FIN11- Algorithmic trading



Magnus Solheim
s145997

Tormod Skov-Skov
s145277

Claude Wilhelm Wiese
s144932

Bendik Kildahl Larsen
s155000

Abstract

This assignment will examine algorithmic trading in the foreign exchange (FX) market, and how to set up and use a trading robot. First we will explain what algorithmic trading is as well as its role in today's markets. Further on, the assignment will give a short description of the FX market and why we used this market to test our robot. We will then explain how trading robots are programmed, elaborate on key parts of our strategy and explain difficulties that occurred. Finally, we will test the robot on three different currency pairs and evaluate its performance.

The reason we wanted to learn more about algorithmic trading robots is because we feel there is limited opportunities to learn more about it in business schools today. After attending the guest lectures with both Rolf Palmer (head of FX at DNB Markets) and Øyvind Schanke (Chief Investment Officer at Norges Bank Investment Management) who talked much about algorithmic trading, it further fueled our interest to learn more about the field and we understood the importance and impact trading robots has on trading in today's markets. We learned that this is a common method of trading, and we think it will be even more important in the future.

Contents

1 Automated trading **2**

2 The software program **2**

3 Generally about an EA **2**

4 The FX market **3**
4.1 Why we chose the FX market 3
4.2 Generally about the FX market 3

5 Technical analysis **3**
5.1 Single moving average (SMA) 3
5.2 Average true range (ATR) 3

6 The EA strategy **4**

7 Challenges that occurred **5**

8 Expectations **6**

9 Results **6**

10 Evaluation of performance **8**

11 Summary/Conclusion **9**

12 Limitations **9**

13 Appendix **11**
13.1 Codes used in the EA 11
13.2 Other figures 14

1 Automated trading

Automated trading systems are used by traders to execute trades without human intervention. By using an automated trading system, traders are able to set specific rules for both trade entries that decides when to buy or sell and trade exits that decide when to close out the position. These are rules which can be performed automatically by computers once they are programmed. It is important that all the rules are part of a strategy. Such strategies could be based on fundamental-, statistical- or technical analysis.

There are several advantages and disadvantages by using an automated trading system. The most fundamental advantage is that trades are performed without humanly emotions (Nasdaq, 2016). When humans trade, they are constantly in danger of letting their emotions influence their trading patterns, and thereby not following their pre-set strategy. This could lead to an even greater loss or a missed opportunity to make profit. By using a computer to perform the trades, no emotions will affect the strategy. Even though this is true, it does not mean that trading with an automated system is completely without emotions. The trading robot has no emotions, but the trader who controls it may be tempted to change its pre-set strategy during unprofitable periods. Another advantage using automated trading is that you can backtest your rules before investing real money. By testing the strategy on historical data, it enables the trader to see if it performed as expected, and evaluate the performance for further improvements. However, there is a risk of over optimization which can lead to an insufficient strategy.

When using a fully automated trading system there is always a risk of mechanical failure. If there is a internet connection loss, some of the orders that should have been sent, may still be on the computer. This can lead to missing- or duplicate orders. Because of this risk, the systems require monitoring, which could be time consuming. Another disadvantage is that an automated trading system is inflexible. It will only execute trades based on the pre set coding and is therefore limited to the implemented logic. This can be a problem during changing market conditions if the trading robots ability to adapt is limited.

2 The software program

When we first decided to make an algorithmic trading robot, we had to decide which software to use. There are several different options with different advantages and disadvantages. We looked into MATLAB and Excel, but the problem here is that the software is not designed to build trading robots and the back-tester needs to be built from scratch. We finally decided to use MetaTrader 4 to build and implement the robot, because this is a free program, and it provides free live and historic data on different asset classes. MetaTrader utilizes a scripting language called MQ4, where we can develop trading robots hereafter referred to as an EA (expert advisor).

3 Generally about an EA

This part describes the different parts of an EA. An EA always consists of three parts; Entry, exit and sizing. The entry part tells the EA what it should look for before entering any trades, the exit tells the EA what it should look for when exiting a trade, and the sizing decides the quantities the EA will make.

From a coding point of view, the EA would operate a little bit differently. There are four main parts; Setup, initialize, start and deinitialize.

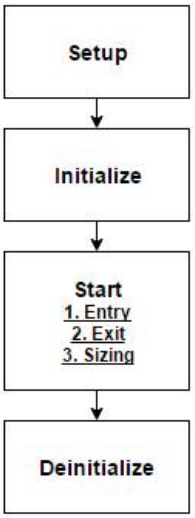


Figure 1: *Shows the different parts of an EA.*

In the setup part, all the things that are required for the EA to work are written. In the initialize part, all the codes will run once, when the EA is executed. It will typically check if the broker is a four or five digit broker. This means if the broker offers four or five decimal quotations. This is important because the robot cannot differentiate the last decimal point if it is not coded, and will therefore also make sizing and set stop-loss levels that are not intended. If the EA wants to trade Yen, a similar code must be included in the initialize part of the robot. This is because Yen is the only major currency pair that uses a three decimal quotation system.

The most important part is the start part. This part of the code will run once the initialize part is done, and then it will run one time, each time there comes in new information. This means that it runs on every tick, synonymous to every time the price of an asset changes. The first thing the EA will do in the start part is to look for positions to be exited and then for any possible entries. If the EA finds an entry, it will thereafter look for the appropriate sizing. In the deinitialize part, all

the codes run once after the EA is stopped. We have not included anything in this part of our code as it is a less important part. Typically, a code that alerts you when the EA is stopped will be included in this part.

A more detailed description about each line of code will be found in Appendix 1. We have inserted comments next to the code so that it is possible to understand the meaning of each line.

4 The FX market

4.1 Why we chose the FX market

When we first started using MetaTrader we opened a demo account with 10,000 USD. The demo account allows us to create our own algorithm, and test them on multiple different currencies as well as commodities. MetaTrader has no historical data on stocks or indexes, and this resulted in us choosing the FX market. An advantage of choosing the FX market is that we do not have to take stock splits, dividends etc. into account which makes our testing and data cleaning process much easier. The FX market is open 24 hours a day, five and a half days a week which makes the statistical data more significant. This is a good thing, because our strategy is based on technical analysis, and with more available data it is easier to observe if patterns actually exists.

