損益	0.00	総利益	0.00	総損失	-0.00
プロフィットファクター		期待利得	0.00		
絶対ドローダウン	0.00	最大ドローダウン	0.00 (0.00%)	相対ドローダウン	0.00% (0.00)
総取引数	0	ショートポジション(勝率%)	0 (0.00%)	ロングポジション(勝率%)	0 (0.00%)
		勝率(%)	0 (0.00%)	負率(%)	0 (0.00%)
		最大 勝トレード	0.00	負トレード	-0.00
		平均 勝トレード	0.00	負トレード	-0.00
		最大 連勝(金額)	0 (0.00)	連敗(金額)	0 (-0.00)
		最大化 連勝(トレード数)	0.00 (0)	連敗(トレード数)	-0.00 (0)
		平均 連勝	0	連敗	0

設定 | 結果 | グラフ | レポート | 操作履歴

F1キーでヘルプが表示されます Default 回線不通!

－ 3. 全ティック (利用可能な最小時間枠を使い全てのティックを生成する。最も正確な方法) ;



－ 4. まとめ・・現状での理解範囲です (間違いがあれば、ご容赦)

I. 大雑把に把握 ; 15Mでのバックテストを想定

バックテストモード	M15の「1足」内に生成される足	バックテスト時の 売買トリガ
始値のみ	・各足の「始値」に斉しい「高値、安値、終値」と「ボリューム=1」を持つ「足」	生成された足の終値 (=始値)
コントロール・ポイント	① 5Mが存在する機関 ; ・ 先ず、5Mの足 (5分目、10分目) が、補間に使用され、 ・ 次に、各5分間の中が、たぶんボリューム数に応じて増減する? 足数で補間され (数個)、コントロール・ポイント (足) として生成されるように見える。 ② 5Mが存在しない期間 ; フラクタル補間で、「12個± (数個) ; ボリューム数考慮」のコントロール・ポイント (足) が生成される。	生成された足の終値
全ティック	① 1Mが存在する機関 ; ・ 先ず、1Mの足 (2分目、3分目、...) が、補間に使用され、 ・ 次に、各1分間の中が、たぶんボリューム数に応じて増減する? 足数で補間され (数個)、コントロール・ポイント (足) として生成されるように見える。 ② 1Mは存在しないが、5Mが存在する機関 ; ・ 先ず、5Mの足 (5分目、10分目) が、補間に使用され、 ・ 次に、各5分間の中が、たぶんボリューム数に応じて増減する? 足数で補間され (数個)、コントロール・ポイント (足) として生成されるように見える。 ③ 1Mも5Mも存在しない機関 ; フラクタル補間で、「12個± (数個) ; ボリューム数考慮」のコントロール・ポイント (足) が生成される。	生成された足の終値

II. 「Bars in test」の読み方； 「15Mでのバックテスト」を想定

※本稿では、バックテスト「レポート」の詳細解説（ネット上に情報は豊富にある）は省略して、特に気をつける部分、および意外な結果が観察された部分のみを、解説します。



A B C D

	「Bars in test」の色	意味
A部	灰色	バックテストの期間指定をしたため、使用されなかった「M15」の期間（部分）
B部	ピンク色	データの補間（フラクタル補間）に、M15自身のデータのみが使われた期間（部分）
C部	濃い緑色	データの補間（フラクタル補間）に、M5（ひとつ下の時間枠）が使われた期間（部分）
D部	明るい緑色	データの補間（フラクタル補間）に、M1（最小時間枠）が使われた期間（部分）

※疑問； なぜか、「始値のみ」の場合に「灰色」の表示が無い！

$$\begin{aligned} \text{※ [モデリング品質； \%]} &= ((0.25 \times \text{B期間の足数「.hst」}) + \\ &\quad (0.5 \times \text{C期間の足数「.hst」}) + \\ &\quad (0.9 \times \text{D期間の足数「.hst」})) \div \\ &\quad (\text{B期間の足数} + \text{C期間の足数} + \text{D期間の足数}) \end{aligned}$$

バックテスト期間全域を「M1足」がカバーする時 [モデリング品質] = 90% (最大値) となる。

※「M1」足の場合でも、「コントロール」と「全ティック」では、補間結果に違いがある。

※最初の100足は「補間」に使われない。(期間指定しない場合で試したのが下記)

(一番過去のデータ；8月19日) 部分；上から [全ティック、コントロール、始値のみ] の順で表示



## - 5. 空のEA

```

//+-----+
//|                                             Empty_EA_1.mq4
//|                                             amenbo
//|                                             泉の森の弁財天池
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"

extern int a=10;
extern double b=3.0;
//
int init()
{
    return(0);
}
//
int deinit()
{
    return(0);
}
//-----
int start()
{
    for(int i=0;i<5;i++)
    {
        double c=b+a*1.2;
    }
    //
    Print("a=", a, " : b=", b);
    return(0);
}
//-----

```



(3) 幾つかの「FXTファイル」観察結果

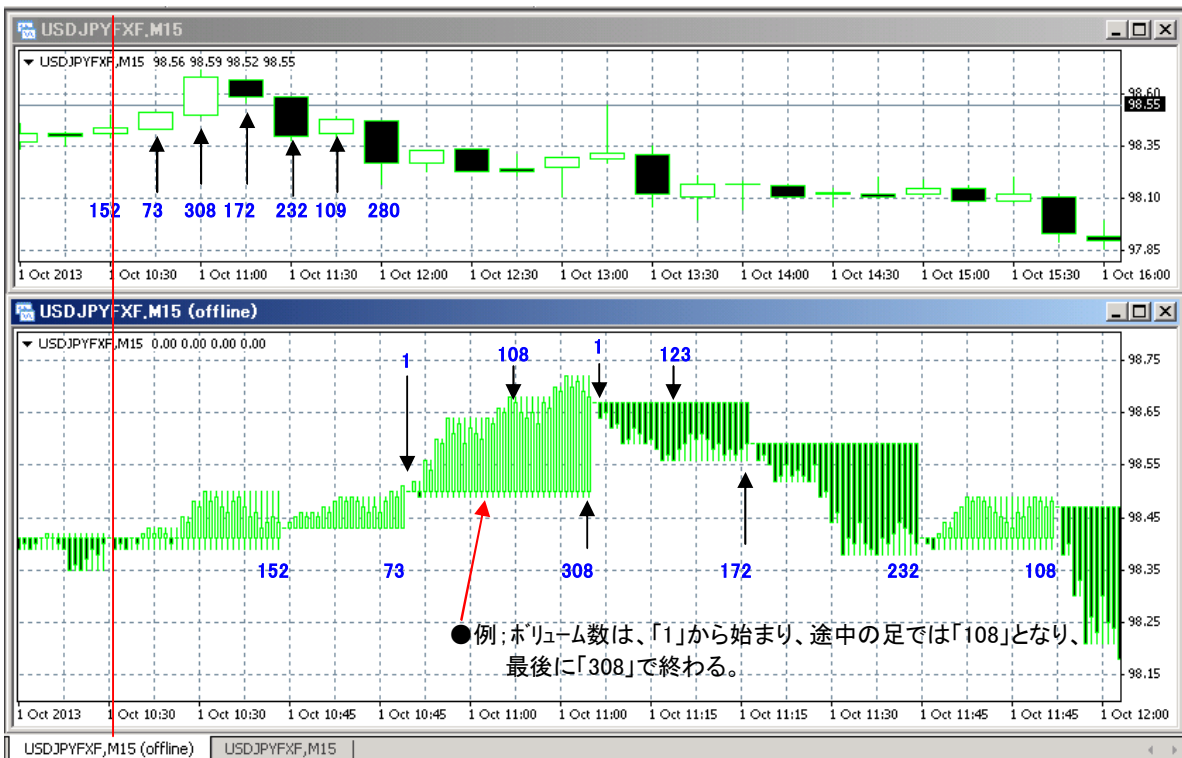
- ・「.fxt」データの生成規則を把握するために、幾つかの「検察結果」を示します。
- ・青い数字=ボリューム数、「コントロール・ポイント」と「全ティック」は最後の補間足での値

－ 1. 「M1 ; 1分足」観察例 ; 上から [始値のみ(.fxt)] → [コントロール・ポイント(.fxt)] → [全ティック(.fxt)]



※最短時間枠の1分足でも、「コントロール・ポイント」と「全ティック」では、補間で生成される足の数が異なる。「全ティック」では、生成される「補間足数」は「ボリューム数」を考慮しているように観える。

－ 2. 「M15 ; 15分足」観察例 ; 上から [通常の足(.hst)] → [コントロール・ポイント(.fxt)]



※「ボリューム数」により「補間足数」は増減している。「108」「123」は途中の補間足の「ボリューム数」。

#### 4. FXTデータ (ファイル) とフラクタル補間

##### (1) 補間 (コントロール) ポイント数の観察

※補間されるコントロール・ポイント数を幾つかの例で観察してみる。

ー 1. 「M15」M5データが存在しない (利用不可) 領域; [通常の足.hst] → [コントロール・ポイント(. fxt)]



※ 「赤い縦線」; カウント起点 (Open=High=Low=Close、Volume=1)

「青い縦線」; 補完されたコントロール・ポイントの終点

※規則性; (推測)

