

Saul Stockhus

AUTOMATED TRADING
SOFTWARE FOR FOREIGN
EXCHANGE

Bachelor's Thesis
Information Technology

May 2011



MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis May 20, 2011
Author(s) Saul Stockhus	Degree programme and option Information Technology	
Name of the bachelor's thesis Automated trading software for Foreign Exchange		
Abstract <p>The Foreign Exchange market is the largest financial market in the world. It is used by banks, governments, corporations, hedge funds and other institutions for a number of purposes, such as currency conversion, hedging and speculation. Due to advances in computing, financial data that could previously only be analyzed using charts or calculated manually can now be run through extensive analyses using special algorithms in order to predict the future.</p> <p>The aim of this study is to research trading in the Foreign Exchange market and to implement an automated trading strategy using a suitable programming language and successfully test it with real market data. Of particular interest was the prospect of discovering new trading methods and tools.</p> <p>During the course of this study, different types of trading strategies as well as new chart types were developed for the NinjaTrader platform. The strategies were tested using historic exchange rates as well as captured live rates. Choosing the type of strategy, trading platform or broker, as well as other considerations such as real life implementation is discussed. The study is useful particularly to IT and business students interested in the currency markets and automated trading in general. The study gives guidelines on strategy development and gives room for improvement in the future.</p>		
Subject headings, (keywords) Foreign Exchange, forex, price analysis, automated trading, algorithmic trading, NinjaTrader		
Pages 42 pages + appendices 2 pages	Language English	URN http://www.urn.fi/URN:NBN:fi:amk-201105208891
Remarks, notes on appendices		
Tutor Reijo Vuohelainen	Employer of the bachelor's thesis Mikkeli University of Applied Sciences	

CONTENTS

1	INTRODUCTION.....	1
1	FOREIGN EXCHANGE	3
1.1	Overview.....	3
1.2	Participants	4
1.3	Financial instruments.....	6
2	TRADING CONCEPTS	6
2.1	Trading with currency pairs.....	6
2.2	Types of brokers	11
2.3	Terminology	12
3	ALGORITHMIC TRADING.....	14
3.1	Overview.....	14
3.2	Technical Analysis.....	14
3.3	Sample strategies	17
3.4	Software	18
3.5	Communication between platform and broker	22
4	PROJECT.....	23
4.1	Project overview	23
4.2	Considerations	23
4.3	Developed code	31
4.3.1	Strategies	31
4.3.2	Bar types.....	33
4.4	Problems	35
4.5	Results.....	36
4.6	Real life implementation.....	40
5	CONCLUSION	40
	BIBLIOGRAPHY	42

APPENDICES

LIST OF FIGURES

Figure 1. Example of a currency pair chart	7
Figure 2. Market Depth window of the ECN broker MB Trading, showing bids on the left and asks on the right.....	12
Figure 3. Bar properties in NinjaTrader	15
Figure 4. Candlesticks	15
Figure 5. OHLC bar.....	16
Figure 6. Candlestick.....	16
Figure 7. Line chart - example of an up trend	16
Figure 8. MetaTrader 4 main window	19
Figure 9. MetaEditor programming environment	20
Figure 10. NinjaTrader 6.5 main window	21
Figure 11. NinjaTrader script editor	22
Figure 12. Market Analyzer window with a custom indicator showing the current spread.....	25
Figure 13. Range bar chart – a built-in bar type in NinjaTrader	27
Figure 14. Renko bar behavior	28
Figure 15. New bar type behavior	28
Figure 16. Output window used for debugging.....	31
Figure 17. Tracker bars.....	34
Figure 18. 60 minute, Tracker and Neko bars compared	35
Figure 19. Strategy #1 - Cumulated profit.....	37
Figure 20. Strategy #1, Profit/Loss Report. Green triangles indicate winning trades and red triangles losing trades.	38

LIST OF TABLES

Table 1. Foreign Exchange trading sessions	3
Table 2. Top 10 currency traders - % of overall volume, May 2010 [Euromoney]	4
Table 3. ISO 4271 codes and currency names for major currencies	7
Table 4. Lot names and their sizes	8
Table 5. Spread cost compared to bar range. The chosen spread is 3 pips.	29
Table 6. Mixed mode operation.....	32
Table 7. Account performance	38

1 INTRODUCTION

The Foreign Exchange market today is the largest financial market in the world and is an important tool for international trade. The market allows currency to be bought and sold and in the process the market determines the relative values of currencies against each other. Businesses use it to pay their workers, governments try to control the exchange rates of their currency in order to maintain the economy and investors convert currency back and forth in order to invest and reap the profits.

The Foreign Exchange did not always have the importance it has today. When the value of a currency was still tied to real commodities, such as gold and silver, there was little risk that holding a currency would affect profits. Foreign Exchange only had a minor part in finance until the late 1960's, when governments with the largest economies agreed to part with the established Bretton Woods fixed-rate system and decided to allow trading to determine currency rates, resulting in a floating exchange rate regime. Uncertainty about exchange rates arose and currency trading boomed as a result (Levinson 2005, 14 - 15). At the time, FX trading was mostly done between dealers using telephones or through voice brokers. Electronic FX trading would not take off until the 1990s, when the Electronic Broking Services (EBS), founded in 1993 by 15 banks, launched their electronic trading platform for spot FX in order to compete with Reuters (Weithers 2006, 67). Today, numerous different trading platforms exist and actively compete against each other but the biggest are still Reuters and EBS, the only difference between them being that they specialize in different currency pairs.

Due to advances in computing, financial data that could previously only be analyzed using charts or calculated manually can now be run through extensive analyses using special algorithms. Banks and hedge funds invest in the newest technology possible in order to gain an advantage over their competitors. With the rise of online trading platforms, retail traders have also been allowed to participate in the Foreign Exchange and compete against the largest financial institutions in the world through brokers and dealers.

The aim of this study is to research trading in the Foreign Exchange market and to implement an automated trading strategy using a suitable programming language and to successfully test it with real market data. Of particular interest was the prospect of discovering new trading methods and tools. In order to understand the subject, the study covers the basics of the Foreign Exchange, trading in general, predicting future prices and things to consider when im-

plementing an automated trading strategy as well as other considerations. Background material for this study has been gathered from mainly books and online sources. Information about the Foreign Exchange was quite easy to find and confirm due to the amount of sources available. As the information has found to be accurate and the subject has been extensively studied, it was possible to reduce the amount of sources in this study without compromising accuracy. This study does not aim to cover all of the topics in great detail and they only serve to help understand the practical part of the work. A large amount of information on the subjects included in this study can easily be found on the Internet, for example, if one chooses to study a particular topic in more detail.

I chose this topic as a result of my curiosity towards the financial markets, starting from the stock market and continuing with Foreign Exchange. At first I was simply interested in earning some extra income while studying. I started by learning the basics of trading, technical analysis, terminology, trading psychology and money management. I tested various trading platforms and practiced manual trading with virtual money on demo accounts. I tried many different trading methods, but eventually grew tired of the fact that every method seemed to fail after working for some period of time. This led me to the conclusion that I had to find proof that any method “worked” before trying it for myself. I had a hard time finding proof, as strategies that wouldn’t work would damage the reputation of the developer. I decided to code my own strategy so I would know its operating principle and could test it myself. In order to do this, I had to research programming, market theory, algorithmic price analysis and have learned much about the trading and the financial markets as a result. Primarily this study is aimed at students in fields of business and information technology but I believe that this study will be useful for anyone interested in trading, the financial markets or Foreign Exchange.

The organization of this study is as follows:

Chapter 1 describes the Foreign Exchange market and its structure. The market participants and financial instruments are introduced.

Chapter 2 describes basic trading concepts and terms that help understand the markets and the way algorithms work. Broker types are introduced and compared.

Chapter 3 introduces algorithmic trading and describes why and by whom it is used. A few simple methods of price analysis as well as sample strategies are demonstrated. A brief overview of platform to broker communication is given.

Chapter 4 describes the project from start to finish. The tools that are used are introduced and major decisions and problems about the implementation of the project are discussed. A few select pieces of code are demonstrated and their operating principle explained. Finally, the code is tested using previously obtained data and the results are discussed.

Chapter 5 concludes the study and explains what has been achieved and learned during the project, discusses the results and gives suggestions on how to improve the study.

1 FOREIGN EXCHANGE

1.1 Overview

The Foreign Exchange (forex, FX) is a worldwide financial market for trading currencies. Despite its name, the market is not centralized but currency is traded over-the-counter between two parties. Over-the-counter means the contract terms as well as the agreed price are not publicly disclosed. Because the Foreign Exchange lacks a central exchange, there is no single price for any given instrument. The prices are very close to one another, however, as any significant price differences would result in arbitrage opportunities. Because of its international nature, the FX market is open 24 hours a day except for weekends (Frankel [referred 10.5.11]). Geographically, the biggest centers of trade are the United Kingdom, the United States and Japan. Practically, the day is divided into three sessions, market being the most active when both the European and North American sessions overlap and the most quiet during the Asian session. (Table 1 below)

Table 1. Foreign Exchange trading sessions

Session	Market location	Hours (UTC)
Asian	Tokyo	13:00-08:00
European	London	07:00-16:00
North American	New York	12:00-20:00

The Foreign Exchange is the largest and most liquid financial market in the world, having an average daily turnover of \$4.0 trillion. The most traded currency is the US dollar, being involved in 84.9% of the daily average turnover in April 2010 (BIS = Bank for International Settlements, 12/2010). For comparison, the average daily turnover of the New York Stock

Exchange in 2009 was roughly \$70 million (World Federation of Exchanges). Other popular currencies include the Euro, the Japanese yen, the Pound sterling and the Swiss franc.

The Foreign Exchange market is used for a number of different purposes, such as currency conversion, hedging and speculation. It is difficult to exactly determine what percentage of overall trade is purely speculation, for example, as there is no central exchange. However, automated trading has increased in importance in the Foreign Exchange markets over the recent years (BIS, 12/2010). The Foreign Exchange market has also seen an increase in retail interest as it has been marketed as an easy way to riches. There have also been numerous cases of fraud committed by smaller, unregulated forex brokers.

1.2 Participants

Market participants in the Foreign Exchange include banks, central banks, corporations, governments, hedge funds and brokers serving their clients. Below, some of them are discussed in detail. This section is based on the material by Gonalo Moreira (Unit 1, Chapter 2).

Banks

The biggest players in the Foreign Exchange are the banks. A large bank may trade billions of dollars in one day. Hundreds of banks participate in forex, either to hedge for currency risks for themselves or their clients or to simply gain profits for their stockholders. It is interesting to note that almost 50% of the trading volume is due to trading done by Deutsche Bank, UBS AG, Barclays Capital and Citi (below).

