SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Faculty of Civil Engineering

Reg. No.: SvF-104289-7478

Methods of algorithmic trading

Master thesis

Study programme: matematické a počítačové modelovanie
Study field: 9.1.9. Applied Mathematics
Training workplace: Department of Mathematics and Constructive Geometry
Thesis supervisor: prof. RNDr. Karol Mikula, DrSc.

Bratislava 2016 **Bc. Michal Krakovský**

Declaration of honour

I hereby declare that I personally prepared the present academic work and carried out myself the activities directly involved with it, with technical assistance of my supervisor and using the referenced sources.

Bratislava, June 2016 ....

Signature

Acknowledgment

I would like to express my thanks of gratitude to my supervisor prof. RNDr. Karol Mikula for his support, patience, and motivation. His guidance helped me in the time of research and writing of this thesis. Furthermore I would also like to thank my parents and friends for their endless love and support.

Abstract

Automated trading is a fast growing industry taking over financial markets all around the world. In this work we studied and tested a standard automated system and its components. Systems which use 2 moving averages or sum of intrinsic mode functions as signal generators were implemented in Matlab. This implementations were used to run 4 experiments. In first experiment varying lengths of in-sample and out-sample periods were tested and results were provided. In second experiment we introduced new condition for optimization procedure and tested it on results obtained from first experiment. It was found that this condition improved the trading of the system. Furthermore we tested if parameters found by the optimization procedure are really optimal to use for trading with a result showing that the optimization procedure does not always find the best parameters to trade. In the last part we used empirical mode decomposition to extract intrinsic mode functions from the data and then used sums of those functions as signal generator. The results shows that it might be superior signal generator than the one composed of two moving averages.

Abstrakt

Automatické obchodovanie sa v poslednej dobe stalo veľmi populárne po celom svete. V tejto práci sme študovali štandardný automatický obchodovací systém a jeho komponenty. Implementovali sme systémy, ktoré využívajú kombináciu dvoch pohyblivých priemerov, alebo sumu intrinsic mode funkcií ako generátor signálu. Tieto implementácie nám poslúžili na vybraté experimenty. V prvom sme hľadali vhodné dĺžky in-sample periódy a out-sample periódy. Potom sme do systému pridali novú podmienku na optimalizované parametre, vďaka ktorej systém dosiahol lepšie výsledky. Ďalej sme testovali či parametre nájdené v optimalizácii sú tie najvhodnejšie na obchodovanie. Pričom sme prišli na to, že optimalizácia ktorú používame nepostačuje na nájdenie najlepších parametrov. V poslednej časti sme použili empirical mode dekompozíciu na

získanie intrinsic mode funkcií z dát a následne sme ich sumy používali ako generátor signálu, ktorý dosiahol lepšie výsledky ako ten vytvorený pomocou dvoch pohyblivých priemerov.

Contents

1 Introduction

The Automated Trading system is a computer program that creates and submits orders to a market center or exchange. First trading system tracks back to 1949 when Richard Donchian created Futires, Inc. which was commodity fund using set of rules to create buy and sell signal [1]. It was very different from trading systems of present since in that times there was no Internet and computers were using ticker tapes. Despite that the idea of such a system was born. Although its popularity increased during the 70s and 80s, only in the late 90s it started to attract more investors, in part due to the wider availability of Internet. Since then, automated trading accounts roughly for 75% of all trades done on financial markets. Therefore, it is quite an interesting topic nowadays, not yet widely covered by scientific research.

1.1 Markets and Data

The financial market is a term which describes any market place where people can trade commodities, securities and their derivatives at low transaction costs. Furthermore, they are defined by basic regulations on trading and transparent pricing, which reflects supply and demand. There are numerous markets around the world such as New York Stock Exchange, NASDAQ, London Stock Exchange group, Japan Exchange Group Tokio as well as decentralized markets, namely the foreign exchange market (FOREX) for trading currencies. In the process of building an automated trading system this plays a vital role because historical data from one of those markets is required to test trading strategies on. This process is called backtesting and provides a way to evaluate if a given strategy can be profitable. The datasets used in this work were provided by the company R7 corp, and consisted of futures of DAX, STOXX50, GBL indexes and FOREX currency pairs. The historical data of those indexes consists of future contracts that have 4 trading seasons denoted by the capital letters H, M, U, Z, which stands for March, June, September and December, respectively. In those months the futures

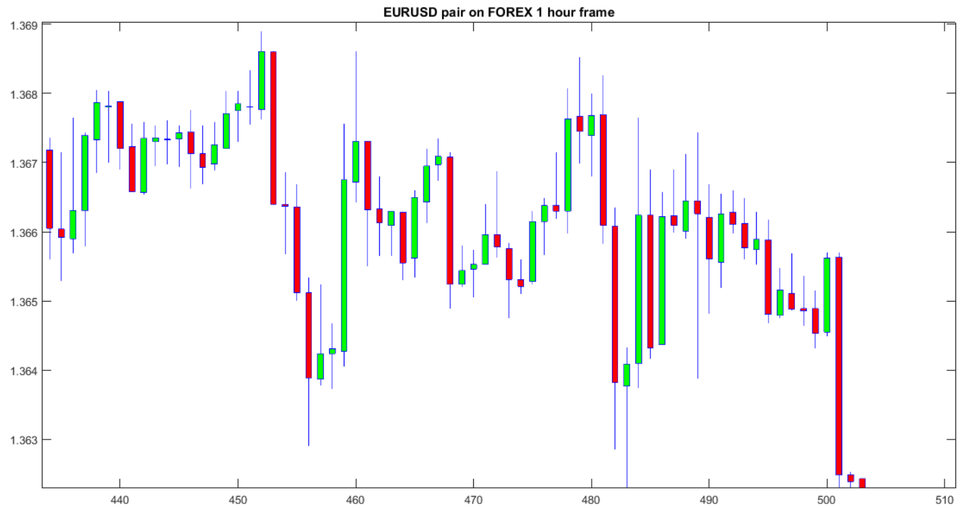


Figure 1: Candle plot of a selection of FDAX prices from 2015

expires and traders then switch to upcoming available in various time frames: 1 min, 5min, 1 hour, 2 hours, 4 hours and daily. The information which the data provides are: opening, closing, lowest and highest prices of the index in a given time frame and as well a volume of trades executed in that frame. Figure 1 shows an example of the so-called candle plot which is a common way of presenting financial data. One bar in the candle plot represents all four data opening, closing, high and low price (Figure 2). The data from FOREX provides the same information on the currency pair in same time frames. The main difference is that FOREX is traded for 24 hours a day except weekends, whereas those indexes are traded within financial markets which operate usually during working hours.

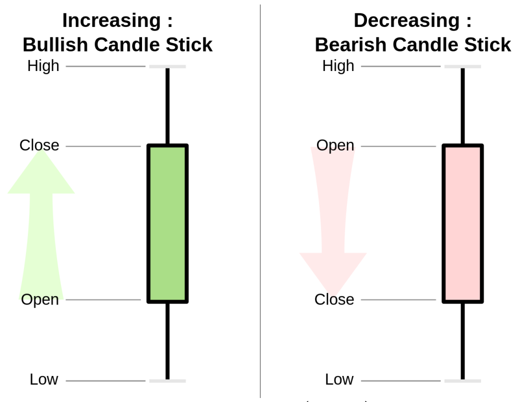


Figure 2: Two types of candles (bars) used in candle plot

2 Example of an automated trading system

In order to achieve the objective of the present work, the company R7 corp. k.s. provided us with an insight into their automated system, which will be described in detail in the present section. This automated trading system is composed of a signal generator, optimization and backtest. The signal generator is a function which only return values $-1, 0, 1$. Upon this values trades are opened and closed. Signal generators usally have some parameter, thus the optimization of those parameters and backtest are important steps and provide a way to set optimal parameters for the signal generator. Backtest is an evaluation of trades done on historical data. Consequently, the data is divided into two parts, namely in-sample and out-sample. On the in-sample part, the optimization is ran and the optimal parameters are found for the signal generator. Then the out-sample part is used to evaluate how would the system trade with those parameters. Afterwards, the profit and loss curve or the so-called equity curve can be drawn as a cumulative sum of the trades over time. In the following example the system will be described and the results will be presented.

