Insert here your thesis’ task.

Czech Technical University in Prague
Faculty of Information Technology
Department of Software Engineering

Bachelor's thesis

ATS - trading strategies based on S/R levels of technical analysis

Petr Jirásko

Supervisor: Ing. Jan Trdlička, Ph.D.

12th May 2015

Acknowledgements

I would like to thank my supervisor, Ing. Jan Trdlička, Ph.D., for his kind guidance and help with the thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 12th May 2015 . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Jirásko, Petr. *ATS - trading strategies based on S/R levels of technical analysis.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2015.

Abstrakt

V posledních letech počítače radikálně změnily téměr každý aspekt lidských činností. Jednou z těchto činností je i obchodování; od směnných obchodů se obchodování vyvinulo až do své nejmodernější podoby - automatizovaného obchodování. Proto má práce představí základy historie a principů vybraných trhů, na kterých se automatizované obchodní systémy dají využít. Také popíše funkci, výhody a nevýhody automatických obchodní systémů. Navíc poskytne základní znalosti potřebné pro vytvoření takového systému, zejména úvod do světa obchodních platforem, výběr konkrétní platformy a základy technické analýzy. Následuje představení, implementace a vyhodnocení několika strategií založených na S/R úrovních technické analýzy.

Klíčová slova Automatické obchodní systémy, supporty a resistance, technická analýza, obchodní platformy

Abstract

In the last few years, computers have radically changed nearly every aspect of human activities. One of them is trading; from bargains in the history trade has developed into its most modern form - automated trading. Therefore, my thesis introduces basics of history and principles of selected markets, where automated trading systems can be utilized. It also describes function, advantages and disadvantages of automated trading systems. Furthermore, it supplies basics of knowledge necessary for creation of such systems, namely a introduction into world of trading platforms, selection a particular platform and basics of technical analysis. Afterwards, a few trading strategies based on S/R levels of technical analysis are introduced, implented and evaluated.

Keywords Automated Trading Systems, Supports and Resistances, Technical Analysis, Trading Platforms

Contents

List of Figures

List of Tables

Chapter 1

Introduction into trading

The history of trade may be as long as the history of the whole humanity. In the last decades, the rapid development of information technologies enabled even small traders to take part in trading. Internet allowed people equipped just with an ordinary personal computer to start trading from home.

A lot of traders still trades manually, however, there is a great possibility to create automated trading systems. Except for the the big players, such as RSJ, a company co-owned by famous Karel Janeček[1], there is also a numerous community of traders using automated trading system.

In the next two sections a brief excursion into history of markets and an introduction into basics of automated trading systems is made.

1.1 Markets history

The history of trading is as old as the humanity itself and begun in the ancient times as a bargain. The most important part for my bachelor thesis is actaully the part of history concerning about foreign exchange markets, stock markets and derivatives markets. Therefore, in the subsections bellow a brief excursion into history and principles of these markets is made. It is not a deep excursion, since its aim is not to depict the complety history and all details about mentioned markets, but to higher the reader's awareness about it.

1.1.1 Foreign exchange markets

The history of currency exchange reaches as far as the second century B.C. Jewish people began to produce own coins, however their duty was to pay the taxes to the Roman empire in Romans' currency. It was not a financial market as we know nowadays, since the Romans determined the currency rate.

In centuries after, the value of the money was in material the were made of, mostly gold, silver or bronze. A change came in 1819 as Great Britain came with Resumption Act claiming the Gold Standard. Since the time the money emitted by the Bank of England were covered by gold. Later in the 19th century other countries joined the Gold Standard.

The Gold Standard was shattered after the First World War and the end of it let to creation of Bretton Woods monetary system in 1944. The International Monetary Fund was established, as well as the World Bank. The system was based on principles such as firm currency rates with maximal deviation of 1 %, preservation of gold as univarsal value and making US dollar the world reserve currency.

In 1971, Richard Nixon, the president of the United States of America, revoked the connection between US dollar and gold. In 1976, in Kingston, Jamaica, new system was introduced. This system cancelled the oficial price of gold and connection between world currencies and gold, excluded gold from foreign exchange reserves and freed currence rates.

The name Forex is an abbreviation word comming from International Interbank FOReign EXchange. The first Forex market was opened 1973 as an OTC market, traded 24 hours 5 days a week.[2]

On an average day, a total volume of trades is around 3 trillion US dollars, which is much more than the volume of trades on NYSE and LSE alltogether. In contrary to stocks, a Forex market does not have any central point. It is a net connecting banks, brokers and traders.

There are 3 groups of currencies. The first group, the major currencies, contains American dollars (USD, \$), Euro (EUR, €), British pound (GBP, £), Swiss franc (CHF) and Japanese yen (JPY, ¥). Those currencies are traded the most. The other 2 groups, side currencies and exotic currencies, have smaller liquidity and therefore have higher spreads and do not offer much possibilities of profiting.

1.1.2 Stock markets

Stock markets begun in the 17th century. The Verenigde Oostindische Compagnie (Dutch East India Company) wanted to gain more capital and issued the world oldest share. The subscription was opened in 6 Dutch cities from the 1st April to the 31th March. Despite the fact the shares were intrinsically same, shares from different cities were not interchangeable.[3] Therefore, the oldest stock exchange in the world, the Amsterdam Stock Exchange was founded.

Stock trade was also founded in America. In 1792, five securities are traded in New York City, of which three are government bonds and two are bank stocks. In 1817 an improvement is made and New York Stock & Exchange Board founded (nowadays known as New York Stock Exchange). The president read a list of traded securities and the traders trade the security currently in turn in 2 sessions a day.[4]

Emitting of stocks is one of ways, how large companies get own capital. A stock constitutes the equity of the stockholder stake.

There are 2 types of stocks. Common stocks gives the holder voting rights. Holders of preferred stocks have restricted voting rights, however, they have guaranteed height of dividends and priority against common stocks holders in case of corporation bankruptcy and therefore preferred stocks are closer to bonds.[5]

A trading of stocks can be done in 2 ways. Directly at the stock exchange or OTC. Trading stocks from a trading platform on own computer is done always done OTC through a broker, which is connected to the stock exchange. The fees are charged per contract.

A corporation, who wants to its stocks to be traded in a stock exchange must meet stock exchange demands about the market value of the corporation and its yearly profit.

The biggest stock exchange in the world is NYSE, where are only corporations with market value higher than 100 million US dollars and yearly profit higher than 10 million US dollars. NYSE is followed by Nikkei in Tokyo, LSE, DAX in Germany and Toronto Stock Exchange.[5]

1.1.3 Futures

The financial derivatives are a vast range of finincial instruments. They have their origin in the middle of 16th century also in the Netherlands as grain traders started to settle forward contracts.[3]

In the United States, in the 19th, the farmers growing grain were selling it bargain-priced in the summer and expensively in the winter. The wealthy enterpreneurs built silos and started to do forward contracts. A forward contract was among a seller and buyer: the amount of grain and price for it were negotiated. However, such a contract did not guarantee the buyer would not buy for a better price somewhere else, as well as the seller would not sell for a higher price to somebody else.

Therefore, exchanges started to collect deposit, that was passed from the side that defaulted to the other side. Along with this, they developed a standardized contract *interchangeable and addressed everything except the price*[6] - a futures contract.

The standardization of contracts lead the nowadays state, when only 3 % of such contracts end up in a physical delivery. Because of such amount of operations, a **clearing** was developed. The exchange steps between two sides of the contract and assumes their obligations in order to make the contract independent of both sides.

Futures contract can be concluded about commodities, as well as i.e. for foreign money (a contract as an insurance against rate volatility) on an exchange. Therefore a broker offering futures must have a direct connection to an exchange.

If a trader wants to buy a future contract, he has to pay to exchange a **initial performance bond** determined by the exchange. Its height is not set as a percentage from the contracts value, however it is derived from the price volatility of the commodity.

1.1.4 Options

Options are financial derivatives traded both, on the stock exchanges as well as OTC. Brokers have the same role as in the case of stocks - to connect the traders to each other.

Options enable the owner to buy (in case of **call option**) or sell (in case of **put option**) underlaying asset for a certain price at the particular time. This price is called **strike price** and the time is called **maturity**.

There are also two types of options according to the possibility of excercision. **American** options can be excercised at any time, **European** only at the maturity time.[7]

1.1.5 CFD

CFD, or Contract For Difference, is a financial derivative, as well as options. It is a speculation on a stock price movement. One side of the contract buys the CFD for a current price of the stock and makes a commitment to pay the difference between current price and the price at the time of maturity.

In case the difference is negative, the trader has to pay it, however, in the opposite case, the trader is in profit.

1.2 Automated trading systems

Computers nowadays bacame involved in nearly every part of our life. Before online trading platforms came, the traders had to be physically in the exchange place. With a trading platform a trader can trade comfortly from his home.

With a trend of automation, also automated trading strategies came. An ATS is actaully a set of exact rules telling certain conditions for entering and leaving trading positions and also the volume and prices of the trade. When the criteria are met, the automated trading system automatically enters a trading position.

1.2.1 Advantages and disadvantages of automated trading

Automated trading differs from the manual trading, in which the trader must manually enter into all positions, or semi-automatic trading systems, where the trader is told about occurance of conditions fulfilment, however, the final decision to enter the trade lies still on him.

There is a lot of advantages of automated trading systems. All major traders, such as banks, use automated trading system. It is nearly impossible to figure out a strategy better than their, but thanks to major traders markets have higher liquidity and therefore a higher possibility of creating a functional trading system. Another advantage is the elimination of so-called human factor. An ATS can not be in stress due to losses; it always enters a trade only when certain criteria are met. Also there is nothing as tiredness and loss of concetration, an ATS can trade the whole market opening time. Also the possibility of relatively quick backtesting and strategies optimization can not be omitted.

Besides advantages automated trading systems have also their disadvantages. Strategies programming is harder discipline than trading itself, because expressing of certain criteria can be not trivial. There is also no measure for catching the fundamental news, for example government proclamations, which perceptibly affect markets. It also happens to a lot of strategies, that they are succesful during backtesting, however in a real market they end up in a loss. Also a risk of market change, that can turn a profiting strategy into strategy with losses, must not be omitted.

1.2.2 Strategies used in automated trading systems

An automated trading system has to be build upon a particular idea. There are several basic kinds of tools of technical analysis used in creation of such systems, namely:

- candlestick patterns (see also 3.3)
- market structure (see also 3.4
- indicators (see also 3.6)
- market profile

Systems based on indicators enter and exit trades, when one or more indicators complies with some condition. Systems based on candlestick patterns uses special candlistick formations; if such a formation appears, it may indicate a change or remaining of a trend. Systems using market structure use different patterns of price development; one of them are S/R levels used in my thesis. Last, but not least, systems based on market profile use the information about past trade volumes as a tool for predicting of future price development.

Chapter 2

Trading platforms

There are more ways, how to connect to your broker. It may provide an own API for, has its own trading platform or support a universal multibroker platform. This chapter concerns especially with multibroker platforms.

A platform is a complex software enabling a trader to connect to brokers, trade, see price development and many more things. In next sections, various basic aspects of platforms are introduced and all 3 platforms are compared.

2.1 Hardware demands and supported operating systems

According to [8], most of the platforms support Windows operating system, some of them support Mac OS, however, nearly no platforms support Linux. It is possible to run the platforms in virtualized operating system, which are not that efficient and hardware demands of every platform must bet taken into account.