4.2 Generally about the FX market

The FX market is a global decentralized market for currency trading. It includes all aspects of trading currencies at current or determined prices. The main participants in the FX market are governments, central banks, banks and other financial institutions, hedgers and speculators (colorado.edu). The FX market is an electronic over-the-counter market where you can trade currencies worldwide 24 hours a day, five and a half days a week. It is the biggest market in the world, in terms of traded volume. In April 2016 the FX market average trading per day was 5.1 trillion dollars (bis.org).

The FX market is one of the least volatile markets. Most of the currency pairs move less than 1% per day. Most FX traders use leverage to increase the value of potential movements in the market. Some brokers offer you leverage as high as 250:1. This means that for every dollar you put into the market, you can trade for 250. Extreme liquidity and the availability of high leverage helped to spur the market's rapid growth and made it ideal for many traders. Positions can be open and closed within seconds or they can be held for months. Because of the market size, none of the largest players are able to move the prices since the currency prices are based on objective considerations of supply and demand (berkeley, 2016).

FX trading are executed using spot transactions or derivatives. The pricing is done using currency pairs and they are quoted either directly or indirectly. In each currency pair, the first currency is the base current and the second is the quote current. GBP/USD=1.8040 is an example of a FX quote, and it represents the exchange rate of the two currencies. The number 4 indicates the Pip value, which is a widely used term in FX trading. The most traded currency pairs are shown in the table below:

EUR/USD
USD/CAD
AUD/USD
USD/CHF
GBP/USD

Table 1: *The five most traded currency pairs*

Currencies always trade in pairs. A trader is long in one currency and short in the other. For example, if a trader sells EUR/USD, he would have exchanged euros for dollar and would now be short in euros and long in dollars. (colorado.edu)

5 Technical analysis

Our strategy is based on technical analysis. Technical Analysis is the forecasting of future financial price movements based on an examination of past and current price movements (Harris, 2003). Technical analysis is not supposed to produce absolute predictions about future price movements, but rather tries to anticipate what is most likely to happen. Before we will explain the actual strategy of the EA, we will explain the indicators that our strategy is based on.

5.1 Single moving average (SMA)

A single moving average (SMA) is the unweighted mean of the previous n data. In financial applications it is normally calculated as the sum of closing prices divided by the number of closing prices. We are using 10- and 40 periods simple moving averages as default in our EA code, but these will be optimized to fit the best possible result for different periods.

$$SMA = \frac{Pm + Pm - 1 + Pm + ... + Pm(n-1)}{n} \quad (1)$$

$$= \frac{1}{n} \sum_{i=0}^{n-1} Pm - i \quad (2)$$

5.2 Average true range (ATR)

Average true range (ATR) is an indicator developed by Welles Wilder (stockcharts, 2016). The indicator measures volatility. The true range (TR) is calculated as the greatest of the following:

1. *Current high less the current low* (3)

2. *Current high less the previous close* (*absolute value*) (4)

3. *Current low less the previous close* (*absolute value*) (5)

This can also be seen in the figure below.

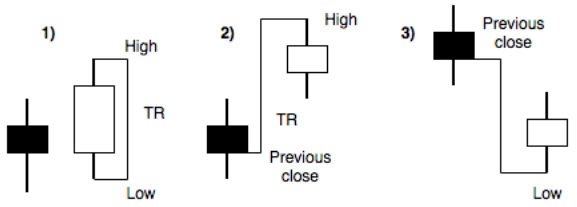


Figure 2: *ATR is the greatest of the following.*

For ATR to make any sense combined with SMA it should be somewhere between the two 10- and 40 day moving averages that we already have decided to use. We have chosen to use a 20 period ATR as the default in our code, but this could also be optimized later. The first 20-period ATR is calculated as the average of the TR values for the last 20 days. Wilder also came up with an idea to smoothen the data by incorporating the previous ATR value. Therefore the current ATR(20) can be calculated like this:

$$Current\ ATR(20) = \frac{Prior\ ATR(20) \times 19 + current\ TR}{20} \quad (6)$$

ATR is not a directional indicator such as MACD (moving average convergence/divergence oscillator) or RSI (relative strength index). Instead, ATR is a unique volatility indicator that reflects the degree of interest or disinterest in a move. Strong moves, in either direction, are often accompanied by large true ranges (TR). This is especially true at the beginning of a move (stockcharts, 2016). Uninspiring moves can be accompanied by relatively narrow ranges. Therefore ATR can be used to validate the enthusiasm behind a move or breakout. A bullish reversal with an increase in ATR would show strong buying pressure and reinforce the reversal. A bearish support break with an increase in ATR would show strong selling pressure and reinforce the support break. In the code, we are also implementing something we call ATR-shift. We are using an ATR-shift of 10 periods. This means that a signal will be triggered if ATR(20) is currently greater than it was 10 periods ago.

6 The EA strategy

The first robot we made did not include ATR in the coding. We implemented a default strategy to the EA where it would entry a long trade when SMA(10) crosses SMA(40) from the bottom and enter a short trade when SMA(10) crosses SMA(40) from the top. The exit rules are similar. We are exiting a long trade when SMA(10) crosses SMA(40) from the top, and exiting a short trade when SMA(10) crosses SMA(40) from the bottom.

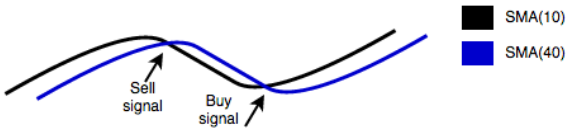


Figure 3: *Example of triggered SMA signals*

The sizing was set to 0.33 lot per trade, and would grow in proportion with the balance in the account. An example would be if our balance grew from our initial balance of 10,000 USD to 20,000 USD the sizing would now be 0.66 lots per trade. The risk per trade is also attached to the balance. We would risk 1% of our portfolio per trade no matter what the balance was. So, if our balance grew to 20,000 USD the risk would still be 1% of the portfolio equal to 200 USD per trade. The stop loss was therefore set to be 30 pips behind the price. The trailing stop loss would also follow the price by 30 pips.