2.1 STline

The first component of the system is the signal generator, which in this case is created by using so-called STline. STline is a piecewise constant function with two parameters:

period and factor. Period in this case influences the number of times frames that are taken into account when calculating the average volatility. Whereas, factor dictates how tight is the STline constructed around the data. The STline surrounds the data from up or down (always only from one side). Also, the SLline is based on ATR which denotes Average True Range. It is a measure of volatility introduced by Welles Wilder in [2]. The ATR is calculated from TR (True Range) which can be written as

$$TR(i) = max[max(high(i) - low(i), |high(i) - close(i-1)|), |low(i) - close(i-1)|],$$

where $high(i)$ and $low(i)$ denotes highest and lowest value in given time frame and $close(i-1)$ stands for closing price in previous time frame. ATR is usually computed as Moving Average of TRs or as Smoothed Moving Average as in this case

$$ATR(i) = \frac{ATR(i-1)(period - 1) + TR(i)}{period}.$$

To use this formula initial value of ATR has to be computed as

$$ATR(period + 1) = \sum_{i=1}^{period} \frac{TR(i)}{n}.$$

Once ATR is obtained it is used to create an envelope around the data with a use of the second parameter - factor.

$$\begin{aligned} UpperBound(i) &= \frac{high(i) + low(i)}{2} + factor * ATR(i) \\ LowerBound(i) &= \frac{high(i) + low(i)}{2} - factor * ATR(i) \end{aligned} \tag{1}$$

The trend variable is defined as

$$trend(i) = \begin{cases} 1 & \text{if } close(i) > UpperBound(i-1) \\ -1 & \text{if } close(i) < LowerBound(i-1) \\ trend(i-1) & \text{if otherwise} \end{cases}$$

The trend provides information regarding whether the data crosses the volatility envelope or not. The update of $UpperBound$ and $LowerBound$ as in (1) every time steps takes into account this conditions

$$\begin{gathered} if\ (trend(i) = -1\ \wedge\ trend(i-1) = 1)\ \wedge\ (UpperBound(i)\ >\ UpperBound(i-1)) \\ then \quad UpperBound(i) = UpperBound(i-1), \end{gathered}$$

and correspondingly for the $LowerBound$,

$$\begin{gathered} if\ (trend(i) = 1\ \wedge\ trend(i-1) = -1)\ \wedge\ (LowerBound(i)\ <\ LowerBound(i-1)) \\ then \quad LowerBound(i) = LowerBound(i-1). \end{gathered}$$

Finally STline is constructed as

$$ST(i) = \begin{cases} UpperBound(i) & \text{if } trend(i) = 1 \\ LowerBound(i) & \text{if } trend(i) \neq 1 \end{cases}$$

In Figure 3, the blue line represents the candle plot of the data and green is the STline with a period and factor equal to 10. In the Figure it is as well possible to see the upper envelope approximately until 600th frame and then the data crosses it, meaning that the STline switches to a lower envelope.

2.2 Positioning

In the context of this work, the term "long" is used when purchasing a stock or currency. When a long position is opened it translates into a buying order to the broker, which results in owning that specific asset or currency for the price of that asset or currency at the time of the purchase. This position is opened in a situation when the price is expected to rise in the future thus generating profit when sold. On the other hand, the term "short" is more complex since it has different meanings for stocks and currencies. Regarding the currency pair, it simply means buying the second currency of the pair.

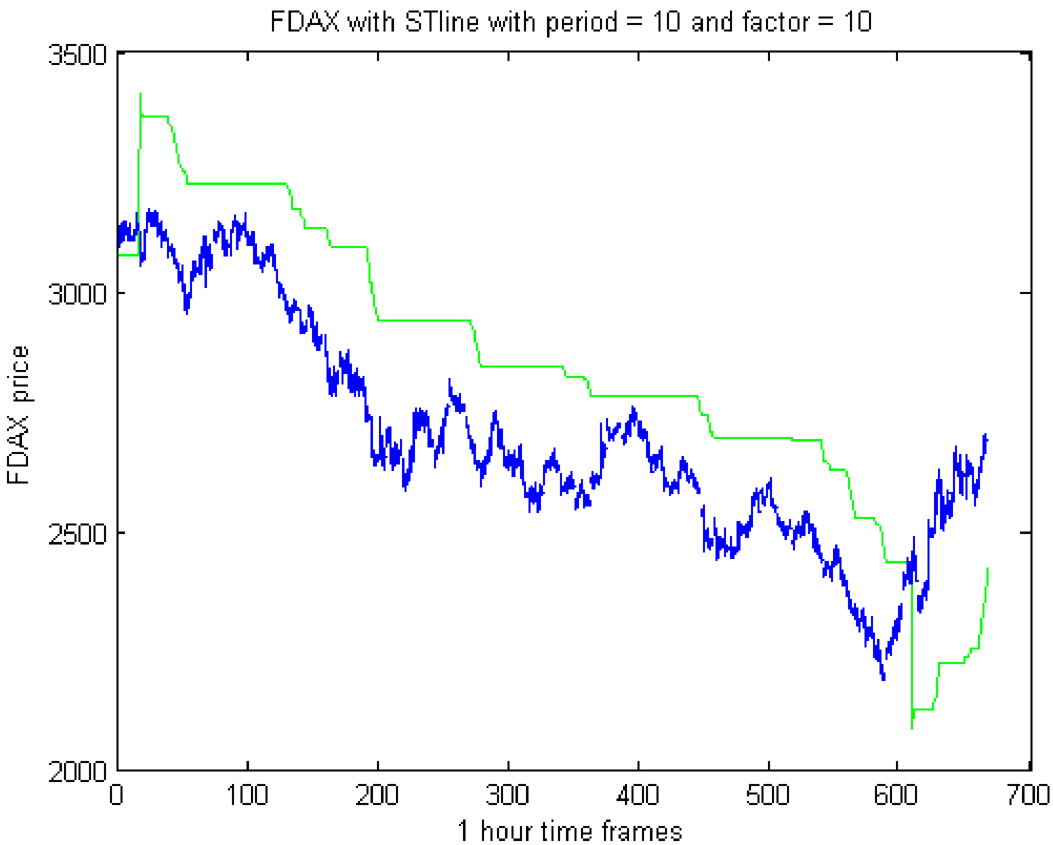


Figure 3: STline with period 10 and 10 and FDAX 1 hour

For example, for the pair EUR-USD when a long position is opened, the euro currency is purchased. Contrarily, when a short position is opened, the USD currency is purchased thus meaning that it is believed that the EUR currency will decrease and therefore short trade will be profitable. The same logic will apply for the term short in stocks. Short position is opened when it is expected that the price of the stock will decrease. It means that when it is opened, the trader borrows a stock from broker and sells it to another trader. Nevertheless, to borrow a stock from a broker, the trader has to have enough capital to guarantee the possibility of buying that stock and return it to the broker at any time. Once the trader decides to close this position he has to buy that stock back (possibly for a lower price which would generate him a profit) and return it to the broker.

2.3 Trading

As mentioned before STLine is used in this system as signal generator. The system opens long position when data crosses the STLine from underneath thus when STLine starts to surround it from bottom. Consequently when the data crosses the STline from above short position is opened. In both cases, if there is still an opened position from a previous trading, it is closed before the new position is opened. This generates profit and loss according to the prices of the asset at the time of the opening and closing of the trade. Furthermore, two additional constant parameters are used while trading. First is called take profit. If the profit of the currently opened trade exceeds this constant, the trade is automatically closed and the profit equal to this constant is earned. As the name of the second parameter, stop loss, suggests, it is used to check if the currently opened trade is not already generating loss higher than this parameter. If so, the trade is closed and the loss equal to this constant is taken.

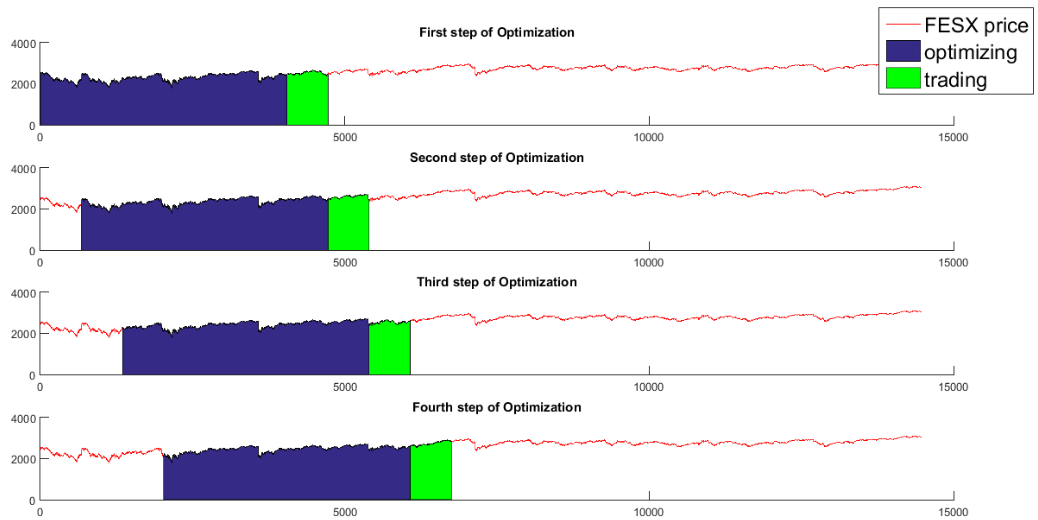


Figure 4: Optimization technique and backtesting

2.4 Optimization and Backtest

The data which was used to backtest is historical data of futures of aforementioned indexes and FOREX pair currencies. These futures have a symbol different from the ones of index FDAX for DAX, FESX for STOXX50 and FGBL for GBL. Historical data from 2003 until the end of 2015 was used. The optimization is done in order to find the optimal parameters for the STLine. Ultimately using the STLine with the optimal parameters should create profit which is the goal of such system. The profit or loss created is checked with backtest, which acts on historical data as if it was new data coming to the system and evaluates how the system would react and perform.