2.1.1 MultiCharts hardware demands and supported operating systems

According to MultiCharts's manufacturer [9], the minimal requirements are: CPU: at least dual core, memory: at least 2 GB RAM, hard-disk space: 500 MB, monitor: minimal resolution 1024 x 768 and operating system: Windows XP, Vista, 7 and 8.

Recommended hardware requirements are: CPU: at least dual core, memory: 4 GB RAM, hard-disk space: 1 GB, monitor: multi-monitor solution (it is possible to display charts etc. on multiple screens) and operating system: Windows XP, Vista, 7 and 8.

As MultiCharts is based on .NET technology, it requires .NET framework 3.5 and 4 in order to run.

2.1.2 NinjaTrader hardware demands and supported operating systems

There was no note about hardware and operation requirements of this platform. Subjectively, the program runs smoothly and I experienced no freezing of the program at all.

From operating systems, NinjaTrader supports only Windows.

2.1.3 MetaTrader 4 hardware demands and supported operating systems

Subjectively speaking, MetaTrader 4 looks slightly archaic. However, it is running smoothly and without any problems even on a virtualized machine. Minimal system requirements are: Windows 2000/XP/Vista, single core CPU 2 GHz or with higher frequency, 1 GB of memory, screen resolution 1280 x 1024 or higher and an internet connection.[10]

2.2 Pricing policies

Price is one of the most important factors when selecting a platform, especially for a trader-beginner.

MultiCharts offers a 30 days trial version with all features. After 30 days, user must buy a full version.[12]

There is a list of features of NinjaTrader containing advanced charting and analysis, live trading, simulated trading, automated trading, atm strategies, chart trader, one cancels other orders, custom indicator/strategy development and hotkeys and 4 types of licenses enabling you to use only particular features [11]:

- Free lincense enables to do everything, except for real trading.
- Direct license enables only advanced charting, live trading and simulated trading.
- Lease license allows to use every feature of the platform. After the end of the lease period, you might not use the software anymore for live trading, but you can continue using it under free license.
- Lifetime license enables, as well as lease license, to do everything.

MetaTrader 4 is completely free of charge for personal purposes, as mentioned explicitely in the EULA.

A comparison of all prices is in table 2.1. Prices stated in the brackets are per month. In the case of NinjaTrader, the first price stays for a signle broker usage, the second one for a multi broker usage.

	MetaTrader 4	**MultiCharts .NET**	**NinjaTrader**
3 months	-	\$297.00 (\$99)	\$180.00 (\$60.00) / \$270.00 (\$90.00)
6 months	-	\$497.00 (\$82)	\$330.00 (\$55.00) / \$495 (\$82.50)
a year	-	\$797.00 (\$66)	\$600.00 (\$50.00) / \$900.00 (\$75.00)
lifetime	free	\$1,497.00	\$995.00 / \$1,495.00

Table 2.1: License pricing of MetaTrader 4, MultiCharts .NET and NinjaTrader. Prices in brackets are per month, in case of NinjaTrader, the first prices are for single broker usage, the second for multi broker usage.

Despite the fact MultiCharts is for relatively high sum of money, AMP Futures brokerage offers MultiCharts .NET Special Edition for free for a life.[13]

2.3 Data feeds and vizualization

The data is the most important issue of trading. Data can vizualized through different types of charts and they are also essential part for backtesting. Therefore the platforms contain at least some form of stored data management.

2.3.1 MultiCharts data feeds and vizualization

MultiCharts offers the widest range of types of charts. Apart from classical bar and candlestick charts it also offers line chart, line break, renko, kagi, point & figure, Haikin-Ashi, volume delta and cumulative delta charts. User can set the gaps between chart elements, the size of elements as well as their color.

Incoming data can be provided by a broker or a data feeder supporting one of 3 APIs, namely CQG, Rithmic+ and TTNet. There are also 29 data feeders, varying in available historical data, both free and paid.[14] I.e. CQG offers tick-by-tick data 50 days in the past, minute data one year back for most of the symbols.

The management of historical data has a different philosophy than in other 2 platforms. There is no history center for managing historical data for backtesting and optimization, however, data are downloaded before testing.

2.3.2 NinjaTrader data feeds and vizualization

NinjaTrader has 4 types of chart: box, candlestick, line on close, HiLo and OHLC. The gaps between elements can not be set as in case of MultiCharts.

Chart settings (including colors) can be saved or loaded as a template as in MetaTrader 4.

Historical data are managed in Historical Data Manager. Data can be imported or exported from/into a .csv file, edited or downloaded from the current connection with broker or data feeder.

Among data feeders mentioned on the website of manufacturer are eSignal, Google Finance, IQFeed from Telvent DTN, Metastock Import, TradeStation and Yahoo Finance.[15]

2.3.3 MetaTrader 4 data feeds and vizualization

MetaTrader 4 offers only 3 types of chart: bar chart, line chart and candlestick chart. Only the size of the whole chart can be changed, the gaps remain always default. Colors of the chart can be set, as well as saved a loaded into/from a template file.

The historical data for backtesting and optimization can be managed in the history center. Data can be directly imported or exported from/to a .csv file. The basic data are provided by the software producer, MetaQuotes Software Corporation, additional data can be downloaded through account from broker.

2.4 Indicators and automated strategies

Indicators and automated strategies are the core elements of automated trading (for details about indicators, see 3.6). Platforms offer different basic indicators and strategies, as well as the philosophy of trading manner differs.

2.4.1 MultiCharts indicators and automated strategies

MultiCharts offers so-called Studies. A Study is an Indicator or a Signal, which is actually a function causing event, that signalizes a particular situation occured, i.e. a price action pattern appeared, and entering trades.

Indicators and signals are not sorted into categories, however there are off course indicators signalizing trend and oscillators. To see the full list please visit [16].

Strategies executions differs from NinjaTrader and MetaTrader, Multicharts has so-called Portfolio Trader, in which you can create or delete Strategies. A Strategy consists of 2 lists: a list of Signals and a list of Instruments. After launching all signals from the list all applied to all instruments from the list.

2.4.2 NinjaTrader indicators and automated strategies

NinjaTrader also offers numerous indicators, which are, as in case of MultiCharts, not listed in categories. Off course, NinjaTrader contains basic indicator, such as MA, Zig-Zag or others. The manufacturer does not list the indicators on its website.

Code of all indicators can be inspected. Modification of them is forbidden, however you can also create your own custom indicators. Creating new strategy or indicator is done through a wizzard setting automatically input parameters.

After installation there are just 4 sample strategies, new can be created.

You can run strategy or add indicator by simple attiching it to the chart.

2.4.3 MetaTrader 4 indicators and automated strategies

MetaTrader 4 offers indicators sorted by categories. All indicators in categories are just in compilated form and it is impossible to edit them or to inspect the code. Only the built-in indicators that are not categorized are possible to edit and inspect code. The categories are following:

- **Trend** (Average Directional Movement, Bollinger Bands, Envelopes, Ichimoku Kinko Hyo, Moving Average, Parabolic SAR, Standard Deviation)
- **Oscillator** (Average True Range, Bears Power, Bulls Power, Commodity Channel Index, DeMarker, Force Index, MACD, Momentum, Moving Average of Oscillator, Relative Strength Index, Relative Vigor Index, Stochastic and Williams' Percent Range)
- **Volumes** (Accumulation/Distribution, Money Flow Index, On Balance Volume, Volumes)
- **Bill Williams** (Accelerator Oscillator, Alligator, Awesome Oscillator, Fractals, Gator Oscillator, Market Facilitation Index)
- **The others** - most important as MACD, OsMA (used in my strategies) or Zig Zag

You can also create a new category and use all of the built-in indicators as functions in your programms.

After installation, there is just a sample strategy (in MetaTrader called Expert Advisor), however, you can also create or delete strategy categories and off course develop new strategies.

Running strategies and indicators is done simply by attaching them to the chart. There is no portfolio trading.

2.5 Programming languages

All platforms use some kind of programming languages to code automated strategies. Some of them use standard programming languages, such as C#, while others use their own programming language.

2.5.1 C#

There are 2 versions of MultiCharts platform. Both offer the same features, the difference is in used programming language. The original version, that is still in use, uses so-called EasyLanguage. The version MultiCharts .NET 64 uses Microsoft's programming language C#. I tested only the .NET version, therefore, there will be no description of EasyLanguage, since C# is a standard programming language.

It is possible to use all features from C#, including features as for example delegates, exceptions, indexers or interfaces.

Off course, there are special libraries for trading written by MultiCharts developer.[17] Following list introduces some of the data types and functions used for studies and indicators programming:

- Classes inherited from class *IndicatorObject* are used for indicators. Only those can create graphical objects implenting interface *IPlotObject*. It has special methods:
  - *Create()* called once in the beginning after constructor.
  - *StartCalc()* called in the beginning to set study variables, can be called more than once in the object life cycle.
  - *CalcBar()* called for every new bar.
  - *Destroy()* called before destructor.
- Classes inherited from class *SignalObject* are used for creating signals. One or more signals run togehter are considered a strategy. Uses the same functions as described above.

2.5.2 NinjaScript

NinjaScript is just a renamed C# with special libraries. Developer in NinjaScript can use all libraries available for C#. The documentation of NinjaScript consists from these topics:

- **Alert & Debug** contains functions for printing output, making logs and alerts.

- **Data** contains predefined variables as in MetaTrader (i.e. arrays High, Low etc.) and functions for getting information about the market.

- **Drawing** contains interfaces and functions for drawing lines, Fibonacci retracements etc.

- **Indicator** has functions s and predefined variables for printing indicators.

- **Indicator Methods** contains predefined indicators (as MACD and others) as functions.

- **Strategy** contains functions for programming strategies, such as sending orders etc.

For the full documentation visit [18].

2.5.3 MQL 4

The MetaQutes language in version 4 is *an object-oriented high-level programming language intended for writing automated trading strategies, custom technical indicators for the analysis of various financial markets. It allows not only to write a variety of expert systems, designed to operate in real time, but also create their own graphical tools to help you make trade decisions*[19] used by MetaTrader 4.

The syntax of MQL 4 is the same as the syntax of C++ with a few exceptions. There is no address arithmetics, no goto operator, no anonymous enumaration can be declared and there is also no multiple inheritance. Since build 600 (I am currently using build 765), there are these basic data types: integers, bool, string (not as in C++, string is not treated as an object, but as an basic data type), floating point numbers, color, datetime and enumerations. There are also complex data types: structures and classes.

MQL 4 contains a relatively small group of built-in fucntions. Firstly, it contains common functions for printing, memory allocation and deallocation and sending mails and notifications. Secondly, it contains special functions for: work with arrays, conversions, math, string operations, account information, current symbol information, chart operations, trade operations.

There are no special objects for expert advisor or indicators, however, there are some special functions common for both, functions only for expert advisors (trade functions) and functions only for indicators (creating graphical objects). To the special functions belong:

- *OnInit()* is called immediately after start of expert advisor or indicator.

- *OnDeinit()* is called immediately before end of expert advisor or indicator.

- *OnTick()* is used only in expert advisors and it is called after every tick.

- *OnCalculate()* is used only in indicators and it is called everytime the indicator values have to be recounted, usually after a new tick.

2.6 Strategies backtesting

The very essential part of creating a strategy is its testing. Testing is executed upon historical data stored in platform. Strategies can also be optimized: the user selects ranges of some parameters and the platform finds out, which parameter settings is optimal. Also a forward testing can be done: strategy is optimized upon historical data from a certain period of time and then backtested upon another to avoid overoptimizing.