However, after backtesting the robot we encountered a problem. The robot triggered a lot of false signals which resulted in a lot of unprofitable trades. To fix this problem we implemented the ATR rule to the robot before allowing it to execute any trades. This rule is based on current volatility being greater than previous volatility. More accurately, if current volatility (ATR(20)) is greater than volatility (ATR(20)) 10 hours ago, the robot will enter a long trade when SMA(10) crosses SMA(40) from the bottom and enter a short trade when SMA(10) crosses SMA(40) from the top. We also changed the exit rules. Now the EA would only exit a trade if it reached the stop loss or the trailing stop loss level. Both of these parameters were still 30 pips behind the price.

We thought this rule would remove much of the false signals because trends form when there is a strong excess of buying or selling pressure. When there is volatility in the market, sellers and buyers tend to disagree on the price, and in many cases one side will essentially lose, and the market will go in the other direction. The market participants on the losing side further amplify this because a lot of traders will take losses and stop loss levels will be triggered. Therefore we believe there to be an increased chance of trends forming when there is increased volatility, and that is what we try to exploit by entering this new trading rule.

7 Challenges that occurred

In this section, we will highlight some of the difficulties that occurred before we could backtest with reliable results. The challenges were not mainly to write the code, but rather to obtain a reliable and accurate data set. A clean data set is critical to perform an accurate back-test. These challenges will be explained separately below.

The data set we used is provided by MetaTrader and it includes the opening- and closing price, highs and lows as well as the volume. MetaTrader does not reveal where their data comes from so it was difficult for us to know how reliable the data was. The fact that the data set was free may also be a concern, as we have learnt that there is a huge industry to buy data from more reliable sources than MetaTrader to ensure that you have clean data. However, it is also common practice to clean the data yourself, which is what we did. We did this by exporting the data set from the desired time period to Excel. We would then look for duplicate, missing and wrong entries. We found some and changed them so that the closing price for one period was equal to the opening for the next period. This makes sense because the closing price of one bar would be equal to the opening for the next bar. The one exception is for the last bar on Fridays and the opening bar on Mondays where the price can differentiate, so naturally, we did not make any changes between these bars.

A more difficult problem to fix, is that the data set may also include spikes and bad ticks. A Spike is when there is a sudden and huge change in the price of an asset and bad tick is equal to an inaccurate or incorrect trade.

An example of a bad tick would be if someone needed to close out their short position when there was low liquidity in the market and made a buy order at X+5 pips when the asset was trading at a price of X. Bad ticks tends to be of extremely low volume. Bad ticks are a problem because even though there has been executed a trade at this level it is impossible for us to act on it and our back-tester cannot differentiate bad ticks from clean data. Therefore, there is a problem that the EA may execute trades at these prices, and thus artificially inflates the performance of our backtesting and making it dangerously unreliable.

The exact same problem occurs when there is a spike in the market. The biggest spikes tend to happen when there are announced big news announcements, but it can also occur during normal market conditions. A recent example of a huge spike without any major macro- economic news, was when the USD/GBP reached its lowest quotations in 30 years on the 7th of October 2016. According to Business insider this was caused by one trader alone (Williams-Grut, 2016). However, the investigation is still preceding. When a spike occurs, there are very few prices where there is any trading. This makes it impossible to enter a trade at these levels for someone without a high-frequency trading infrastructure. The back-tester cannot differentiate spikes from clean data so the same problem occurs as for bad ticks.

Our solution to deal with both spikes and bad ticks, was to analyze and backtest on hourly periods instead of shorter timeframes. This means that one bar is equal to one hour. By doing this, we are minimizing the effect that both spikes and bad ticks will have on our EA, because we are reducing the ability of the EA to overreact on such events. However, the problem does not disappear completely, but it was our best solution to encounter the problem.

The last problem that we want to highlight is widening of the spread. Brokers and liquidity suppliers may widen the bid/ask spread (BBO) when there is low liquidity in the market. An example would be ahead of non farm-payrolls, a monthly released report of the total number of paid U.S. workers of any business, where brokers and liquidity providers typically would protect themselves from spikes in prices by widening the spread. When the spread widens it can trigger the stop loss, which again will result in unprofitable trades. When we are backtesting in MetaTrader we are using a fixed spread, and this can give us unreliable results as this would not be the case in real life trading. As we found no optimal solution to fix this problem, we sat the fixed spread in the back-tester to 2.0. As the average spread for the currencies that we use is much smaller than this, we believe that if we still get a profitable EA with the 2.0 spread it will withstand widening of the spread also in real life.

EUR/USD	0.2
EUR/GBP	0.8
USD/CHF	0.95

Table 2: *Average spread last quarter of 2016*

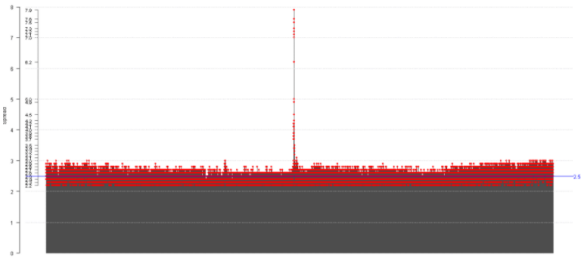


Figure 4: *An example of widening of the spread for EUR/GBP on the 10.05.2012. The extreme widening of the spread only lasts for a brief moment.*

Even though it is possible that some errors is still in the dataset after this procedure, we believe it contained good enough information for our purpose.

8 Expectations

Our expectation is to make a profit that is greater than what you would get at a risk-free interest rate, which is 0.81% based on a 1 year US treasury bill yield (Bloomberg, 2016). The risk free rate is a rate with no variance, but as no such rate exists in practice it is customary to use the interest rate associated with the U.S treasury bills, as the market does not recognize the possibility of the U.S defaulting on its standing debt obligations (Brealey, 2014). There will be no point entering this type of market without achieving a greater return. An additional expectation is that the robot should follow our strategy and never deviate from it.

We should also take into consideration some of the big macro events that happened during our two periods of historical data. In the period from 2014.11.30 - 2015.11.30 the Swiss Franc soared by 20% in one day compared to USD (economist,2015). This can be seen as an unusual event, and it could impact the robots trading patterns. The same goes for Brexit and the Trump election in the period from 2015.11.30-2016.11.30.