Table 2. Top 10 currency traders - % of overall volume, May 2010 [Euromoney]

Rank	Name	Market share
1	Deutsche Bank	18.06%
2	UBS AG	11.30%
3	Barclays Capital	11.08%
4	Citi	7.69%
5	Royal Bank of Scotland	6.50%
6	JPMorgan	6.35%
7	HSBC	4.55%

8	Credit Suisse	4.44%
9	Goldman Sachs	4.28%
10	Morgan Stanley	2.91%

The banks conduct their trading by having a proprietary dealing desk that is responsible for executing orders, dealing and managing risks. In practice, trading takes place in the interbank market, a network of banks and financial institutions. The interbank has the lowest spreads due to very high liquidity and sometimes the spread can even reach zero. Banks also have the most amount of insider information about the market as they have access to their clients' positions. One can easily see why banks have the advantage over retail traders.

Central banks

Central banks use the Foreign Exchange to enforce monetary policies, to manage their currency reserves and to keep the fluctuation of their currency rates in check by the means of an intervention. Sometimes even the announcement of a possible intervention may cause the markets to go in the intended direction.

Corporations

International firms use the Foreign Exchange for trade, to pay workers and receive payments from clients. Finally, all of their earned profits must be converted to their home currency. They are at the mercy of the current exchange rate and thus may resort to using forex derivatives to hedge against currency risk.

Speculators

A speculator is simply anyone who plans to gain profit from forex without using it for business reasons. Speculators include banks, hedge funds, institutions and retail clients.

Investors

Businesses may wish to own property, companies or other assets in other countries. Investors deal with bonds, shares and other types of securities offered in foreign countries. All of these activities require obtaining the currency of the foreign country. (Levinson, 18)

1.3 Financial instruments

Money exchanges hands in the Foreign Exchange by the means of different financial instruments. These instruments are traded in different markets such as the spot, forward, swap, future and option markets. In this study, the spot market is of particular interest, as it is the one the project will be using. In the forex spot, or cash market, money is exchanged between two parties and has a two-day delivery date. This is the market that is used when exchanging currency at a bank before going abroad, for example. Most of retail forex trading is done using the spot market via online trading platforms. The price determined by the spot market is called the spot price (Levinson, 15).

2 TRADING CONCEPTS

2.1 Trading with currency pairs

Currencies in spot FX are traded against each other in practice by using currency pairs. A currency pair is usually denoted as XXXYYY or XXX/YYY, where XXX is the base currency and YYY the counter-currency. For example, the most popular currency pair is EUR/USD (Figure 1). Currencies in this notation follow the ISO 4217 standard and are always three letters long. A *quote* is a currency pair followed by an exchange rate, for example EUR/USD 1.3505. In this case, the quote means that one euro is worth 1.3505 US dollars. The inverse of the pair, USD/EUR, can be derived from this quote by calculating

$$\frac{1}{\left(\frac{EUR}{USD}\right)} = \frac{1}{1.3505} \approx 0.7404.$$

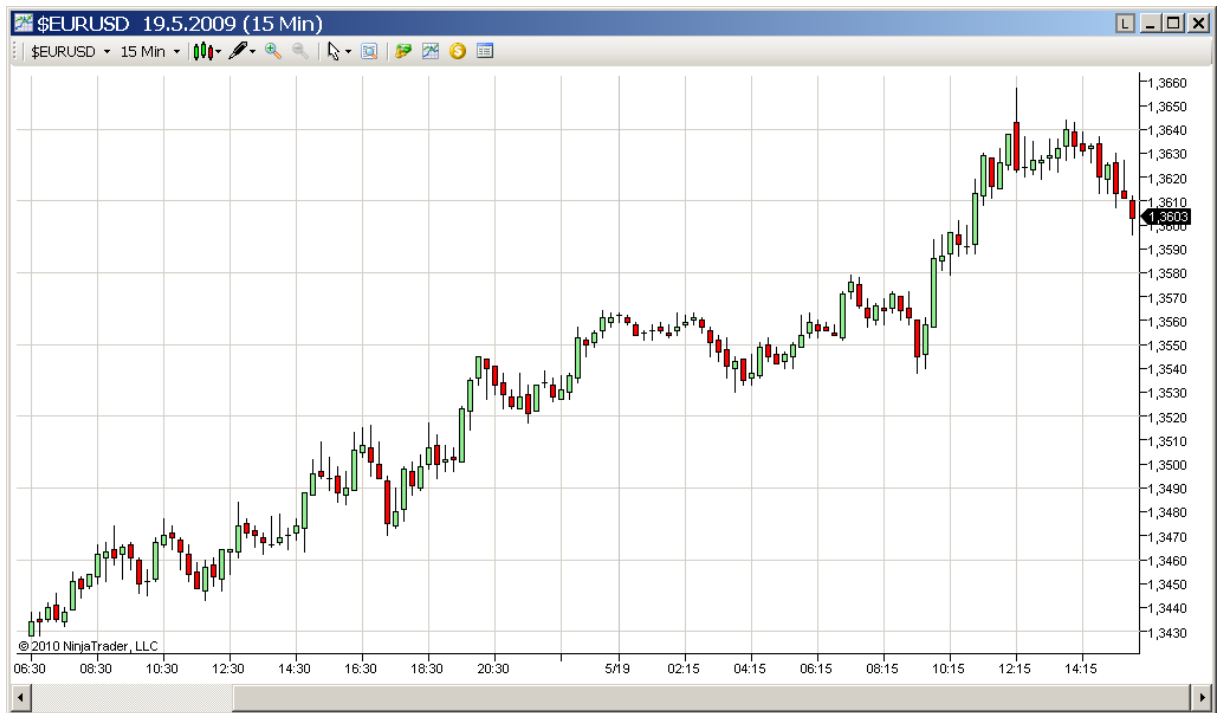


Figure 1. Example of a currency pair chart

The most traded currency pairs called the majors are AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF and USD/JPY. Currency pairs which do not involve the US dollar are called cross-currency pairs or “crosses”. These pairs can be traded by using the major pairs. For example, going long on EUR/AUD would involve going long EUR/USD and short AUD/USD. Because trading these pairs involves two trades, the spreads of cross pairs tend to be larger. Due to currency pairs being simply ratios, the exchange rate of any cross pair can be calculated using the majors. (Weithers, 84 - 87)

Table 3. ISO 4271 codes and currency names for major currencies

Alphabetic Code	Currency
EUR	Euro
USD	United States dollar
AUD	Australian dollar
NZD	New Zealand dollar
GBP	Pound sterling
JPY	Japanese yen
CHF	Swiss franc

CAD	Canadian dollar
-----	-----------------

The smallest increment the quote of a currency pair can move is called the *pip* (percentage in points). For example, a quote change of EUR/USD from 1.3000 to 1.3001 is a one pip move. Different currency pairs may have different pip values. For example, the USD/JPY pair has a pip value of 0.01 instead of 0.0001. In recent years, some platforms have increased quote precision to 1/10th of a pip. In trading terms, 1/10th of a pip is sometimes called a *sub-pip*.

When considering trading, a quote actually includes two prices, *bid* and *ask*. The bid is the highest price buyers are willing to pay for the instrument. Conversely, ask is the lowest price sellers are willing to accept for the instrument. The difference between the two is called the *spread* (Weithers, 89). In the FX market, the spread is usually measured in pips. If one were to buy a currency pair and then instantly close it, one would instantly take a pip loss equal to the spread. This is called *spread cost* and can be considered the cost of trading. Spread in the FX market is never fixed and constantly changes depending on supply and demand. For example, during large news events such as the NFP (Non-Farm Payroll), the spread can significantly increase due to the high volume, as market makers try to decrease their risk in the markets.

Currency pairs are either bought or sold in *lots*, a measure of the amount of base currency one is either buying or selling. One lot is worth 100 000 units of base currency. Market makers have later coined terms for lot sizes less than 100 000 units (Table 4 below).

Table 4. Lot names and their sizes

Name	Lot Size (in units)
Lot	100 000
Mini lot	10 000
Micro lot	1 000
Nano lot	100

Order types

In practice, currency pairs are either bought or sold by using orders sent to the broker. Many types of orders exist and are useful in different situations. A few of them are listed below.

A *market order* is an order for immediate execution, regardless of price. Market orders are susceptible to slippage. Market orders are often used when getting filled is more important than the price. Market orders are said take liquidity.

A *limit order* is an order which is only executed at the specified price or better. There is no guarantee of execution. Limit orders are often used when price is more important than getting filled. Limit orders stay on the broker's server or ECN. Limit orders are said to provide liquidity.

A *take profit* (TP) is special order that resides on the broker's server. When the price reaches the desired profit target, the TP closes the trade. In an ECN, the TP could also be implemented as a limit order.

A *stop loss* (SL) is special order that resides on the broker's server. It closes the trade and takes a loss using a market order when the price reaches the stop loss level.

Money Management

When trading, it is not enough to know whether to buy or sell; it is important to consider how much to buy or sell. In trading terms, this is called money management. The position size used in a trade is determined by the trader's risk appetite. For example, conservative traders may only risk 1-2% of their account per trade. By not risking much in a single trade, the overall edge of the strategy used by the trader can work to his/her advantage. How much should a trader risk per trade? This is a difficult question and an answer, provided by probability theory, the Kelly criterion is discussed next.

The Kelly criterion is a formula used to determine the optimum bet size for a series of bets in order to get optimal growth. First used by gamblers, it has later been adapted to financial markets. The formula is as follows:

$$f^* = \frac{bp - q}{b}$$

The formula gives the optimum percentage of the account to bet as a fraction as the result. Variable b is the net odds received on the bet, p the probability of winning and q the probability of losing, which is $1 - p$.

An example:

The analysis of trading strategy shows that the strategy wins 55% of the time, with the win/loss ratio being 1.10. The variable b would be 1.10, p would be 0.55 and q would be $(1 - p) = 0.45$. By inserting into the formula, one gets:

$$\frac{(1.10 * 0.55 - 0.45)}{1.10} = 0.14$$

In this case, the Kelly formula suggests betting 14% of the account size on each trade. In practice, this is a huge amount of risk. The Kelly formula only considers the long-term profits and does not take into account how one gets there, including draw downs and volatility. Temporary losses could severely damage one's account: Only three consecutive losses of 14% results in an account size of 63.6% of the original. Another problem in the Kelly formula is that odds in the real market are never static and market conditions change, and thus it is impossible to accurately predict the future performance of a strategy. Due to these concerns, most people do not use the Kelly formula as it is but divide the bet size by two, for example. (The Kelly Criterion, Part I, Basic Kelly Math [referred 19.05.2011])

It is important to note, that money management alone cannot provide an edge in the markets. Some money management techniques, such as martingale, try to achieve that but will eventually result in a huge loss. Martingale is actually more of a betting strategy than a money management strategy. It simply involves increasing the position size of a losing trade, hoping that the price will eventually get back to where it was. In the end, it is up to the trader to decide how much one wants to risk per trade.