2.6.1 MultiCharts strategies backtesting

MultiCharts makes testing strategies in a different way than NinjaTrader and MetaTrader 4 do. Strategy testing is located in Portfolio Manager. For a chosen strategy componed from chosen signals and instruments a backtesting, optimazation and forward testing can be done.

After testing the user can see the portfolio performance report. This report can be exported as a .xcl file and contains Strategy Analysis (performance summary, ratios, equity curve), Trade Analysis (list of trades), Periodical Analysis (profits and losses in partucular days or certain periods).

Optimization offers two possibilities. The first, exhaustive optimization is a brute-force optimization known from the other platforms. Exhaustive optimization finds the absolutelly best solution, however, running lowspeed due to trying every possible combination of parameters. The second, genetic optimization, is much faster.

2.6.2 NinjaTrader strategies backtesting

NinjaTrader offers backtesting, brute-force optimization and also forward testing as MultiCharts. In contrary with MultiCharts, only one strategy upon one instrument can be done in one time.

After testing, a strategy output is made and contains:

- Summary containing important testing output, such as gross profit, gross loss, maximal drawdown etc.
- Chart displaying the price development and closed trades.
- Graphs displaying i.e. drawdowns, net profits for a certain period etc.
- Executions of trade orders.
- Trades with statistics for every signle trade.
- Periods showing profits, losses etc. for a chosen period.
- Settings of the used strategy.

2.6.3 MetaTrader 4 strategies backtesting

In MetaTrader for a user can do backtesting and optimizing on available historical data.

Testing can be done in more precision settings: by using just by opening prices, at control points base on lower timeframe or by every tick. The more precise method, the longer the testing takes.

There is also a possibility of testing in visual mode, when all the trades are displayd on the chart.

After testing an equity curve and report with strategy results displayed.

2.7 Chosen platform

The selection is made according to aspects described in previous sections. For every section there is chosen the winning platform and arguments for the selection.

- **Hardware demands: MetaTrader 4** has the lowest hardware demands from all 3 platforms. Since I do my thesis in virtualized Windows 7 64 bit, this fact is quite important. However, during testing of MultiCharts and NinjaTrader I expected no problems with freezing

- **Pricing policies: MetaTrader 4** is for free, unlike MultiCharts and NinjaTrader. There is also no possibility of getting one of those platforms for free, since AMP Futures, which offers a free licence for use with their brokerage, offers only futures and my trading system will trade Forex markets.

- **Supported brokers: MetaTrader 4** is supported by X-Trade Brokers. I chose this broker dispate the fact it does not have lowest spreads, however, there is no minimal deposit rule and also a lot of webinars, platform tips. Also all customer services are provided in Czech.

- **Data feeds and vizualization: MultiCharts** has the best selection of data feeders.

- **Indicators and automated strategies: MetaTrader 4** has all indicators sorted by categories and contains all indicators necessary for my trading systems.

- **Programming language: MultiCharts** signals and indicators are written in pure C#. For this language there is a good documentation and a lot of helpfull topics on StackOverflow.

- **Strategies backtesting: MetaTrader 4** has the visual mode of backtesting, where you can see all chosen indicators and executed trades, which is very helpfull and is in my opinion bigger advantage than portfolio testing of MultiCharts.

According to arguments served in the list above, I choose MetaTrader 4 as my trading platform.

Chapter 3

Introduction into technical analysis

In this chapter, an introduction into technical analysis is made. My strategies described in 4 are based only technical analysis, which is actually just a price action found in the charts.

3.1 Terms clarification

There is a lot of specific terms in trading. Therefore, I delivered a list of the most basic terms with their clarification:

- A **bid price** is also called buy price. In Forex, it is a price of the base currency expressed in the ammount of quote currency. For example, if a bid price of EUR/USD is 1.3653, you can buy one Euro for 1.3653 US Dollars.

- An **ask price**, also called sell price, is higher than bid price. In Forex markets ask price is a price expressed in quote currency, for which you can sell one unit of the base currency. For example, if the ask price of EUR/USD is 1.3655, you can sell one Euro fo 1.3655 dollars.

- A **lot** is the standardized and minimal unit of trade volume. In case of stocks it is the minimal number of stocks you can purchase. In terms of options, a lot represents the number of contracts contained in one derivative security.[20] In Forex, a lot corresponds to 100,000 units of base currency. There are also smaller units, a minilot (corresponds to 10,000 units of base currency) and a microlot (corresponds to 1,000 units of base currency).

- A **spread** is the difference between ask and bid price expressed in pips. Spread is actually fee for the broker and therefore it is set by broker.

Depending on the broker and type of account, spread can be fix or flexible. High liquidity bring lower spread and vice versa.

- A **leverage** is an instrument helping smaller traders to invest. It is a ratio, i.e. 100:1, between the money lent by the broker to the trader and trader's money. I.e, with ratio 100:1, the trader can trader trade with 100,000 currency units even though he has only 1,000 units available. To cover broker's risks, after entering a position, a **margin** is blocked on the traders ammount. If the loss reaches the margin, a **margin call** is made. The trader has to add more money to his account or accept the loss.

- A **pip** concerns only Forex and is the smallest possible price change. A **pip value** is a product of pip and lot.[21]

3.2 Candle stick chart

A candlestick chart is a chart depicting price of some symbol in a selected time frame (i.e. 1 minute, 1 hour or 1 month). It consists of so-called candlesticks. Each candlestick represents price development during the time period chosen by time frame. Each candle stick represents 4 significant values (therefore, it is sometimes called OHLC):

- **open** represents the open price (price in the beginning of the chosen time period)
- **high** represents the highest price reached during the chosen time period
- **low** represents the lowest price reached during the chosen time period
- **close** represents the close price (price at the end of the chosen time period)

There are 2 main types of candlesticks:

- a **bullish** candlestick, which open price is lower than the close price and represents increasing of the price. It is usually depicted with green or white body.
- a **bearish** candlestick, which open price is higher than the close price and represents decreasing of the price. The color of its body is usually red or black.

Each candlestick consists of following main parts:

- **upper shaddow**
- **body**

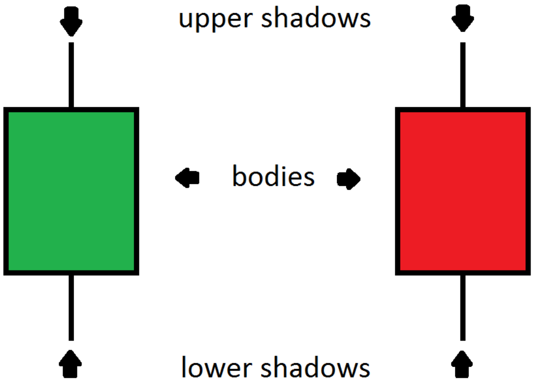

Figure 3.1: Bullish and bearish candlestick.

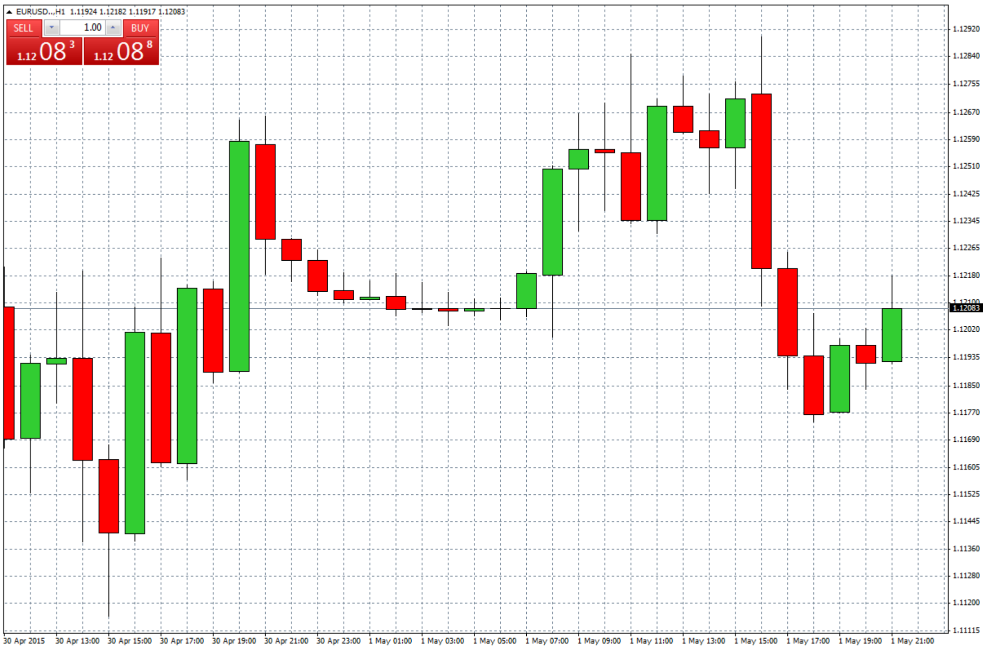

Figure 3.2: One hour time frame candlestick chart of EUR/USD

- **lower shaddow**

In case of a bullish candlestick the lowest price within the body represents the open price and the highest price within body represents the close price. In case of a bearish candlestick is the situation opposite. A candlestick with no upper shaddow has a shaven head, a candlestick with no lower shaddow has a shaven bottom.[22]

A candlestick chart does not give any evidence about the volume of trades, as well as the bar chart.

3.3 Candlestick patterns

This chapter concerns about candlestick reversal patterns. These patterns consist only from one or two significant candlesticks. Occurence of such a pattern may indicate a change of a trend, as well as it might not. In following subsections, the most important candlestick reversal patterns are described.[22]

3.3.0.1 Hammer and hanging man

A hammer may indicate a change in the direction of a bearish trend, as well as a hanging man may indicate the change of a bullish trend.3.3 Both of those patterns can be recognized thanks to these signs:

- There should be no or very short upper shadow.
- The lower shadow should be minimally twice as long as the real body.
- The body color does not matter.
- *The real body is at the upper end of the trading range.*[22]

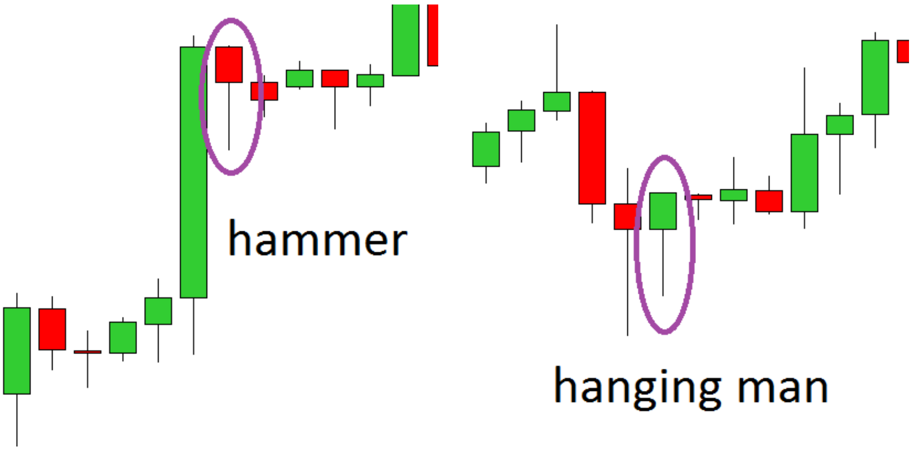

Figure 3.3: Hanging man and hammer candlestick patterns.