The optimization procedure used can be described in follwing steps:

1. Starting from the oldest date, take 6 seasons of futures data (in case of FOREX it is 1.5 year). This part of data will be called in-sample.

2. Find the best parameters of STLine according to objective function (see next section) in that period.

3. Use the STline with those parameters found to trade in the next season (3 months in case of FOREX). The data used for this step is called out-sample.

4. Add the data from out-sample to in-sample and put the oldest season from in-sample away so number of 6 seasons in in-sample is preserved.

5. Repeat the procedure from step 2 until the latest date in the data is reached.

6. Evaluate the total profit and loss. This means checking which trades were done in out-sample periods.

Figure 4 shows 4 steps of optimization and backtest.

2.5 Objective function

The crucial component of optimization is the objective function, since it defines which parameters will be chosen as optimal. The objective function which is maximized with respect to parameters p and f, representing period and factor

$$\begin{aligned} & ProfitAndLoss(p, f) - maxDrawdown(p, f) - k * numberOfTrades(p, f) + \\ & \min_{p_n, f_n \in N} (ProfitAndLoss(p_n, f_n) - maxDrawdown(p_n, f_n) - k * numberOfTrades(p_n, f_n)). \end{aligned}$$

$ProfitAndLoss$ stands for the sum of the results of trades which were performed, $maxDrawdown$ is the biggest loss which was caused by consecutive negative trades and k is the constant which gives weight to the $numberOfTrades$. All of those variables are functions of p and f since those parameters of STLine decided how trades were executed. The set N consists of values of p_n and f_n which are from the neighborhood of p and f. Namely it consists of 4 elements because only one value for each parameter from both sides is chosen to represent the neighborhood. Constructing objective function in this manner puts emphasis not only on the highest profit, but it also tries to decrease the biggest drawdown and number of trades. Moreover, it is stabilized with the min term, which assures that a small variation of parameters would not radically change the output.

2.6 Genetic Algorithm

In this section genetic algorithm as used in the described system will be presented. The objective function as defined earlier is used in the genetic algorithm where it is usually called fitness function. Optimal values of period and factor are searched within bounds $period \in [10,\ 100]$, $factor \in [1,\ 10]$ and $period$ is always incremented by 5 and factor by 0.1 inside the bounds. The algorithm can be written in steps:

1. Generate initial population.
2. Fitness function is evaluated upon the population and the best pairs the so called elite kids are selected for new generation.
3. Crossover kids (pairs) are generated and added to the new generation.
4. Mutation kids (pairs) are generated and added to the new generation.
5. The process is repeated from step 2 until stopping criterion is reached.

Population in this case consists of period and factor pairs which are generated randomly within the bounds. In this case, it contains 25 pairs. Only one elite kid is selected for a new generation in this system. Crossover kids are constructed from the best couples from previous iteration. They are constructed as mean value of addition of two parents with $rand(0,1)$ and $(1 - rand(0,1))$ weights. Parents are selected based on their fitness function result from the previous generation. In this case, the first crossover kid (pair) is a combination of the first and second best pairs. The second is a combination of third and fourth and so on. 12 crossover kids are added to population in the described system. And the remaining 12 places in new population are filled with so called mutation kids which are created from a pair (parent) from previous population. The parameter value of this parent are slightly randomly altered. Those parents are selected randomly, however parents which present a better fitness function have higher probability of being selected. The process of creation of a new generation stops when

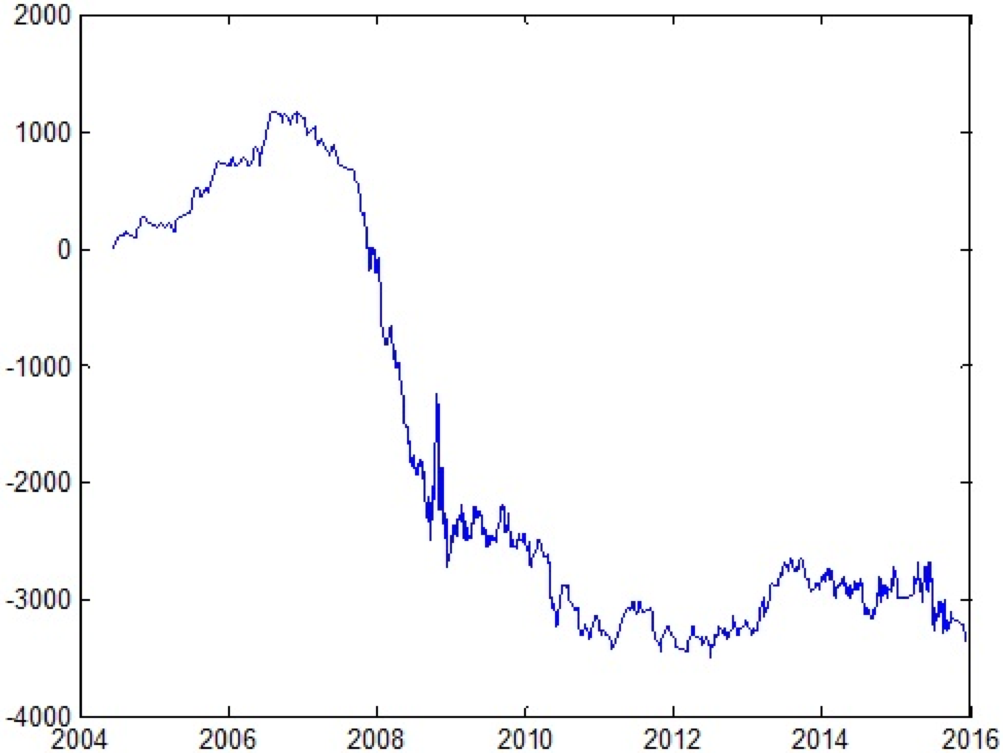


Figure 5: Performance of trading system developed by R7 (Equity curve)

the prescribed number iterations is reached. The elite kid pair of the last generation is selected as optimal.

2.7 Results

Unfortunately, the aforementioned system does not yield satisfactory and robust results thus it cannot be applied on actual markets. Figure 5 shows how this system would perform if it would start trading from the 2004U contract (third contract of the year) until 2015Z (last contract in 2015). This is one of the many unsatisfactory results which this system is prone to.

In Figure 5 on the y axis we can see profit in the point of FESX index (each point has value of 10€). It can be seen that the initial progress of the system looks promising until one part where many negative trades have been performed.

Season	Period	Factor	SL	TP	Profit & Loss	# trades
2004U	35	7,2	2	8	52,33025759	5
2004Z	60	6,6	2	8	124,7026654	6
2005H	60	6,6	2	8	-53,40201513	8
2005M	60	6,6	2	8	129,3039133	4
2005U	45	6,8	2	8	211,984724	8
2005Z	45	6,8	2	8	208,5296767	6
2006H	30	6,6	2	8	66,751598	8
2006M	95	6,3	2	8	156,7773623	10
2006U	50	6,9	2	8	166,7151907	5
2006Z	75	6,3	2	8	45,52970762	10
2007H	100	6,5	2	8	-290,8708216	11
2007M	15	9,5	2	8	-38,00269746	6
2007U	20	9,7	2	8	-251,4964394	5
2007Z	15	2,5	2	8	-730,7036784	46
2008H	85	5,9	2	8	-795,6214439	12
2008M	15	1,7	2	8	-827,379848	73
2008U	95	1,4	2	8	-490,8945821	86
2008Z	45	1	2	8	-329,4647005	114

Table 1: Detailed results obtained from FESX

Table 1 presents detailed figures from the beginning of the trading until the end of the last contract in 2008. The columns present optimal values of the parameters for a given season, stop loss and take profit (in this case constant), profit generated and number of trades in the season. It can be seen that from season 2007H the system started to have substantial losses, which overall affected this system to be unprofitable after a short time.

