UNIVERZITA MATEJA BELA V BANSKEJ BYSTRICI

EKONOMICKÁ FAKULTA

DESIGN AND IMPLEMENTATION OF AUTOMATED TRADING SYSTEMS

DIPLOMOVÁ PRÁCA

159625f7-db46-421a-b47e-ed37d6a29e1c

2010 **Bc. Michal Kecera**

UNIVERZITA MATEJA BELA V BANSKEJ BYSTRICI

EKONOMICKÁ FAKULTA

DESIGN AND IMPLEMENTATION OF AUTOMATED TRADING SYSTEMS

Diplomová práca

159625f7-db46-421a-b47e-ed37d6a29e1c

Študijný program: Financie, bankovníctvo a investovanie

Študijný odbor: 3.3.6. financie, bankovníctvo a investovanie

Pracovisko: Katedra financií a účtovníctva

Vedúci diplomovej práce: doc. RNDr. Pančík Juraj, CSc.

Stupeň kvalifikácie: inžinier (v skratke „Ing.“)

Dátum odovzdania práce: 12.11.2010

Dátum obhajoby:

Banská Bystrica 2010 **Bc. Michal Kecera**

Abstrakt

Bc. KECERA, Michal: Design and Implementation of Automated Trading Systems. [Diplomová práca]. Univerzita Mateja Belu v Banskej Bystrici. Ekonomická fakulta; Katedra financií a účtovníctva. Vedúci práce: Pančík Juraj, doc. RNDr., CSc. Banská Bystrica, 2010. 70 s.

Práca zhrňuje základné poznatky o automatických obchodovacích systémoch. Popisuje prekážky a problémy pri vývoji, pričom následne poskytuje riešenia. Práca prezentuje metodológiu pre vývoj a implementáciu. Zaoberá sa meraním výkonnosti, money manažmentom a problémom optimalizácie a robustnosti pri automatických obchodovacích systémoch. Ponúka zhrnutie známych metód a prezentuje aj nové a upravené metódy pre riešenie daných problémov. Následne ponúka výsledky systémov vyvinutých pomocou danej metodológie a s použitím spomínaných postupov.

Kľúčové slová: Automatický obchodovací systém. Money manažment. Optimalizácia. Meranie výkonnosti. Position sizing. Robustnosť.

Abstract

Bc. KECERA, Michal: Design and Implementation of Automated Trading Systems. [Diploma thesis]. Matej Bel University in Banska Bystrica; Department of Finance and Accounting. Supervisor: Pancik Juraj, doc. RNDr., CSc. Banska Bystrica, 2010. 70 s.

This diploma thesis gets together the basics about automated trading systems. It describes the obstacles and pitfalls during development and offers solutions to these. This thesis presents a methodology for design and implementation. It is concerned with system performance measurement, money management, and optimization and robustness issues. It offers a summary of widely used methods while presenting some adjustments to these and new methods to solve above mentioned issues. Subsequently it offers the results for systems developed using methodology and principles described here.

Key words: Automated trading system. Money management. Optimization. Performance measurement. Position sizing. Robustness.

Foreword

This diploma thesis is focused on the subject of automated trading systems. Considerable part of total transaction volume on exchanges is driven by institutional algorithmic trading. Today's more accessible technology and connection to marketplace through brokers makes this type of trading relevant for retail traders as well.

Having actively and discretionary traded for my own account for two years I moved from this type of trading to development of automated trading systems. During 18 months I was able to develop several systems that I started trading. The out-of-sample backtested results are presented.

The main goal of this thesis is to offer a basic framework for systems development, focusing on principles and methodology and highlighting several issues/pitfalls connected to the subject. The partial goals are:

a) Presenting a reasonable performance measurement method for systems comparison and assessment.
b) Tackling the issue of robustness and optimization
c) Finding a suitable money management technique

Theoretical part of this thesis is covered in Chapters 1 and 2 with the analytical part and solutions presented for every major goal in Chapters 3, 4 and 5.

The contribution of this thesis is in the refinement of existing performance metrics to make them more dependable for development and comparison of systems, the introduction of robustness measurements and a different take on money management - focusing on drawdown estimation.

I would like to express my gratitude towards my supervisor Dr. Juraj Pancik for his guidance. I would also like to state that I wrote this thesis independently, with guidance from diploma thesis supervisor and I used literature referenced at the end.