3.3.0.2 Engulfing

The engulfing, unlike hammer or hanging man, is a formation of 2 candlesticks. The figure on the left in 3.4 shows a bearish engulfing - the market is trending up and the bullish candlestick body is engulfed by the second bearish candlestick. The right figure shows the bullish engulfing, exactly opposite to the bearish.

The recognition of an engulfing has following steps:

- The market has a clear trend, up or down. It does not matter, whether the trend is only a short-term or long -term
- The body of the second candlestick engulfs the body of the first. Shadows do not have to be engulfed.
- The colors of candle sticks must be opposite of opposite color.
- The two colors rule mentioned above does not have to be obeyed, if the body body of the first candlestick is much smaller than the second one. After a long downtrend a tiny bullish body engulfed by a huge bullish body can announce a bottom reversal. Vice versa, after a long uptrend a tiny bearish body engulfed by a huge bearish body can signalize a top reversal.

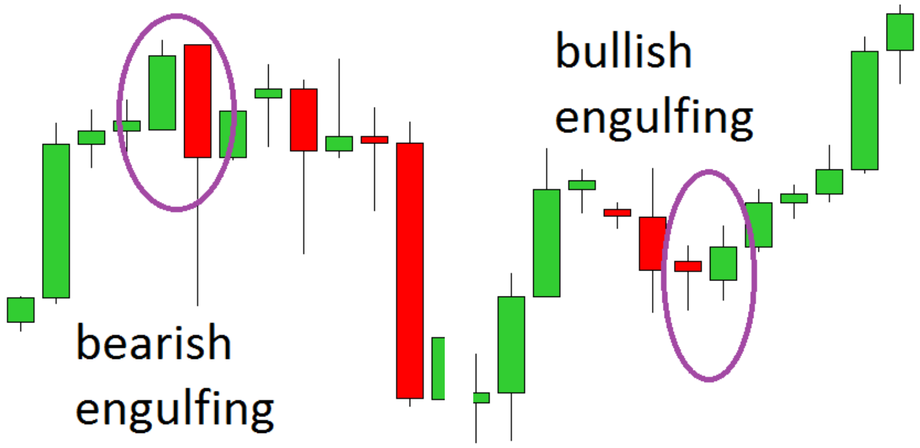

Figure 3.4: Bearish and bullish engulfing candlestick patterns.

3.3.0.3 Dark-cloud cover and piercing

The dark-cloud pattern consists of 2 candlesticks. The second one opens higher than the first high, however, it closes close to the low. The second candle stick close must be within the first body. Following 2 factors intensifies the importance of the pattern:

- *The greater the degree of penetration of the black real body's close into the prior white real body, the greater the chance for a top.*[22]
- If the second bearish body opens above a major resistance level and than falls down, it prooves the market will not end up in an uptrend.

The piercing pattern is an opposite pattern to dark-cloud cover. The importance of the pattern has the same factor as above mentioned dark-cloud cover, however, these factors are just upside down.

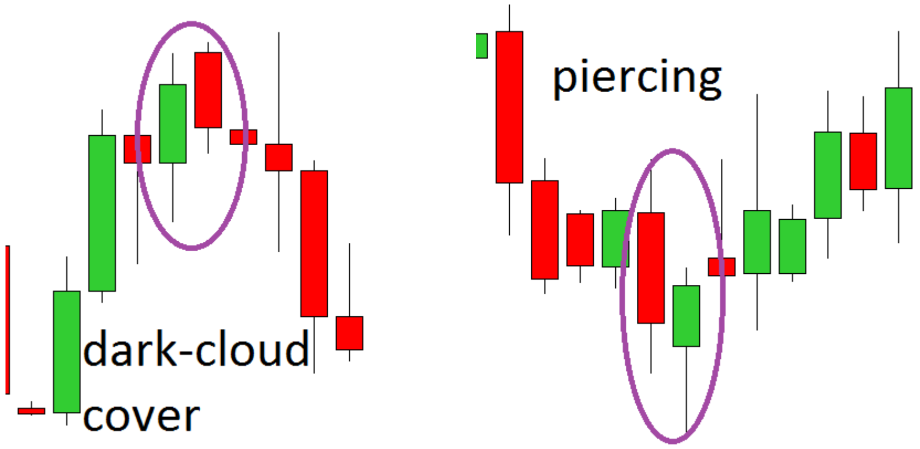

Figure 3.5: Dark-cloud cover and piercing candlestick patterns.

3.3.0.4 Stars

Stars are single candlestick patterns characterized by a small body. There are 4 types of stars:

- evening star
- morning star
- doji star
- shooting star

A **morning star** is a bottom reversal pattern. It comes after a bearish candlestick with bigger body, however, it is much lower (there is a vertical gap). It is followed by a bullish candle stick.

The opposite reversal pattern is an **evening star**. It comes after an uptrend, after a bullish candlestick with bigger body. There is also a gap between preceeding and following candlesticks. An evening star is followed by a bearish candlestick.

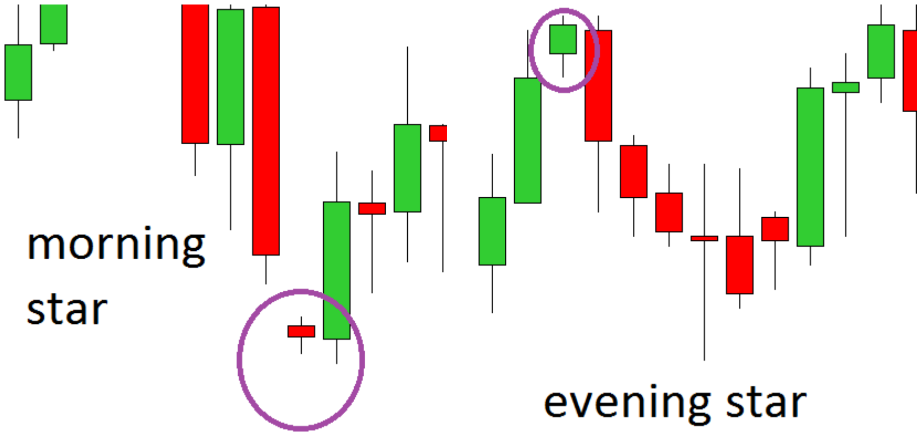

Figure 3.6: Morning and evening star candlestick patterns.

There are also morning and evening stars called **doji**. The vertical gaps between preceeding and proceeding candle sticks are usually higher. Doji star has nearly no body and usually indicates a change of a trend. An evening doji is placed left in the figure 3.7.

In the same figure on the right, there is also a **shooting star**. A shooting star is actually an inverted hammer. The color of the body is not important. This pattern is not as strong signal as an evening star.

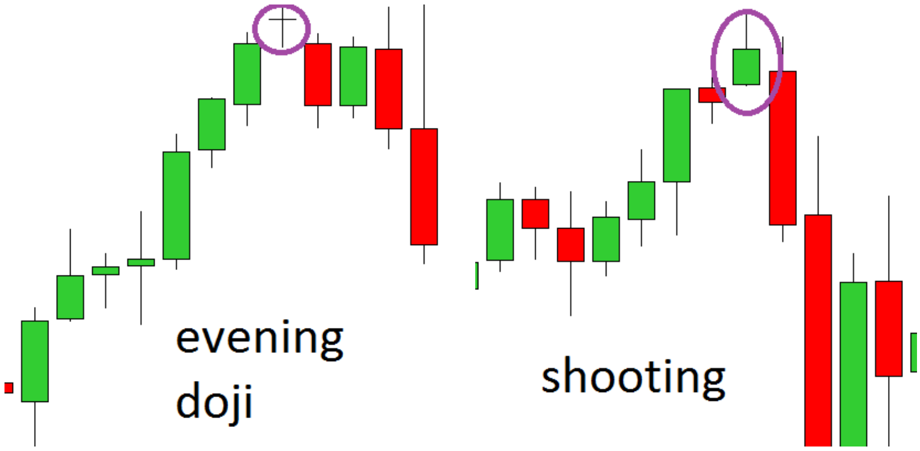

Figure 3.7: Evening doji star and shooting star candlestick patterns.

3.4 Supports and Resistances

Both, supports and resistances, are just price levels in the graph of a symbol. **Support** is a level, beneath which the most of the traders do not want to sell anymore. On the contrary, **resistance** is a level, above which most of the

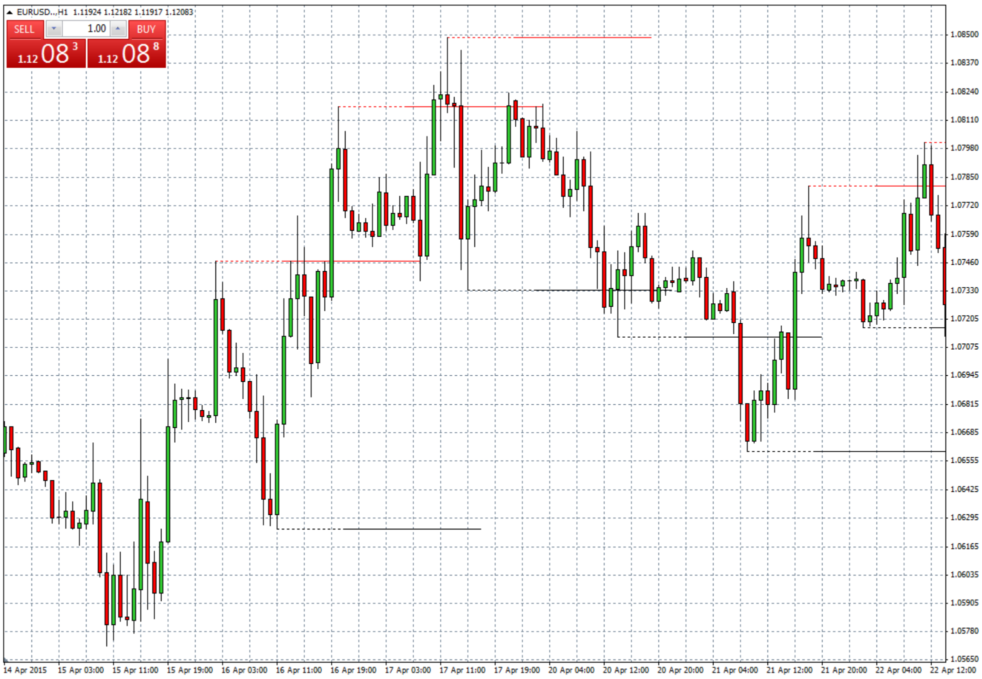

Figure 3.8: Support and resistances found by my custom indicator

traders does not want to buy. These levels are predicted according to the past prices on the graph (the price has allready touched the support or resistance level and the trend of the market changed in the past). [23]

When support or resistance level is reached, there are two possible ways of market development. Firstly, a breakout can happen and the trend of the market stays. Secondly, the trend can change from bearish to bullish and vice versa. The higher time frame, the more testifying such levels are. [24]

In the following subsections, some of ways leading to discovery of such S/R levels are described.

3.4.1 Basic recognition of S/R levels

The easiest way of finding S/R levels is to choose those peaks complying with a chosen criteria, namely the width of the peak (all candle sticks in the distance $\frac{n-1}{2}$ must be smaller than the peak, n is the width of the whole peak) and the depth of the peak.

S/R levels found by this method are displayed in figure 3.8 and are created by my custom indicator described in 4.2.1.