3 Hilbert-Huang Transform

One part of this work is dedicated to testing possible usage of Hilbert-Huang transform in automated trading system. It is a method for analysing nonlinear and non-stationary data developed by Huang at al. [4]. The method consists of two parts. First is so called Empirical Mode Decomposition and the other Hilbert Transform.

3.1 Empirical Mode Decomposition

The Empirical Mode Decomposition will be ouf our primary interest since it provides a way to decompose data. The idea behind EMD comes from an assumption that any data consists of different intrinsic mode of oscillations. Intrinsic mode function (IMF) is simply defined as a function that satisfy two following conditions,

1. Within the whole domain the number of extrema and the number of zero crossing of the function must be either equal or differ at most by 1

2. At any point of a domain the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is 0.

Example of such a function can be seen in Figure 6.

As the name indicates this method is empirical and until now justified only by examples.The decomposition to intrinsic mode functions is based on so called sifting process where individual mode functions are extracted from the data one by one. Let $X(t)$ denote a data in time t. Then first step is to construct maxima and minima envelopes around the data and compute their mean value which will be denoted as $m_1(t)$. The envelopes are created by connecting local extrema by splines. Then $m_1(t)$ is subtracted from the data,

$$X(t) - m_1(t) = h_1(t).$$

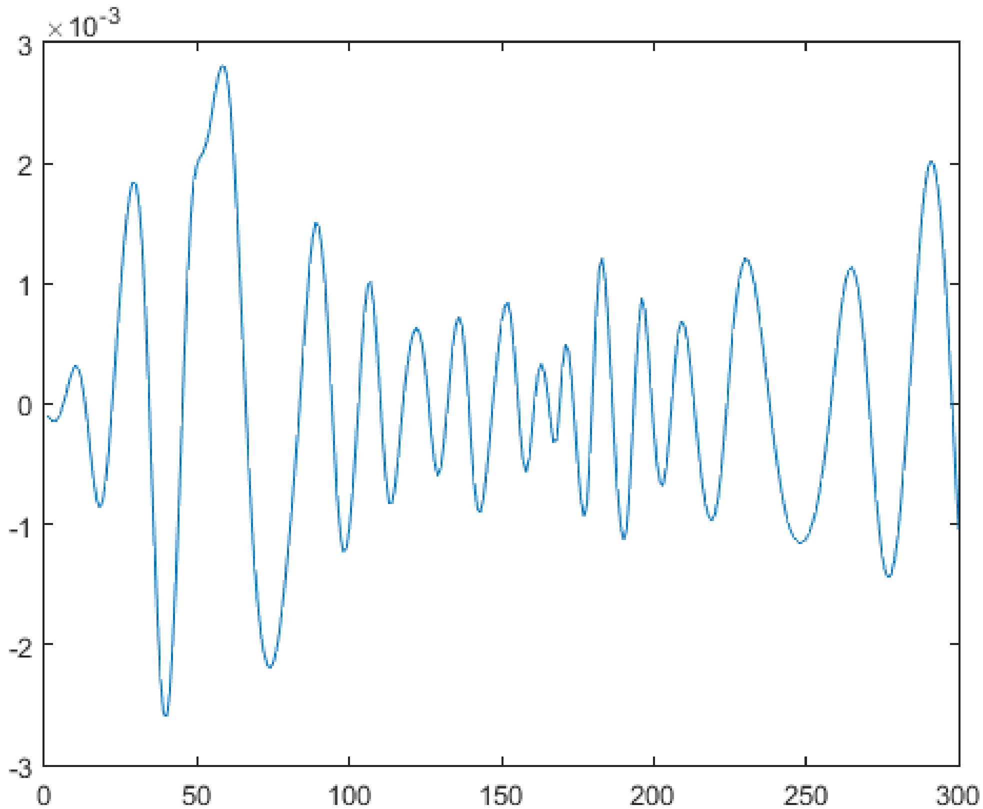


Figure 6: Intrinsic mode function extracted from FOREX data.

Then the envelopes of $h_1(t)$ are created and this simple process is repeated,

$$h_1(t) - m_{11}(t) = h_{11}(t).$$

The process is repeated until what remains satisfies conditions of intrinsic mode function. It will be denoted as $c_1(t)$. Then $c_1(t)$ is separated from the data and residuum $r_1(t)$ is obtained as

$$X(t) - c_1(t) = r_1(t).$$

And again the process continues with $r_1(t)$ until another IMF $c_2(t)$ is found. If the remaining data $r_2(t)$

$$r_2(t) = r_1(t) - c_2(t)$$

is IMF the process is ended. If not the process again continues with $r_2(t)$. This process will ultimately find $c_k(t)$ which when subtracted from $r_{k-1}(t)$ will give new residuum which will either be equal to zero or bare trend. Then the process is ended and all intrinsic mode function are found.

In Figure 7 decomposed data is shown together with its intrinsic mode functions. As designed empirical mode decomposition first extracts higher detail of data (second-forth subplot in Fig. 7) and later the IMFs get coarser (last 3 in Fig. 7). The usage of this method as signal generator and its results will be described in section 4.2.

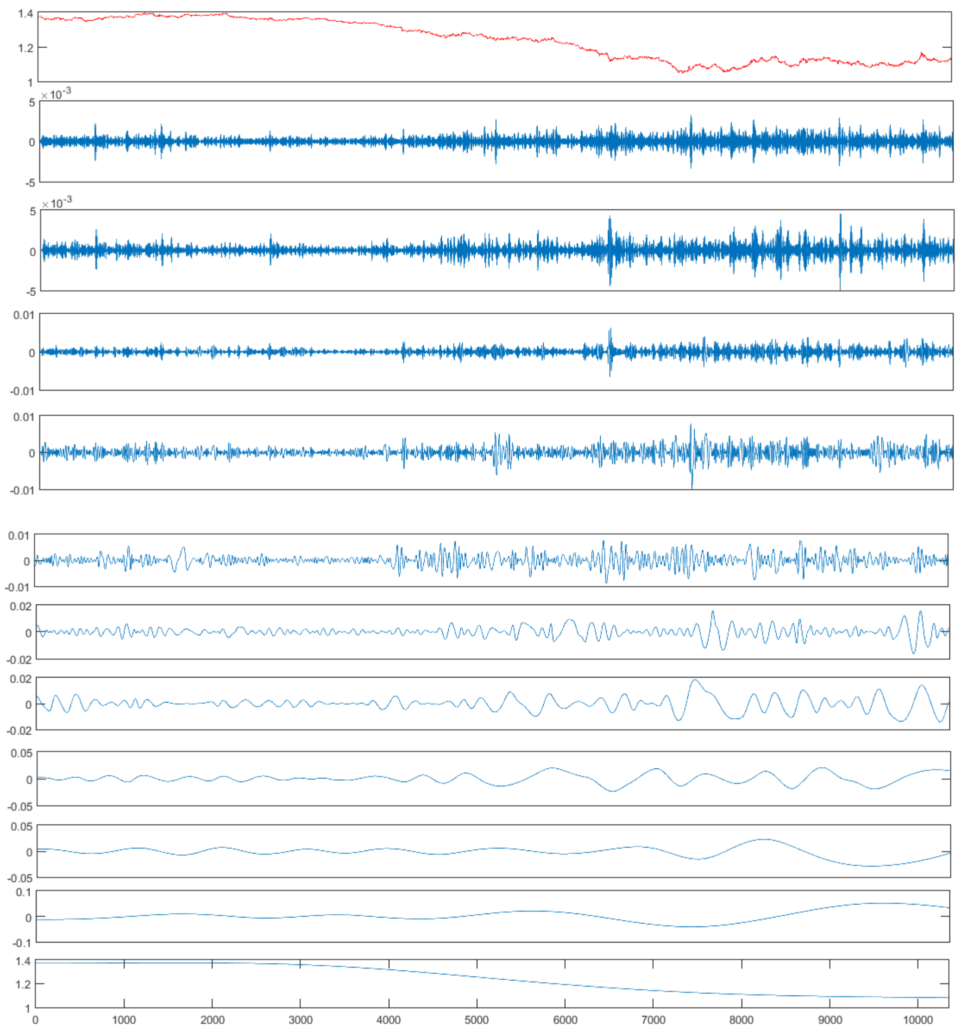


Figure 7: In top of the figure we can see data (red) which was decomposed into 11 IMFS which can be seen under it, starting with first IMF until the last, which represents the trend.