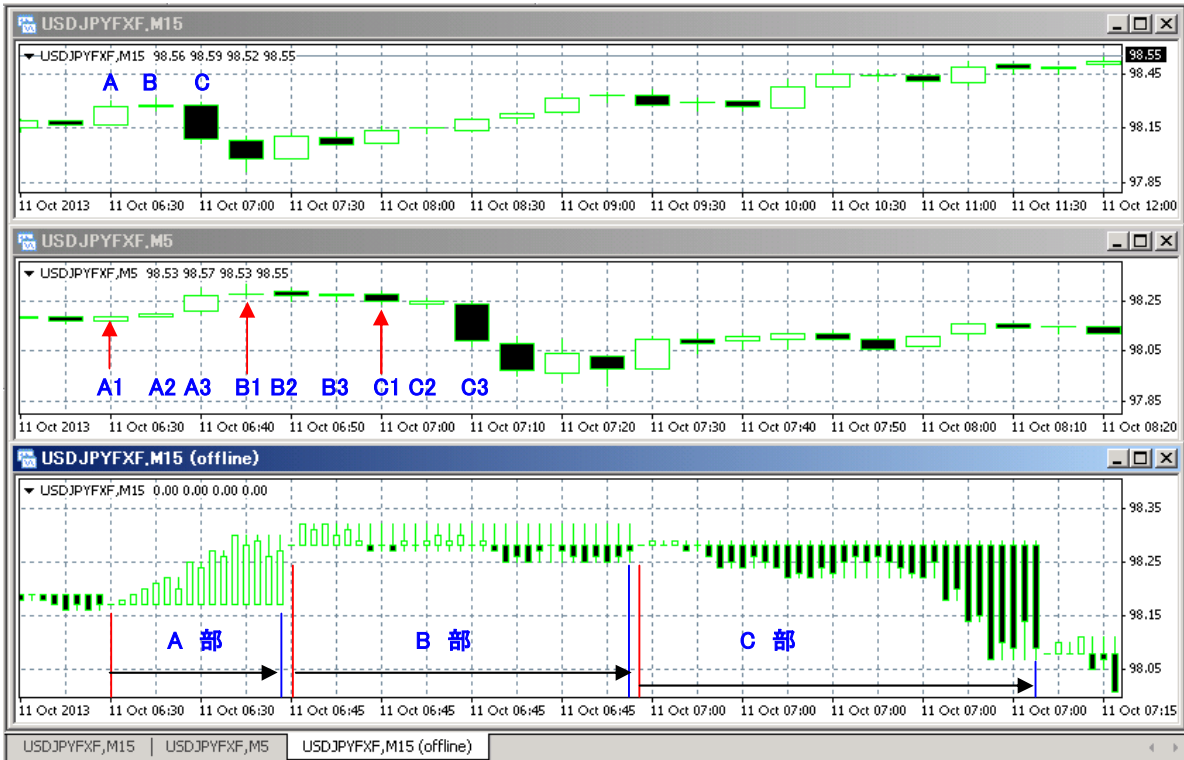
①補間前のM15.hst「1足」が、補間後のM15.fxtでは「12足(個)±数足」に増えた。

②それでは、全く参照するM5足の存在していない領域で、

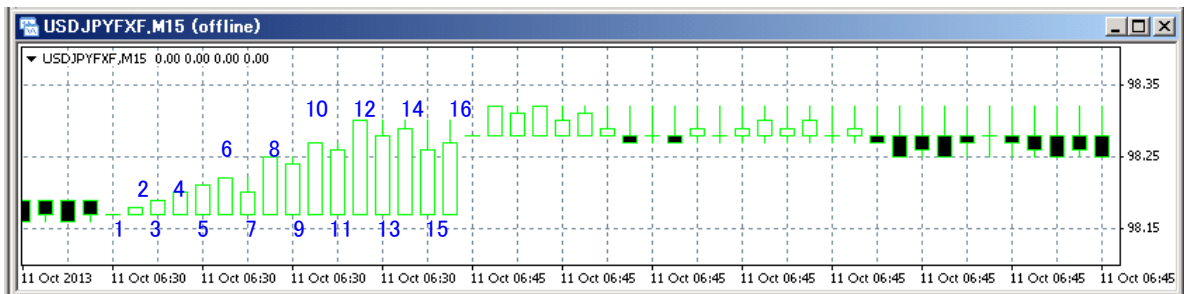
「如何なる根拠に基づいて、12個の補間ポイント(コントロール・ポイント)を生成するか」が問題となるわけで、

その根拠が、「フラクタル補間」である、と思ったのですが!

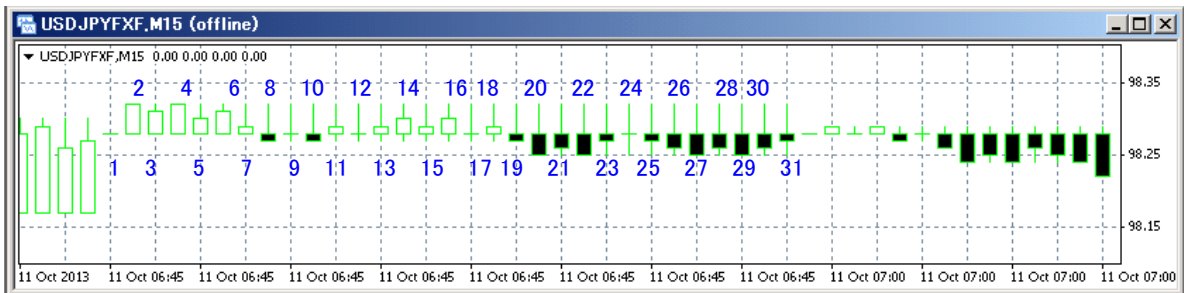
ー 2. 「M15」 M5データが存在する領域 ; M15.hst→M5.hst→M15.fxt コントロール・ポイント



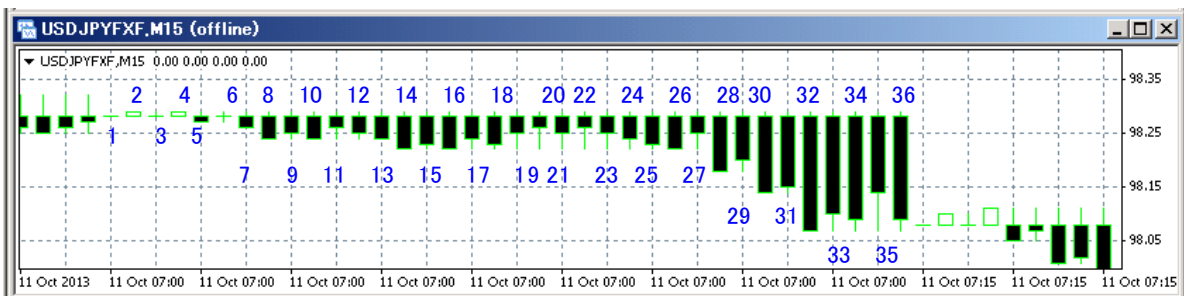
A部 (06:30);



B部 (06:45);



C部 (07:00);



<まとめ>

M15 .hst	A部				B部				C部			
足	A				B				C			
Time	06:30				06:45				07:00			
Open	98.17				98.28				98.28			
High	98.30				98.32				98.29			
Low	98.17				98.25				98.07			
Close	98.27				98.27				98.09			
Volume	60				239				243			
M5 .hst	A部				B部				C部			
足	A1		A2	A3	B1		B2	B3	C1		C2	C3
Time	06:30		06:35	06:40	06:45		06:50	06:55	07:00		07:05	07:10
Open	98.17		98.19	98.21	98.28		98.29	98.28	98.28		98.24	98.24
High	98.19		98.20	98.30	98.32		98.30	98.28	98.29		98.26	98.25
Low	98.17		98.19	98.20	98.27		98.25	98.25	98.29		98.22	98.07
Close	98.19		98.20	98.27	98.28		98.27	98.27	98.25		98.25	98.09
Volume	15		7	38	91		78	70	55		45	143
累計			22	60			169	239			100	243
M15 .fxt	A部				B部				C部			
コントロール・ポイント No.	1	3	4	16	1	12 = B1	23	31	1	12 = C1	23	36
Time	06:30	06:30	06:30	06:30	06:45	06:45	06:45	06:45	07:00	07:00	07:00	07:00
Open	98.17	98.17	98.17	98.17	98.28	98.28	98.28	98.28	98.28	98.28	98.28	98.28
High	98.17	98.19	98.20	98.30	98.28	98.32	98.32	98.32	98.28	98.29	98.29	98.29
Low	98.17	98.17	98.17	98.17	98.28	98.27	98.25	98.25	98.28	98.29	98.22	98.07
Close	98.17	98.19	98.20	98.27	98.28	98.28	98.27	98.27	98.28	98.25	98.25	98.09
Volume = 累計	1	18	22	60	1	91	169	239	1	55	100	243

※規則性 ; (推測)

