

MetaTrader5 と Python による自動売買

富山県立大学 情報基盤工学講座
1515010 大谷和樹

指導教員：奥原浩之

1 はじめに

これまではBlockly によるビジュアルプログラミングを模索していた。外部のファイルからデータを読み出して入力として利用しようとしていたが、Blockly の外部からデータを受け取る方法が探した限りでは見つからなかった。他の方法として、JavaScript を使って1から全てのシステムを作る方法もあったが、後半年と少しで完成させることは難しいと判断し、テーマの変更をすることにした。変更後のテーマは今まで専門ゼミ等で引き継がれてきたMetaTrader5 (MT5) と Python を利用して自動売買を行うシステムの改良となった。

2 今までの引き継ぎ内容

専門ゼミで、平松君がこのテーマについての引き継ぎを行って Wiki のまとめておいてくれたので、それを参考にして実際に平松君が作成したプログラムが動くのか確認した。必要な Python のライブラリと MT5 のアプリをダウンロードし、Wiki に挙げられていたプログラムを実際に走らせてみた。ここで、大まかな売買の流れを説明すると、MT5 からデータをもってきて、program.py の中にある実際に売買を行う際に使うルールと照らし合わせ、売買の基準を満たしたら Trade.mq5 で実際に MT5 内での売買を行っている。今回のプログラムでは単純移動平均を使用したゴールデンクロスで売買を行っている。実際にこのプログラムを1時間動かした時の損益の結果が以下の図1である。

時間	銘柄	チャート	タイプ	数量	価格	決済逆指値(円)	決済損益(円)	時間	価格	損益	変更
2020.06.17 07:08...	usdjpy	630481671	buy	0.1	107.279			2020.06.17 07:10:41	107.283	40	0.00%
2020.06.17 07:11...	usdjpy	630483323	buy	0.1	107.282			2020.06.17 07:12:00	107.217	-50	-0.00%
2020.06.17 07:14...	usdjpy	630484602	buy	0.1	107.273			2020.06.17 07:14:54	107.260	-50	-0.00%
2020.06.17 07:16...	usdjpy	630485411	sell	0.1	107.260			2020.06.17 07:17:12	107.260	0	0.00%
2020.06.17 07:17...	usdjpy	630486256	sell	0.1	107.252			2020.06.17 07:19:12	107.246	-60	-0.01%
2020.06.17 07:19...	usdjpy	630487645	sell	0.1	107.238			2020.06.17 07:20:29	107.245	-70	-0.01%
2020.06.17 07:22...	usdjpy	630489130	buy	0.1	107.244			2020.06.17 07:23:13	107.240	-40	-0.00%
2020.06.17 07:23...	usdjpy	630489787	buy	0.1	107.241			2020.06.17 07:25:17	107.251	-100	-0.01%
2020.06.17 07:26...	usdjpy	630491616	buy	0.1	107.247			2020.06.17 07:28:43	107.244	-30	-0.00%
2020.06.17 07:30...	usdjpy	630494464	buy	0.1	107.248			2020.06.17 07:31:31	107.247	-100	-0.00%
2020.06.17 07:32...	usdjpy	630495412	buy	0.1	107.249			2020.06.17 07:33:17	107.238	-110	-0.01%
2020.06.17 07:34...	usdjpy	630496464	buy	0.1	107.243			2020.06.17 07:34:21	107.233	-100	-0.01%
2020.06.17 07:36...	usdjpy	630496866	buy	0.1	107.239			2020.06.17 07:35:29	107.238	-110	-0.01%
2020.06.17 07:36...	usdjpy	630497656	buy	0.1	107.227			2020.06.17 07:36:21	107.223	-40	-0.00%
2020.06.17 07:36...	usdjpy	630497971	buy	0.1	107.228			2020.06.17 07:37:41	107.226	-20	-0.00%
2020.06.17 07:38...	usdjpy	630498837	buy	0.1	107.222			2020.06.17 07:39:05	107.217	-50	-0.00%
2020.06.17 07:40...	usdjpy	630500195	buy	0.1	107.224			2020.06.17 07:41:11	107.223	-10	-0.00%
2020.06.17 07:42...	usdjpy	630501150	buy	0.1	107.227			2020.06.17 07:43:23	107.217	-100	-0.01%
2020.06.17 07:44...	usdjpy	630502289	buy	0.1	107.233			2020.06.17 07:46:19	107.227	-60	-0.01%
2020.06.17 07:45...	usdjpy	630503497	buy	0.1	107.233			2020.06.17 07:47:09	107.228	-50	-0.00%