Trade example

EUR/USD is currently at 1.3000/1.3002. A trader decides to go long as the trend seems to be upwards. He sends a buy market order for 1 lot, experiences no slippage and thus buys 100 000€ and sells \$X, where X is the amount of dollars worth 100 000€, found by calculating $1.3002 * 100\ 000€ = \$130\ 020$. EUR/USD goes up to 1.3048/1.3050. The trader decides

to take the profit and close the trade. He closes the trade with a sell market order, and thus sells 100 000€ and buys \$X, where $X = 1.3048 * \$100\,000 = \$130\,048$. The profit he made was $\$130\,048 - \$130\,020 = \$28$.

2.2 Types of brokers

It is not possible for a retail trader to participate in the FX market without a broker, as banks require bank accounts with six-digit numbers before they will consider anyone to trade with them. Also, the minimum trade size in the interbank is 10 lots or 1M units, which to a retail trader is massive. There are two types of brokers for retail forex, market makers and ECNs (Electronic Communications Network). This section is based on material by Grace Cheng.

Market makers get their name from the fact that they provide both bid and ask prices at the same time and thus create a market. Market makers are the counter-party in each trade and thus they are in conflict of interest with the trader. If a trader is buying a currency pair, the market maker is the one selling it to him. Market makers charge for their services by giving the trader a worse spread than in the real market. The advantage for the trader is that the spreads remain mostly fixed. They also might use tricks such as re-quotes, slippage and delays in order to gain an advantage over the trader. Some have even resorted to canceling profitable trades. Trading using market makers during important news events is problematic and sometimes platforms may even freeze during these events, as the amount of trading activity surges. The systems used to filter quotes and to execute orders simply cannot keep up.

ECN-based brokers use an electronic exchange to facilitate trading. The ECN is connected to banks, market makers and other liquidity providers, as well as other traders and passes their prices to an electronic exchange. Customers then place orders on the exchange and the matching engine of the broker matches and executes the trades. The broker itself isn't involved in the trade and thus is not in conflict of interest with the trader. The payment for the broker's service is not included in the spread and thus the broker charges for each trade using a commission, its size depending on the volume of the customer's order. ECNs have an advantage over market makers over transparency, as everyone can see everyone else's orders and their sizes (Figure 2).

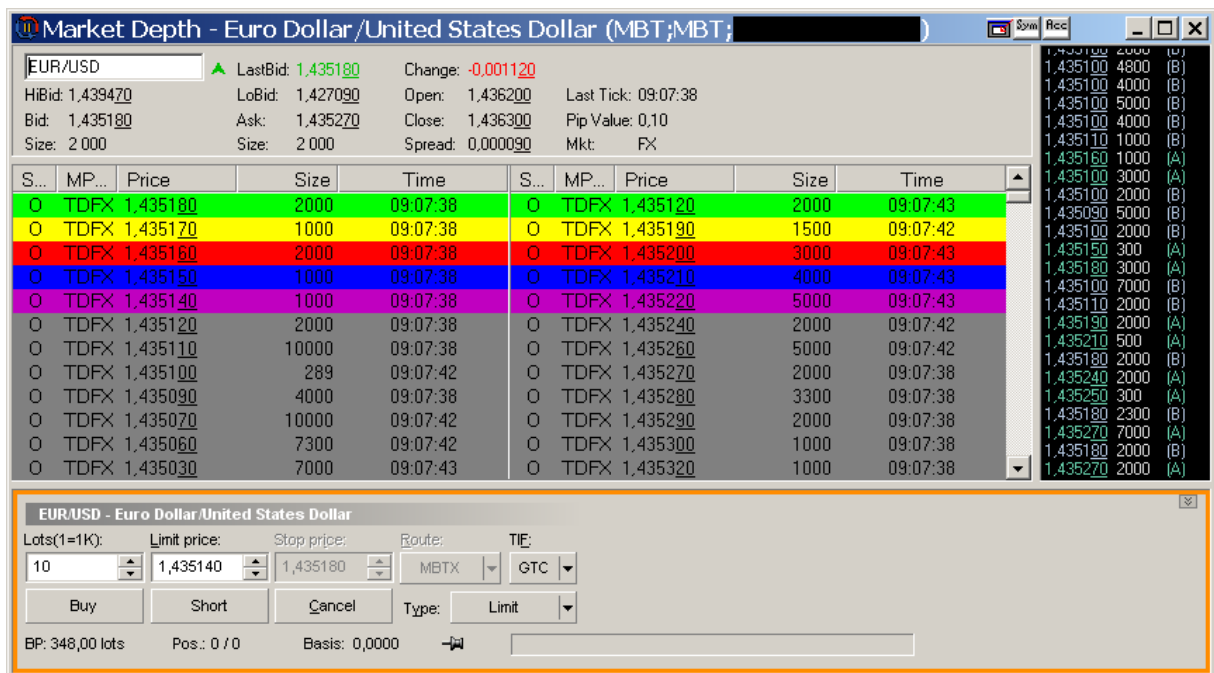


Figure 2. Market Depth window of the ECN broker MB Trading, showing bids on the left and asks on the right

2.3 Terminology

Backtesting – testing a trading strategy on prior time periods in order to test its effectiveness. Without the ability to backtest, one would have to test a trading strategy in real time, taking days, weeks or even months, depending on the strategy.

Fundamental analysis – a discipline of analyzing the market and making trading decisions based on fundamental market data, such as earnings in the case of stocks or interest rates, in the case of forex.

Leverage – allows a trader to trade with more money than he has. Leverage is usually denoted as a ratio, such as 100:1, in this case meaning that one can trade with 100 times the amount one has available for trading.

Liquidity – a measure of the ability to either buy or sell the financial instrument at any given time.

Long – buying the base currency and selling the counter-currency

Lot – the standard amount of base currency the instrument is traded with. One lot is equal to 100 000 units.

Range – a market condition where the price seems to move sideways and does not have an underlying trend. In trading terms, the price can also be said to be *ranging*.

Short – selling the base currency and buying the counter-currency

Slippage – the difference between the estimated transaction cost one expects and the actual transaction cost one gets. Example: a buyer sees a market quote of EUR/USD at 1.3000 bid, 1.3001 ask and decides to send a buy market order to the broker. It takes some amount of time for the order to reach the broker's server. While the order is being processed by the broker, the market has moved to 1.3002/1.3003. The order finally gets filled. Thus, the buyer gets a price of 1.3003 for the order and experienced a slippage of 2 pips. Slippage might also happen due to the size of the order, as there might not be enough liquidity to fill the entire order at the price the customer wanted.

Technical analysis – a discipline of analyzing the market using price, volume and other information. The underlying principle of technical analysis is that the price reflects all relevant information and due to this, it is not necessary to look at fundamental data. Technical analysis believes that traders tend to produce patterns, such as trends or concepts such as support & resistance that can be used to predict the future price.

Tick – a quote change

Trend – a concept of technical analysis, the tendency of the price to move in one direction and resist returning to previous levels. A price reaching higher highs and higher lows (highs and lows being points where the price seems to turn) could be said to be trending upwards.

Volume – the amount of trading activity in a certain amount of time. As there is no central exchange in forex, volume is difficult to determine. One alternative is to use the amount of ticks in a certain time period to analyze and compare trading activity between different time periods.

3 ALGORITHMIC TRADING

3.1 Overview

Algorithmic trading consists of using a software program to make trades on a market using some special algorithm. After analyzing price the algorithm makes decisions about the trade: direction, timing, quantity, order type etc. Algorithmic trading is used by banks and hedge funds, for example. Due to free market platforms such as MetaTrader and brokers with low capital requirements, almost anyone can participate in algorithmic trading nowadays. Below is a list of some pros and cons of algorithmic trading.

Advantages

- no human errors
- no subjectivity
- based on statistics
- takes little time once operational

Disadvantages

- requires good programming skills
- requires accurate historic data
- requires analysis

3.2 Technical Analysis

Technical analysis is simply the study of past prices in order to predict future prices using tools such as charts and indicators. Technical analysis started from Dow Theory developed by Charles Dow in the beginning of the 20th century. Dow Theory included such trading concepts as trends, divergence and support/resistance, for example.

Numerous approaches to analyzing price in order to forecast it have been developed. The most common method used in technical analysis is to divide the price movements into certain slices of time, called bars.

Bars

A bar is a visual representation of a block of time or price movement. Bars usually contain information such as open, close, high and low, volume and time (Figure 3). Different types of bars exist and are used by chartists, such as Japanese candlesticks (Figure 4), OHLC bars (Figure 5), Kagi, Point & Figure, Renko etc.

\$EURUSD	
\$EURUSD / 60 Min	
Date	21.5.2009
Time	16:00:00
Price	1,3556
Open	1,3758
High	1,3779
Low	1,3736
Close	1,3743
Volume	1073

Figure 3. Bar properties in NinjaTrader

Bar information

Open – The price at the time the bar was created

Close – The price at the time the bar was last updated

High – The highest price the bar reached

Low – The lowest price the bar reached

Volume – (see 2.3 → Volume)

Time – The time the bar either opened or closed, depending on the platform

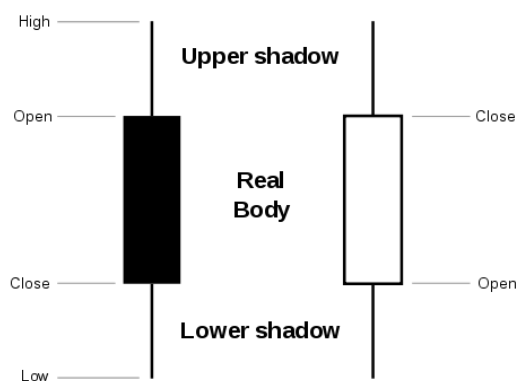


Figure 4. Candlesticks

Candlesticks (Figure 4, above) are normally drawn with the real body either filled-in or clear. If the open is higher than the close, the real body is filled-in; if not, the real body is kept clear. Candles can also be colored, e.g. green and red.



Figure 5. OHLC bar

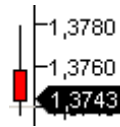


Figure 6. Candlestick

Charts

Charts are simply a group of bars. Many types of charts have been developed over the years. Maybe the most simplistic chart type is the *line chart* (Figure 7 below). It only uses one data point, the close for each bar (although the bar itself is not visible).



Figure 7. Line chart - example of an up trend

A chart helps to see what is happening with the price. In the above figure, the chart seems to indicate that the price is trending upwards, as the price is making higher highs and higher lows.

Indicators

In order to provide accurate buy/sell signals, visual software tools called indicators have been developed and are used in technical analysis. The most basic indicator used in financial applications could be the Simple Moving Average (SMA), simply being the mean of the previous X data points, usually bars. Different types of indicators exist and are used for measuring trends, momentum, volume, volatility and other aspects of price. Indicators are normally used by traders to assist in making trading decisions, but indicators can also be used in automated trading strategies.