①各足の Volume に注目すると、「B部とC部」にはコントロール・ポイント生成規則性が観られる。

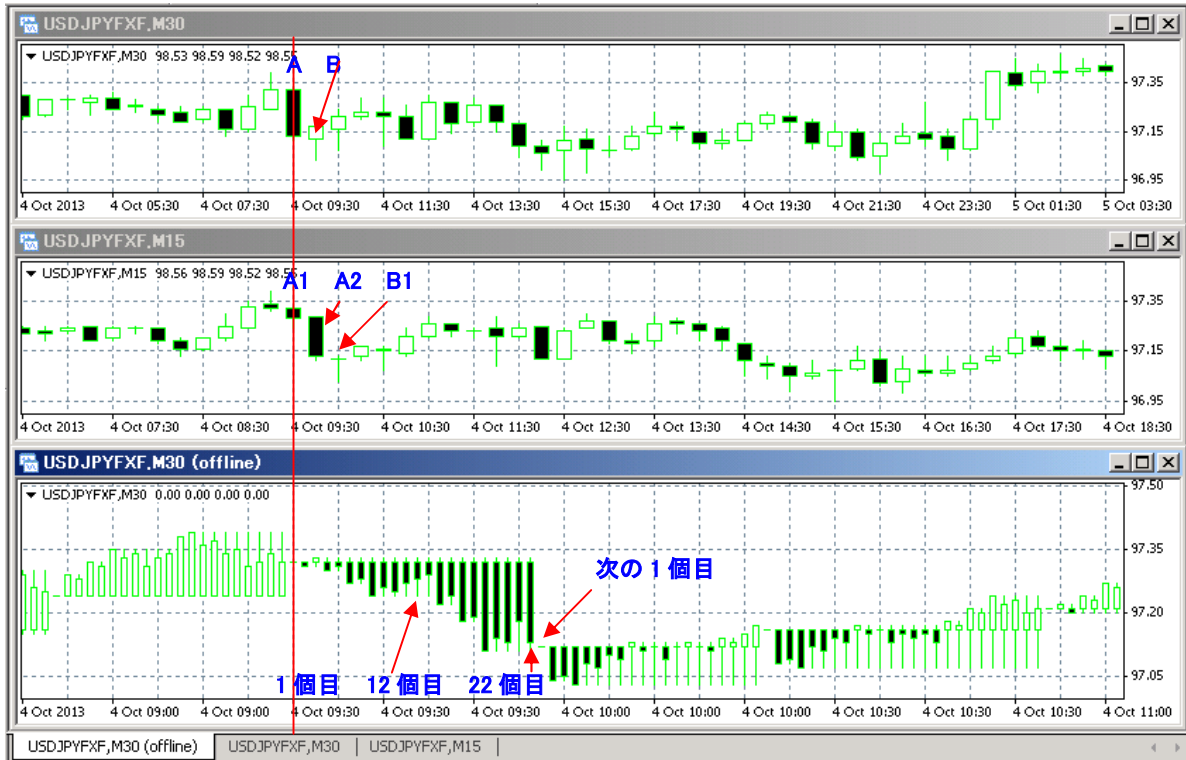
コントロール・ポイント No.	1	2~11	12	13~22	23	24~	終端	足総数	参考 Volume
B部		10個生成	=B1足	10個生成		24~30 ; 7個生成	31 =B足	31個	239
C部		10個生成	=C1足	10個生成		24~35 ; 12個生成	38 =C足	36個	243

②Volume 数が多いほうが、生成コントロール・ポイントは多いが比例関係ではない。(規則性は不明)

③A部 (足総数=22個、Volume=60) ; 規則性が読取れず。

※「生成足 (コントロール・ポイント)」は、フラクタル補間によるのだろうか??

－ 3. 「M30」M15データが存在している領域；M30.hst→M15.hst→M30.fxt コントロール・ポイント



<まとめ>

M30.hst	足	A								B		
	Time	09:30								97.12		
	Open	97.32								97.17		
	High	97.33								97.03		
	Low	97.11								97.12		
	Close	97.13								97.12		
	Volume	420								304		
M15.hst	足	A1		A2						B1		
	Time	09:30		09:45						97.12		
	Open	97.32		97.29						97.13		
	High	97.33		97.29						97.03		
	Low	97.24		97.11						97.12		
	Close	97.28		97.13						97.12		
	Volume	168		252						304		
M30.fxt	コントロール・ポイント	1個目		12個目				22個目		次の1個目		
	Time	09:30		09:30				09:30		10:00		
	Open	97.32		97.32				97.32		97.12		
	High	97.32		97.33				97.33		97.12		
	Low	97.32		97.24				97.11		97.12		
	Close	07.32		97.28				97.13		97.12		
	Volume	1		168				420		1		

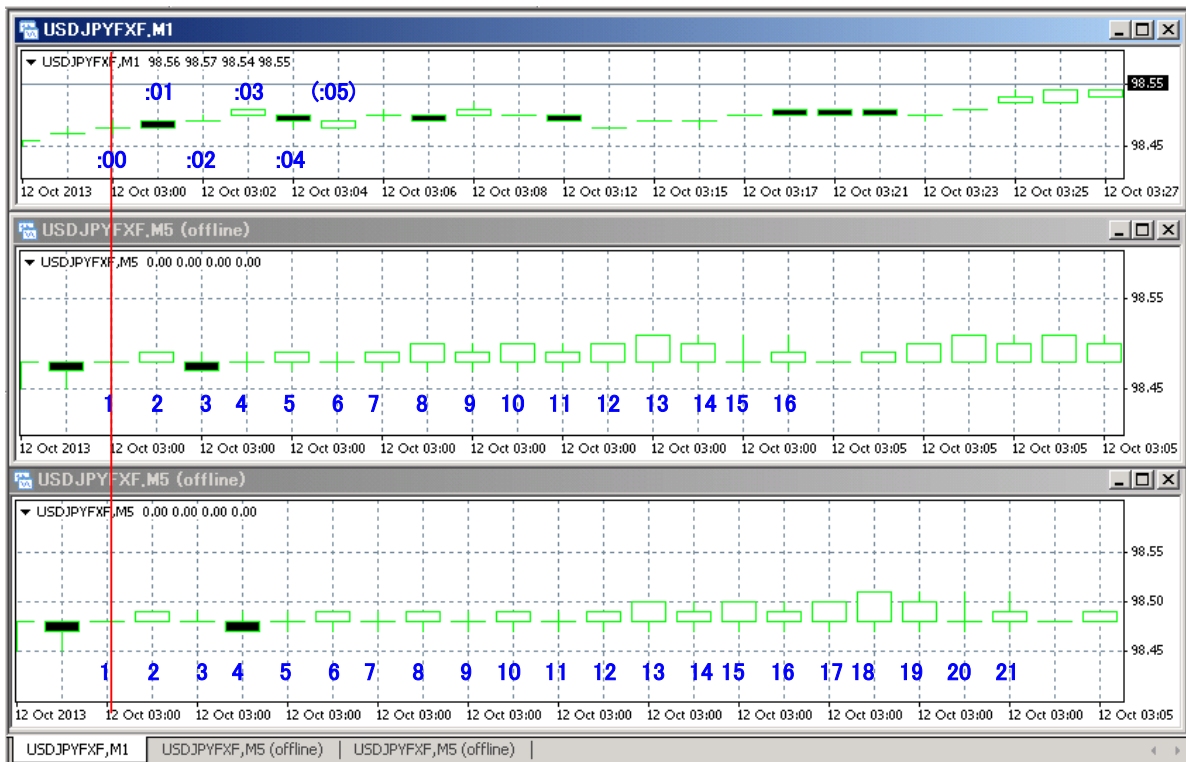
※規則性；(推測) M30.fxtのコントロール・ポイントの数(補間のされかた)は下記。

コントロール・ポイントNo.	1	2~11	12	13~21	22(終端)	足総数	Volume
		10個生成	=A1足	9個生成	=A足	22個	420

- ・ 1個目；Volume=1、(Open=High=Low=Close=97.32)という略「始値のみ」の値
- ・ 12個目；M15の足そのものを、中間の補間ポイントとして使用
- ・ 22個目；M30の足そのもの
- ・ 2~11個目、13~21個目；補完して生成している(フラクタル補間なの??)

－ 4. 「M5」 M1 データが存在している領域；

・ 上から、[M1 足.hst] → [M5 コントロール・ポイント.fxt] → [M5 全ティック.fxt]



M1 通常の足.hst	Time		:00	:01	:02	:03	:04	:05
	Open							
	Close		98.48	98.48	98.49	98.51	98.49	98.49
	Volume		6	6	8	7	5	8
	累積 volume			12	20	27	32	40
M5 コントロール.fxt	コントロール・ポイント No.	1	4	6	11	13	16	
	Open	98.48						
	Close	98.48	98.48	98.48	98.49	98.51	98.49	
	Volume	1	6	12	20	27	32	
M5 全ティック.fxt	コントロール・ポイント No.	1	5	11	16	18	21	
	Open	98.48						
	Close	98.48	98.48	98.48	98.49	98.51	98.49	
	Volume	1	6	12	20	27	32	

(2) 補間点の Tick (Bid) 値は Close 値である・・Print での確認結果

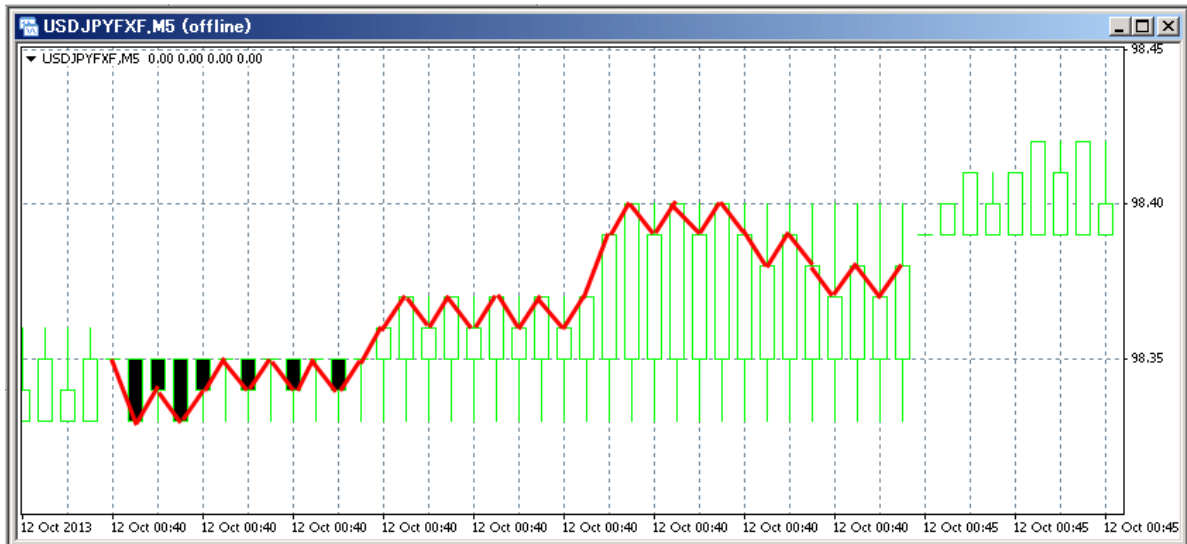
・ F X T データを、チャート表示した方は次のような疑問を持ちませんでしたか？