4 Trading system analysis

As shown in Figure 5 the system described in section 2.7 is mostly unprofitable. However it can be seen that for some seasons it was profitable. Therefore, our aim is to analyze properties of trading system similar to the one described described in 2.7. This analysis will be divided into 2 parts:

1. Analysis of the optimization procedure

2. Intrinsic mode functions as signal generator

4.1 Analysis of the optimization procedure

The optimization procedure has many components such as length of in-sample period and length of out-sample period or objective function. All of them influence the system in some way. In this section the influence of lengths of in-sample and out-sample periods will be analysed. The logic behind the optimization as it was described in section 2.4 comes from idea that, if optimal parameters (which would generate profit) are found in some period (in-sample) of historical data, those parameters can be applied to the future (out-sample) and generate profit in real time. If this assumption would be true in any circumstances the results of the system would be positive, generating profit. However, they are not. Thus further analysis of the conditions under which such a system can be profitable is required. The questions which are addressed are:

1. How often is it necessary to recalibrate the parameters of the system?

2. How long should be the period on which the recalibration is done, i.e., length of in-sample data?

3. Could be there more than one set of optimal parameters?

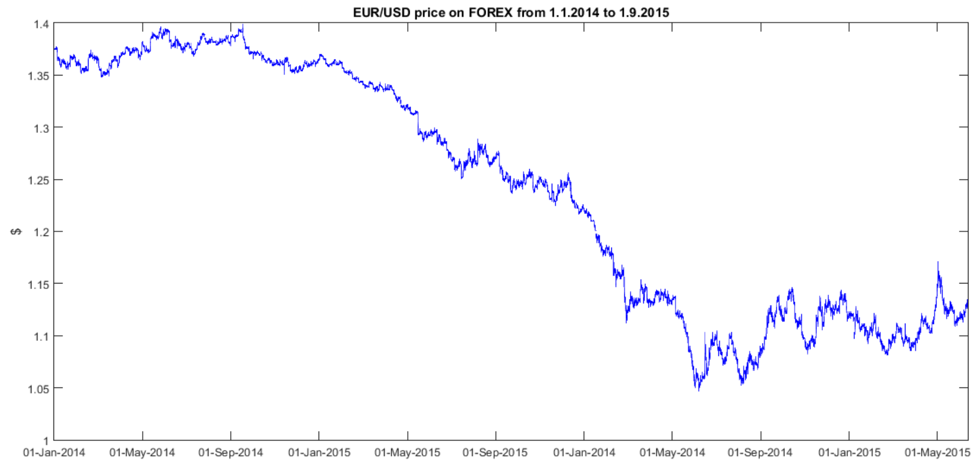


Figure 8: EUR/USD currency pair 1hour time frame

4.1.1 Simplified system settings

Our implementation of trading system was programmed in Matlab. The data used for the analysis were EUR/USD FOREX currency pair, since the data of futures sometimes suffers from jumps between changes of the season and this adds more complexity into the problem. Those data are from 1.1.2014 to 1.9.2015 with time frame of 1 hour and are shown in the Figure 8.

For all analysis always only the opening price (when trades are started) and closing price (when they are closed) is used. Furthermore simpler version of signal generator was used, particularly two moving averages(2 MA). For further simplification take profit and stop loss parameter (as explained in section 2.3) were not used, thus leaving the decisions solely on the moving averages. This signal generator is shown in Figure 9.

In this case 2 moving averages of data are used to generate signal. The idea behind this generator is that one of the moving averages has shorter window, thus responding to the changes in data faster. The other one has longer window and resembles longer term changes. Once the shorter moving average crosses longer one from bellow, long contract is opened. It is expected that there is short-term trend of rising price which is described by the shorter moving average. And conversely when the shorter crosses the longer from above the price is expected to decrease. In Figure 9 long signal is

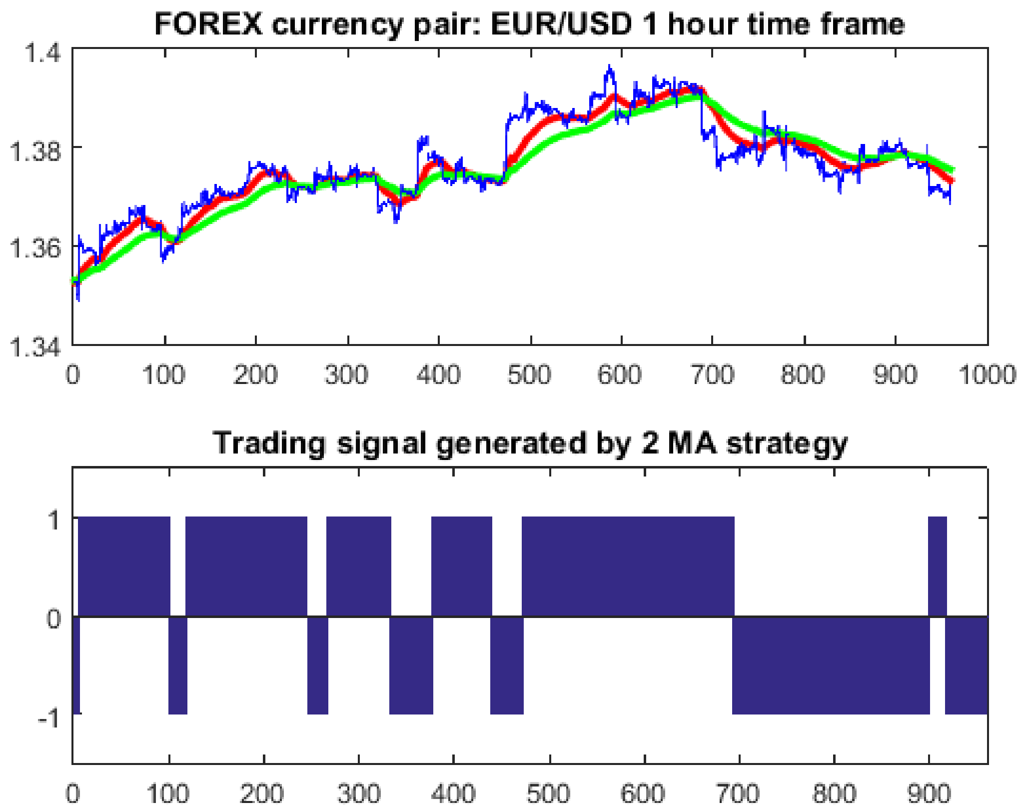


Figure 9: Upper subplot shows data (blue) with shorter moving average (red) and longer moving average (green) and bottom subplot shows signal generated by those moving averages.

represented by bars with value 1 and short with value -1. This signal generator thus have 2 parameters l_1 - length of the window of the shorter moving average and l_2 - length of the window of longer moving average and in this sense is similar to STline. Both STline and moving average are called technical indicators in financial world. In our approach the exponential moving average is used. As opposed to simple where all the past values have the same weight, in exponential moving average data in the window are weighted exponentially, putting higher weight to more recent data than to older ones.

The exponential moving average used in this work was proposed by [3]. Since the implementation is done in Matlab, function tsmovavg which is part of financial time series toolbox, is used for this computation. The interval where l_1 is sought is $[1, 50]$ and for l_2 is $[51, 120]$. These intervals were set empirically. Furthermore objective function is simplified compared to one from section 2.5. Now it will only take into account overall profit in the in-sample period.

4.1.2 Parameter recalibration

In this section we study behaviour of the system when lengths of in-sample and out-sample periods are varying. In-sample length will be varied in interval $[100, 500]$ with increment 50 and the out-sample length in interval $[100, 300]$ with same increment. The combination of those different lengths will create 45 pairs (9 different in-sample lengths and 5 different out-sample). Optimization procedure and backtest will be similar to the one described in section 2.4, here it will be done in this way:

1. Starting from the oldest date take data of length of in-sample period,
2. Find the best parameters $l1$ and $l2$ for moving averages according to objective function (giving maximal profit) in that period.
3. Use two moving averages with those parameters found to trade in the out-sample period.

	in-sample length	**out-sample length**	**total profit out-sample**
1	500	300	**0.01648**
2	400	300	**0.00494**
3	350	100	**0.001**
4	150	200	**-0.00931**
5	100	150	**-0.01007**
6	450	300	**-0.0125**
7	500	250	**-0.01832**
8	450	150	**-0.01879**
9	250	300	**-0.02031**
10	350	200	**-0.02098**

	avg # of trades in-sampe	**average profit in-sample**	**# of trades out-sampe**
1	11.6875	0.025655938	275
2	8.9375	0.02173875	227
3	7.285714286	0.020978878	273
4	2.86	0.013042	322
5	2.313432836	0.009407164	465
6	9.125	0.024211563	238
7	11.18421053	0.024198947	229
8	9.138461538	0.024916154	256
9	5.333333333	0.017632727	255
10	7.734693878	0.02144449	281

Table 2: Ten best results of the system for varying in-sample and out-sample lengths.

4. Add the data from out-sample to in-sample and put away the oldest data from in-sample (keep the length of in-sample constant).