3.3 Sample strategies

Trend

Trend trading tries to take advantage of an instrument that is trending; that is, generally moving in one direction. The presumption is that the price will continue to move in that direction. The problem with this type of strategy is that there is always a trend in any timeframe or price scale and thus it may be difficult to decide which trend to follow.

Counter-trend

Counter-trend trading is the opposite of trend trading. It takes advantage of an instrument that seems to be ranging and is more likely to revert to a mean, for example a moving average.

Carry

The carry trade takes advantage of interest rate differences between currencies. A currency with a low interest rate is sold and a currency with a high interest rate is bought at the same time. As the carry trade takes a long time to become profitable, the outcome of the trade may depend entirely on whether the currency pair in question appreciates or depreciates. As such, carry trades work the best with currency pairs that do not move much.

Basket

Basket trading simply means using multiple instruments at the same time. For example, an algorithm might predict that the US dollar will strengthen against other currencies and decide to go long on every pair that has the US dollar as a base currency and vice versa.

Seasonal

A seasonal trading strategy takes advantage of a phenomenon called the *calendar effect*. The strategy might look for tendencies during a certain weekday or a certain month. (Chan, 143) This strategy is more commonly used in stock or commodity trading but can be adapted to forex. For example, the strategy could look at each of the sessions in the forex 24-hour cycle and try to find patterns of behavior in each of them.

3.4 Software

Numerous software platforms for algorithmic trading exist. A couple of them, the ones later referred to in this study will be introduced. In addition to platforms designed solely for trading, software packages such as R and MATLAB can be used for algorithm development as well as actual execution of orders.

MetaTrader

Developer: MetaQuotes Software Corp.

Home page: <http://www.metaquotes.net>

Platform: Microsoft Windows

MetaTrader supports forex, CFD and futures markets. The software is a package of both server and client components. The software is licensed to brokers who then provide it to their clients. The MetaTrader Client Terminal (Figure 8), used for technical analysis, trading and algorithmic trading development is free.

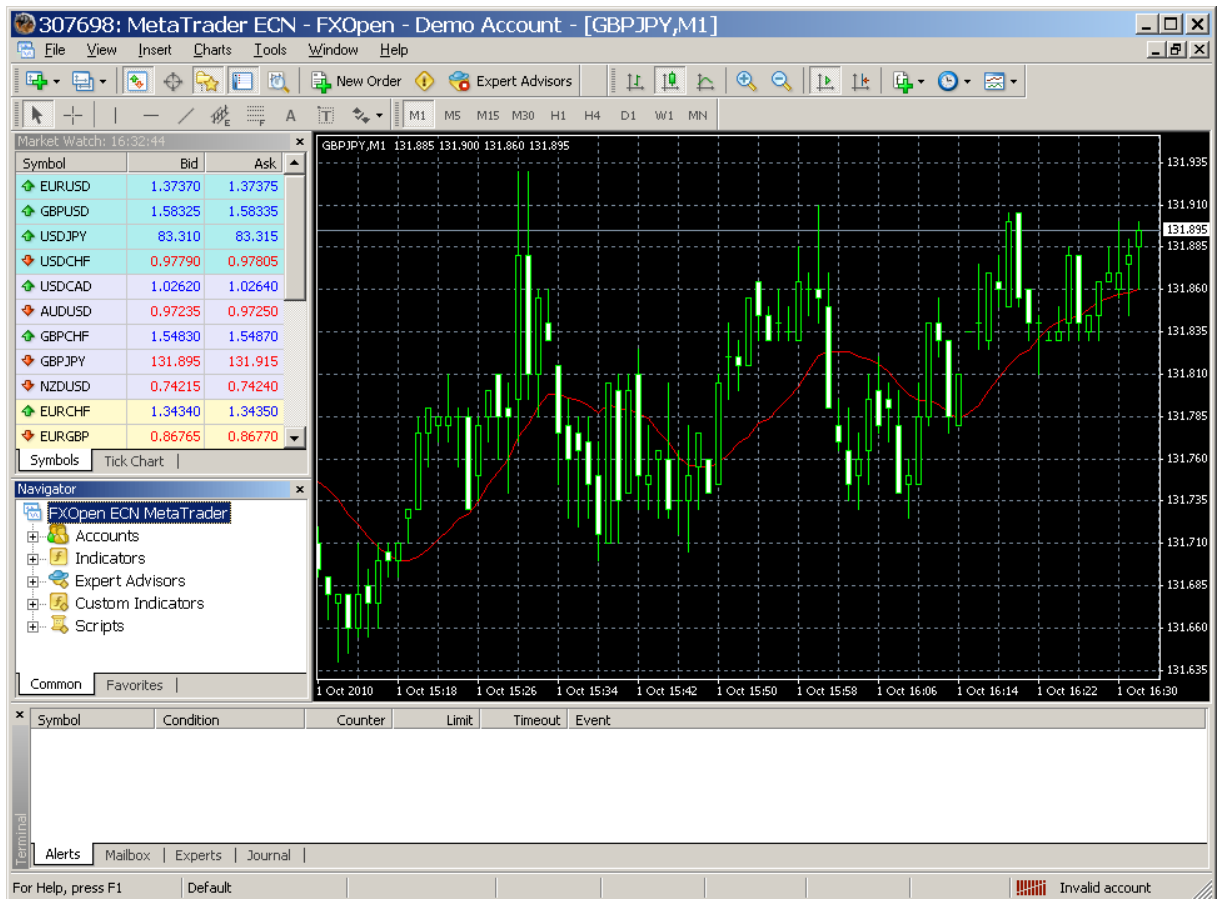


Figure 8. MetaTrader 4 main window

Features

- Automated trading using Expert Advisors
- Development
 - MQL 4 programming language, similar to C
 - MetaEditor programming environment (Figure 9)
 - Possible to develop Expert Advisors, scripts and indicators
- Analytical tools
 - Nine timeframes
 - More than 50 built-in indicators
 - Large database of custom indicators and tools
- Strategy tester
 - Tick-by-tick modeling
 - Expert Advisor optimization

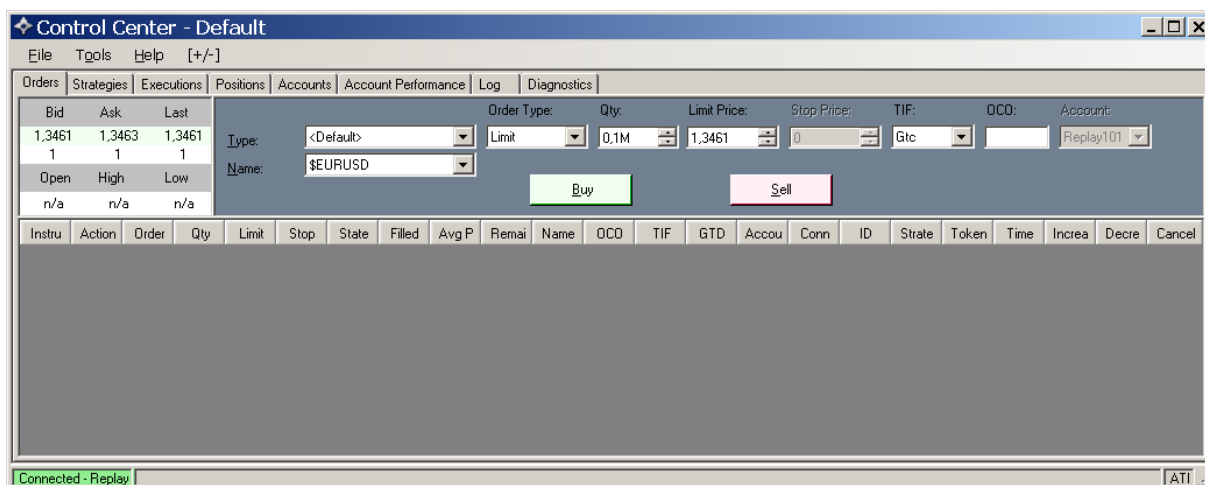
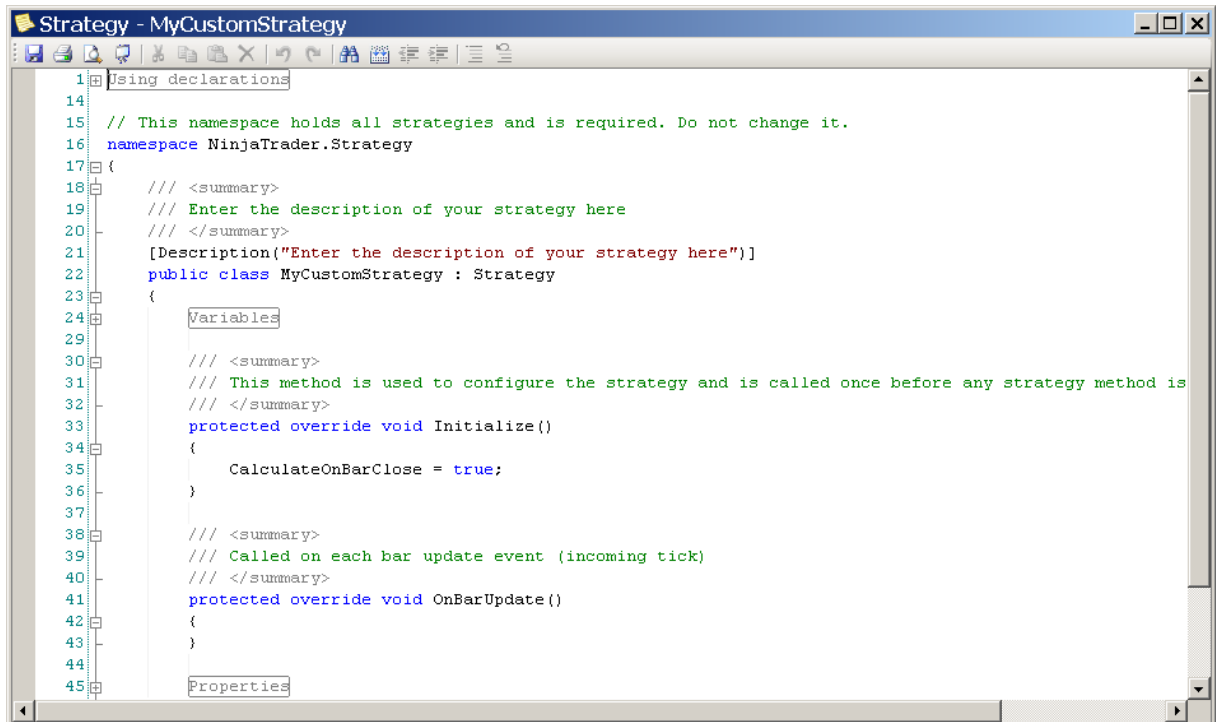