『この奇妙な「足の列」は、何??』

補間で生成された、F X T データ (コントロール・ポイント、又はティック) は、  
バックテスト時にどのように機能するのだろうか？

※結論から言うと、各「足の列 (コントロール・ポイント)」の Close 値が、Bid (=Tick) として扱われます。(下図参照)

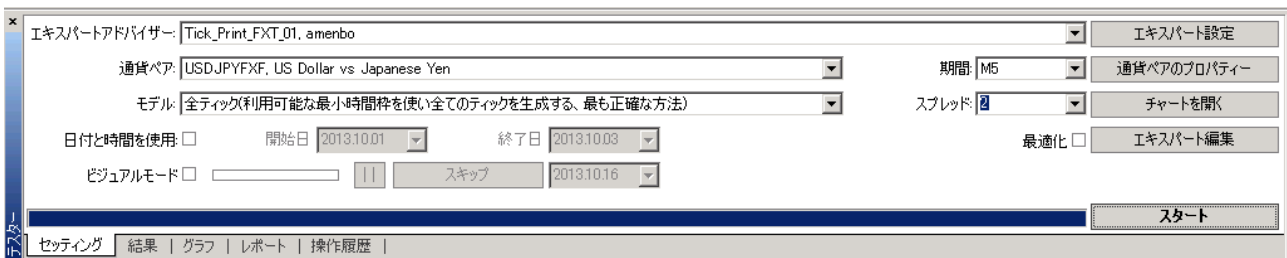
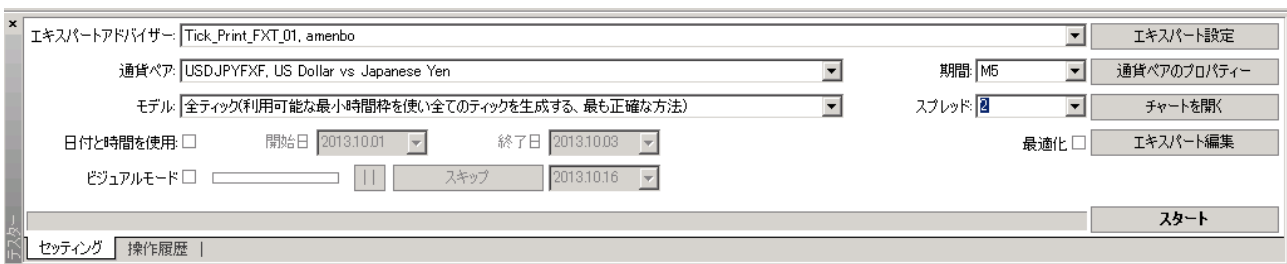
ー 1. 図示 (チャート) ; M5 全ティック.fxt 「2013. 10. 12. 00:40~00:45」 区間



※赤線で示す「F X T データ (足) の Close 値」が、Bid (=Tick) となります。(ラインチャート)

ー 1. 上記を確認してみる

①設定 (実施) 画面 ;



## ②確認コード； 上記の確認用のMQL4コード（EA）

```
//-----
// Tick_Print_FXT_01.mq4
// amenbo
// 泉の森の弁財天池
//-----
#property copyright "amenbo"
#property link "泉の森の弁財天池"
//
int tick;
static datetime lastbar;
//
datetime from_=D' 2013.10.12 00:40'; //←●ここを変更して調査範囲を変える
datetime to_=D' 2013.10.12 00:45'; //←●ここを変更して調査範囲を変える
//
int init()
{
    tick=0;
    lastbar=Time[1];

    return(0);
}
//
int deinit()
{
    return(0);
}
//
//プリントする「M5足」の時間を指定する
//
int start()
{
    if((from_ <= TimeCurrent()) && (TimeCurrent() <= to_))
    {
        if(IsNewBar())
        {
            Print("Number of ticks in minute = ", tick);
            Print("===== START =====");

            tick=0;
        }

        tick++;

        Print("TimeMinute(**) : tick --> ", TimeMinute(TimeCurrent()), "分 : ", tick, "回");
        Print("Open=", Open[0], ":High=", High[0], ":Low=", Low[0], ":Close=", Close[0], ":Volume=", Volume[0]);
        Print("Bid=", Bid, ":Ask=", Ask);

    }
    //
    return(0);
}
//=====
bool IsNewBar()
{
    datetime curbar = Time[0]; // Open time of current bar
    if (lastbar!=curbar)
    {
        lastbar=curbar;
        return (true);
    }
    return(false);
}
//-----
```



## ③出力結果；

```

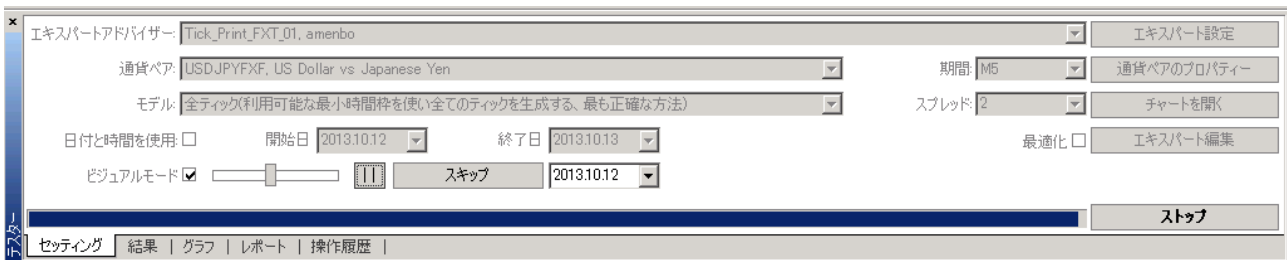
22:39:36 Tick_Print_FXT_01: loaded successfully
22:39:41 TestGenerator: spread set to 2
22:39:52 TestGenerator: unmatched data error (low value 98.65000 at 2013.08.05 23:20 is not reached from the least timeframe,
low price 98.66000 mismatches)
22:39:53 Tick_Print_FXT_01 test started
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Number of ticks in minute = 0
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: ===== START =====
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 1回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.35:Close=98.35:Volume=1
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.35:Ask=98.37
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 2回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.33:Volume=2
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.33:Ask=98.35
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 3回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.34:Volume=3
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.34:Ask=98.36
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 4回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.33:Volume=4
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.33:Ask=98.35
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 5回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.34:Volume=5
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.34:Ask=98.36
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 40分 : 6回
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.35:Volume=7
22:39:56 2013.10.12 00:40 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.35:Ask=98.37
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 41分 : 7回
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.34:Volume=8
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.34:Ask=98.36
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 41分 : 8回
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.35:Volume=9
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.35:Ask=98.37
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 41分 : 9回
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.34:Volume=10
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.34:Ask=98.36
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 41分 : 10回
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.35:Volume=11
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.35:Ask=98.37
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 41分 : 11回
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.34:Volume=12
22:39:56 2013.10.12 00:41 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.34:Ask=98.36
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 42分 : 12回
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.35:Low=98.33:Close=98.35:Volume=14
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.35:Ask=98.37
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 42分 : 13回
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.36:Low=98.33:Close=98.36:Volume=16
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.36:Ask=98.38
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 42分 : 14回
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.37:Low=98.33:Close=98.37:Volume=17
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.37:Ask=98.39
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 42分 : 15回
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.37:Low=98.33:Close=98.36:Volume=19
22:39:56 2013.10.12 00:42 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.36:Ask=98.38
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 43分 : 16回
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.37:Low=98.33:Close=98.37:Volume=23
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.37:Ask=98.39
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 43分 : 17回
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.37:Low=98.33:Close=98.36:Volume=24
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.36:Ask=98.38
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: TimeMinute(***) : tick --> 43分 : 18回
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Open=98.35:High=98.37:Low=98.33:Close=98.37:Volume=25
22:39:56 2013.10.12 00:43 Tick_Print_FXT_01 USDJPYFXF,M5: Bid=98.37:Ask=98.39