9 Results

In this section we will present the results. We tested the EA on the currency pairs listed below.

EUR/USD
EUR/GBP
USD/CHF

Table 3: *The three currency pairs used*

First we tested our initial strategy on the three currency pairs in the time period from 2014.11.30 to 2015.11.30. After reviewing the results with the initial parameters, we then optimized the EA parameters for the period. These parameters were; stop loss, trailing stop loss, SMA(10), SMA(40), ATR(20) and ATR- shift.

Default parameter settings	
StopLoss (pips)	30
Trailing stop (pips)	30
sma short	10
sma long	40
atr period	20
atr shift	11

Table 4: *Shows the default parameter settings for the EA*

These where the parameters we optimized for the EA. All the currency pairs started with the same default settings as shown in table 4.

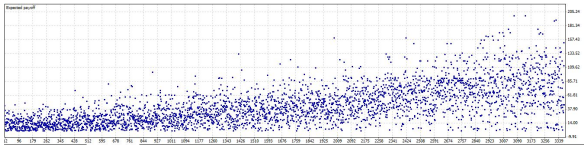


Figure 5: *Shows all the different combinations of the parameters with the expected payout per trade along the vertical axis. The horizontal axis shows the number of tests run.*

We chose the combination of parameters that maximized the profit for the time period. We went through the same process for the three currency pairs and the optimized parameters are shown in table 5. We then tested the optimized parameters in the time period 2015.11.30 to 2016.11.30

Optimized parameter settings

	EUR/USD	**EUR/GBP**	**USD/CHF**
StopLoss	15	10	10
Trailing stop	60	37	67
sma short	17	16	17
sma long	63	20	56
atr period	22	28	40
atr shift	11	5	17

Table 5: *Shows the optimized parameter settings the EA uses on the different currency pairs*

	EUR/USD			EUR/GBP			USD/CHF		
Results	**2015 original**	**2015 optimized**	**2016**	**2015 original**	**2015 optimized**	**2016**	**2015 original**	**2015 optimized**	**2016**
Initial position	10000	10000	10000	10000	10000	10000	10000	10000	10000
Total net profit	-498,24	10809,6	-984,61	-9,46	2696,43	6082,83	-1870,55	20187,03	-2819,49
Total trades	43	24	69	7	20	240	90	65	60
Expected payoff per tra	-11,59	450,4	-14,27	-1,35	134,82	25,35	-20,78	310,57	-46,99
Short positions	20	11	34	3	9	115	46	29	29
Long positions	23	13	35	4	11	125	44	36	31
Profitable trades	15	8	14	3	10	59	32	7	7
Loss trades	28	16	55	4	10	181	58	58	53
Largest profit trade	704,58	3515,94	981,79	122,14	1076,57	2967,09	453,79	8802,5	1119,18
Largest loss trade	-102,35	-223,56	-102	-127,73	-163,77	-217,18	-110,19	-366,29	-101,66
Average profit trade	139,33	1660,93	298,44	97,07	404,05	558,75	93,2	4733,17	227,44
Average loss trade	-92,44	-154,86	-93,87	-75,16	-134,41	-148,53	-83,67	-223,19	-83,24
Final position	9501,76	20809,6	9015,39	9990,54	12696,43	16082,83	8129,45	30187,03	7180,51

Figure 6: *Shows results and key figures for all the currency pairs*

In figure 6 one can observe the results and key figures for each currency pair. Most pairs did not generate a profit with the original settings. When we optimized the parameters, the EA performed better, as we expected. As one can observe from the EUR/GBP pair, the optimized 2015 numbers performed better in 2016 than they did in 2015. An interesting thing we observed is the imbalance between profitable trades and unprofitable trades. For each currency pair most of the trades ends in losses and only a few make a profit. Despite that most of the trades ends in losses, the *average profit trade* is higher than the *average loss trade*. This indicates that the profit optimization is probably not the best solution when it comes to programing the robot, and we will talk more about different solutions in terms of optimizing criteria in the evaluation part of the paper. Another interesting aspect is that long and short positions are equally weighted. This is a good sign that indicates that the robot is able to catch both upwards and downwards trends in the market and act accordingly. That is consistent with our strategy and confirms that it works properly.

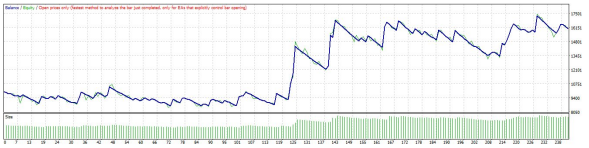


Figure 7: *Optimized balance on EUR/GBP from 25.11.2015 - 25.11.2016*

Figure 7 shows the P&L results and volume traded when the EA traded EUR/GBP in 2016 based on the optimized parameters from 2015. The green line shows our equity, and shows the account balance plus the floating P&L of our open positions. The volume per trade grows in proportion with the balance and is shown as the green lines at the bottom of the figure. The EA performed best on this currency pair and made a profit of approximately 60 percent. In the first half of the time period, the EA underperformed and made a loss. After the first half, our EA performed a series of profitable trades.

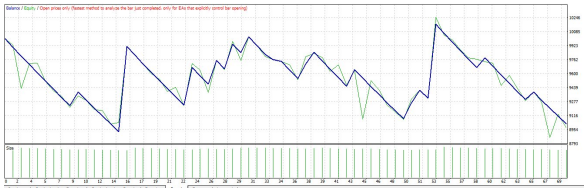


Figure 8: *Optimized balance on EUR/USD from 25.11.2015 - 25.11.2016*

Figure 8 shows the balance and volume traded on the currency pair EUR/USD. The balanced reached a positive P&L only once during the time period. The EUR/USD decreased a lot in November and this may be a result of the presidential election in the United States. Our EA did clearly not manage to catch the downward trends in the Dollar vs. the Euro.