3.4.2 Fibonacci S/R levels

Fibonacci retracements are one of the technical analysis methods to find S/R levels. The method uses the Fibonacci sequence with the definition

$$f_0 = 0, f_1 = 1, f_{n+1} = f_{n-1} + f_n, \forall n \in N^+$$

which corresponds with the sequence

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots$$

The key Fibonacci ratios are derived from formula

$$F(n) = \left(\frac{1+\sqrt{5}}{2}\right)^{-n}, \forall n \in N$$

representing the approximation of the ratio between f_{j-n} and f_j for $j \in N$:

$$F(n) = \left(\frac{f_{j-n}}{f_j}\right)$$

The key Fibonacci ratios are $F(0) = 1$, $F(1) = 0.618$, $F(2) = 0.382$, $F(3) = 0.236$ and $\lim_{n\to\infty} F(n) = 0$. The 0.5 ratio is the ratio between the 3^{rd} and 4^{th} member of the sequence and *is used because of the overwhelming tendency for an asset to continue in a certain direction once it completes a 50 % retracement.*[25]

For an unclear reason, these ratios can predict a level, at which the price development direction changes.

3.4.2.1 Fibonacci retracement patterns

The idea of usage Fibonacci retracement patterns lies in the usage of a trendish market. In a trendish market (the trend can be confirmed by indicators, such as MA or Stochastic, see 3.6), WOLOG in a bullish one, the trader's aim is to find a low and high swing as depicted below in the figure 3.9, where you can see displayed Fibonacci levels.

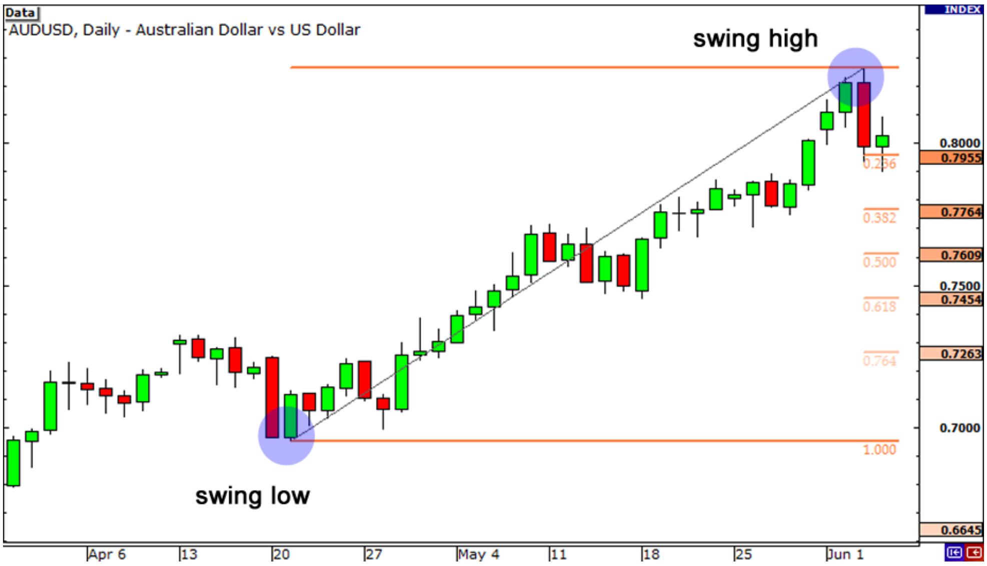

Figure 3.9: A low and high swing on a daily chart of AUD/USD[25]

It is expected, the price will retrace after reaching a Fibonacci level, therefore a trader should buy at Fibonacci's support and sell on Fibonacci's resistance. The theoreticle principle is depicted in figure 3.10 for both, a bullish and a bearish trend.

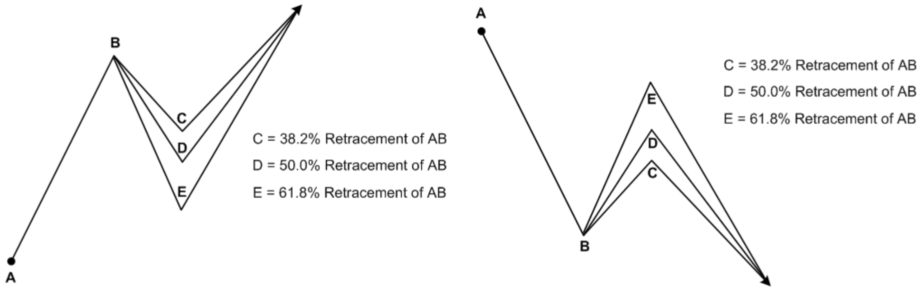
B
C
D
E
A
C = 38.2% Retracement of AB
D = 50.0% Retracement of AB
E = 61.8% Retracement of AB
A
C = 38.2% Retracement of AB
D = 50.0% Retracement of AB
E = 61.8% Retracement of AB
E
D
C
B

Figure 3.10: Representation of a retracement of an uptrend and a downtrend[25]

Coming back to example with a bullish trend, a further price development of AUD/USD confirmed Fibonacci retracement levels. As you can see in the figure 3.11, the price, after reaching a high swing, was rather decreasing, however, on Fibonacci's 0.382 level it started to increase again.

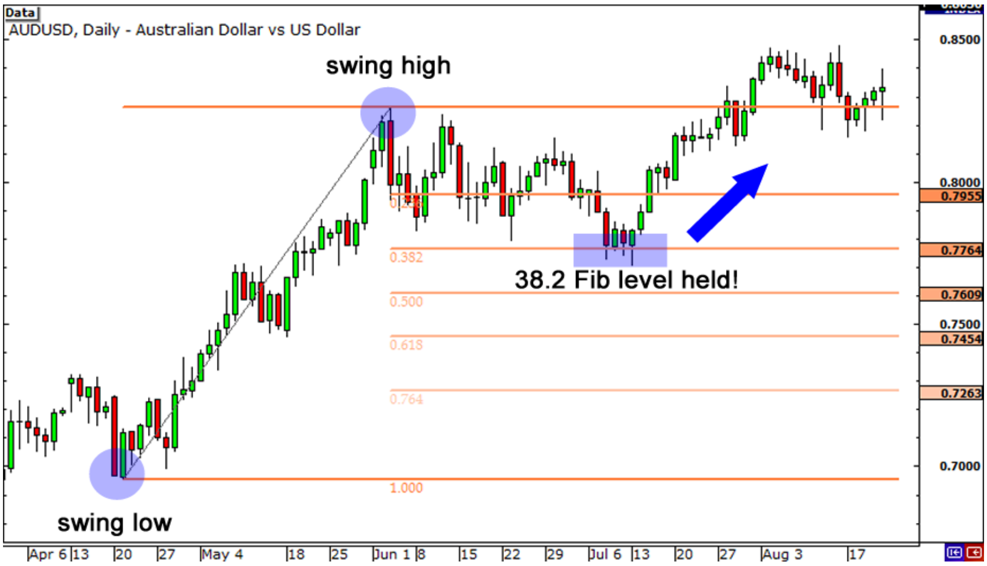

Figure 3.11: Further development of a daily price of AUD/USD confirming Fibonacci's retracement levels[25]

A figure displaying the same situation for a bearish trend would look the same as the figure for a bullish trend, just upside down, therefore I ommit it.

3.5 Types of orders

To understand, how different types of closing a trade are made, it is usefull to see the sequence of trade steps.

There are 3 parts involved in a trade: the program sends a command to the client's terminal (still on the user's computer). The terminal then sends an order to a server. In the figure 3.12 there are following steps of an order:

- t_0 - expert advisor is launched.
- t_1 - program forms a trade request and passes the control to the client's terminal.
- t_2 - terminal overtakes the control.
- t_3 - terminal evaluates the command. If the command is invalid (i.e. there is not enough equity on the account), the terminal returns control to the program.
- t_4 - program might receive control back, if the command was refused. The program gets this information at this point.
- t_5 - if the order request was correct, the server receives the request at this point.

- t_6 - server sends information, whether the request was accepted or rejected. Rejection can occur due to change of price, however, this depends on the type of order.

- t_7 - terminal receives the response from the server.

- t_8 - terminal has displayed the information about trade and now passes the control back to the program.

- t_9 - program overtakes the control back (if it did not so at the t_4).

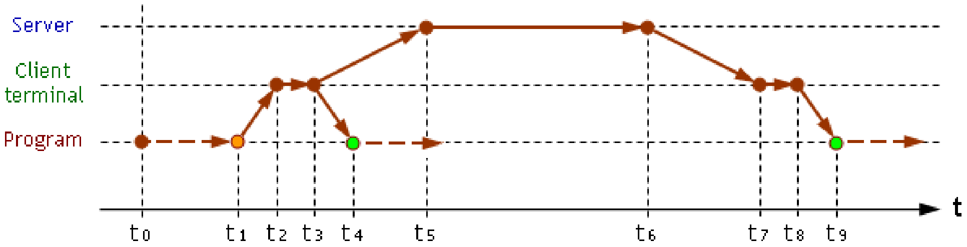

Figure 3.12: Time sequence in making a trade [26]

As seen in the description above, the price can change during sending of the order. Therefore there are different order types as described in following subsections.[27]

3.5.1 Market order

A market order is the most basic kind of a trade order. It instructs the broker to buy or sell at the best price available, in an ideal case to buy at bid price or to sell at ask price. However, due to volatility of the market the real price of the order can be far from the price intended.

3.5.2 Limit order

A limit order is used in situation, when a correct price is more important than the execution of the order. It tells the broker to execute the trade at price chosen or better, which means to buy at specified price or lower and to sell at specified price or higher. The trader must be on the correct side of the market to avoid bad execution of limit order, i.e. if a trader sends a limit order to buy a stock for $6,23, while its current price is $6,10, the order will be executed immidiately.

3.5.3 Stop order

A stop order is fulfilled only when a stop price level is hit. It works in an opposite direction than limit order. A buy stop level is above the current market price, the sell stop level is below the market price. Stop orders are mostly used to set stop loss.

3.5.4 Conditional order

Conditional orders must be placed before entering the trade. They are the most basic way of trading automation. Two most basic conditional order types are OCO (order cancels order) and OSO (order sends order).

An OCO order enables the trader to send more orders at one time. When one of sent orders is fulfilled, the others are cancelled. A good example can be a so-called bracket order: a stop order for stop loss, a limit order for take profit and an order to cancel one of them.

An OSO consits of a primary order, which can send one or more secondary orders, when some criteria are met.

3.6 Indicators

Indicator is a mathematical calculation based on price/volume development helping in predicting future prices. In the subsections below the most important indicators are described.

3.6.1 Moving average

The aim of moving average indicator is to filter out noises from a random price fluctuations. It is mostly used to identify the trend direction. The method of moving average generates a series of avarages from a moving intervals.

A MA can be one-sided or both-sided. The both-sided moving average uses values from both sides (preceeding and following values), the one-sided only preceeding values.

There are more methods of counting moving average, from which the two most common are SMA (simple moving average) and EMA (exponential moving average).

In case of SMA all values are of the same importance. For both-sided SMA the formula is

$$\overline{y_t} = \frac{1}{2m+1} \sum_{i=-m}^{m} y_{t-i}$$

and for one-sided

$$\overline{y_t} = \frac{1}{2m+1} \sum_{i=0}^{2m} y_{t-i}$$

where m is a parameter chosen by the user.

An EMA the nearest values have the highest importance. The formula for counting EMA is

$$\overline{y_t} = \sum_{i=0}^{t-1} w_i y_{t-i}$$

where

$$w_i = \alpha(1-\alpha)^i$$

3.6.2 OsMA

OsMA (oscillator of moving average) belongs to the group of indicators called oscillators, which oscillate around 0. Basically, OsMA shows the momentum of price action.