```

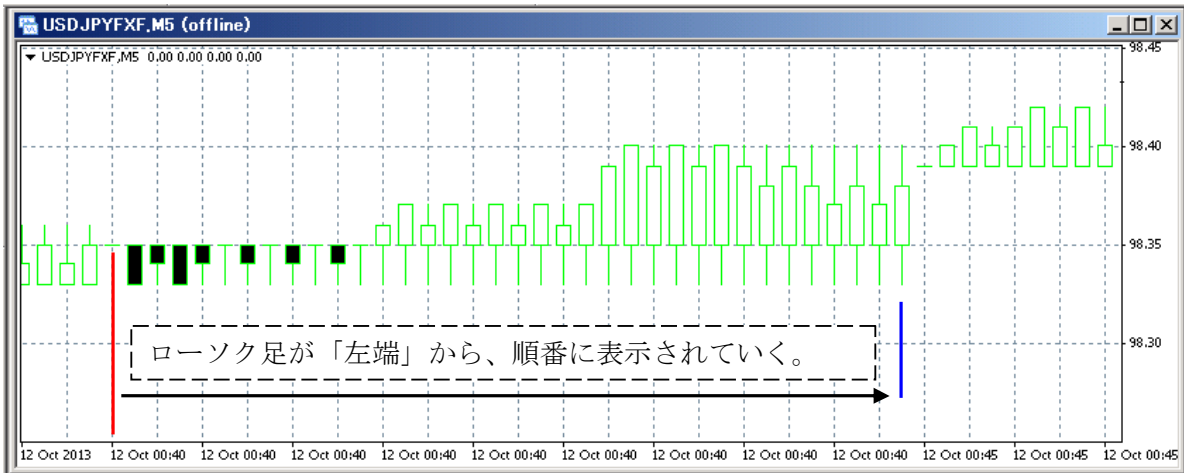


(3) ビジュアルモードで確認してみる ; M5全ティック.fxt

ー 1. 設定 ; ビジュアルモードのスピードを [設定バー] で、ゆっくりを設定する。



ー 2. 動作 ; [スタート] 後に、[スキップ] して、目的の期間を表示させる。



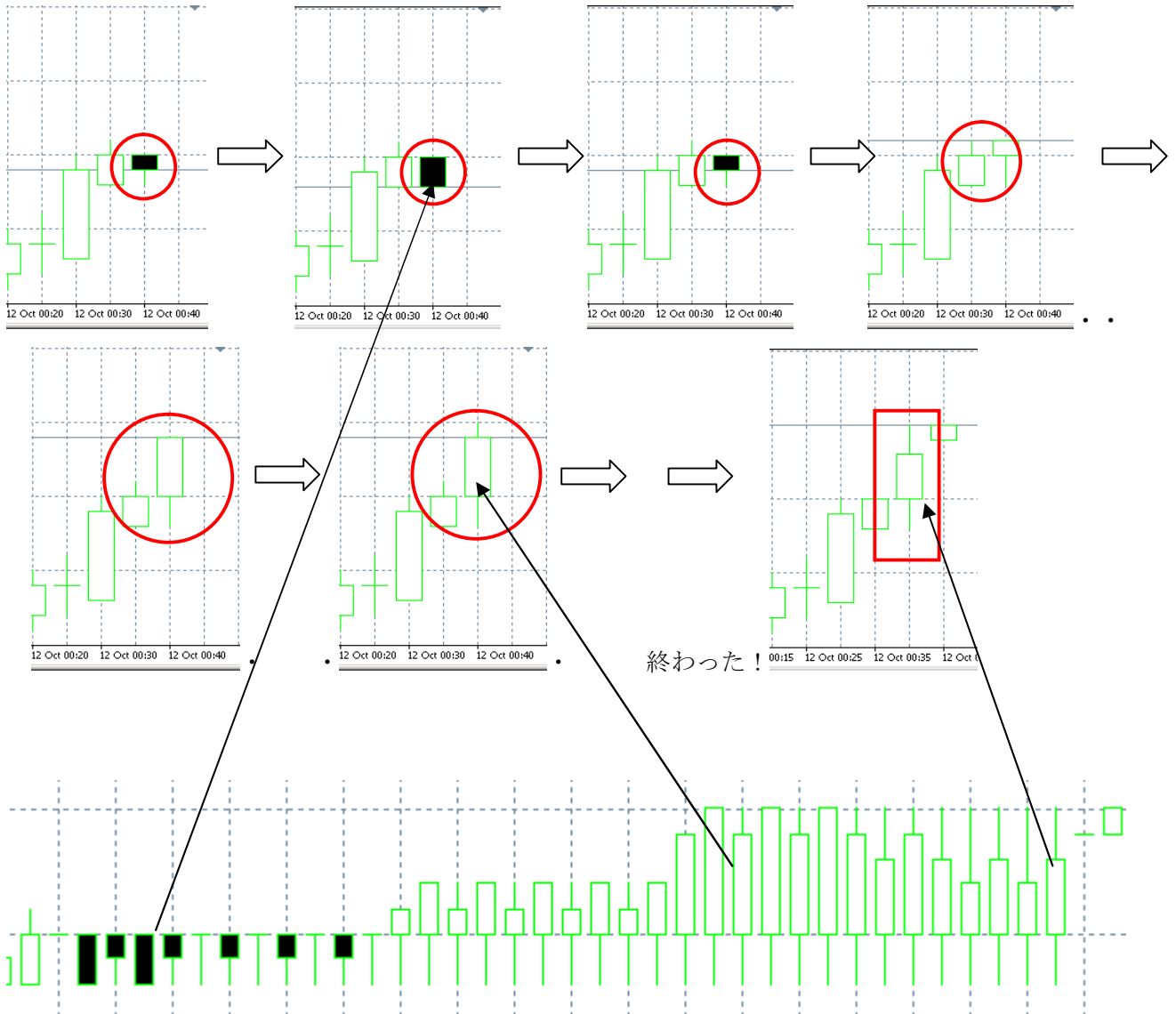
①スタート



② 1ティック進んだ (1コントロール・ポイント)



③その後の、ティック進行を追いかけた



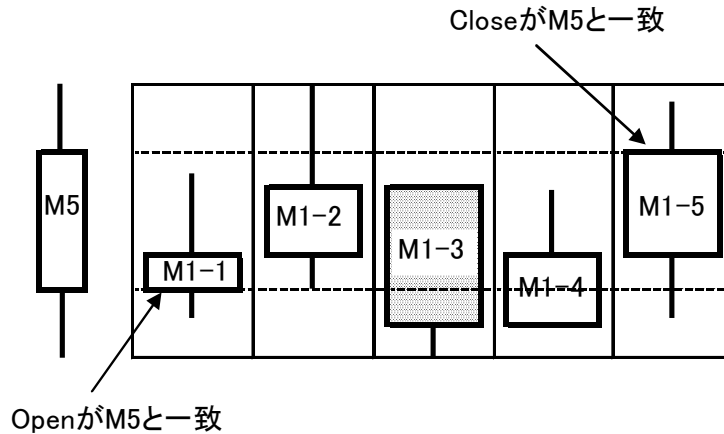
※要するに、FXTデータ（ファイル）のコントロール・ポイント（ティック）は、ビジュアルモードでティックとして観察される。

(4) コントロール・ポイントの生成過程イメージ (アメンボの推測)

※M5 (.hst) から、コントロール・ポイントや全ティックで、バックテスト用の擬似ティックを生成する場合で考察します。

※本「推測」は、記述済み各種「観察結果」を総合的に考察した結果です。(間違いあれば、ご容赦)

－ 1. 考察・・・M5 (.hst) とM1 (.hst) の関係例

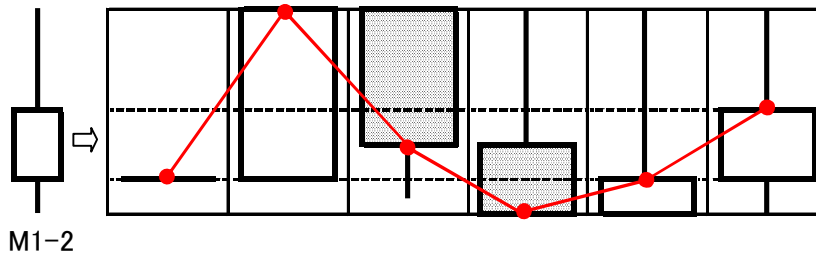


※M5 の Volume = 「(M1-1)～(M1-5)」 の Volume 値合計

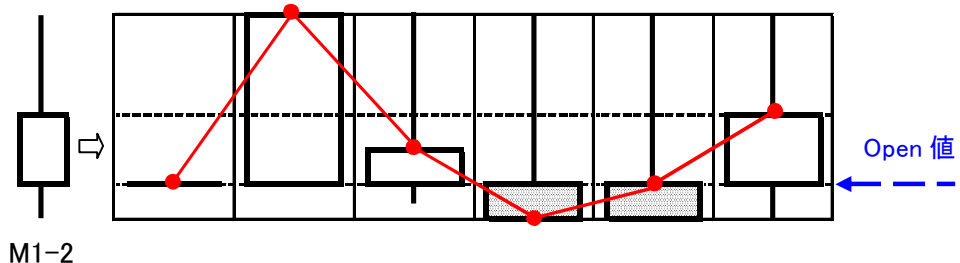
－ 2. 考察・・・「2番目のM1足」を抜き出して、補間を試みる。

※補間の値 (Bit=Close) は決定済みと仮定します。

①通常の足が生成されるイメージ (上記のM5とM1の関係) で補完すると



②アメンボが推測する「コントロール・ポイント」生成方法で表現すると



※全て、「M1-2」の Open 値から、

各コントロール・ポイントの Close 値までの差分で表現される。

※「●; 赤丸」がティックの動きを表す。

※推測根拠; 「F X Tデータ」と「ビジュアル・モード」での表示による

## (5) フラクタル補間とは？

<「参照・引用」文献>

※これまで、MQL 投稿は学術書ではありませんから投稿記事を執筆するに際して「参考・引用」文献名称を書いたことはありませんでした。(それにアメンボは MQL4 に関しては実証主義ですので)ただ、課題によっては、諸兄の更なる解析を可能とするために「参考・引用」文献を記載しておくほうが良いのではと、と考え始めました。

本件がまさにその「課題」と判断し、有用と思われる資料名のみ掲載しておきます。

(ネット検索すれば、直ぐに手に入ると思います)

- ①Strategy Tester: Modes of Modeling during Testing 2005.09.13 MetaQuotes Software Corp.
- ②Articles4\_En Peculiarities.pdf
- ③The Algorithm of Ticks' Generation within the Strategy Tester of the MetaTrader 5 Terminal  
MetaQuotes 2 June, 2010

② ; 「フラクタル補間」について、アメンボが知り得た唯一の資料です。

(他にもあるのかも? でも、この解説が簡単すぎて、未だに良く判らない。)

③ ; 「3-5-3」 ウェーブ・モデル等の解説書。

(一読すると、このアルゴリズムは MT5 でのみ使用され、Mt4 では関係ないように思えた、**が実は!**)

#### ー 1. 上記の資料から要約 (意識)

※補間とは、「1 足」内を更に細かく分割して、擬似 Tick (Close 地) 点を生成すること。

( 「Tick=コントロール・ポイント」と見做しても OK )