5. Repeat the procedure from step 2 until the latest date in the data is reached.

6. Evaluate the total profit and loss. This means checking which trades were done in out-sample periods.

This optimization and backtest procedure with one particular set of in-sample and out-sample lengths (as parameters) will be called system. Thus 45 systems were tested. They were ordered by the profit which they generated over the period of used data and best 10 of those systems are shown in Table 2.

In first two columns of upper part of Table 2 lengths of in-sample and out-sample

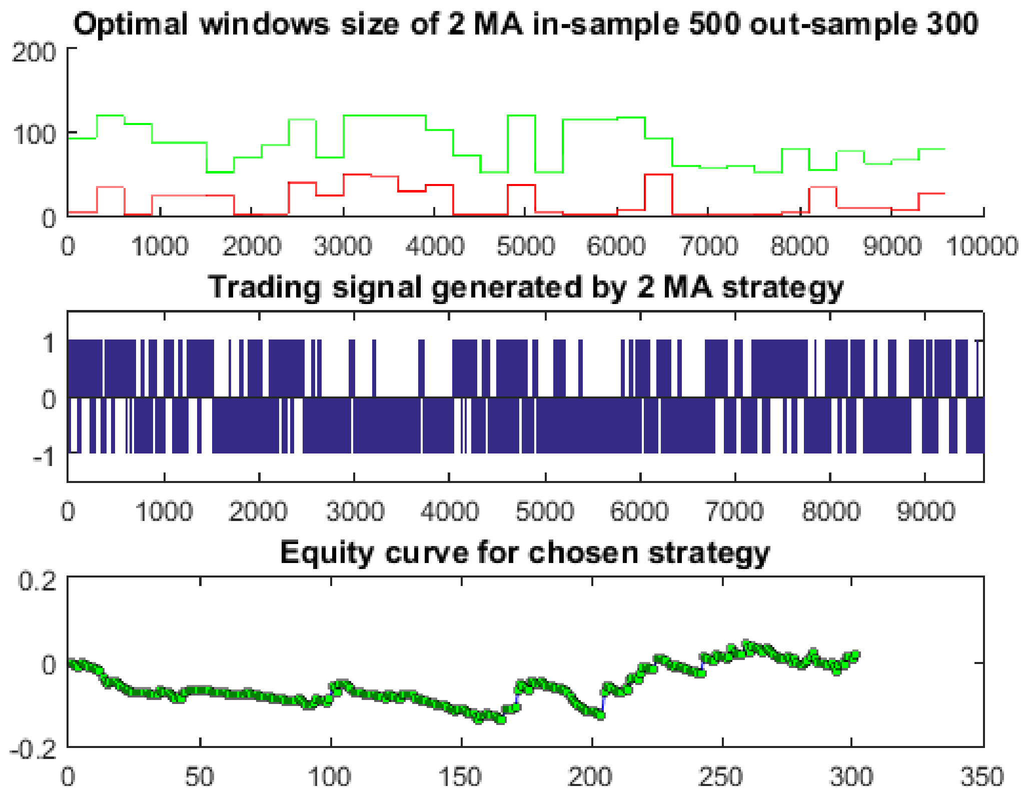


Figure 10: Best performing system (500) in-sample (300) out-sample

periods are shown and in third is total profit which was generated in all out-sample periods. In table below first column shows average number of trades in one in-sample period and in second column of this table is average profit generated in one in-sample period. In third column we can see total number of trades which would be done in all out-sample periods. The system with best performance was one which had in-sample $length = 500$ and out-sample $length = 300$. That system results are plotted in Figure 10. The first subplot represents evolution of parameter l_1 (red) and l_2 (green), the second shows signal which was generated by moving averages with those parameters. The last subplot shows equity curve of this system.

4.1.3 Optimization with last trade taken into account

When the end of out-sample period is reached new parameters l_1 and l_2 are sought with optimization.The system described in 4.1.2 closes any open trades before it starts this search. In this section we propose that, instead of closing opened trades before optimization we take them into account. This is best described by an example. Imagine that the last trade was a long trade opened 3 time frames before optimization. This means that the last part of signal is $[1, 1, 1]$. Now when new parameters are sought, only the ones which have the same array in that particular position (parameters which would open the same trade for that data), qualify for the optimization. To compare this new approach with the old one we took 10 best systems from section 4.1.2 and added this last trade condition into them. Table 3 shows results of those improved systems, ordered by the best profit.

Except last three systems, all the systems improved the final result. Best improvement can be seen on system with lengths 350 in-sample and 200 out-sample, which was the worst out of the 10 in section 4.1.2 and ended up being best after the improvement. What is interesting is that the average profit in-sample is generally lower due to the restriction of the last trade. This can be seen as well on the number of trades in-sample which are generally lower too. Figure 11 demonstrates comparison for results of system (with in-sample length = 350 and out-sample length = 200) with and without the last trade taken into account in optimization.

It is evident that this simple improvement has substantial effect on the system. The difference can be seen in the first subplot of the (a) and (b) figures. The second system parameters l_1 and l_2 suffers from less jump which seems to have positive effect on the results.

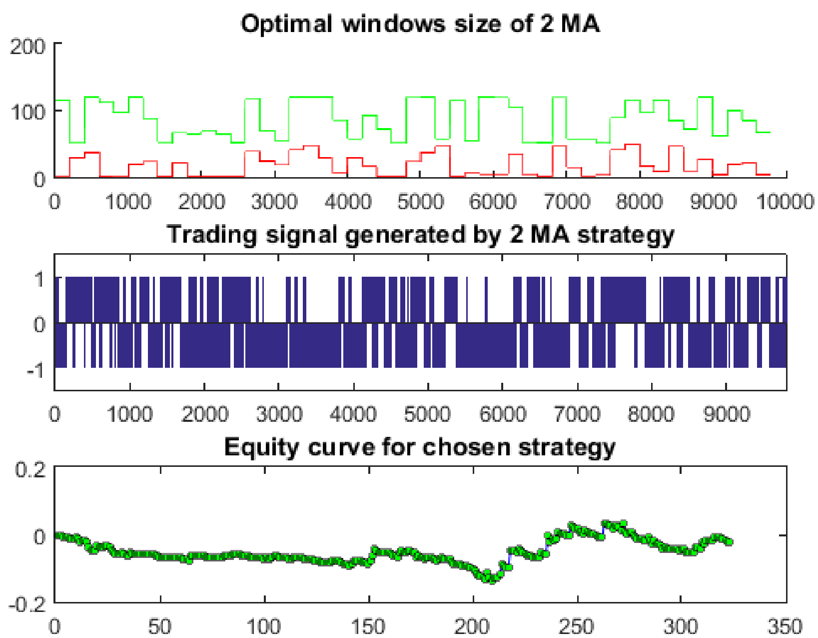
4.1.4 Less optimal parameters

In this part the optimization was done as in section 4.1.2 and with the set of 5 best system from that section. However, now in the optimization instead of selecting the

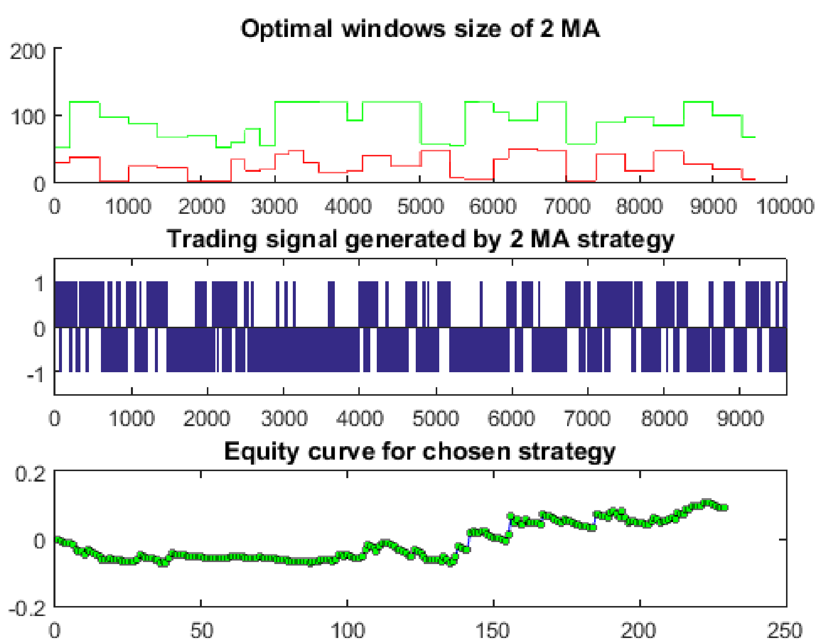
	in-sample length	**out-sample length**	**total profit out-sample**
1	350	200	**0.09353**
2	100	150	**0.08732**
3	350	100	**0.0387**
4	450	300	**0.03631**
5	500	250	**0.02231**
6	500	300	**0.01855**
7	400	300	**-0.00057**
8	150	200	**-0.03248**
9	250	300	**-0.05415**
10	450	150	**-0.05505**

	average # of trades in-sampe	**average profit in-sample**	**# of trades out-sampe**
1	3.387755102	0.013228163	185
2	1.402985075	0.00648791	296
3	4.255102041	0.015579286	211
4	5.3125	0.017660938	244
5	7.026315789	0.015273421	235
6	6.5	0.017143438	238
7	4.71875	0.016070938	236
8	1.84	0.0091998	288
9	2.696969697	0.012560303	200
10	5.323076923	0.016009538	224

Table 3: Ten best results of the system for varying in-sample and out-sample lengths improved with last trade condition.