Basically, OsMA is nothing else than a difference between long term moving average and a short term average.

OsMA indicator is used in strategies based on divergences described in **??**

Chapter 4

Strategies design

There are numerous strategies, some of them more profitable, some of them suitable just for a very few markets and some not profitable at all.

4.1 Requirements

The requirements for my automated trading strategies are in the following categories:

- **Platform and programming language:** As my selected platform is MetaTrader 4, the strategies will be written in MQL4.
- **Broker:** X-Trade Brokers were selected as the broker supplying data, upon all the strategies will be tested.
- **Markets:** Strategies will be written for Forex markets, namely pairs GBP/USD and EUR/USD.
- **Core of strategies:** Strategies will have to be based on S/R levels of technical analysis.
- **Parameters:** Strategies will have to have at least two parameters, so they can be optimized.
- **Testing and optimizing:** Strategies will be tested and optimized upon historical data in 5 minutes timeframe. Firstly, they will be optimized by selecting and optimizing at least two parameters upon data from the 1st of January 2013 to the 31st December 2013. Afterwards, they will be test upon the data from the 1st of January to the 31st December 2014.

4.2 Indicators

Technical indicators are mathematical calculation based on price/volume development helping in predicting future prices. Therefore, additional custom indicators necessary for implementation of chosen strategies are described in next subsections. All parameters are named exactly as in source files of indicators.

4.2.1 Indicator of S/R levels based on swings

There are more ways of looking for S/R levels. One of them mentioned in 3.4.1 is to find tops (in case of resistances) or bottoms (in case of supports) of peaks. I will implement an indicator displaying S/R levels, as seen in figure 4.1 with following parameters:

- **MountainWidth** is a number of candlesticks on both sides, that must be lower (in case of resistances) or higher (in case of supports) than the peak.
- **CandleSticksNumber** is a number of candlesticks in the past taken into account.
- **Delta** is the width of one side of S/R price level neighborhood. Candlesticks with one part inside of this neigborhood and the other part outside are considered to be touching the S/R line, candlesticks with one part outside of the neighborhood and the other part on the other side of this neighborhood are considered crossing the S/R price level.
- **MaxCrosses** is the maximal number of crosses allowed for a valid S/R price level.
- **MinTouches** is the minimal number of touches making the S/R price level valid.

The indicator displays the S/R level as a line. When the S/R line is still not valid, the line is dotted. In the moment the S/R level becomes valid, the line changes to solid. Supports are colored red, resistances black. For every S/R level, a number describing number of touches and number of crosses of the S/R level is displayed.

4.2.2 Indicator of neighborhoods of S/R levels

In order to help vizualizing the strategy of price correction in the neighborhood of S/R level, I will implement an indicator displaying neighborhoods of a S/R level nearest to the current price. It will display inner neighborhood (colored dark orange lines in figure 4.2), outer neighborhood (colored blue), line connecting averages of high and low prices (green) and checks in the places

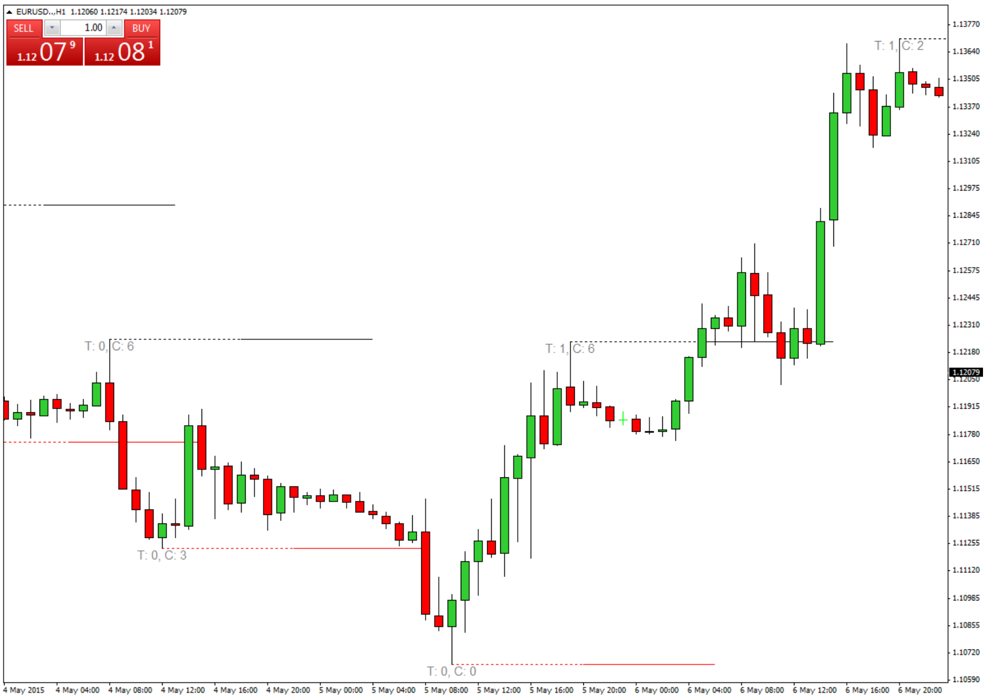

Figure 4.1: Indicator of S/R levels based on swings

of candlesticks (red). It will also display S/R levels. The indicator will have the same parameters as 4.2.1 and additionally:

- **Delta1** is the one side width of the inner S/R line neighborhood in pips.
- **Delta2** is the one side width of the outer S/R line neighborhood in pips.

4.3 Strategies

All my strategies will be based, according to the requirements, on S/R level recognition. I will implement altogether 4 strategies by combining 2 strategies of trade entering and 2 strategies of trade exiting.

There will be 2 ways of recognizing a possibility of entering either a short or a long position, price correction in the neighborhood of S/R level and pattern recognition in the neighborhood of S/R level. All strategies will entered by market order (see also 3.5).

Also there will be 2 ways of exiting a trade: a fix stop loss and take profit and trailing stop.

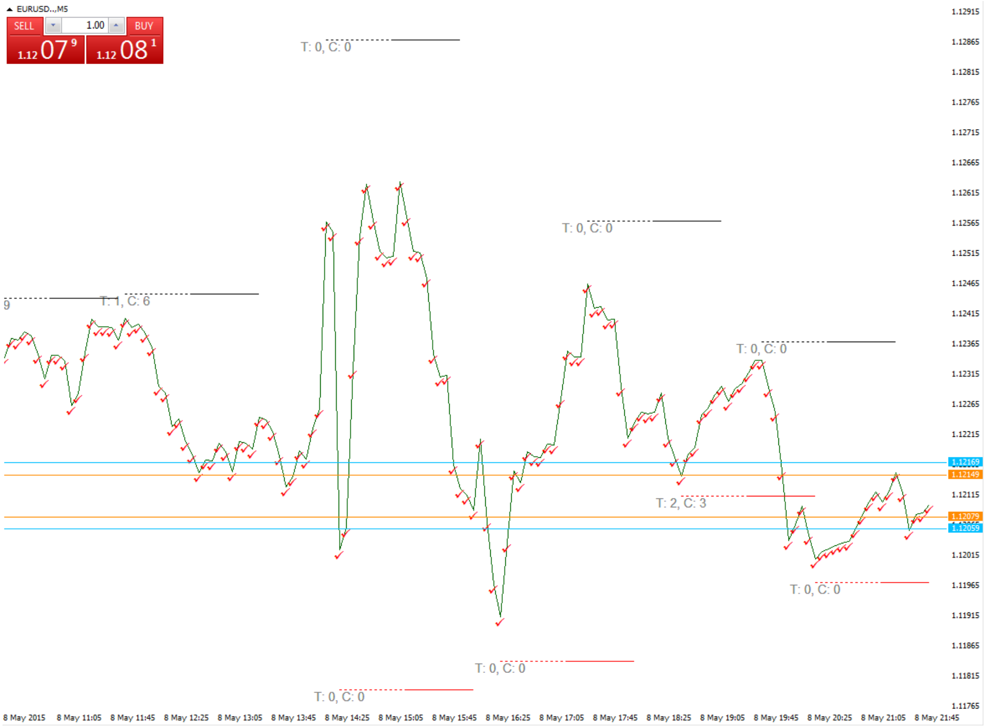

Figure 4.2: Indicator of neighborhoods of S/R levels

4.3.1 Exiting of strategies

Once a position is opened, it necessarily has to be correctly exited. There are more ways of strategy exiting, from which I will use 2.

The first is exiting through firmly set take profit and stop loss price levels. If the market trend goes in the wrong direction (it decreases when a trader is in a buy position or increases, when trader is in sell position) and the stop loss level is hit, order is automatically closed in order to prevent higher losses. If the market trend goes in the right direction, the order is closed, when the take profit level is hit. This strategy of exiting is the simpliest to program, however, in case of a long trend this strategy takes just a small part of the potential profit.

The second uses a trailing stop. After entering a trade, only a stop loss is set in some distance. If the trend goes in the wrong direction and the stop loss is hit, the position is closed with loss. In an opposite case, if a difference between stop loss and current price is higher than set, the stop loss is moved towards the current price and therefore goes with trend. If a trend changes after longer period, the stop loss is hit.

There are following parameters used by both strategies of trade exiting:

- **Lots** is a volume of the trade in lots.

- The volume of the trade is equal to product of account free margin and **MaximumRisk** divided by 1000.
- If there are more losses in row, the volume of the transaction is reduced by the product of trade volume and the number of losses in row diveded by the **DecreaseFactor**.
- **Slippage** is the maximum difference in pips between order price and the market price.

Parameter **Stop Loss** and **TakeProfit** are used just by fixed stop loss and take profit strategy and represent the value of stop loss and take profit in pips. **TrailingStop** is used by trailing stop exit strategy and represents the minimal distance between stop loss and current price in pips.

4.3.2 Price correction in the neighborhood of S/R level

The first two strategies are based on a reversal pattern of price correction in the neighborhood of a S/R level and price reflection.

In the figure 4.3 there is displayed the price mirroring. On the left, a touch of S/R level after a bullish trend is depicted (the price does not exceed the upper border of inner neighborhood of the S/R level). The strategy counts with an idea, that after reaching the lower border of outer neighborhood the trend becomes bearish. On the right side of the figure a simmilar situation is depicted for a bearish trend.

Another possibility of price development after touching the S/R level is depicted in the figure 4.4. On the left side, the price after touching the S/R level reaches a level beneath the lower border of inner neighborhood of the S/R level, however, it remains above the lower level of outer neighborhood and then again increases. If the price increases above the higher border of outer neighborhood, a buy order is send. On the right side of the figure, there is depicted the same idea for an initially bearish trend.