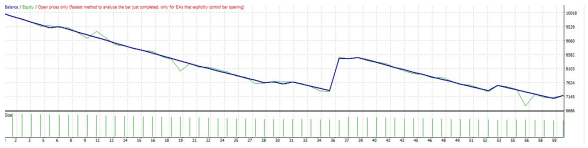


Figure 9: *Optimized balance on EUR/USD from 25.11.2015 - 25.11.2016*

Figure 9 shows the P&L and volume traded for USD/CHF. As one can observe, this development is not profitable, and our balance reached a loss of approximately 30 percent. The period consisted of many trades that made a small loss because it reached the stop-loss boundaries. This is a great example of how we do not want our EA to perform, and it shows that our strategy and choice of optimizing all the parameters only based

on the profit from the previous year was not the best solution.

10 Evaluation of performance

The results we got from the optimized pairs varied a lot as can be seen from the balance figures above. However, the first thing we wanted to check was if the robot executed trades as we wanted it too. An advantage when using MetaTrader is that we can get a visual graph on how the EA performed as well as when it performed trades. An example can be seen in the figure below.



Figure 10: *Visual view of backtesting on the EUR/USD currency pair with default settings*

Figure 10 shows two trades performed by the EA with the default settings. The first trade is a short trade, which can be seen as the red arrow on the figure. The trade gets closed when it reaches the orange arrow going in the opposite direction. It gets closed at the stop-loss value 30 pips above the initial sell price and is therefore an unprofitable trade. The next trade is a long trade, which can be seen as the blue arrow. This is a profitable trade, and it gets cancelled out by the trailing stop-loss, which can be seen as the other orange arrow at the right-end side of the figure. The short-trade is executed right after the sma(10) (red line) crosses the sma(40) (yellow line) from the top. This is according to our strategy. Further on, we can see that the ATR(20) (blue line) is greater when the trade is executed than it was 10 hours ago, which is also according to the strategy.

After reviewing the complete dataset, we can see that the EA traded according to how it was set up. In other words, we could not observe any problems with the coding. The optimized EA gave varying results for the different currency pairs. The EUR/GBP was the only pair that resulted in a positive P&L. The results were stable until June of 2016 before the balance suddenly skyrocketed. We believe the EA managed to hit the Brexit almost perfectly which again resulted in a few very profitable trades. However, we are not sure whether this was pure luck or if the EA with the optimized settings generally manages to capture macroeconomic events in a good manner.

The other pairs performed poorly during the period. The worst performer was the USD/CHF, which ended the period approximately 30 percent down. In the first half of the period, the balance goes linearly down. This indicates that the EA did not make any profitable trades, with a few exceptions, and the positions got closed by hitting the stop loss boundaries. As we can see from figure 6, the EA made 7 profitable trades and 53 losses. If this was to be considered a good strategy, the profitable trades would have to cover the many losses. This does not seem to be the case, and the EA did not meet our expectations. On the other hand, our EA performed as it should. It takes a lot of trades before our balance reach a critical point, and if we were trading in real time, we could stop the EA and change some of the parameters and probably avoid the big loss.

After analyzing the data, we believe the EA got optimized on the wrong assumptions. By optimizing the parameters on profit from the previous year, it was not necessarily the best strategy to use on the next year. The most volatile periods for the different currency pairs happens when there is uncertainty about any macroeconomic events which may cause unpredictable price fluctuations.

There were several unusual events that occurred in the tested period that could impact the EUR/USD price, e.g. monthly released reports like the US Trade balance report, reports from the ECB (European Central Bank) or FED (Federal Reserve). The EUR includes many countries, but Germany, Spain, Italy and France stands for approximately 75% of the total GDP. This means that news announcements and statistics related to these countries will impact both the price and the volatility of the currency pair. This is not a problem for our initial strategy, because it will detect when a trend forms based on the single moving averages and the ATR, and execute trades accordingly. It does however become a problem when we optimize the way we did, because it is based on that the exact same news announcements will happen every year, which is highly unrealistic.

We believe we could have done the optimization part better by extracting a part of the data set that we knew would be as free of unusual events and other factors as possible and then optimize. This way the EA would optimize under more normal conditions. There would still arise a few problems with this method. The time period would have needed to be big enough for us to analyze the results and we would need to know that the market conditions was as normal as possible. One could argue that normal market conditions does not exist and that there will always be some factors that causes price fluctuations. One problem is that all the currency pairs correlate to some extent, so if some big news announcement happens in one currency pair this would affect all the currency pairs in different ways. Even though there is a lot of difficulties with using this method for optimizing as well, we believe this would be better as it would extract the most volatile parts of our initial data set. The EA would no longer try to optimize to win these volatile trades, but would rather try to find a strategy

that would produce stable profitable trades.

We thought a great deal about how to gain an edge from big news announcements with our strategy, but have not found any reliable way to profit from such trades. Therefore, we think the best strategy would be to turn off the EA before big news events to prevent taking on too much risk.

Even though the code in our EA did not include any mistakes, it could have been done in a more robust way. Generally, the more robust an EA is, the better. By robust, we mean its ability to adapt to different conditions in the market. All the parameters that we have implemented in this code are fixed values. This means that the EA will always do exactly what we have programmed it to do. We have learned that it is possible to let the EA take decisions on its own based on conditions in the market by implementing a code that would allow the EA to operate more robustly. An example would be our sizing algorithm that was set at a fixed and attached level to what the balance was. A better solution could be to let the EA base its sizing to the current volatility in the market. When there is more volatility in the market, our equity curve will have bigger swings as well, and essentially our trading system will become one that only depends on volatility. This means that the most profitable trades must come from the most volatile periods. A more elegant solution could be to let the EA base its sizing on the current volatility in the market, and thereby making large orders when there is low volatility and small orders when there is high volatility. This way the EA could potentially make more stable returns by cancelling out the volatility effect. Unfortunately we were not able to implement such a code in the EA.

11 Summary/Conclusion

In this paper we have showed how to build an EA and explained in detail how to prevent unreliable back-testing results in the best possible way. We have also tested the EA on three different currency pairs with varied results. The only profitable result we got was from the EUR/GBP pair. However, this pair generated a 60% return over a one year period starting 2015.11.30 ending 2016.11.30.

When testing the EA on the two other currency pairs it showed negative results as well as a disturbingly bad trading pattern. The EUR/USD pair generated a 10% loss while the USD/CHF pair generated a 30% loss. The ratio of profitable to non profitable trades was as low as 13.2%. Our back-testing could also have been done more accurately by testing on a data set with more normal market conditions, or by choosing the optimized parameters by including other criteria than to only maximize the profit.