#### コントロール・ポイント (最も近い短期の時間枠を使用する) ;

・最も近い短期の時間枠の .hst 足データが「ある領域」では、これを利用して補間点を生成する。

例 ; M1 5 ならば、M5 データがある領域

・最も近い短期の時間枠の .hst 足データが「無い領域」では、直近の 1 2 足のデータを利用して補間を行う。

例 ; M1 5 ならば、M5 データが無い領域では、直近の 1 2 個の M1 5 足 (の Close 値) のパターンを使って、「足」と「次の足」の間を補間する。(同じパターンを繰り返す)

#### 全ティック (その領域で存在する最小時間枠 .hst 足データを利用する) ;

・その領域で存在する最も短期の時間枠 .hst データを利用する。

例 ; M30 なら、M1、M5、M15 足データ (Close 値と Volume) を利用して補完点を生成する。  
補間点と補間点の間は、更に補間して tick を生成する。  
重複したデータは、Volume 値を参照して篩いにかける。

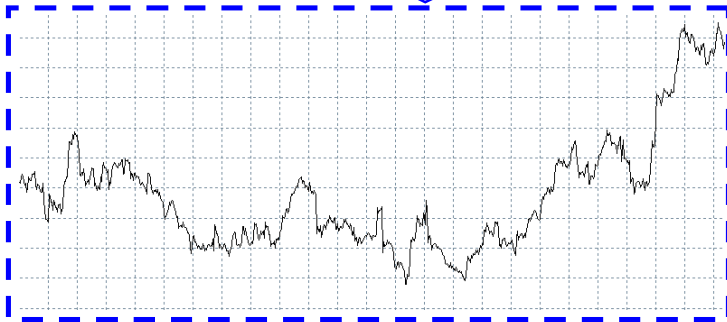
M30 のデータしか存在しない領域ではモード「全ティック」では、  
2 回コントロール・ポイントを生成したようなイメージで補間点が発生する。

ー 2. 「フラクタル」って何だ？

- ・為替チャートで解説すると、事象が「ランダム」状態にあるならば(明確なトレンドが無いなら)、チャート全体を見渡しても、一部を拡大しても「同じ様なパターン」が出現するということです。  
例；



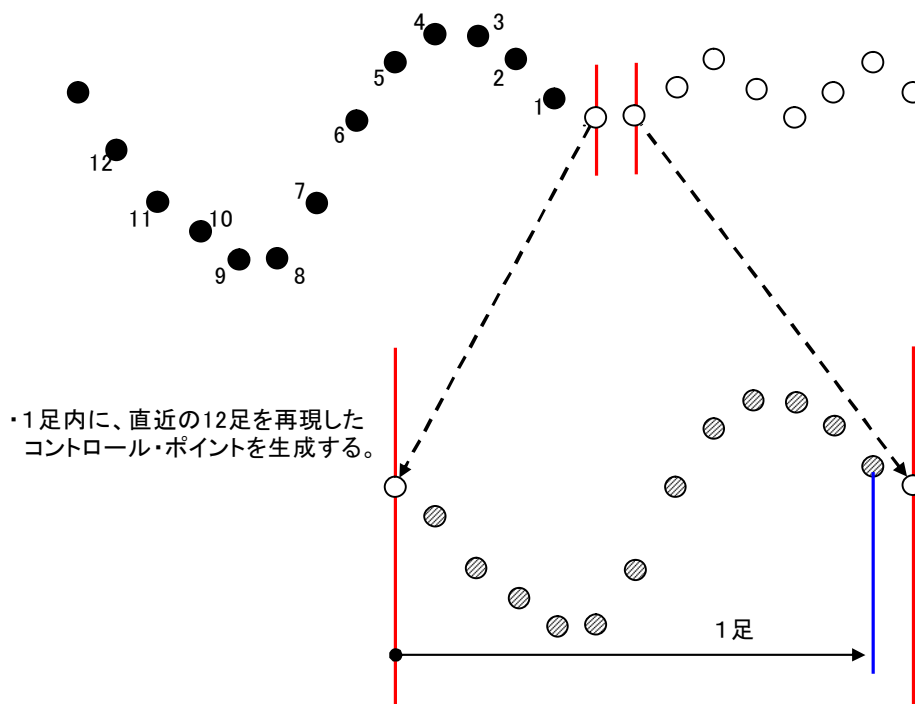
窓枠内を拡大



・ランダム事象であれば、拡大しても、同じ様なパターンが出現する(フラクタル)、と、言われているんですが、明確な例を探すのは困難。

ー 3. フラクタル補間があるなら、こんなイメージのはず

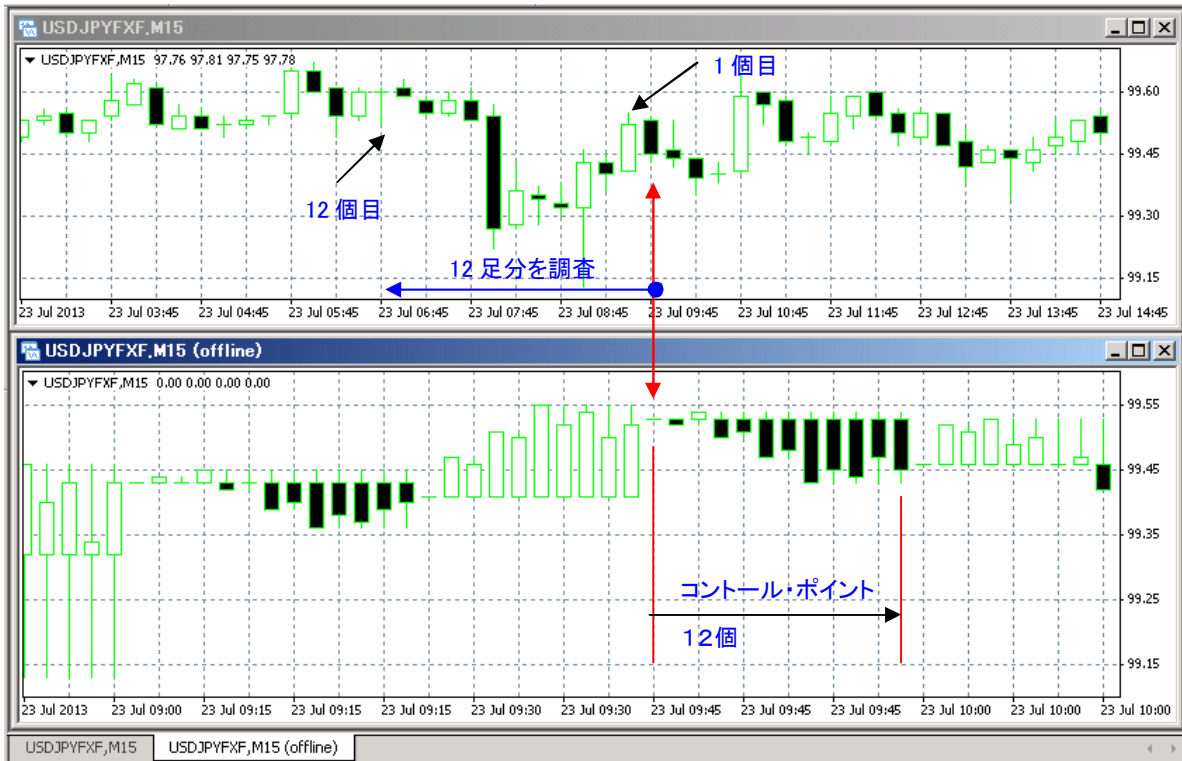
- 例；M5.hst データの存在しない領域で、M15「コントロール・ポイント」モデルを使い、バックテストした時のイメージ。



(6) フラクタル補間を探したけれど！ 見つからない、そして

ー 1. 測定する範囲 M1 5足.hst と、M1 5コントロール・ポイント.fxt

- ①足 [2013.09.23 09:45] の、直近12個分の足
- ②足 [2013.09.23 09:45] のコントロール・ポイント (12個)