The strategy can be widely parametrized. Along with parameters descripted in 4.2.1 and 4.2.2, there are parameters (the candlesticks are indexed from right to left, so the newest candlestick has index 0):

- **MinP** is the minimal number of candlesticks, that are trendish in the beginning, so in case of a bullish trend, the price with index i is higher than price with index $i + 1$. Vice versa, in case of a bearish trend, the price with index i is lower than the price with index $i+1$. This parameter is used for both situations, price reflection as well as price correction.
- **MinO** and **MaxO** represent the minimal and maximal number of candlesticks, that must be within the inner neighborhood of the S/R level after the period defined by **MinP**. This parameter is used also in both situations. If a price reflection occurs (the next candlestick is beneath

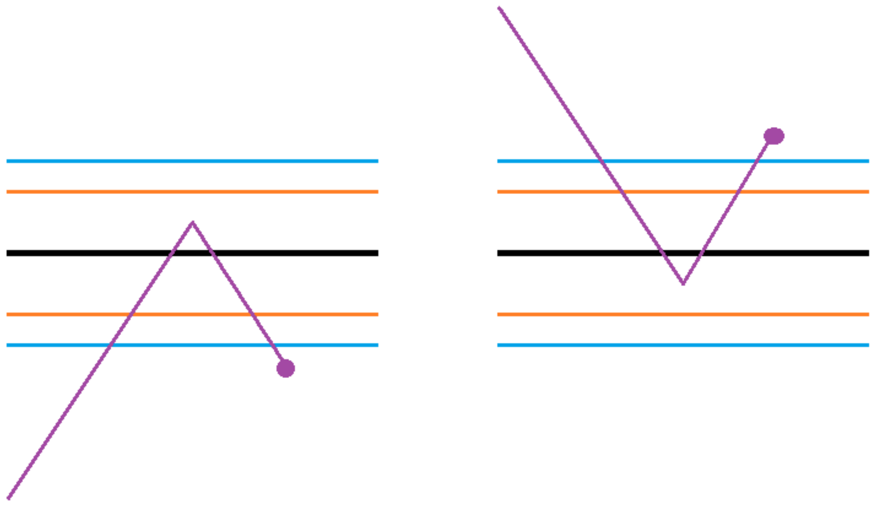

Figure 4.3: Price mirroring

or above the outer neighborhood of the S/R level), the next parameters are ommited and trade entered.

- **MinN** and **MaxN** are the minimal and maximal counts of candlesticks comming after the period defined by **MinO** and **MaxO** that must be between the lower border of inner neighborhood and the lower border of outer neighborhood or between the higher border of inner neighborhood and the higher border of outer neighborhood. These parameters are used just in case of price correction.
- **MinM** and **MaxM** represent the minimal and maximal counts of candlesticks comming after the period defined by **MinN** and **MaxN** that must be within the inner neighborhood of the S/R level. These parameters are used just in case of price correction. The next price has to be outside of the outer neighborhood in order to enter a trade.

4.3.3 Pattern recognition in the neighborhood of S/R level

As mentioned in 3.3, there are candlestick patterns hanging man and hammer that may indicate a change of trend direction. The strategy of pattern recognition in the of S/R level enters a trade, when the first closed candlestick

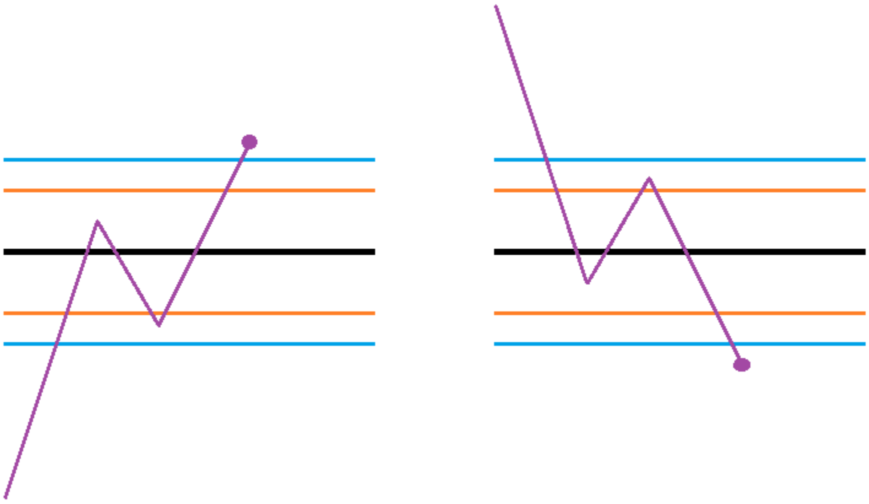

Figure 4.4: Price correction

complies the pattern and the current price is in selected neighborhood of a valid S/R level. A buy position is opened, when a hammer appears after a bearish trend, a sell position is opened, when a hanging man appears after a bullish trend.

The strategy will have the same parameters as 4.2.1 and additionally:

- **MaxUpperShadowBodyRatio** indicates the maximal ratio between the upper shaddow size (which should be as small as possible) and the body size.
- **MinLowerShadowBodyRatio** indicates the minimal ratio between the lower shaddow size and the body size. According to [22], the lower shaddow size should be minimally twice larger than the body size.

Chapter 5

Strategies implementation

As the preferred platform for the implementation is MetaTrader 4, the MQL language is used. In next sections, the logic of MQL programs, implementation idea, implentation of trade entering and exiting and implementation of particular classes are described.

5.1 Logic of MQL

In MQL, there are .mqh source code files, that can be included in any other file, and special files for expert advisors and indicators.

EA and indicator files have some common attributes. Namely the possibility to add input parameters and special event handling functions. Common special event handling functions for EA and indicator files used in my strategies are *OnInit()* and *OnDeinit()*. Their function is simmilar to constructors and destructors. *OnInit()* is called immediately after launching the strategy or indicator to initialize all necessary objects and variables and *OnDeinit()* is called immidiately before the end.

Special function *OnTick()* is typical only for expert advisors and it is called immediately after a new chart tick. Special function *OnCalculate()* is used only in indicators and it is called everytime a the indicator values have to be counted again, usually after a new tick.

There are also trade functions, that can be called only from expertadvisors, from the most important namely *OrderSend()* for sending a new order and *OrderModify()* for the modification of the order. Functions for creating and modifying of graphical objects are logically available only in indicators.

5.2 Implementation idea

As MQL is an object-oriented language, I decided to implement classes, that can be reused in both, indicators and expert advisors.

I will implement include files in following categories:

- *Common* contains classes substituting missing standard library in MQL, namely *SmartArray* and *List* and also basic class *Object* enabling the instances of inherited classes to be put into *SmartArray* and *List*, since MQL4 does not support class templates .
- *Enums* with enumerations.
- *GraphicalObjects* contains classes for displaying various graphical objects, that are possible to put into or removed from a container.
- *IndicatorAux* contains classes, that are used directly in indicators and expert advisors. These objects are described in the next sections, as they are the most important part of my implementation.

I also implemented a file with common functions, that are used directly in indicators and expert advisors outside of objects.

5.3 Implementation of trade entering and exiting

On every chart tick in *OnTick()* event handler a function *countOpenedOrders()* is called. It returns number of opened orders. If there is an order, no new order is send. If 0 is returned, *CheckForOpen()* is called. This function receives from one of particular objects described below a signal - to buy, to sell or to make no order.

As described in 4.3.1, there are two type of exiting. If the position exit is made as fix stop loss and take profit, values of them are counted as

$$stopLossPrice = AskPrice - StopLoss \cdot PipSize$$

in case of a buy operation and

$$takeProfitPrice = AskPrice + TakeProfit \cdot PipSize$$

in case of a sell operation. **TakeProfit** and **StopLoss** are input parameters. Then a *sendOrder()* function is called and order done and the position is automatically closed when it reaches stop loss or take profit.

In case of closing the position by trailing, stop loss is not counted and *sendOrder()* is send without a take profit, stop loss is set the same as before. On every tick, if there is an opened position, function *modifyTrailingStop()* checks the current price. If the price exceeds stop loss, the position is closed with a loss, however, in case the stop loss is nearer to the current price than necessary (set by the parameter **TrailingStop**), *OrderModify()* is called and stop loss moved further in the direction of trend.

5.4 Implementation of particular classes

As mentioned before, the idea is to use particular classes, that are reusable in indicators and expert advisors. In next subsections, usage and public functions of these classes are described, as well as auxiliary classes used in them.

5.4.1 Swinger

Swinger is a class for finding and displaying of S/R levels. It is used in both indicators and all strategies.

It contains 3 auxiliary classes: *SwingerPeakFinder* for finding and filtering peaks of S/R levels, *SwingerSettings* for storing the settings and *SwingerSRLevel*, which represents a S/R level and contains index of the candlestick, where S/R level is, information whether it is support or resistance, number of level crosses and touches.

Swinger has following public functions, all return void and have no parameters:

- *void clear()* erases all saved peaks, as well as all displayed S/R indicator lines.
- *void searchPeaks()* finds all price peaks in the chart and stores them in a *SmartArray* as instances of *SwingerSRLevel*.
- *void filterPeaks()* filters found peaks according to parameters stored in *SwingerSettings*.
- *void print()* prints all S/R levels in a text form into terminal.
- *void display()* displays all S/R levels stored.

5.4.2 Patterner

Patterner serves for finding candlestick patterns in the chart, namely hammer and hanging man. It contains one auxiliry class *PatternerSettings* with all necessary parameters.

Patterner has only one public method *isTherePattern (int index)*. It returns *PATTERN_ENUM* with the kind of pattern on the candlestick with given *index* - hammer, hanging man or no pattern.

5.4.3 OneTwoThree

OneTwoThree class used in strategy Price correction in the neighborhood of S/R levels (see 4.3.2) and Indicator of neighborhoods of S/R levels (see 4.2.2). It contains only one auxiliary class *OneTwoThreeSettings* with parameter settings. It is class inherited from *Swinger* and has all its functionality, in addition, *OneTwoThree* has following public methods:

- *bool isUptrend()* returns true, if the candlesticks before the first candlestick with index 0 form an uptrend, false otherwise.
- *bool isDowntrend()* returns true, if the candlesticks before the first candlestick with index 0 form a downtrend, false otherwise.
- *void clear()* clears all graphical objects in the chart and also removes all stored S/R levels.
- *void displayNearestDeltas()* displays all indicator lines as displayed in the figure 4.2.
- *TRADE_TO_MAKE tradeToMake()* return kind of position, that should be opened - buy, sell or none.
- *bool isPriceInSrLevel()* returns true, if the current price is inside of the S/R level neighborhood.

Chapter 6

Strategies performance

.

There are alltogether 4 strategies, that will be tested, namely price correction in the neighborhood of S/R level with fix stop loss and take profit (named PC Fixed in the tables, file *OneTwoThreeFix*), price correction in the neighborhood of S/R level with trailing stop (named PC Flexible in the tables, file *OneTwoThreeFlexible*), pattern recognition in the neighborhood of S/R level with fix stop loss and take profit (named P Fixed in the tables, file *PatternsFix*) and pattern recognition with trailing stop (named P Flexible in the tables, file *PatternsFlexible*).

6.1 Compliance with requirements

The strategies comply with the requirements set in 4.1. They are made for the MetaTrader platform, tested on an account from X-Trade Brokers on GBP/USD and EUR/USD markets with spread 2 pips and 10,000 USD initial deposit. All 4 strategies are built upon S/R levels and have more than 2 parameters to be optimized through.

6.2 Optimizing

In the next subsections, optimization of strategies with selected optimized parameters is described. All optimisations were done on the EUR/USD market with spread 2 and timeframe 5 minutes for the whole year 2013.

6.2.1 Price correction in the neighborhood of S/R level with fix stop loss and take profit

The strategy was optimized with following firmly set parameters (to see the parameters meaning, please see 4.3.2): CountedCandleSticks: 150, MountainWidth: 8, Delta: 5, MinM: 3, MaxM: 6, MinN: 3, MaxN: 6, MinO: 3, MaxO:

Profit ($)	**Trades**	**Delta1**	**Delta2**
109.55	86	9	14
16.88	108	10	14

Figure 6.1: Price correction in the neighborhood of S/R level with fix stop loss and take profit optimization best and worst result, tested on EUR/USD for the whole year 2013

Profit ($)	**Trades**	**Delta1**	**Delta2**
698.87	91	9	13
18.88	91	8	11

Figure 6.2: Price correction in the neighborhood of S/R level with trailing stop optimization best and worst result, tested on EUR/USD for the whole year 2013

6, MinP: 5, MaxCrosses: 3, MinTouches: 0, StopLoss: 150, TakeProfit: 150, Lots: 0.1, MaximumRisk: 0.02, DecreaseFactor: 0, Slippage: 3.