Further on we should have made the EA more robust so that it would be able to deal with different market conditions in a better way. By having a more robust EA it would be easier to use it on several currencies, and it would be able to make profitable trades under different market conditions.

We experienced that is is difficult to make the EA produce stable and profitable results and we learned that the returns produced by the EA also depends greatly on the trader that uses it. The trader needs to decide when the EA is allowed to make trades, and when there is appropriate to make changes in the codes. Therefore, we believe that it is critical that the trader understands when there might be an edge in the market as well as when the smartest move is to stay out. In general we believe that the perfect strategy over the long term is very difficult if not even impossible to obtain, and therefore we think that the returns of the EA always will depend greatly on the choices made by the trader.

Working with this paper has been both interesting and learnful for all the team members. We have been able to learn a lot from our mistakes during this period and we think that we have become better algorithmic traders than we were before.

12 Limitations

There is no simple way of making an EA that generates money more often than it loses. Even experts that uses all their time and knowledge struggle to do this consistently. Our knowledge and insight about algorithmic trading and the financial markets, are limited, and this has affected the depth as well as the results of our research. During the data gathering process we were able to find data sets for all the relevant time periods, but it was not a completely clean data set, which have had an impact on the results of our EA. There were also other criteria we wanted to apply in our strategy, but were not able to do this because of our limited programming skills.

References

[1] Harris, L. *Trading & Exchanges. Market microstructure for Practioners. Oxford University press,* 2003.

[2] Brealey, R. Myers, S. Myers, and Marcus, A. *Fundamentals of Corporate Finance.* New York, 2014

[3] Grøtte, O. (2012). *Aksjekjøp og Daytrading (2006 ed.).* Oslo: Hegnar Media

[4] Lyons, R. K. (2001). *The Microstructure Approach to Exchange Rates. Retrieved November 2016, from the MIT Press.* Available at: `http://faculty.haas.berkeley.edu/lyons/docs/bookch1.pdf`

[5] Bloomberg. (2016, December). *Retrieved December 2016, from U.S generic Govt 10 year yield analysis.* Available at: `https://www.bloomberg.com/quote/USGG10YR:IND`

[6] Buttonwood. (2015, January). *Retrieved November 2016, from Going cuckoo for the Swiss.* Available at: `http://www.economist.com/blogs/buttonwood/2015/01/currencies`

[7] Colorado EDU. (2016). *The Foreign Exchange Market. Retrieved November 2016, from Colorado EDU.* Available at: `http://www.colorado.edu/Economics/courses/boileau/4999/sec3.PDF`

[8] Michael. (2012, October). *Retrieved November 2016, from Spread widening during news and low liquidity periods.* Available at: `https://raidenworks.com/2012/10/29/spread-widening-during-news-and-low-liquidity-periods/`

[9] Nasdaq. (2016). *Retrieved November 2016, from Advantages of Algorithmic Trading* Available at: `http://www.nasdaq.com/forex/education/advantages-of-algo-trading.aspx`

[10] Stockcharts. (2016). *Retrieved November 2016, from from Average True Range (ATR).* Available at: `http://stockcharts.com/school/doku.php`

[11] Williams-Grut, O. (2016, December). *Citi is being investigated for its role in the pound's 'flash crash'. Retrieved December 2016, from Business Insider* Available at: `http://www.businessinsider.com/ft-citi-japan-investigated-sterling-pound-fx-trading`