Figure 10. NinjaTrader 6.5 main window

Features

- Strategy analyzer
 - High performance backtesting
 - Walk forward and genetic optimization
 - Basket testing
- Trade performance
 - Over 80 individual metrics
 - Multiple performance graphs
- Development
 - Popular non-proprietary C# programming language
 - Event driven model
 - Multi-time frame and multi-instrument
 - Modern script editor with inline syntax checking and Intellisense (Figure 11)
 - Always runs compiled code not interpreted for the highest possible performance
- Live Real Time Trade Simulator
 - Full order and account simulation
 - Realistic order fill algorithm
- Market Replay
 - True tick by tick replay at user defined speeds
 - Replay multiple markets simultaneously
- Custom development
 - Develop your own indicators

- Develop your own automated strategies
- Support for advanced programming



```

1  Using declarations
14
15  // This namespace holds all strategies and is required. Do not change it.
16  namespace NinjaTrader.Strategy
17  {
18      /// <summary>
19      /// Enter the description of your strategy here
20      /// </summary>
21      [Description("Enter the description of your strategy here")]
22      public class MyCustomStrategy : Strategy
23      {
24          Variables
29
30          /// <summary>
31          /// This method is used to configure the strategy and is called once before any strategy method is
32          /// </summary>
33          protected override void Initialize()
34          {
35              CalculateOnBarClose = true;
36          }
37
38          /// <summary>
39          /// Called on each bar update event (incoming tick)
40          /// </summary>
41          protected override void OnBarUpdate()
42          {
43          }
44
45          Properties

```

Figure 11. NinjaTrader script editor

3.5 Communication between platform and broker

Trading platforms have to communicate to the broker in order to execute trades. Brokers usually have their own API (Application Programming Interface) that programmers must implement in the trading platforms in order to trade with that broker. It is interesting to note that such APIs can also be obtained by traders. Some brokers, however, may charge a fee for it or require a certain amount of trading volume in a month. Due to the fact that each broker may have their own API, it is time-consuming to create a new trading platform from scratch. In order to help with this problem, a general protocol, called FIX was developed.

According to the FIX organization, FIX is a “series of messaging specifications for the electronic communication of trade-related messages”. It was developed together with banks, brokers, exchanges, industry utilities and associations, institutional investors and information technology providers. The FIX protocol is a free and open specification. It is used by “virtually every major stock exchange and investment bank”. (FIX Organization [referred

19.05.2011]) An open-source cross-platform implementation of FIX has been released, called QuickFIX (Appendix 1).

4 PROJECT

4.1 Project overview

The aim of the project was to implement an automated trading strategy using a suitable automated trading platform. In the process, different subjects were researched and various problems were faced. Below is a list of things one had to keep in mind during the project.

Things to consider

- Which strategies to implement?
- Which types of orders to use?
- When to trade?
 - Market activity
- Which instruments?
 - Majors vs. crosses
- How often / how many trades?
- How much to bet on each trade? (money management)
- How costly?
 - Spread costs
- How to test?
 - Where to find market data?
- What are the risks involved?
- How to analyze results?
 - Does the strategy have an edge?

First, however, a trading platform had to be chosen.

4.2 Considerations

Choosing a platform

A number of things were considered when searching for the platform. Below is a list of the most important ones.

The platform

- should be affordable to a retail customer
- should support a suitable broker
- should be free for testing purposes
- should be able to mimic real market conditions well
- should be easy to learn and code with

Many different platforms were considered in the course of this study but a closer look will be taken at only the final two candidates: MetaTrader and NinjaTrader. At the time, MetaTrader was at version 4 and NinjaTrader at version 6.5. Newer versions of these two, NinjaTrader 7 and MetaTrader 5 have been later released.

MetaTrader is probably the most popular algorithmic trading platform for retail users as it is completely free. It supports dozens of brokers. Due to its large user base it has a large library of custom code written for it. However, when considering it as a platform choice, there was a worry that it would not be able to mimic real market conditions as it lacks the type of testing which takes the spread into account. Another thing to note about MetaTrader is that the brokers it supports and partners with are mostly market makers.

NinjaTrader, compared to MetaTrader, supports real-time testing with real market spreads. It also allows one to record market data live and later replay it with various speeds, the maximum being 500 times real-time. The interface seemed more intuitive than MetaTrader. NinjaTrader seemed to consider novice coders as well and had a strategy wizard to get started with. Unfortunately, NinjaTrader requires a license for live trading but as strategy development and testing is completely free, this was not deemed an issue.

In the end, NinjaTrader was chosen as the platform, mainly due to its ability to simulate real market conditions.

Choosing a broker

Choosing the broker in this project was very much tied to the choice of platform. NinjaTrader only has a couple of forex brokers to choose from, but one of them, MB Trading seemed interesting as it is an ECN-based broker. It also had the lowest minimum account deposit as well as lot size (1000 units), suitable for beginning traders and for testing strategies in real life

without significant risks. Due to the advantages of an ECN, such as transparency and non-marked up prices, MB Trading was chosen as the broker.

Retrieving test data for market replay

As the most interesting aspect of NinjaTrader was its real-time testing based on real market conditions, the option of recording real market quotes and replaying them was explored. Market quotes from both MB Trading and GAIN were recorded. MB Trading required a demo account to be created, GAIN did not. Recording the market quotes was done as follows.

First, market replay recorder was enabled:

Tools→Options→Data, setting “Run market replay recorder” was checked on.

When recording market quotes from MB Trading, a setting in NinjaTrader had to be changed in order to accommodate 1/10th pip accuracy:

“Quote currencies (FX) in” in Tools→Options→Misc was set to “TenthPip”.

A Market Analyzer window (Figure 12) was opened and the currency pairs were added by using an instrument list. A column with an indicator using the price was added, otherwise NinjaTrader would not record anything. Then it was simply the case of connecting to the broker as the forex market opened on Monday and letting it run until the weekend.



Instrument	AskPrice	BidPrice	LastPrice	SpreadDisplay
\$AUDCAD	0,8965	0,8956	0,8956	0,0009
\$AUDJPY	74,45	74,40	74,40	0,05
\$AUDNZD	1,2884	1,2864	1,2864	0,002
\$AUDUSD	0,7755	0,7750	0,7750	0,0005
\$CADCHF	0,9597	0,9590	0,9590	0,0007
\$CADJPY	83,09	83,03	83,03	0,06
\$CHFJPY	86,61	86,57	86,57	0,04
\$EURAUD	1,7598	1,7586	1,7586	0,0012
\$EURCAD	1,5768	1,5758	1,5758	0,001
\$EURCHF	1,5124	1,5119	1,5119	0,0005
\$EURGBP	0,8809	0,8806	0,8806	0,0003

Figure 12. Market Analyzer window with a custom indicator showing the current spread

Retrieving test data for backtesting

MB Trading didn't have any historic currency rate data for download but another broker, GAIN did. The data was freely available at <http://ratedata.gaincapital.com/> as zipped .csv files. It was organized in directories in the following manner:

Year\Month\[currency_pair]_Week#.zip

For example: 2010\05 May\EUR_USD_Week1.zip

CSV (comma-separated value) files are simply text files with .csv file extension, having some text-based values separated by a comma. It is useful as a method to store information in a very simple format and can be used to import and export data between applications.

Example of GAIN .csv format:

```
674344297,EUR/USD,2008-12-28 17:00:03.000,1.405700,1.407100,D
```

The same example in NinjaTrader 6.5 .txt format:

```
20081228 170003;1.405700;1
```

As can be seen above, GAIN data includes ask as well as bid, but NinjaTrader 6.5 only supports bid for historical data. Downloading each of these files manually was decided not to be an option, as it would take a very long time. Therefore, a command-line batch file for Windows using the software *Wget* (Appendix 1) was created to download the files automatically. All the files from years 2006 to 2009 were downloaded, taking roughly 5 GB of space. Batch files were then created to unzip these files and run third-party software called *NTDataConverter* (Appendix 1) and *Simple Search & Replace* (Appendix 1) to convert the .csv files into a format NinjaTrader could import. Some errors, such as misnamed files were found in the .zip files and were corrected manually. It was also noted that the data from May 2009 onwards seemed to use a different format. It was decided not to convert the data in question.

Choice of bar types

In the project's first strategy prototypes, time-based bars were used to make trading decisions. This posed certain problems, however. There is no way to know beforehand how much the price will move in a certain period of time. This makes controlling one's risk and trying to predict price difficult. Therefore an alternative was considered: using only the price. It was

found that NinjaTrader had a built-in bar type, “Range” (Figure 13), which only uses the price. It has one parameter, the range (the “height”) a bar has to reach before a new bar is started. In the case of forex, this is measured in pips.



Figure 13. Range bar chart – a built-in bar type in NinjaTrader

Another bar type seemed interesting, called “Renko”. (Figure 14) As with range bars, renko bars only use the price. The way the bars are generated is asymmetrical. An example scenario: Renko bars of 20 pips are used. Price starts at 0 pips. Price goes up to +20 pips. An “up bar” is drawn. Price goes up to +40 pips. A second “up bar” is drawn. The price goes back to +20 pips. One would expect a bar to be drawn, but this is not the case. Finally, price goes back to +-0 pips and a “down bar” is finally drawn. During the course of this project, I developed a new bar type (Figure 15) based on the concept of renko. It gets rid of the asymmetric behavior.

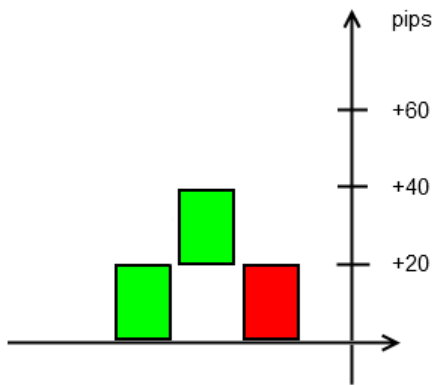


Figure 14. Renko bar behavior

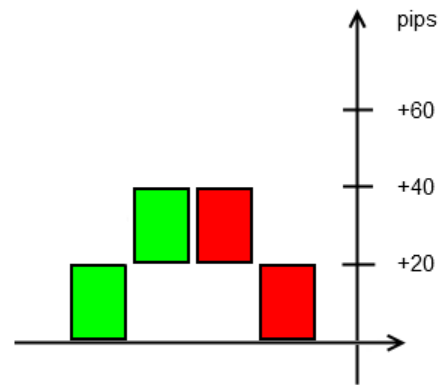


Figure 15. New bar type behavior

Choice of order type

In the beginning, only market orders were tested. It became clear that using them caused the profit ratio to worsen considerably due to spread cost in lower timeframes. After modifying the code to use limit orders, there was a noticeable improvement in profit factor when dealing with low bar ranges. However, using limit orders causes the amount of trades to decrease, as orders are not guaranteed to be filled.

A practical problem arose due to limit order behavior. Keeping track of positions when using market orders was easy because as soon as the order was sent, one could be sure that the order would be filled. Sending a limit order, however, did not guarantee a fill and therefore a different approach to tracking positions had to be taken. A NinjaTrader method for experienced programmers called `OnExecution()` was found. It is called each time an order is filled and can be overridden. The method was modified to look for the position variable of the order that was filled and modify it accordingly, solving the problem (Appendix 2.1).