ー 2. ざっくり検証1； 直近12足と、コントロール・ポイントを比較



※全然、似てない！！、「フラクタル補間」はどこにいった？？



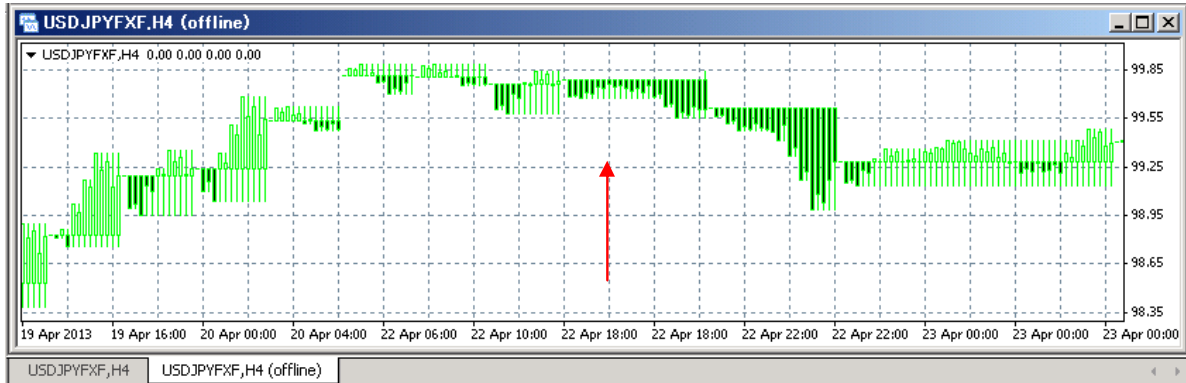
－ 3. チョット、寄り道

データ；

1 H ; 2013.04.22 20:00～

4 H ; 2012.05.10 20:00～

4 Hコントロール・ポイント.fxt ; 1 Hデータの切り替わりポイント (04.22 20:00 近辺)

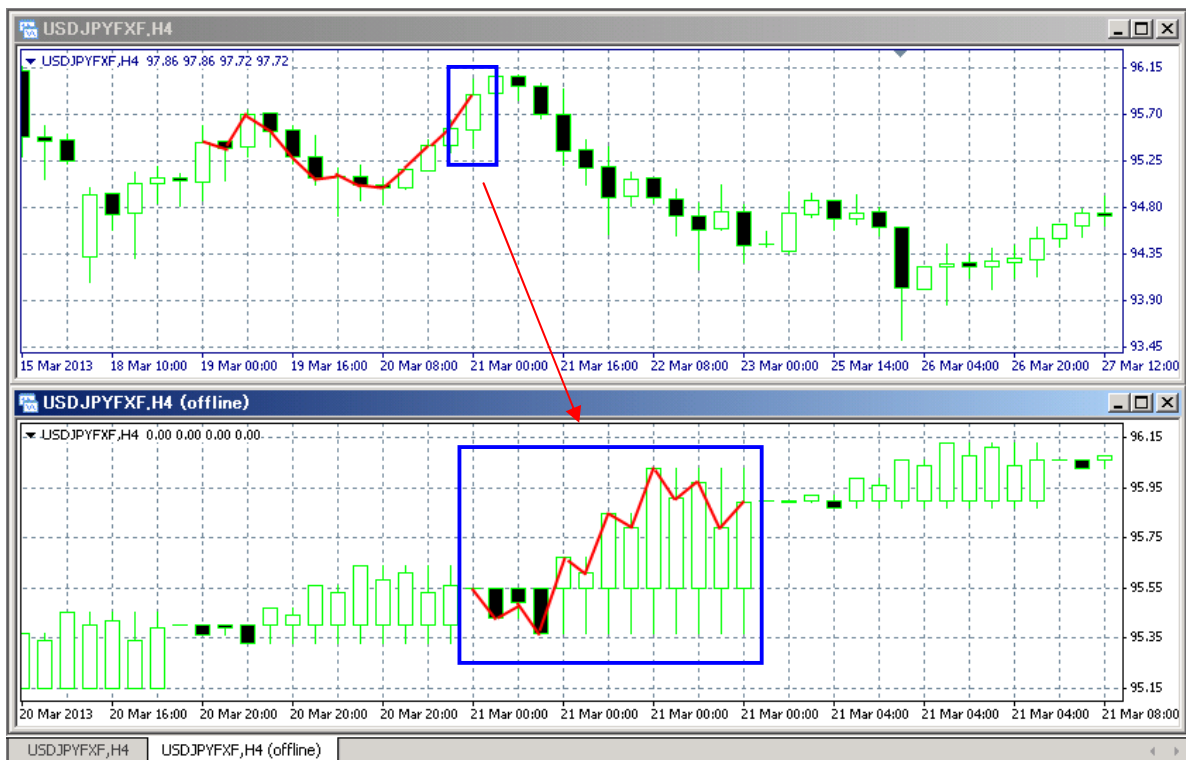


※ 「H1 データの在る」領域を境として、コントロール・ポイント数が、大きく変化している。

－ 4. 本題に戻る； ??何処に、フラクタル補間があるのか、また探す！！

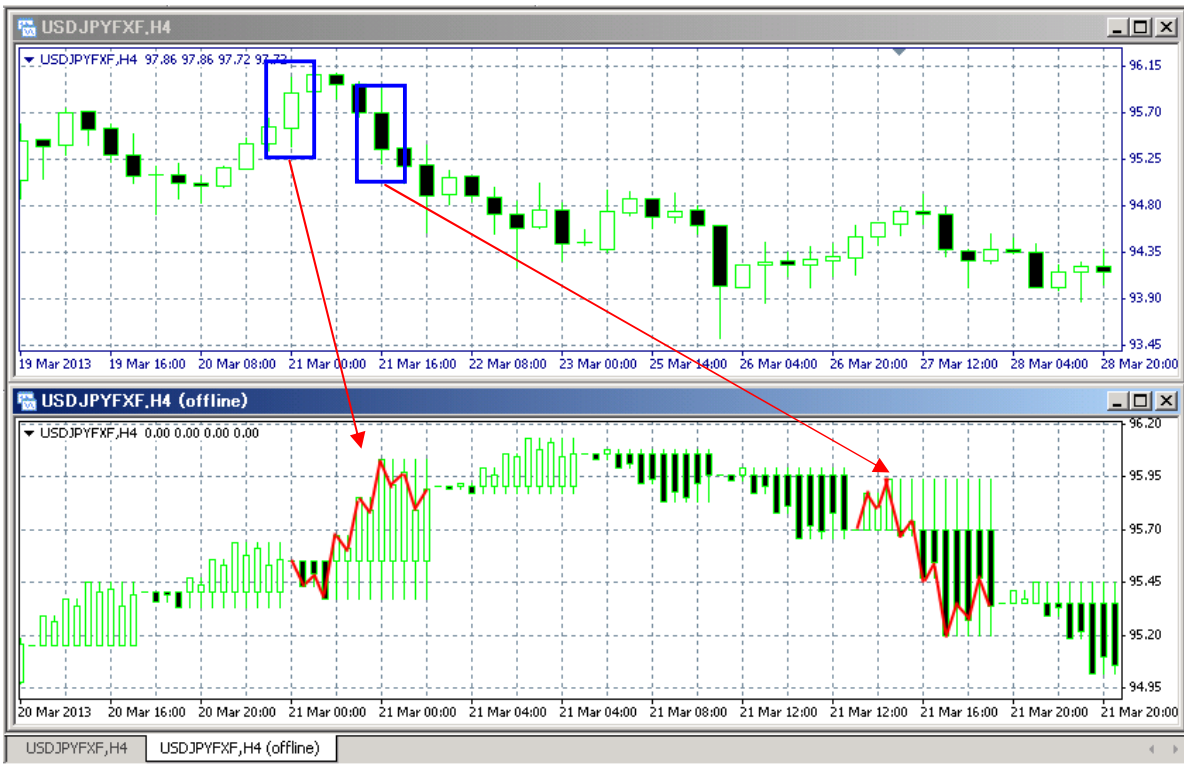
1H データの無い領域；

上から M4 足.hst→M4 コントロール・ポイント.fxt (H1 データの無い領域)

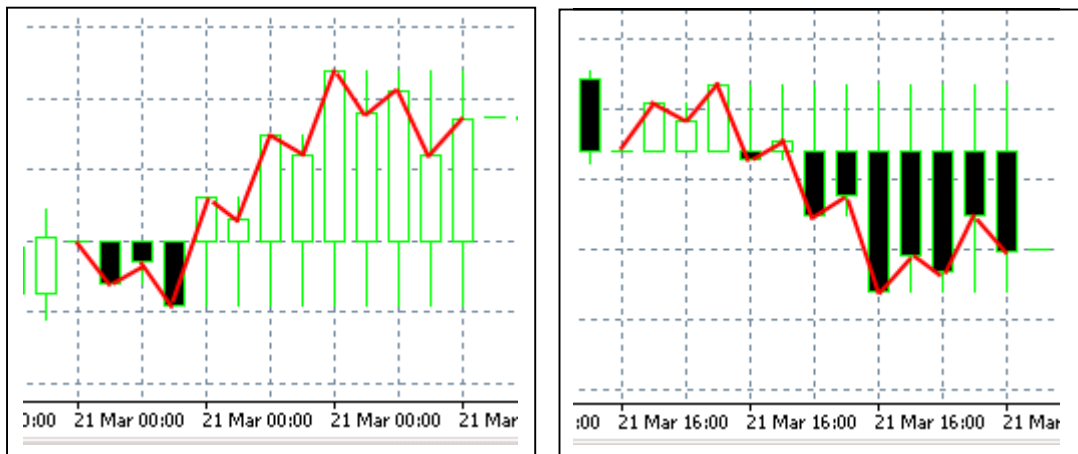


※直近の先行する 1 2 足の Close を結んだ波形と、コントロール・ポイントの波形は、やっぱり、全然似ていない！！

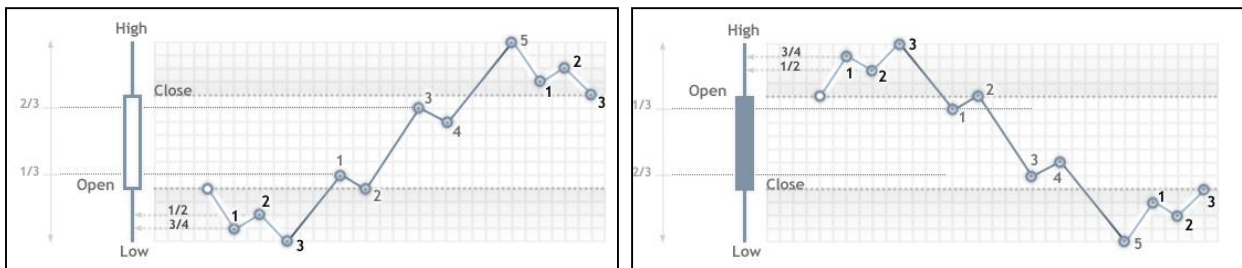
もう一度、波形を見直す；上からH4 足.hst→H4 コントロール・ポイント.fxt (H1 データの無い領域)



拡大して切り出す；



「3-5-3モデル」と比較する； 何と、ポイント数が「1個」違うだけで、あとはソックリ！



－ 5. まとめ； 4 H足の場合で考察

結果一覧；

	補間方法	測定領域		コントロール・ポイント数； 測定値	備考
1H データが 無い領域	「3-5-3 モデル」 と推測中	2013.03.20 00:00 ～2013.03.20 20:00	5 足	13、13、13、13、10	
1H データが 在る領域	(未解析)	2013.04.23 04:00 ～2013.04.24 00:00	5 足	47、48、47、51、49	4×12=48

※測定は「Tick\_Print\_FXT\_01.mq4」による。

A. 補間方法 (モード「コントロール・ポイント」の場合で確認)

※確認できた補間方法は「3-5-3 モデル」。

「フラクタル補間」されている例は、結局「見つからず」。

もしかしたら、上記の「(未解析)」部分にあるのだろうか??