13 Appendix

13.1 Codes used in the EA

```
    /*
     THE ENTRY RULES:
     If Current Volatility (ATR(20)) is greater than Volatility (ATR(20)) 10 hours ago:
     Enter a long trade when SMA(10) crosses SMA(40) from bottom
     Enter a short trade when SMA(10) crosses SMA(40) from top

     THE EXIT RULES:
     Exit a trade 30 pips behind the inital entry price.
     Trailing stop of 30 pips

     EA SIZING RULE:
     1 Lot
  */

#define SIGNAL_NONE 0 // This is the setup part of our code. Here we declare the different variables used in our EA and define
#define SIGNAL_BUY   1 // the signals. Here we have defined a buy signal as a "1" and a sell signal as a "2" and so on.
#define SIGNAL_SELL  2
#define SIGNAL_CLOSEBUY 3
#define SIGNAL_CLOSESELL 4
extern int MagicNumber = 12345; // These are external variables used as the input parameters in the EA.
extern bool SignalMail = False; // Bool and int are different datatypes and the red words are what we called the variables.
extern double Lots= 1.0; // The green numbers is our default codes which has to be fixed here to change the default settings, but
extern int Slippage = 3; // they can still be modified in the backtester as we did.
extern bool UseStopLoss=True;
extern int StopLoss=30;
extern bool UseTakeProfit=False;
extern int TakeProfit=0;
extern bool UseTrailingStop=True;
extern int TrailingStop=30;
extern int sma_short= 10;
extern int sma_long = 40;
extern int atr_period= 20;
extern int atr_shift = 11;
extern bool isSizingOn=true;
extern double Risk=1;
int P=1;
int Order=SIGNAL_NONE;
int Total,Ticket,Ticket2;
double StopLossLevel,TakeProfitLevel,StopLevel;
double sma10_1,sma10_2,sma40_1,sma40_2;
double atr_current,atr_past;
bool isYenPair=false;
// This is the initialize (init) part of the code. Here we account for the brokers who use different decimal quatations.
int init()
  {

   if(Digits==5 || Digits==3 || Digits==1) P=10;else P=1;

   return(0);
  }
// This is the deinit part of our code. We have not included anything in this part.
int deinit()
  {
   return(0);
  }
// Here begins the start part of the code. This is where we implement the entry- and exit signals as well as the sizing.
int start()
  {

   Total = OrdersTotal();
   Order = SIGNAL_NONE;

/* Here we are setting up the different single moving averagers. There is a mode
      called sma in the program and it we are setting it to calculate the sma
      on closing prices.
   */
   sma10_1 = iMA(NULL,0,sma_short,0,MODE_SMA,PRICE_CLOSE,1);
   sma10_2 = iMA(NULL,0,sma_short,0,MODE_SMA,PRICE_CLOSE,2);
   sma40_1 = iMA(NULL,0,sma_long,0,MODE_SMA,PRICE_CLOSE,1);
   sma40_2 = iMA(NULL,0,sma_long,0,MODE_SMA,PRICE_CLOSE,2);
```

```
71
72// Here we are defining our ATR values.
73   atr_current=iATR(NULL,0,atr_period,1);
74   atr_past=iATR(NULL,0,atr_period,atr_shift);
75
76// Here we define the minimum stop level.
77   StopLevel=(MarketInfo(Symbol(),MODE_STOPLEVEL)+MarketInfo(Symbol(),MODE_SPREAD))*P;
78
79   if(StopLoss<StopLevel) StopLoss=StopLevel;
80   if(TakeProfit<StopLevel) TakeProfit=StopLevel;
81
82   if(isSizingOn==true)
83     {
84      Lots=Risk*0.01*AccountBalance()/(MarketInfo(Symbol(),MODE_LOTSIZE)*StopLoss*P*Point); //Point will be equal to 0.0001
85/* We tried adjusting for YEN pairs here, and the code does not show any errors, but for some reason it does not adjust the sizing
86      properly when we are backtesting the results.
87   */
88      if(isYenPair==true) Lots=Lots*100;
89
90      Lots=NormalizeDouble(Lots,2); // Here we are rounding up the lot size to the second decimal point.
91     }
92
93// Here the EA checks if the position is right.
94   bool IsTrade=False;
95
96   for(int i=0; i<Total; i++)
97     {
98      Ticket2=OrderSelect(i,SELECT_BY_POS,MODE_TRADES);
99      if(OrderType()<=OP_SELL && OrderSymbol()==Symbol() && OrderMagicNumber()==MagicNumber)
100       {
101        IsTrade=True;
102        if(OrderType()==OP_BUY)
103          {
104
105           //This is the part of the code which tells when the EA should exit a trade.
106           if(sma10_2>sma40_2 && sma40_1>=sma10_1) Order=SIGNAL_CLOSEBUY;  // Rule to EXIT a Long trade
107
108           if(Order==SIGNAL_CLOSEBUY)
109             {
110              Ticket2=OrderClose(OrderTicket(),OrderLots(),Bid,Slippage,MediumSeaGreen);
111              if(SignalMail) SendMail("[Signal Alert]","["+Symbol()+"] "+DoubleToStr(Bid,Digits)+" Close Buy");
112              IsTrade=False;
113              continue;
114             }
115           //This is the trailing stop rule for exiting a long trade
116           if(UseTrailingStop && TrailingStop>0)
117             {
118              if(Bid-OrderOpenPrice()>P*Point*TrailingStop)
119                {
120                 if(OrderStopLoss()<Bid-P*Point*TrailingStop)
121                   {
122                    Ticket2=OrderModify(OrderTicket(),OrderOpenPrice(),Bid-P*Point*TrailingStop,OrderTakeProfit(),0,MediumSeaGreen);
123                    continue;
124                   }
125                }
126             }
127           else
128
129           if(sma40_2>sma10_2 && sma10_1>=sma40_1) Order=SIGNAL_CLOSESELL;  // Rule to EXIT a Short trade
130
131           if(Order==SIGNAL_CLOSESELL)
132             {
133              Ticket2=OrderClose(OrderTicket(),OrderLots(),Ask,Slippage,DarkOrange);
134              if(SignalMail) SendMail("[Signal Alert]","["+Symbol()+"] "+DoubleToStr(Ask,Digits)+" Close Sell");
135              IsTrade=False;
136              continue;
137             }
138           //This is the Trailing stop rule for exiting a short trade
139           if(UseTrailingStop && TrailingStop>0)
140             {
```

```
            if(OrderOpenPrice()-Ask>(P*Point*TrailingStop))
              {
               if((OrderStopLoss()>(Ask+P*Point*TrailingStop))||(OrderStopLoss()==0))
                 {
                  Ticket2=OrderModify(OrderTicket(),OrderOpenPrice(),Ask+P*Point*TrailingStop,OrderTakeProfit(),0,DarkOrange);
                  continue;
                 }
              }
           }
        }
     }

/* Here is the entry rules. We enter the volatility rule first and then the sma rules. Both of them has to be true for
      the EA to execute any trades. So if our volatility rule is greater now than in the past which is set to 10 hour ago, and if
      sma 10 is equal to or greater than sma 40 a buy signal is triggered.
   */
   if(atr_current>atr_past)
     {

      if(sma40_2<=sma10_2 && sma10_1>=sma40_1) Order=SIGNAL_BUY; // Rule to ENTER a Long trade

      if(sma10_2<=sma40_2 && sma40_1>=sma10_1) Order=SIGNAL_SELL; // Rule to ENTER a Short trade

     }

//Buy signal has been triggered here, but this rule checks wheater the account has enough money on it to execute an order.
   if(Order==SIGNAL_BUY)
     {
      if(!IsTrade)
        {
         if(AccountFreeMargin()<(1000*Lots))
           {
            Print("We have no money. Free Margin = ",AccountFreeMargin());
            return(0);
           }

         if(UseStopLoss) StopLossLevel=Ask-StopLoss*Point*P; else StopLossLevel=0.0;
         if(UseTakeProfit) TakeProfitLevel=Ask+TakeProfit*Point*P; else TakeProfitLevel=0.0;
Ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,Slippage,StopLossLevel,TakeProfitLevel,"Buy(#"+MagicNumber+")",MagicNumber,0,DodgerBlue);
         if(Ticket>0)
           {
            if(OrderSelect(Ticket,SELECT_BY_TICKET,MODE_TRADES))
              {
               Print("BUY order opened : ",OrderOpenPrice());
               if(SignalMail) SendMail("[Signal Alert]","["+Symbol()+"] "+DoubleToStr(Ask,Digits)+" Open Buy");
                 } else {
               Print("Error opening BUY order : ",GetLastError());
              }
           }
         return(0);
        }
     }

//It checks the exact same thing for a sell order.
   if(Order==SIGNAL_SELL)
     {
      if(!IsTrade)
        {
         //Check free margin
         if(AccountFreeMargin()<(1000*Lots))
           {
            Print("We have no money. Free Margin = ",AccountFreeMargin());
            return(0);
           }

         if(UseStopLoss) StopLossLevel=Bid+StopLoss*Point*P; else StopLossLevel=0.0;
         if(UseTakeProfit) TakeProfitLevel=Bid-TakeProfit*Point*P; else TakeProfitLevel=0.0;

Ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,Slippage,StopLossLevel,TakeProfitLevel,"Sell(#"+MagicNumber+")",MagicNumber,0,DeepPink);
         if(Ticket>0)
           {
            if(OrderSelect(Ticket,SELECT_BY_TICKET,MODE_TRADES))
              {
               Print("SELL order opened : ",OrderOpenPrice());
               if(SignalMail) SendMail("[Signal Alert]","["+Symbol()+"] "+DoubleToStr(Bid,Digits)+" Open Sell");
                 } else {
               Print("Error opening SELL order : ",GetLastError());
              }
           }
         return(0);
        }
     }

   return(0);
  }
```