Another problem with limit orders is that big events, such as news releases happen in forex. On occasions, prices have moved several hundred pips without stopping. If caught on the wrong side of such a move, using limit orders would not have guaranteed a way to exit the trade.

If one chooses to use higher time frames or bar ranges and thus make fewer trades, the difference between the two orders becomes minimal and using market orders can be justified due to the lower cost. (Table 5)

Table 5. Spread cost compared to bar range. The chosen spread is 3 pips.

Bar range (pips)	Spread cost (%)
8	37.50%
16	18.75%
32	9.38%
64	4.68%
128	2.34%

The strategies in this project were modified to be able to use both market and limit orders, based on user preference. Market orders could be used for trend trading and limit orders for counter-trend trading, for example.

Types of strategies

The most important strategy types to implement in the trading strategy seemed to be trend and counter-trend as these were the conditions the market was most likely to be at any time. Another strategy that seemed interesting was basket trading. Other types of strategies based on market fundamentals such as interest rates were deemed too difficult to implement.

When to trade?

Because the strategies would only use the price, the question of when to trade became somewhat irrelevant as the strategies would simply not care about the amount of volatility. However, market activity does affect the spreads. Therefore, a spread check was implemented in the strategies: if the spread was too high for the bar range, the bar range would simply not be used.

Money management

Different types of money management were tested and as they seemed not to provide an edge, it was decided to simply use the same unit (lot) size for each trade. This later lead to an idea: instead of trading separately for each bar range, practically running separate strategies, all of these strategies could be joined together, resulting in a single decision on whether to go long,

short or stay out. What one would then be betting is that all of the bar ranges combined would be profitable.

Position tracking

By default, NinjaTrader keeps track of orders using a convenience layer, leaving the technical aspect of order management hidden. This became a problem when trying to use multiple bar ranges in a single currency pair. The strategies would have to know whether each bar range was long, short or had no position and send separate orders for each range. A solution was found by using IOrder objects that contain information about an order. In addition, an integer array for positions in each bar range was created. For example, if the position of the bar range is short 1000 units, the appropriate array slot would have -1000. Zero indicates no position.

Below is an excerpt from the code. A long limit order is sent to the broker and the result is stored into an IOrder object. entryOrders are of the IOrder object type.

```
entryOrder[u,t]=EnterLongLimit(i, true, quantity[u,t], GetCurrentAsk()-benefitb,
u.ToString()+", "+t.ToString());
```

IOrders, as well as most variables in the code are arranged in two-dimensional arrays, first index (u) being the currency pair and the second index (t) being the bar range.

Debugging

A way to output information was found by using the output window of NinjaTrader (Figure 16), resembling a console. The way most debugging was done during the project was to indicate certain key points in the code with Print(), helping to see where the code didn't behave as expected.

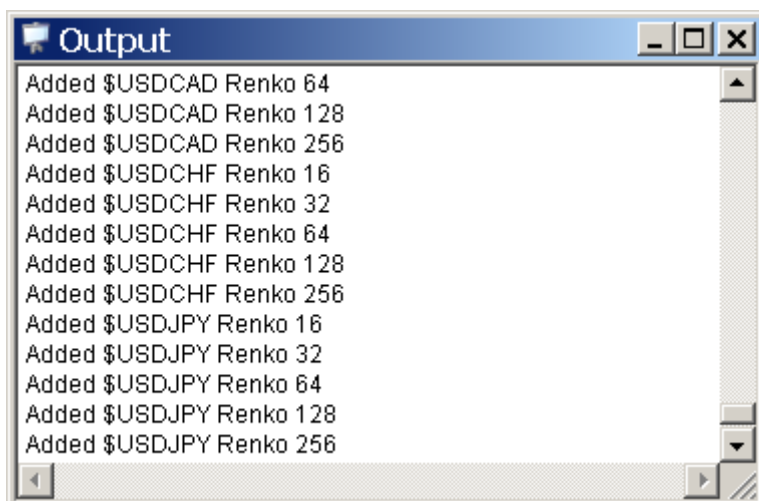


Figure 16. Output window used for debugging

4.3 Developed code

During the course of completing the project, a number of different pieces of NinjaScript code were written, such as different strategies, code analyzing the price and custom bar types. Only a few of them will be explained in greater detail.

4.3.1 Strategies

Both of the strategies explained in this section have the properties below.

- Can be in trend and counter-trend mode
- Unit (lot) size can be set
- Minimum bar range can be set
- Can use both limit and market orders
- Use multiple currency pairs
- Use multiple bar ranges

Multi-scale price-based strategy #1

This strategy is a combination of a number of different methods. It can be used in both trend and counter-trend mode. It does resemble a basket strategy in the sense that it uses multiple currency pairs at the same time, but this is mostly to make it easier to manage, as in Ninja-

Trader version 6.5, starting a new strategy for each currency pair was difficult and time-consuming. In version 7 of NinjaTrader this has been made easier.

The strategy can be thought of as a combination of multiple strategies working with different bar ranges using the new Neko bar type (see 4.3.2). These bar ranges can then be given assigned priorities when deciding whether to go long or short. By default, bar ranges are increased in power of two. For example: 32, 64, 128 etc. Using this method, the strategy is easily modified to either prefer lower or higher price scales or a combination of both.

The basic logic of the strategy is very simple. If the bar's close is higher than the open, buy; if lower, sell and the opposite when in counter-trend mode. In addition to this mode, I developed a type of mixed mode, which changes between trend and counter-trend based on a guess about the market conditions. The mixed mode looks at the latest two bars and makes a decision according to Table 6 below.

Table 6. Mixed mode operation

Current bar	Previous bar	Mode
Up	Down	Counter-trend
Up	Up	Trend
Down	Up	Counter-trend
Down	Down	Trend

Basket strategy #1

The basic idea of this strategy is to look at separate currencies instead of currency pairs. This gives new trading possibilities, such as ranking currencies against each other to find out which ones are strongest and which the weakest or to create bar series for currencies, similar to the U.S. dollar index which measures the relative strength of the U.S. dollar against other major currencies. The operating principle of this strategy is demonstrated below.

An array of eight integers, for each of the eight most popular currencies is created:

AUD	EUR	GBP	NZD	CAD	CHF	JPY	USD
-----	-----	-----	-----	-----	-----	-----	-----

-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---

If, for example, AUD/USD moves up 20 pips, the array is updated for both AUD and USD:

AUD	EUR	GBP	NZD	CAD	CHF	JPY	USD
20	-	-	-	-	-	-	-20

If AUD/CAD were to move down 20 pips, the array would be as follows:

AUD	EUR	GBP	NZD	CAD	CHF	JPY	USD
0	-	-	-	-20	-	-	-20

As the strategy keeps running, eventually a clearer picture of the strengths of currencies is created. The strategy goes through every currency pair and checks if the base currency is either stronger or weaker than the counter-currency in the array and then either decides to go long or short or not to trade, if the integers are equal. Interestingly, it is not actually required to use the counter-currencies to trade this table, as appropriate positions for all the majors can be derived from the table. Trading with only the majors saves in spread costs and reduces redundant trades.

Using this type of strategy with only the major currency pairs causes the strategy to depend on the strength of the US Dollar. However, using the strategy with cross-currencies as well makes the strategy a bit more useful.

4.3.2 Bar types

Creating new bar types in NinjaTrader was not officially supported. Information about creating them had to be found on the forums and by inspecting the original code for official bar types and modifying them. NinjaTrader 6.5 keeps built-in bar types in a file named “@BarsTypes.cs” in the directory:

C:\Documents and Settings\\My Documents\ <NinjaTrader Folder>\ bin\Custom\Type
(Windows XP)

Neko

Neko (Figure 15) is simply a modified renko bar with symmetrical behavior and it only measures price. The basic idea of this bar type is to give even odds in trading. When trading this bar type, the win/loss is always at 1.0, meaning that the only thing that can change is whether or not price tends to either trend or range. This helps to determine whether a certain bar range is either more likely to trend or not.

Tracker

Tracker (Figure 17) is a bar type that does not update based on price. Instead, it looks for a specified amount of ticks in a row in one direction and then creates a new bar. The basic principle is that if there are many ticks in one direction, it is less likely to be due to chance. This bar type is not really meant for automated trading but might be useful in technical analysis to potentially identify the long-term trend.

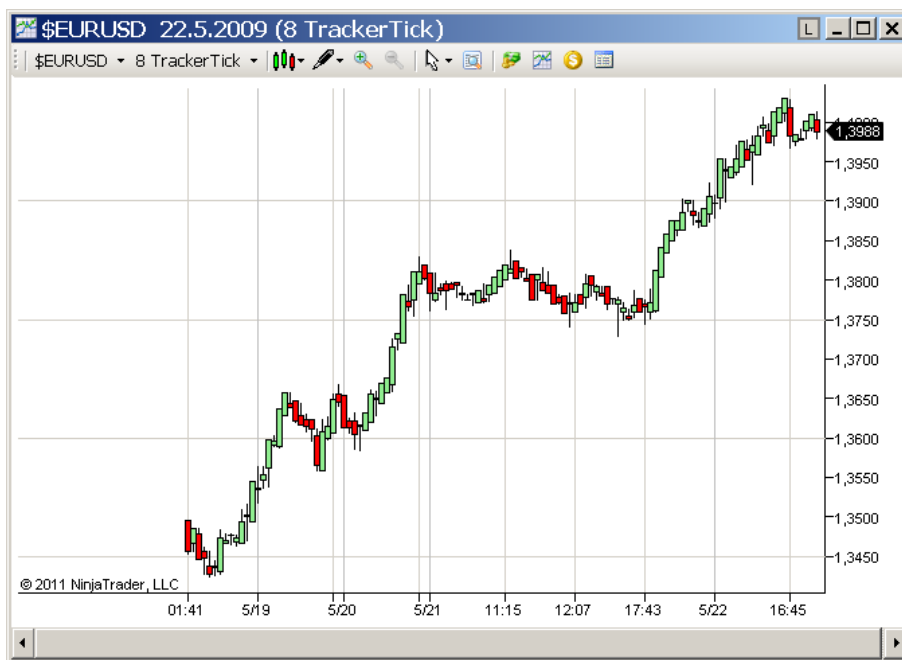


Figure 17. Tracker bars

Figure 18 below shows three types of charts. The top one uses regular 60 minute bars. The second one is Tracker with a parameter of 8 (ticks). The bottom one is Neko with a parameter of 16 (pips). It is important to note that one should not compare the bar types horizontally, but rather vertically as the new bar types only take price into account and thus do not align themselves to the topmost time-based bars.