The optimized parameters were **Delta1** with values from 7 to 13 with step 1 and **Delta2** with values from 10 to 15 with step 1. The best profit and the worst profit can be seen in table 6.1.

6.2.2 Price correction in the neighborhood of S/R level with trailing stop

The strategy was optimized with the same firmly set parameters with same values as the previous, except for parameters TakeProfit and StopLoss. In addition, the extra parameter TrailingStop was set to 100 pips.

The optimized parameters had the same range and values as by previous strategy. Results can be seen in the table 6.2.

6.2.3 Pattern recognition in the neighborhood of S/R level with fix stop loss and take profit

The strategy was optimized with following firmly set parameters (to see the parameters meaning, please see 4.3.3): CountedCandleSticks: 150, MountainWidth: 8, Delta: 7, MaxCrosses: 3, MinTouches: 0, StopLoss: 105, TakeProfit: 8, Delta: 7, Lots: 0.1, MaximumRisk: 0.02, DecreaseFactor: 2 and Slippage: 3.

The optimized parameters were **MaxUpperShadowBodyRatio** (labeled as MUSBR) with values from 0.1 to 0.5 with a step 0.1 and **MinLowerShadowBodyRatio** (labeled as MLSBR) with values from 1.5 to 2 with step 0.1. The best profit and the worst profit can be seen in table 6.3. There was no linear relation between parameter values and profit.

Profit ($)	Trades	MUSBR	MLSBR
386.20	36	0.4	1.6
135.70	35	0.2	1.5

Figure 6.3: Pattern recognition in the neighborhood of S/R level with fix stop loss and take profit optimization best and worst result, tested on EUR/USD for the whole year 2013

Profit ($)	Trades	MUSBR	MLSBR
214.18	39	0.5	1.6
3.70	35	0.2	1.5

Figure 6.4: Pattern recognition in the neighborhood of S/R level with trailing stop optimization best and worst result, tested on EUR/USD for the whole year 2013

Indicator	PC Fixed	PC Flexible	P Fixed	P Flexible
Initial deposit	10,000	10,000	10,000	10,000
Total net profit	-31.02	267.05	-224.23	-376.66
Gross profit	875.34	1,129.96	834.02	590.61
Gross loss	-907.06	-862.90	-1058.25	-967.26

Figure 6.5: Strategies performance on EUR/USD for the year 2014

6.2.4 Pattern recognition in the neighborhood of S/R level with trailing stop

The strategy was optimized with the same firmly set parameters with same values as the previous, except for parameters TakeProfit and StopLoss. In addition, the extra parameter TrailingStop was set to 100 pips.

The optimized parameters were also **MaxUpperShadowBodyRatio** (labeled as MUSBR) with values from 0.1 to 0.5 with a step 0.1 and **MinLowerShadowBodyRatio** (labeled as MLSBR) with values from 1.5 to 2 with step 0.1. In the table 6.4 a best and worst profit can be seen. There was also no linear relation between parameter values and profit.

6.3 Forward test

The results from optimization were used as input into forward test. The strategies were launched with the best settings according to optimization, however, upon data from the whole year 2014. They were launched on market EUR/USD, upon which the optimazation was done. They were also launched on GBP/USD to see the difference. The results can be seen in tables 6.5 and 6.6

Indicator	**PC Fixed**	**PC Flexible**	**P Fixed**	**P Flexible**
Initial deposit	10,000	10,000	10,000	10,000
Total net profit	-93.10	-210.52	1.19	86.35
Gross profit	610.11	538.65	946.89	999.37
Gross loss	-703.21	-749.17	-945.70	-913.30

Figure 6.6: Strategies performance on GBP/USD for the year 2014

6.4 Results interpretation

As seen from the results of forward testing, there is a large risk of overoptimalization - chosing optimized parameters from the past does not automatically mean success in the future.

Despite the fact the strategies meet the criteria set in 4.1, they can not be labelled as succesful. The problem might be in underestimating nuances in the market, insufficient design or in fact a good trading algorithm must be much more complicated.

Conclusion

In the first part of my thesis I succeeded to introduce history and basic principles of the markets, as well as to describe automated trading systems, choose a suitable platform for creating trading strategies and deliver the necessary introduction into technical analysis.

In the second part I designed and implemented 4 trading strategies based on S/R levels of technical analysis, which were tested upon 2 Forex markets.

The results of the tests showed, that implementing of a trading strategy is not a trivial issue. It demands good knowledge of the markets, trading platform, technical analysis. Trading is a vast field and understanding it into details requires many years of experience.

I see the contribution of my thesis in fact the reader can gain a compact basic awareness about trading, technical analysis and trading strategies implementation.

Bibliography

[1] RSJ a.s. Výroční zpráva 2013. [online], 3 2014. Avilable from: http://www.rsj.com/documents/vyrocni-zprava-2013.pdf

[2] Zelina I. Forex - History and Current Development. [online], 4 2014, Available from: https://is.bivs.cz/th/16083/bisk_b/Igor_Zelina_BP-_final.pdf

[3] Petram, L. O. The world's first stock exchange: how the Amsterdam market for Dutch East India Company shares became a modern securities market, 1602-1700. [online], 2011. Available from: http://dare.uva.nl/document/2/85961

[4] Intercontinental Exchange, Inc. New York Stock Exchange History. [online], 2015, [Cited 2015-05-02]. Available from: http://www1.nyse.com/about/history/timeline_events.html

[5] Vozárová P. Akciové a devizové trhy. [online], 3 2014. Available from: https://edux.fit.cvut.cz/archive/B132/BI-FTR/_media/lectures/prednaska3.pdf

[6] CME Group Inc. An introduction to futures and options. [online], 2006. Available from: http://www.cmegroup.com/files/intro_fut_opt.pdf

[7] Hull, *J. C. Option, Futures and Other Derivatives*, volume 7. Pearson Prentice Hall, 2009, ISBN 978-0132777421, page 6

[8] FindTheBest.com, Inc. Best Trading Software - Compare Features & Reviews. [online], 2015, [Cited 2015-04-25]. Available from: http://trading.softwareinsider.com/

[9] MultiCharts LLC. Tech Specs. [online], 2015, [Cited 2015-04-28]. Available from http://www.multicharts.com/net/tech-specs/

[10] FXDirectDealer, LLC FXDD MetaTrader 4 Forex Trading Software Platform. [online], 2015, [Cited 2015-03-14]. Available from: `http://www.fxdd.com/us/en/platforms/standard-spreads/metatrader-4/`

[11] NinjaTrader TM, LLC NinjaTrader Charting Software & Trading Platform: Purchase Options. [online], 2015, [Cited 2015-03-23]. Available from: `http://www.ninjatrader.com/purchase.php`

[12] MultiCharts, LLC Purchase MultiCharts. [online], 2015, [Cited 2015-03-23]. Available from: `http://www.multicharts.com/net/purchase/`

[13] AMP Futures Trading MultiCharts Special Edition - AMP. [online], 2015, [Cited 2015-04-10]. Available from: `http://www.ampfutures.com/platforms/ampmc-net-free/`

[14] MultiCharts, LLC Supported Data Feeds. [online], 2015, [Cited 2015-04-25]. Available from: `http://www.multicharts.com/market-data-feeds-net/`

[15] NinjaTrader TM, LLC, Futures Broker: NinjaTrader Online Futures Brokerage. [online], 2015, [Cited 2015-04-24]. Available from: `http://www.ninjatrader.com/Brokerage.php`

[16] MultiCharts, LLC Technical Analysis and Technical Analysis Software. [online], 2015, [Cited 2015-04-25]. Available from: `http://www.multicharts.com/technical-analysis/`

[17] MultiCharts, LLC MultiCharts .NET Programming Guide. [online], 2015, [Cited 2015-04-25]. Available from: `multichartscsharp`

[18] NinjaTrader TM, LLC NinjaTrader Version 7. [online], 2014, [Cited 2015-05-02]. Available from: `http://www.ninjatrader.com/support/helpGuides/nt7/index.html?welcome_to_productname_version.htm`

[19] MetaQuotes Software Corp. MQL4 Reference. [online], 2015, [Cited 2015-04-25]. Available from: `http://docs.mql4.com/`

[20] Investopedia, LLC. Lot Definition. [online], 2014, [Cited 2015-04-30]. Available from: `http://www.investopedia.com/terms/l/lot.asp`

[21] Baloga, M. Návrh a optimalizace investiční strategie na forexu s implementací zásad money managementu. [online], 5 2010. Available from: `http://digilib.k.utb.cz/bitstream/handle/10563/11443/baloga_2010_dp.pdf`

[22] Nisson, S. *Japanese Candlestick Charting Techniques.* New York Institute of Finance, 1991, ISBN 0-13-931650-7.

[23] Thomsett, M. C. *Support et Resistance Simplified.* Marketplace Books, 2003, ISBN 978-1592800674, pages 17 - 19.

[24] Centrum finančního vzdělávání, s.r.o. 12. Support a resistance. [online], 12 2004, [Cited 2015-03-21]. Available from: `http://www.financnik.cz/komodity/manual/komodity-support-resistance.html`

[25] Gaucan, V. How to use Fibonacci retracement to predict forex market *Journal of Knowledge Management, Economics and Information Technology*, 2 2011. Available from: `http://www.scientificpapers.org/wp-content/files/1134_How_to_use_Fibonacci_retracement_to_predict_forex_market.pdf`

[26] MetaQuotes Software Corp. MQL4 Tutorial. [online], 2015, [Cited 2015-05-02]. Available from: `http://book.mql4.com/`

[27] Investopedia, LLC. Introduction To Order Types. [online], 2015, [Cited 2015-05-02]. Available from: `http://www.investopedia.com/university/intro-to-order-types/`

Appendix A

Acronyms

NYSE New York Stock Exchange

LSE London Stock Exchange

OTC Over The Counter

CFD Contract For Difference

ETF Exchange-Traded Funds

WOLOG Without Loss of Generality

ATS Automated Trading System

EA Expert Advisor

EULA End-User License Agreement

MQL MetaQuotes Language

S/R Support and Resistance

OsMA Oscillator Of Moving Avarage

MACD Moving Average Convergence/Divergence

MA Moving Average

OCO Order Cancels Order

OSO Order Sends Order

SMA Simple Moving Average

EMA Exponential Moving Average

Appendix **B**

Contents of enclosed CD

```
src ......................................the directory of source codes
├── impl ...................the directory of source codes of my strategies
│   ├── experts .......................the directory containing strategies
│   ├── include .........................the directory with source codes
│   │   ├── common .....................the directory with other libraries
│   │   ├── enums .......................the directory with enumarations
│   │   ├── graphicalobjects ........the directory with graphical objects
│   │   └── indicatoraux.............the directory with auxiliary classes
│   └── indicators ..................the directory containing indicators
└── thesis....................the directory of source codes of this thesis
    └── pictures.........................pictures included in the thesis
text .......................................the thesis text directory
└── BP_Jirasko_Petr_2015.pdf...........the thesis text in PDF format
readme.txt ......................the file with CD contents description
```