B. 補間数

※「Open=High=Low=Close,Volume=1」を含めて「1 2個」が基本。(「3-5-3 モデル」の場合)

より短期のタイムフレームのデータが利用出来る場合；

例えば4 Hの場合だと、1 Hのデータが4個つかえるので、  
補間数は「1 2個の4倍」となる。

5. 補足 ; 測定結果の追加

(1) M1、M5データの無い領域で、M15のコントロール・ポイントと、全ティックを比較する

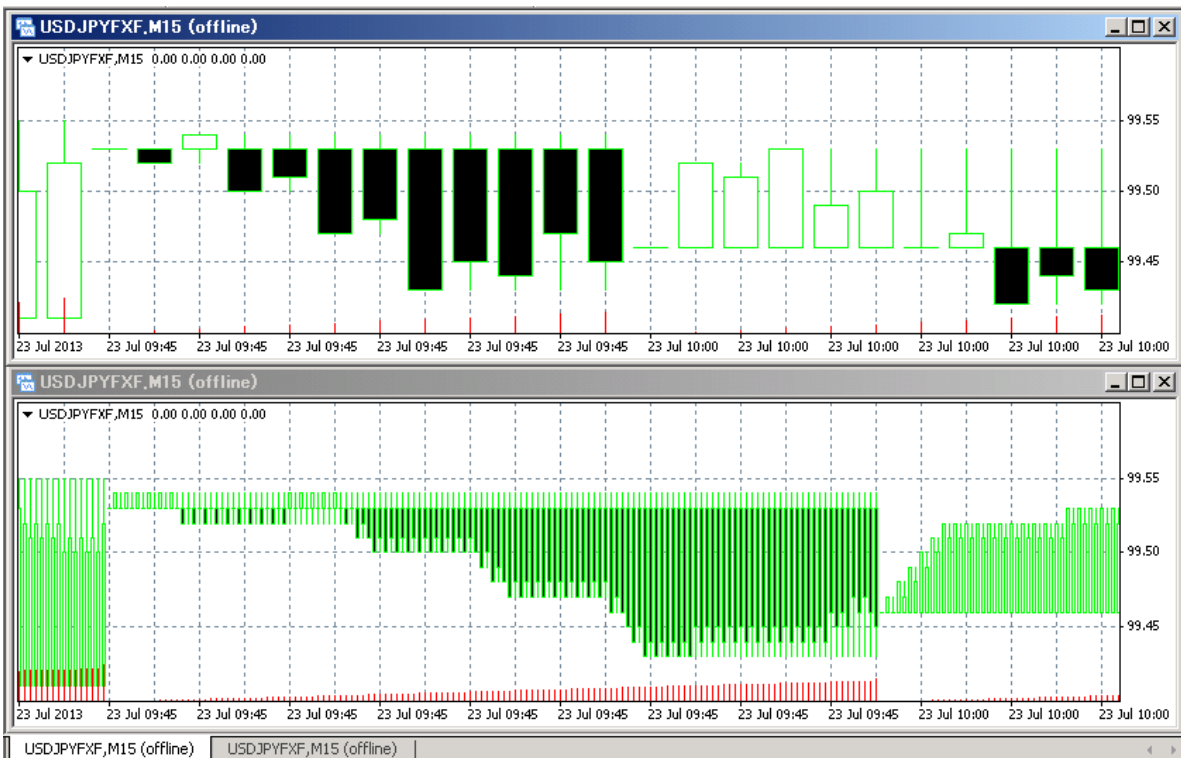
○アメンボが推測する「生成規則」を述べると、  
 短期時間枠 (time frame) 足データ (M1、M5) が存在しない領域では、  
 ①モード [コントロール・ポイント] でバックテストすると、  
 ⇒ M15の1つの足内に、平均「12個」のTickが生成される  
 ②モード [全ティック] でバックテストすると、  
 ⇒ M15の1つの足内に、平均「 $12 \times 12 = 144$ 」個のTickが生成されるはず

○実際の観測結果 ; 「2013.07.23 09:45」のM15足をバックテストした  
 モード [コントロール・ポイント] ⇒ 12個のTick生成  
 モード [全ティック] ⇒ 137個のTick生成 ・ ・ 144個には7個不足  
 (詳細は、下図と、カウント用MQLコードを走らせた結果を参照)

※M1.hstデータは「2013.10.17 09:53~」、M5.hstデータは「2013.10.09 19:10~」であり、  
 「2013.07.23 09:45」の時点にはデータ存在しない状態で、  
 M15の1足「2013.07.23 09:45」のFXTデータを観察した。

M15「2013.07.23 09:45」FXTデータ ;

上から、[M15.fxt コントロールポイント] → [M15.fxt 全ティック]



ティックをカウントする；

<MQL4コード；カウントに利用したコード>

- ・以下の調査は「Tick\_Print\_FXT\_01.mq4」（P15）を使用（バックテスト）して実施した。

<M15コントロール・ポイントの場合の出力；¥test¥logs内のログ>

```

23:10:22 Tick_Print_FXT_01 USDJPYFXF,M15: loaded successfully
23:10:22 TestGenerator: spread set to 2
23:10:22 TestGenerator: unmatched data error (high value 97.36000 at 2013.10.09 19:00 is not reached from the least timeframe,
high price 97.33000 mismatches)
23:10:22 Tick_Print_FXT_01 test started
23:10:22 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Number of ticks in minute = 0
23:10:22 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: ===== START =====
23:10:22 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 45分 : 1回
23:10:22 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.53:Low=99.53:Close=99.53:Volume=1
23:10:22 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.53:Ask=99.55
23:10:22 2013.07.23 09:47 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 47分 : 2回
23:10:22 2013.07.23 09:47 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.53:Low=99.52:Close=99.52:Volume=27
23:10:22 2013.07.23 09:47 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.52:Ask=99.54
23:10:22 2013.07.23 09:48 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 48分 : 3回
23:10:22 2013.07.23 09:48 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.52:Close=99.54:Volume=40
23:10:22 2013.07.23 09:48 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.54:Ask=99.56
23:10:22 2013.07.23 09:50 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 50分 : 4回
23:10:22 2013.07.23 09:50 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.5:Close=99.5:Volume=53
23:10:22 2013.07.23 09:50 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.5:Ask=99.52
23:10:22 2013.07.23 09:51 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 51分 : 5回
23:10:22 2013.07.23 09:51 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.5:Close=99.51:Volume=66
23:10:22 2013.07.23 09:51 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.51:Ask=99.53
23:10:22 2013.07.23 09:52 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 52分 : 6回
23:10:22 2013.07.23 09:52 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.47:Close=99.47:Volume=79
23:10:22 2013.07.23 09:52 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.47:Ask=99.49
23:10:22 2013.07.23 09:53 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 53分 : 7回
23:10:22 2013.07.23 09:53 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.47:Close=99.48:Volume=92
23:10:22 2013.07.23 09:53 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.48:Ask=99.5
23:10:22 2013.07.23 09:55 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 55分 : 8回
23:10:22 2013.07.23 09:55 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.43:Close=99.43:Volume=105
23:10:22 2013.07.23 09:55 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.43:Ask=99.45
23:10:22 2013.07.23 09:56 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 56分 : 9回
23:10:22 2013.07.23 09:56 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.43:Close=99.45:Volume=118
23:10:22 2013.07.23 09:56 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.45:Ask=99.47
23:10:22 2013.07.23 09:57 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 57分 : 10回
23:10:22 2013.07.23 09:57 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.43:Close=99.44:Volume=131
23:10:22 2013.07.23 09:57 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.44:Ask=99.46
23:10:22 2013.07.23 09:58 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 58分 : 11回
23:10:22 2013.07.23 09:58 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.43:Close=99.47:Volume=144
23:10:22 2013.07.23 09:58 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.47:Ask=99.49
23:10:22 2013.07.23 09:59 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 59分 : 12回
23:10:22 2013.07.23 09:59 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.43:Close=99.45:Volume=156
23:10:22 2013.07.23 09:59 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.45:Ask=99.47
23:10:22 2013.07.23 10:00 Tick_Print_FXT_01 USDJPYFXF,M15: Number of ticks in minute = 12
23:10:22 2013.07.23 10:00 Tick_Print_FXT_01 USDJPYFXF,M15: ===== START =====
23:10:22 2013.07.23 10:00 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 0分 : 1回
23:10:22 2013.07.23 10:00 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.46:High=99.46:Low=99.46:Close=99.46:Volume=1
23:10:22 2013.07.23 10:00 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.46:Ask=99.48

```

<M15膳ティックの場合の出力；¥test¥logs内のログ>

```

23:10:42 TestGenerator: spread set to 2
23:10:46 TestGenerator: unmatched data error (high value 97.36000 at 2013.10.09 19:00 is not reached from the least timeframe,
high price 97.33000 mismatches)
23:10:46 Tick_Print_FXT_01 test started
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Number of ticks in minute = 0
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: ===== START =====
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 45分 : 1回
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.53:Low=99.53:Close=99.53:Volume=1
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Bid=99.53:Ask=99.55
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: TimeMinute(***) : tick --> 45分 : 2回
23:10:46 2013.07.23 09:45 Tick_Print_FXT_01 USDJPYFXF,M15: Open=99.53:High=99.54:Low=99.53:Close=99.54:Volume=2

```