(a)



(b)

Figure 11: (a) System without check for last trade (b) Improved system with recalibration with last trade in account

parameters with highest profit we select parameters with lower profit. First we take those 5 selected systems and in optimization we select parameters which would generate second highest profit in in-sample as optimal. After that we do it again, this time selecting the parameters which would end up third in optimization. We do this until the sixth most optimal parameters. In Table 5 results of those systems are presented. In third column the number denotes the rank of selected parameters in optimization, starting from 0 which stands for the parameters with highest profit and ending with 5 which stands for parameters which are sixth in means of highest profit.

In case of first two system (with in-sample lengths of 500 and 400 and out-sample lengths 300) the best parameters translated into the best out-sample profit too. However in 3 other cases the best parameters performed relatively bad compared to others. For example for system 350 100 the parameters which would create highest profit would be the second best in-sample and right after those would be sixth best in-sample. This shows that finding the best parameters for in-sample simply does not imply that those parameters will be the best in out-sample as well.

4.2 Intrinsic mode functions as signal generator

As described in section 3.1 EMD can decompose data into intrinsic mode functions. Those functions then can be summed together to reconstruct the data. However if only some part of the intrinsic mode functions are summed together the result is smoothed data (see Fig. 12). For this experiment improved implementation of EMD was used which was developed by Flandrin at al. in [5]. In this section two different levels of smoothing (trends) which are obtained by different number of IMFs summed (for example sum of 3 IMFs for smoother and sum of 6 IMFs for coarser trend) are used as signal generator with same logic applied as with moving averages. When the coarser one crosses the smoother one from underneath long position is opened and when the coarser crosses the smoother from above short position is opened. Again similar optimization procedure as in section 4.1.2 is used, now with the difference that instead of searching

in-sample length	out-sample length	Optimality of parameters	total profit out-sample
500	**300**	**0**	**0.01648**
500	300	**2**	-0.02958
500	300	**5**	-0.02966
500	300	**3**	-0.04242
500	300	**1**	-0.05424
500	300	**4**	-0.06982
400	**300**	**0**	**0.00494**
400	300	**1**	-0.0586
400	300	**5**	-0.08158
400	300	**3**	-0.08224
400	300	**2**	-0.0949
400	300	**4**	-0.14234
350	**100**	**1**	**0.03674**
350	100	**5**	0.0236
350	100	**4**	0.0029
350	100	**3**	0.00196
350	100	**0**	0.001
350	100	**2**	-0.00582
150	**200**	**1**	**0.03269**
150	200	**2**	0.03105
150	200	**3**	0.02779
150	200	**5**	0.00771
150	200	**4**	-0.00521
150	200	**0**	-0.00931
100	**150**	**2**	**0.08111**
100	150	**4**	0.07479
100	150	**1**	0.05749
100	150	**3**	0.05687
100	150	**5**	0.03477
100	150	**0**	-0.01007

Table 4: System using less optimal parameters

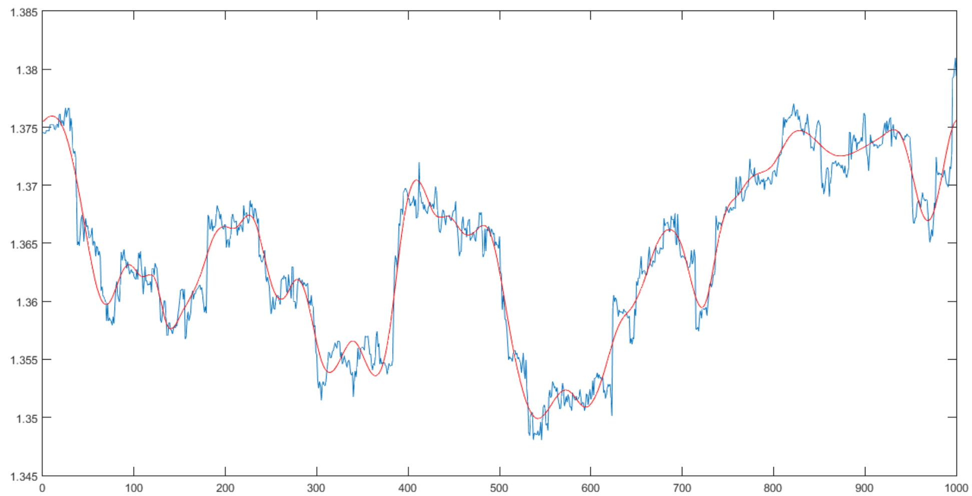


Figure 12: Part of data (blue) with sum of 5 imfs (red)

for l_1 and l_2 (lengths of 2 moving averages) we search (in in-sample period) how many IMFs should be summed for the smoother trend and how many should be summed for the coarser trend to produce highest profit. This numbers of sums are then used when trading in out-sample period. For example let us assume that it was found that in-sample period best profit would be generated by sum of 5 IMFs as coarser trend and sum of 3 IMFs as smoother trend. Figure 13 shows example of 2 different sums of IMFs used as signal generator with signal generated by them.

In Table 6 we can see results of such system again for 10 best systems from the first experiment. The best system has in-sample length 450 and out-sample length 150 and can be seen in Figure 14.

Even though the results looks promising it has to be noted that all the tests are computed without taking into account transaction costs. Thus results with higher number of trades would perform worse in real trading. In these results we can see high number of trades due to instability of EMD when it comes to computing IMFs every time frame (see Fig. 15) as opposed to when we compute them in in-sample where all the data is know.

On the other hand in in-sample where all the data is known the IMFs are computed

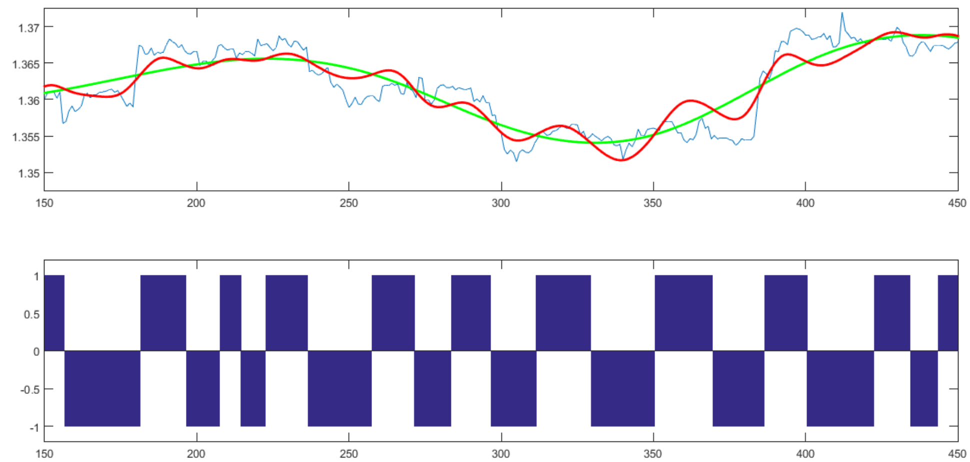


Figure 13: In upper sub plot we can see data (blue) sum of 3 IMFs (green) and sum of 6 IMFs (red) and in subplot bellow signal generated by those sums is shown.

	in-sample length	**out-sample length**	**total profit out-sample**
1	450	150	**0.31694**
2	400	300	**0.28747**
3	450	300	**0.26169**
4	500	300	**0.24752**
5	100	150	**0.17519**
6	500	250	**0.17433**
7	250	300	**0.07138**
8	350	200	**0.04935**
9	150	200	**0.03437**
10	350	100	**-0.03427**

	average # of trades in-sampe	**average profit in-sample**	**# of trades out-sampe**
1	1.092307692	0.012190769	1121
2	1.787878788	0.01282697	1025
3	1.1875	0.0136625	902
4	1.1875	0.013595625	913
5	1.647058824	0.001368529	1249
6	1.076923077	0.013453846	928
7	2.424242424	0.006446364	933
8	1.204081633	0.010901429	976
9	0.98	0.0046422	1215
10	1.191919192	0.009637071	1222