図 1: 引き継いだプログラムの実行結果

3 進捗状況

今回の研究では、より良い指標の組み合わせ等を見つけるために複数の指標で売買を行えるプログラムを作成する必要がある。そのため Python で簡単にテクニカル指標が計算できる「pyti」というライブラリを使用することにした。pyti には 45 の指標があり今回はその中でまず MACD という指標についてのルールを作成することにした。指標の計算にはしばしば start, high, low, close, の情報が必要になるので、過去のこれらのデータを MT5 からもらってくるプログラムをサイトを参考にして作成した。プログラムの実行結果が次の図2である。

このプログラムでは1日ごとのそれぞれのデータを表示している。今回の MACD では close の値を使うので、この値を使って計算を行おうとしたが、参考にしていたサイトの pyti のプログラムではうまく動かなかった。

time	open	high	low	close	tick_volume	spread	real_volume
0 2020-04-07	1.07520	1.09259	1.07834	1.08906	89661	0	0
1 2020-04-08	1.08983	1.09026	1.08206	1.08552	78654	0	0
2 2020-04-09	1.08559	1.09515	1.08409	1.09285	62410	0	0
3 2020-04-10	1.09286	1.09522	1.09185	1.09347	49605	0	0
4 2020-04-13	1.09428	1.09676	1.08925	1.09120	89885	0	0
5 2020-04-14	1.09120	1.09870	1.08987	1.09798	113863	0	0
6 2020-04-15	1.09794	1.09997	1.08562	1.09066	100096	0	0
7 2020-04-16	1.09076	1.09111	1.08166	1.08386	123356	0	0
8 2020-04-17	1.08386	1.08925	1.08119	1.08739	126445	0	0
9 2020-04-20	1.08825	1.08965	1.08411	1.08607	92579	0	0
10 2020-04-21	1.08611	1.08801	1.08164	1.08564	187627	0	0
11 2020-04-22	1.08563	1.08851	1.08032	1.08329	82377	0	0
12 2020-04-23	1.08226	1.08466	1.07560	1.07768	123860	0	0
13 2020-04-24	1.07768	1.08257	1.07269	1.08244	108051	0	0
14 2020-04-27	1.08208	1.08602	1.08117	1.08308	95439	0	0
15 2020-04-28	1.08275	1.08881	1.08096	1.08182	77849	0	0
16 2020-04-29	1.08180	1.08595	1.08179	1.08728	74700	0	0
17 2020-04-30	1.08709	1.09724	1.08328	1.09507	84471	0	0
18 2020-05-01	1.09480	1.10108	1.09343	1.09802	93522	0	0
19 2020-05-04	1.09864	1.09743	1.08956	1.09002	54761	0	0
20 2020-05-05	1.09054	1.09257	1.08257	1.08398	58158	0	0
21 2020-05-06	1.08397	1.08564	1.07818	1.07920	54633	0	0
22 2020-05-07	1.07925	1.08339	1.07664	1.08319	60714	0	0
23 2020-05-08	1.08332	1.08795	1.08156	1.08388	48524	0	0
24 2020-05-11	1.08249	1.08903	1.08005	1.08009	49534	0	0
25 2020-05-12	1.08038	1.08852	1.07843	1.08469	54386	0	0
26 2020-05-13	1.08464	1.08963	1.08115	1.08172	58328	0	0
27 2020-05-14	1.08167	1.08238	1.07748	1.08042	53397	0	0
28 2020-05-15	1.08051	1.08095	1.07888	1.08187	90568	0	0
29 2020-05-18	1.08644	1.09269	1.07908	1.09130	48887	0	0
30 2020-05-19	1.09145	1.09759	1.09021	1.09217	51891	0	0

図 2: 各データの取得結果

4 おわりに

研究テーマを新しく設定し、完成に向かって動き出した。今回は今まで引き継がれていたものの動作確認とテクニカル指標の1つである MACD の実装を目標に行ったが、MACD に関しては実際に使用できるところまで至らなかった。今後はとりあえず MACD のプログラムを完成させ実際に MT5 で売買を行ってみることに、その他の指標についてのプログラムの作成を進めていく。

参考文献

- [1] <https://blog.novonovo.jp/python/pyti/> のインジケータを使ってチャートを分析する/, Accessed: June 17, 2020.
- [2] <https://www.mql5.com/ja/docs/integration/python-metatrader5/mt5copyratesrange-py>, Accessed: June 17, 2020.