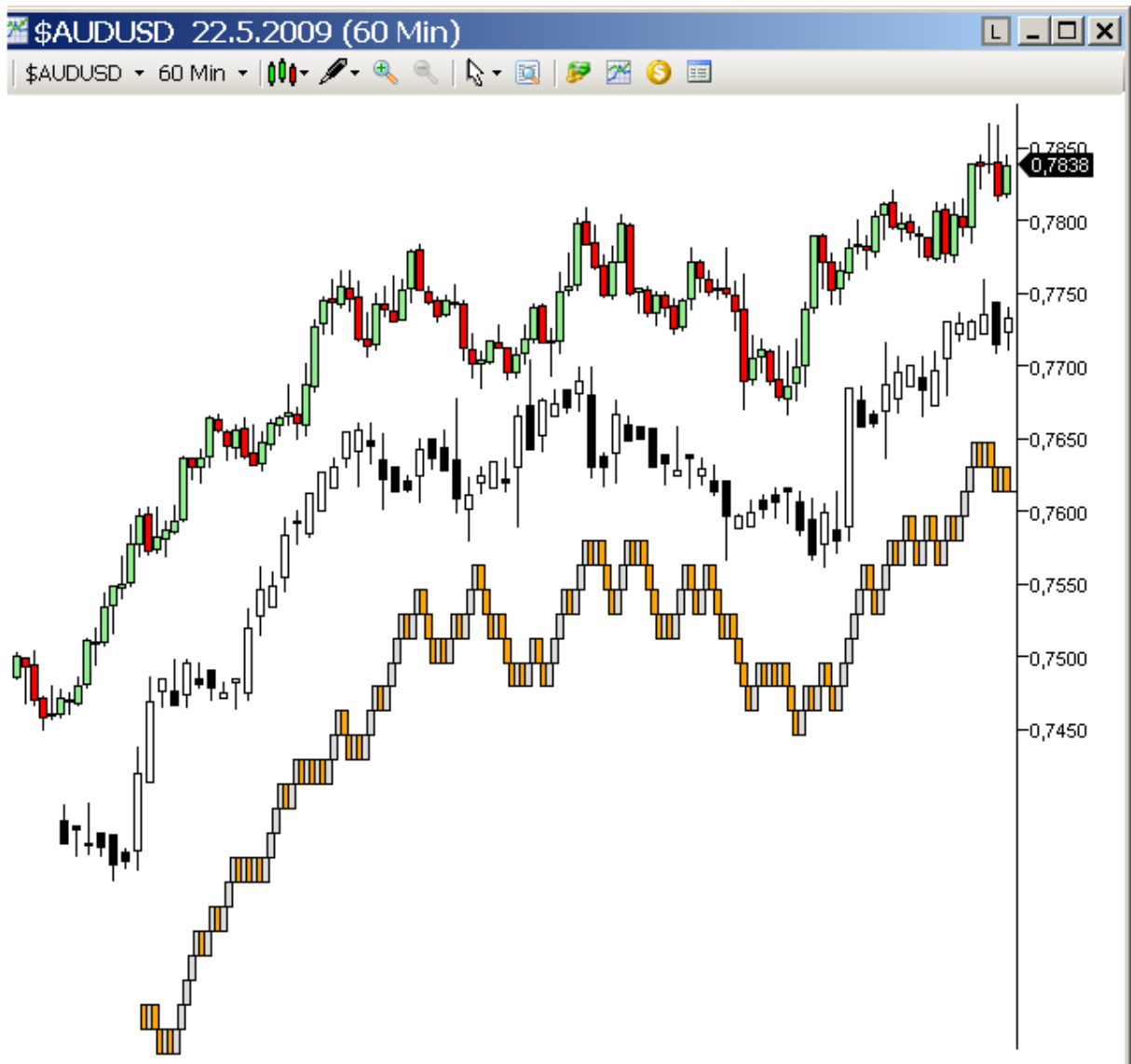


Figure 18. 60 minute, Tracker and Neko bars compared

4.4 Problems

One of the problems in backtesting is that the historic data only includes the bid. If no spread cost is taken into account, one may get overly optimistic results and conclude that the strategy has an edge (Chan, 152). Backtesting also does not take things such as network delay, slippage into account and if the strategy uses limit orders, backtesting cannot tell whether an order would be filled or not, it can only guess. Fortunately, Market Replay has an algorithm built-in that emulates network delays and fills. Another problem with backtesting is that it seems to have problems with strategies using multiple currency pairs, such as the basket strategy. For the basket strategy, market replay was preferred.

Testing in market replay is quite CPU-intensive. As the developed strategies use multiple currency pairs, testing took a long time: approximately one hour for 5 days (one trading week) of data. NinjaTrader also has a maximum market replay speed of 500x (1x being real time) and thus 5 days of data will take approximately $\frac{5 \cdot 24 \cdot 60 \text{ minutes}}{500} = 14.4$ minutes at minimum.

Memory usage was also a problem as NinjaTrader 6.5 has a memory limit of ~1.4 GB. Using multiple currency pairs and loading multiple bar series consumed much memory and market replay would run into the memory limit. This was mostly due to the design of version 6.5, as it keeps all of the previous bars in memory, regardless of whether the strategy uses them or not. Version 7 would later come out and allow for a 256-bar limit, but in the mean time, strategies were modified to use less memory using the following method. The strategy would take in tick data and manually create bar data that would be placed in an array. Then the amount of previous bars could be limited to only a few or just one. Another way to minimize memory use was to limit the amount of currency pairs and to use bars with a higher pip range.

4.5 Results

Strategy used: Multi-scale price-based strategy #1

Testing parameters:

- Backtest (using GAIN data)
- Time period: 1 January 2009 to 31 January 2009
- Spread not taken into account

Strategy parameters:

- Trend
- Bar type: Neko
- Unit size: 1
- Uses majors
- Bar ranges starting from 32, increasing with power of two
- Prefers higher bar ranges

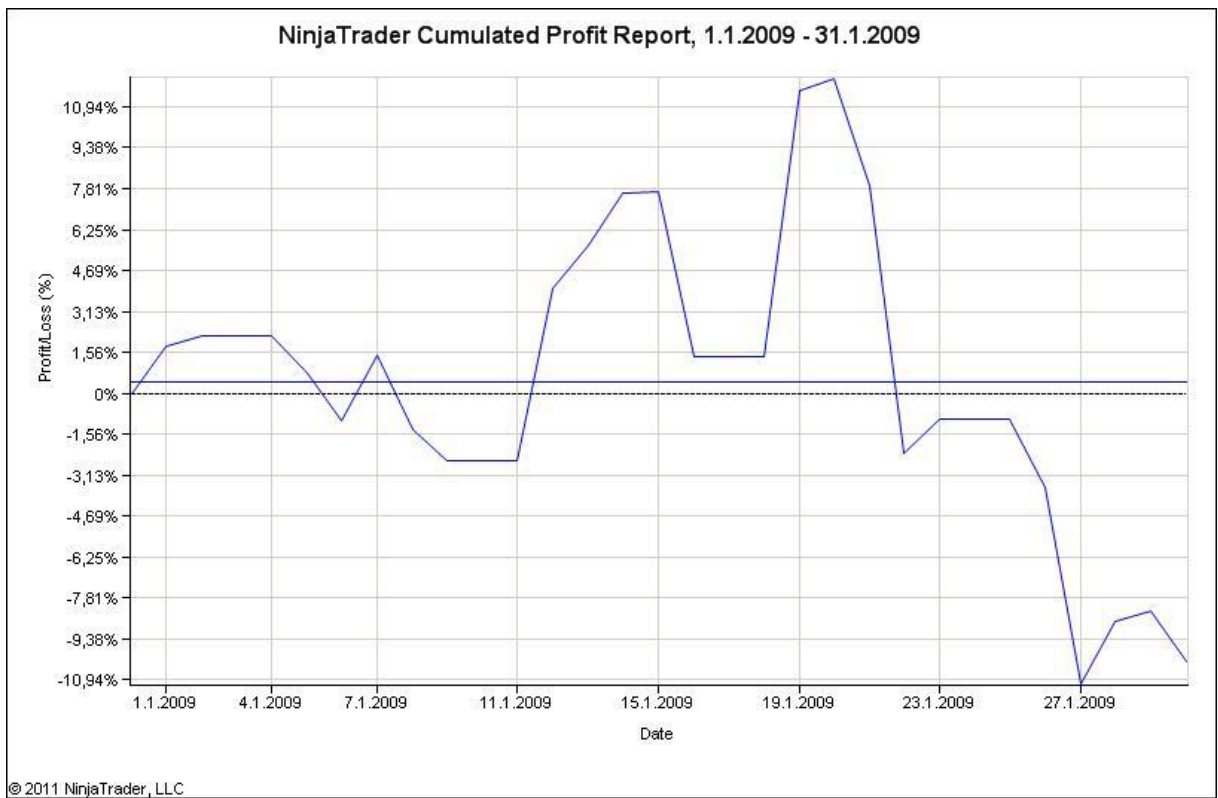


Figure 19. Strategy #1 - Cumulated profit

As can be seen above, the strategy results seem quite random and no underlying edge can be seen.

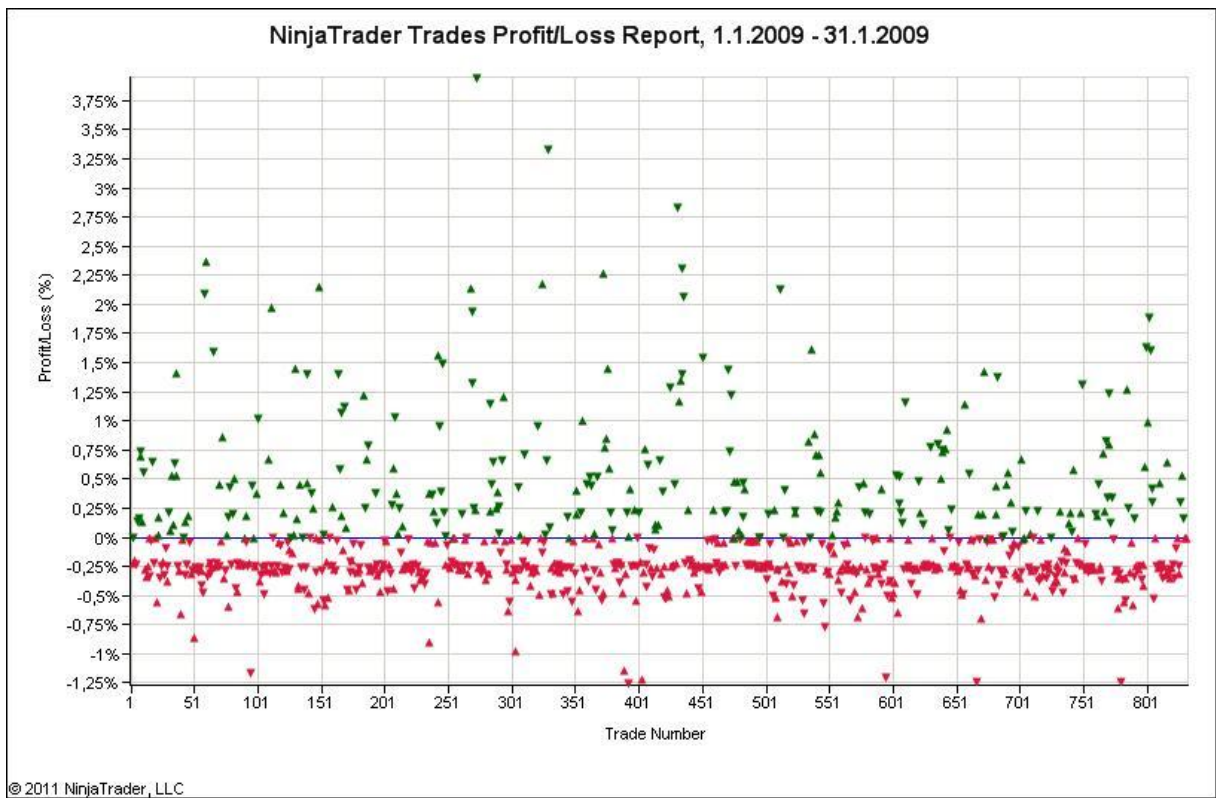


Figure 20. Strategy #1, Profit/Loss Report. Green triangles indicate winning trades and red triangles losing trades.