13.2 Other figures



Figure 11: *Example of buy a buy trade and the trailing stop loss.*

This figure shows a buy trade indicated by the blue arrow. The trailing stop loss is the stapled red line. The trailing stop loss is above or initial buy price, which means that this is a profitable trade for the EA.



Figure 12: *Example of a short trade and the stop loss*

The figure shows a short trade indicated by the red arrow. Here, the stop loss is above the short price, and it will get triggered if the price reaches the stop loss.

Pass	Profit	Total trades	Profit factor	Expected Payoff	Drawdown $	Drawdown %	Inputs
1	118.06	6	1.43	19.68	368.93	3.57	StopLoss=30; TrailingStop=59; sma_short=11; sma_long=44; atr_period=38; atr_shift=15; MagicNumber=12345; Signal...
2	346.02	8	5.44	43.25	372.22	3.50	StopLoss=30; TrailingStop=61; sma_short=14; sma_long=49; atr_period=30; atr_shift=10; MagicNumber=12345; Signal...
3	247.05	7	22.77	35.29	171.83	1.70	StopLoss=45; TrailingStop=39; sma_short=13; sma_long=48; atr_period=30; atr_shift=10; MagicNumber=12345; Signal...
4	209.10	8	2.23	26.14	240.58	2.36	StopLoss=40; TrailingStop=70; sma_short=19; sma_long=32; atr_period=34; atr_shift=15; MagicNumber=12345; Signal...
5	118.83	8	1.57	14.85	402.40	3.94	StopLoss=35; TrailingStop=76; sma_short=18; sma_long=33; atr_period=36; atr_shift=11; MagicNumber=12345; Signal...
6	167.31	15	1.48	11.15	408.74	3.92	StopLoss=55; TrailingStop=74; sma_short=13; sma_long=21; atr_period=28; atr_shift=5; MagicNumber=12345; Signal...
7	46.13	20	1.10	2.31	336.00	3.29	StopLoss=40; TrailingStop=32; sma_short=20; sma_long=22; atr_period=28; atr_shift=18; MagicNumber=12345; Signal...
8	111.19	13	1.11	8.55	1371.05	11.94	StopLoss=10; TrailingStop=44; sma_short=14; sma_long=24; atr_period=32; atr_shift=12; MagicNumber=12345; Signal...
9	1.69	9	1.01	0.19	201.45	1.97	StopLoss=35; TrailingStop=51; sma_short=11; sma_long=24; atr_period=38; atr_shift=21; MagicNumber=12345; Signal...
10	145.13	13	2.33	11.16	188.58	1.85	StopLoss=50; TrailingStop=68; sma_short=18; sma_long=22; atr_period=38; atr_shift=13; MagicNumber=12345; Signal...
11	67.21	5	2.31	13.44	158.98	1.56	StopLoss=55; TrailingStop=44; sma_short=20; sma_long=42; atr_period=30; atr_shift=5; MagicNumber=12345; Signal...
12	148.33	11	1.85	13.48	252.04	2.43	StopLoss=55; TrailingStop=60; sma_short=19; sma_long=30; atr_period=38; atr_shift=11; MagicNumber=12345; Signal...
13	107.17	7	2.42	15.31	229.12	2.26	StopLoss=55; TrailingStop=67; sma_short=18; sma_long=34; atr_period=26; atr_shift=6; MagicNumber=12345; Signal...
14	456.84	22	1.90	20.77	696.36	6.56	StopLoss=25; TrailingStop=57; sma_short=18; sma_long=20; atr_period=32; atr_shift=18; MagicNumber=12345; Signal...
15	216.01	7	4.53	30.86	199.24	1.95	StopLoss=45; TrailingStop=49; sma_short=15; sma_long=37; atr_period=28; atr_shift=5; MagicNumber=12345; Signal...

Figure 13: *Example of different combinations of optimized parameters*

This column to the right shows different combinations of optimized parameters. The profit column shows the profit from the optimized period. We chose the parameters that resulted in most profit over the period and used these parameters for the next period.

Bars in test	6749	Ticks modelled	12936	Modelling quality	n/a
Mismatched charts errors	0				
Initial deposit	10000.00			Spread	2
Total net profit	6040.15	Gross profit	32595.94	Gross loss	-26555.78
Profit factor	1.23	Expected payoff	25.17		
Absolute drawdown	1360.52	Maximal drawdown	4198.36 (23.90%)	Relative drawdown	23.90% (4198.36)
Total trades	240	Short positions (won %)	115 (20.00%)	Long positions (won %)	125 (28.80%)
		Profit trades (% of total)	59 (24.58%)	Loss trades (% of total)	181 (75.42%)
	Largest	profit trade	2935.68	loss trade	-214.88
	Average	profit trade	552.47	loss trade	-146.72
	Maximum	consecutive wins (profit in money)	4 (1342.06)	consecutive losses (loss in money)	14 (-2402.71)
	Maximal	consecutive profit (count of wins)	5101.05 (3)	consecutive loss (count of losses)	-2402.71 (14)
	Average	consecutive wins	2	consecutive losses	5

Figure 14: *Example of a report after the optimizing*

This figure shows an example of a report after we back-test our EA on a desired period.