Table 5: Systems with IMFs as signal generator

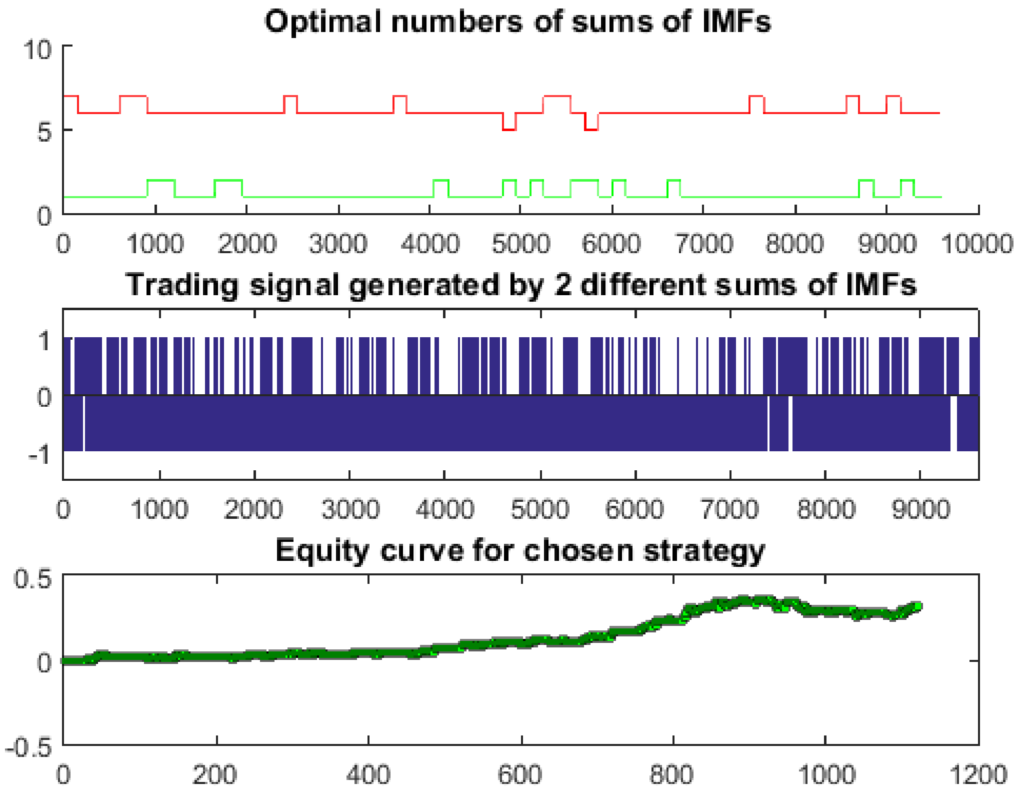


Figure 14: Best strategy basted on imf with 450 in-sample and 150 out-sample

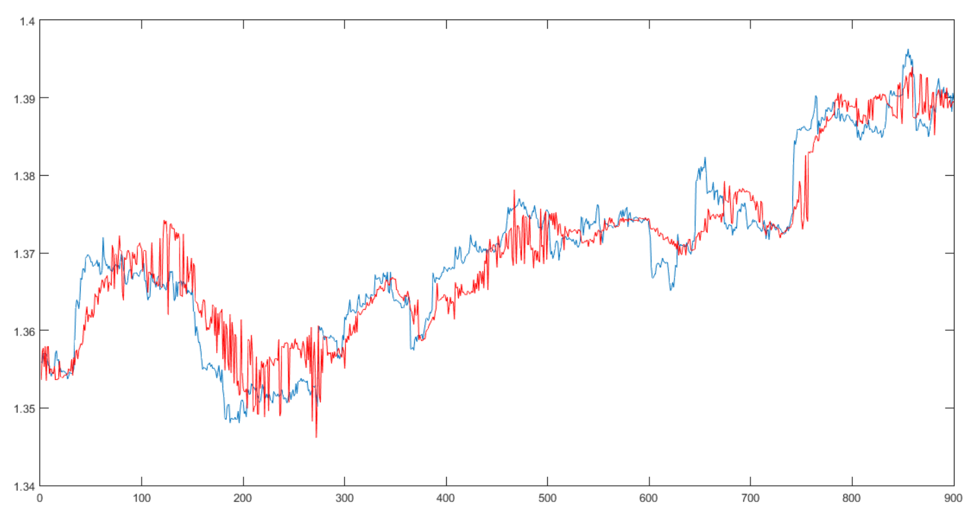


Figure 15: Data (blue) and sum of IMFs with jumps (red)

without jumps thus creating much smaller number of trades there.

5 Conclusion

This work served us as a valuable entry point to the topic of automated trading. We described one of the approaches and used it for our implementation. From the results which were obtained, we can say that finding optimal lengths for in-sample and out-sample periods could be proposed as another optimization problem. Furthermore if the condition where we try to preserve opened trade in time of optimization is added to the system it has positive impact on results of the system. However it was also found that the optimization procedure as designed now, fails to find always the best set of parameters for trading in future, which makes this approach to automated trading questionable. In the end we presented system which used sum of intrinsic mode functions as signal generator. This system had the best results from all tested systems and it could possibly be used in actual trading. However further research of this system would be required to make it usable on financial markets.

Resumé

V tejto práci sme sa zaoberali automatickým obchodovaním. Predstavili sme jeden obchodovací systém a objasnili sme jeho komponenty. Tento systém sa skladá z generátoru signálu, ktorý udáva akým spôsobom sa budú vykonávať obchody. Opísali sme takzvanú STLine, ktorá sa ako takýto generátor používa. Ďalej sme vysvetlili typy pozícií pri obchodvaní. Následne sme predstavili optimalizačnú procedúru, ktorá má za úlohu hľadať správne parametre napríklad pre STLine, vďaka ktorým má systém generovať profit. To či systém ho naozaj aj generuje sa testuje takzvaným backtestom. Na optimalizáciu sa dá využiť napríklad genetický algoritmus, ktorý sme popísali. Následne sme predstavili výsledky takéhoto systému, ktoré však demonštrujú, že takto zostavený systém nie je použiteľný na obchodovanie v realite. V ďalšej časti skúmame dĺžky dát

na ktorých je vhodné optimalizovať a taktiež dĺžky dát na ktorých je potom možné, parametre nájdené pri optimalizácii, použiť. Podarilo sa nám nájsť 10 kombinácií dĺžok, ktoré mali najlepšie výsledky. Tieto výsledky však tiež neboli veľmi priaznivé, keďže viac ako polovica týchto vybratých systémov, by nás priviedla k strate. Preto sme v ďalšej časti navrhli vylepšenie optimalizácie, ktoré spočívalo v tom, že keď hľadáme nové optimálne parametre, zahŕňame do tohto procesu aj trade, ktorý bol v čase optimalizácie otvorený. Vďaka tomuto vylepšeniu sa výsledky siedmych týchto systémov zlepšili. Ďalej sme skúmali ako dobre vie procedúra optimizácie určiť parametre, vďaka ktorým by sme mali získať profit. Robili sme to spôsobom pri ktorom sme namiesto najlepších parametrov v optimalizácii, vyberali tie menej optimálne. Zistili sme, že optimálne parametre nemajú vždy najlepšie obchodovacie výsledky. Tým pádom sa dá tento prístup optimalizácie a následného obchodovania označiť, za nie úplne vhodný. Posledná časť sa zaoberala obchodovaním pomocou takzvaných intrinsic mode funkcií, ktoré získavame pomocou empirical mode dekompozície. Súčtom týchto funkcií sa dá nahradiť trend s rôznou mierou zhladenia. Takto vytvoríme dva rôzne trendy dát, ktoré následne využijeme ako generátor obchodovacieho signálu. Výsledky získané takýmto systémom vyzerajú nádejne. Bolo by ich však najprv treba ešte overiť na rôznych iných dátach, kým by sme ich mohli použiť v realite.

References

[1] Automated Trading, Posted by Automated Trading January 13, 2014 `http://www.automatedtrading.com/2014/01/13/history-trading-systems/`

[2] Wilder, J. Welles. *New Concepts in Technical Trading Systems.* Greensboro, N.C.: Trend Research, Print, 1978

[3] Hunter, J. Stuart Princeton, NJ, *The Exponentially Weighted Moving Average* Journal of Quality Technology,Vol. 18, No. 4, pp. 203-210, 1986

[4] Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis.* Proc. R. Soc. London, Ser. A, 454, 903–995, 1998

[5] G. Rilling, P. Flandrin, P. Goncalves, *On Empirical Mode Decomposition and its Algorithms* IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing NSIP-03, 2003