The figure above shows in a visual form that the strategy makes lots of small losses and fewer bigger wins, balancing each other out.

Table 7. Account performance

Performance	All Trades	Long Trades	Short Trades
Total Net Profit	-0,12	-0,12	0
Gross Profit	1,63	0,76	0,87
Gross Loss	-1,75	-0,88	-0,86
Commission	0	0	0
Profit Factor	0,93	0,86	1
Cumulated Profit	-10,24 %	-13,34 %	3,57 %
Max. Drawdown	-20,98 %	-18,04 %	-11,88 %
Sharpe Ratio	1	1	1
Start Date	1.1.2009		
End Date	31.1.2009		
Total # of Trades	830	413	417
Percent Profitable	31,20 %	30,51 %	31,89 %

# of Winning Trades	259	126	133
# of Losing Trades	571	287	284
Average Trade	-0,01 %	-0,03 %	0,01 %
Average Winning Trade	0,58 %	0,53 %	0,63 %
Average Losing Trade	-0,28 %	-0,28 %	-0,28 %
Ratio avg. Win / avg. Loss	2,07	1,89	2,25
Max. conseq. Winners	8	6	7
Max. conseq. Losers	20	12	13
Largest Winning Trade	3,95 %	2,37 %	3,95 %
Largest Losing Trade	-1,25 %	-1,22 %	-1,25 %
# of Trades per Day	28,71	14,29	14,46
Avg. Time in Market	196,3 min	187,5 min	205,0 min
Avg. Bars in Trade	1941,8	1839,9	2042,8
Profit per Month	-10,40 %	-13,54 %	3,64 %
Max. Time to Recover	9,24 days	24,13 days	9,23 days
Average MAE	0,26 %	0,27 %	0,26 %
Average MFE	0,47 %	0,44 %	0,50 %
Average ETD	0,48 %	0,48 %	0,49 %

Analysis

NinjaTrader provides statistics for strategies in the Account Performance tab. It has three modes: currency, percent and points. The above table shows the results in percentage mode. The results suggest that the strategy was not profitable during the time period. Not only that, but the results seem random. As the backtest in this case was set not to include a spread, taking it into account would result in lower performance.

Explanations of basic statistics

Cumulated profit – “This statistic returns a value representing a summation of all the profits earned by all your trades. It can be interpreted as a performance measure for your strategy.”

Profit Factor – Measures the profitability of the strategy. For example, if the profit factor is 2.0, the strategy doubles the money invested. If below 1.0, the strategy is not profitable.

Max. Drawdown – The lowest point the equity reached during the time period.

Profit per Month – An estimate on the amount of profit in a month.

4.6 Real life implementation

There are considerations that need to be taken into account when planning to run strategies in real life.

- Is there enough money available to open an account?
- If using a commission-based broker, would the strategy be profitable?
- Is the minimum unit (lot) size of the broker small enough to run the strategy with minimal risk?
- What if a power outage happens?
 - Should one use a dedicated server?
- What if there is a bug in the code?
- What if the connection to the broker is disrupted, does the strategy recover well?
 - Are the market positions between strategy and broker synchronized?
- What should the strategy do during the weekends?
- Should major news events be ignored?

Due to time constraint in this project, real life implementation will be left for the future.

5 CONCLUSION

The aim in this study was to “research trading in the Foreign Exchange market and to implement an automated trading strategy using a suitable programming language and to successfully test it with real market data”. Of particular interest was to see if new trading ideas could be thought of and used in creating the strategy.

In order to accomplish the aim of the study, different subjects related to Foreign Exchange, trading in general and algorithmic trading was researched. Different considerations, such as platform choice, strategy and broker types have been discussed. In the end, many working automated strategy prototypes and new bar types for NinjaTrader have been developed of which only a few have been explained in this study. Historic market data for testing was found and imported into the trading platform using a custom script. Live market data was gathered from two brokers using a tool called Market Replay. Strategies and bar types was tested using the gathered data. A lot of time was spent debugging the code, testing strategies and finding solutions. Problems encountered during the project have been discussed.

Although the practical part of the study was completed and tested, the results regarding profitability did not seem positive. It seems that trying to profit from a market that seems to resemble a random walk is quite difficult. In the beginning of the project, I was quite eager to create a profitable strategy and easily spent hours testing different strategies and parameters. It would seem that strategy development requires a more systematic and calculated approach not affected by emotions. I found that even though I was testing the strategies with virtual money, I still experienced emotions such as happiness and frustration. As time went on, developing strategies eventually became sort of an obsession and a lot of time was wasted, so to speak, compared to eventually lackluster results. Trading these strategies in real life would probably require discipline and emotional control, which aren't really easy for me personally.

The study can be improved upon in the future. The strategies could be adapted to other markets, such as stock markets. Different, more advanced methods, such as wavelets, multifractals, neural networks and genetic algorithms could be studied and implemented. The code could be updated for newer version of NinjaTrader or even ported to other platforms and strategies modified for live trading. Due to time constraints of this study, these will all have to be left for the future.

All in all, the study gives guidelines to beginning developers of automated strategies. I hope that the personal experiences, problems and solutions offered in this study as well as the new ideas generated can give inspiration to anyone interested in the topic.

BIBLIOGRAPHY

Books

Chan, Ernest P. 2009. Quantitative Trading - How to Build Your Own Algorithmic Trading Business. New Jersey. John Wiley & Sons, Inc.

Levinson, Marc 2005. Guide to Financial Markets. London. Profile Books Ltd.

Weithers, Tim 2006. Foreign Exchange – A Practical Guide to Financial Markets. New Jersey. John Wiley & Sons, Inc.

Electronic sources

BIS = Bank for International Settlements 2010. Triennial Central Bank Survey – Report on global foreign exchange market activity in 2010. Updated 1.12.2010. [Referred 14.02.2011]

Available in PDF-format:

<URL:<http://www.bis.org/publ/rpfx10t.pdf>>.

Cheng, Grace 2006. Forex Trading: Market Maker Vs. ECN. No update information available. [Referred 12.03.2011] Available in www-format:

<URL:<http://www.investopedia.com/articles/forex/06/ECNmarketmaker.asp>>.

Euromoney. FX Poll 2010. 2010. No update information available. [Referred 12.03.2011]

Available in www-format:

<URL:<http://www.euromoney.com/Article/2473939/Foreign-exchange-survey-2010-results.html>>.

FIX Organization. No update information available. [Referred 19.05.2011] Available in www-format:

<URL:<http://www.fixprotocol.org/what-is-fix.shtml>>.

Frankel, Jeffrey A. 2008. Foreign Exchange. The Concise Encyclopedia of Economics. 2008. Library of Economics and Liberty. No update information available. [Referred

10.05.2011]. Available in www-format:

<URL:<http://www.econlib.org/library/Enc/ForeignExchange.html>>.

Moreira, Gonçalo. FXStreet Learning Center. No update information available. [Referred

10.05.2011] Available in www-format:

<URL:<http://www.fxstreet.com/education/learning-center/default.aspx>>.

The Kelly Criterion, Part I, Basic Kelly Math. Author unknown. Updated 28.09.2006. [Referred 19.05.2011] Available in www-format:

<URL:http://www.castrader.com/2006/09/the_kelly_crite.html>.

World Federation of Exchanges. Number of trading days, average daily turnover and average value of trades. No update information available. [Referred 14.02.2011] Available in xls-format:

<http://www.world-exchanges.org/files/statistics/excel/EQUITY609.xls>

APPENDICES

1 Software

Wget

“GNU Wget is a free software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive commandline tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.”

Website: <http://www.gnu.org/software/wget/>

NTDataConverter

“NTDataConverter is a data converter for NinjaTrader v6.5. The aim of NTDataConverter is to be able to convert various historical data format to NinjaTrader import file format.”

Website: <http://algo-traders.blogspot.com/2010/07/ntdataconverter.html>

Simple Search & Replace (SSR)

“Sometimes you just need a very simple tool to search and replace text in text files. This is such a tool. It is built to be used in batch files so it uses a simple method to replace quotation marks (") for the text arguments.”

Website: <http://www.programmersheaven.com/download/56619/download.aspx>

QuickFIX

“QuickFIX is a full-featured open source FIX engine, currently compatible with the FIX 4.0-5.0 spec. It runs on Windows, Linux, Solaris, FreeBSD and Mac OS X. API's are available for C++, .NET, Python and Ruby. QuickFIX/J is available for Java.”

Website: <http://www.quickfixengine.org/>

2 Code

2.1 OnExecution()

```
protected override void OnExecution(IEExecution execution)
{
    // for every tracked position...
    for(int t=0;t<entryOrder.GetLength(0);t++)
    for(int u=0;u<entryOrder.GetLength(1);u++)
    {
        if(entryOrder[t,u] != null)
            if(entryOrder[t,u].Token==execution.Order.Token)
            {
                // if the order has been filled...
                if (execution.Order.OrderState == OrderState.Filled)
                {
                    if(execution.Order.Action==Action.Buy
                    || execution.Order.Action==Action.BuyToCover)
                    {
                        // if a buy, add the amount of units the order was filled
                        position[t,u]+=execution.Order.Quantity;
                    }
                    if(execution.Order.Action==Action.Sell
                    || execution.Order.Action==Action.SellShort)
                    {
                        // if a sell, subtract the amount of units the order was filled
                        position[t,u]-=execution.Order.Quantity;
                    }
                }
                //quit loop
                t=entryOrder.GetLength(0);
                u=entryOrder.GetLength(1);
            }
    }
}
```