Michal Kecera, Banska Bystrica, 20th July 2010

Content

Abstrakt 3

Abstract 4

Foreword 5

1. Automated trading system 11

 1.1. The industry 12

 1.2. Technology requirements 14

 1.3. Data requirements 15

 1.4. Knowledge and skills requirements 15

 1.5. Time-frame 15

 1.6. Trading vehicles 17

 1.7. Long or short 17

 1.8. Components 18

2. Pitfalls 22

 2.1. Data integrity 22

 2.2. Survivorship bias 22

 2.3. Data-Snooping Bias 23

 2.4. Look-Ahead Bias 24

 2.5. Assumptions about trading costs 24

3. Performance measurement 28

 3.1. Net Profit 28

 3.2. % of Winning Trades 29

 3.3. Profit factor 30

 3.4. Exposure % 31

3.5. Compounded Annual Return (CAR) 31

3.6. Sharpe Ratio 33

3.7. Ulcer Index and Ulcer Performance Index 35

3.8. Recovery factor 37

3.9. Modified Recovery factor 37

3.10. CAR/Maximum Drawdown 39

3.11. Modified CAR/Maximum Drawdown 39

4. Money Management 42

4.1. Fixed equity position sizing 42

4.1.1. Constant units 43

4.1.2. Constant Value 44

4.1.3. Constant Risk 45

4.1.4. Constant Volatility 46

4.2. Variable equity position sizing 47

4.2.1. Martingale system 47

4.2.2. Anti-martingale system 49

4.3. Drawdown Estimation 53

5. Optimization and robustness 57

5.1. Exhaustive optimizers 58

5.2. Smart optimizers 58

5.3. Robustness and Over-fitting 59

5.3.1. Walk-Forward Analysis 61

5.3.2. Visual inspection of robustness 63

5.3.3. Robustness measurements 66

6. Systems performance results 69

Resume.. 74

References.. 75

Figures list

Figure 1: Percentage of algorithmic trading volume Source: (www.aitegroup.com)...... 14
Figure 2: Buy and Sell signals on QQQQ chart with daily bars...................................... 19
Figure 3: Rules transforms market time series to time series of signals Source: (David Aronson, 2007, p.17) 20
Figure 4: Signal and Position Size time series.. 20
Figure 5: Without and with transaction costs 26
Figure 6: Equity curve of two systems 28
Figure 7: System with 100 % of winning trades... 30
Figure 8: Two systems with same CAR 32
Figure 9: Two systems with nearly the same Sharpe Ratio 34
Figure 10: Sharpe A = 1.94 Sharpe B = 3.6.. 35
Figure 11: System A has higher UPI than System B 37
Figure 12: Systems with and without compounding... 38
Figure 13: Rolling MCARDD 40
Figure 14: Rolling MCARDD histogram 41
Figure 15: Constant units position sizing.. 43
Figure 16: Constant value position sizing... 45
Figure 17: Constant risk position sizing 46
Figure 18: Martingale vs. Normal system... 48
Figure 19: Probability of runs of losing trades 49
Figure 20: Anti-Martingale vs. Normal system 50
Figure 21: Compounding on 1, 2 and 3 %.. 51
Figure 22: Net profit under different fractions in % 52
Figure 23: Histogram with ARF from sampling.. 54
Figure 24: Maximum drawdown histogram from sampling 55
Figure 25: Optimized system with three parameters 60
Figure 26: Next ten years of trading the same system 61
Figure 27: Walk-Forward Analysis 62
Figure 28: WFA OOS results with arbitrary 1 and 0.5 year IS and OOS periods 63
Figure 29: 3D graph with two parameters and CAR/MDD fitness function 64

Figure 30: 3D graph with two parameters and CAR/MDD fitness function 65
Figure 31: Two system with simple robustness measurements 66
Figure 32: Histogram of optimization results 67

1. Automated trading system

In this introductory chapter we will try to explain what an automated trading system is. There are many expressions that can be used as synonyms. Trading algorithm, automated trading strategy or mechanical trading system they all describe something which falls under the category of quantitative or algorithmic trading. “Quantitative trading, also known as algorithmic trading, is trading of securities based strictly on buy/sell decisions of computer algorithms. The computer algorithms are designed and perhaps programmed by the traders themselves, based on the historical performance of the encoded strategy tested against historical financial data.” (Ernest P. Chan, 2009, p. 1) While this definition reflects accurately what quantitative trading represents we should add that it doesn’t have to apply exclusively on buy/sell decisions. Decisions about allocation based solely on computer algorithms can be also included under this definition. By allocation we mean answer to the question of how much of our capital should we invest in particular period of time. Katz presents a more detailed description:

“A complete, mechanical trading system, one that can be tested and deployed in a totally objective fashion, without requiring human judgment, must provide both entries and exits. To be truly complete, a mechanical system must explicitly provide the following information:

1. When and how, and possibly at what price, to enter the market.
2. When and how, and possibly at what price, to exit the market with a loss.
3. When and how, and possibly at what price, to exit the market with a profit.” (Jeffrey Owen Katz et al., 2000, p. 11)

We could add that for an automated trading system to be complete the system should also provide the information about the position size or in other words number of shares with which it will enter the market. Weissman highlights the separation from human discretion: “Mechanical trading systems can be defined as methods of generating trading signals and quantifying risk that are independent of an individual trader’s discretion. Although the advantages in utilizing a mechanical trading system are manifold, most market participants agree that their greatest benefit is the tempering of destructive trader “emotionalism”—which is considered to be the enemy of all successful market participants—from the decision-making process." (Richard L. Weissman, 2005, p. 5)

Now we can explore the relationship between technical analysis, fundamental analysis and quantitative trading which is accurately summarized by Chan. These types of analysis are more known to the retail investor. "Granted, a strategy based on technical analysis can be part of a quantitative trading system if it can be fully encoded as computer programs. However not all technical analysis can be regarded as quantitative trading. For example, certain chartist techniques such as "look for the formation of head and shoulders pattern" might not be included in a quantitative trader's arsenal because they are quite subjective and may not be quantifiable. Yet quantitative trading includes more than just technical analysis. Many quantitative trading systems incorporate fundamental data in their inputs: numbers such as revenue, cash flow, debt-to-equity ratio and others." (Ernest P. Chan, 2009, p. 1)

In conclusion we can say that automated trading system is a set of rules that can be programmed as algorithm, which makes decisions about allocation of resources or position sizing. This decision needs to be made without any further human intervention besides programming. This includes binary decisions as entry and exit rules. This is consistent with the decision about positions size as position size of zero is also possible and would result in no trade, having similar effect as binary buy/sell rules. We will get back to this further in the thesis.

1.1. The industry

In this section we will try to present an estimate about the size of algorithmic trading in today's marketplaces. We will focus mainly on markets in United States as they are the most liquid and developed markets in the world. We would have to have exact information about each transaction created by these programs to accurately determine how big quantitative trading really is. This is not readily available so we will stick with estimations from various sources. Let's identify who could use quantitative trading. By whom we mean what type of market participant:

1. Mutual funds

2. Banks
3. Market makers
4. Hedge funds
5. Retail investors

All these can and in fact do use algorithmic trading each in a way that benefits their needs. Several are defined below:

- Algorithms can be based on price, volume, timing.
- Slicing a big order into many small orders to hide large order from market participants or to minimize its market impact.
- Benchmark algorithms to achieve a specific benchmark (for example, the volume weighted average price over a certain time period
- Gamer sniffs out large orders and then try to use that knowledge to trade against the block at a profit.
- Sniffer is an algorithm to detect the presence of algorithmic trading and the algorithms they are using.
- Artificial Intelligence algorithms of different kind. A program can read news and web blogs searching for certain factors (hurricanes, wars, economic or political events, public opinions) and make trading decisions based on that. (www.selectorweb.com)

Various sources state that algorithmic trading now accounts for about 40 % to 60 % of all volume on NYSE and NASDAQ. Goldberg states that: “For example, an estimated 71 % of fund managers and 93 % of hedge funds used algorithms in 2006 Algorithmic trading is not just a US story. The Deutsche Boerse says that about 40 % of equity turnover in Germany is accounted for by algorithmic traders. It looks like algorithmic trading has become more than a niche strategy: it’s a global phenomenon.” (Richard Goldberg, 2008, p.22). That is a relatively sizeable share. The growth has been steadily positive since the inception of this type of trading as found by Aldridge: “The proportion of buy-side traders using algorithms in their trading increased from 9 % in 2008 to 26 % in 2009, with algorithms at least partially managing 40 % of the total order flow, according to 2009 Annual Algorithmic Trading Survey conducted by the TRADE Group." (Irene Aldridge, 2009, p.17)

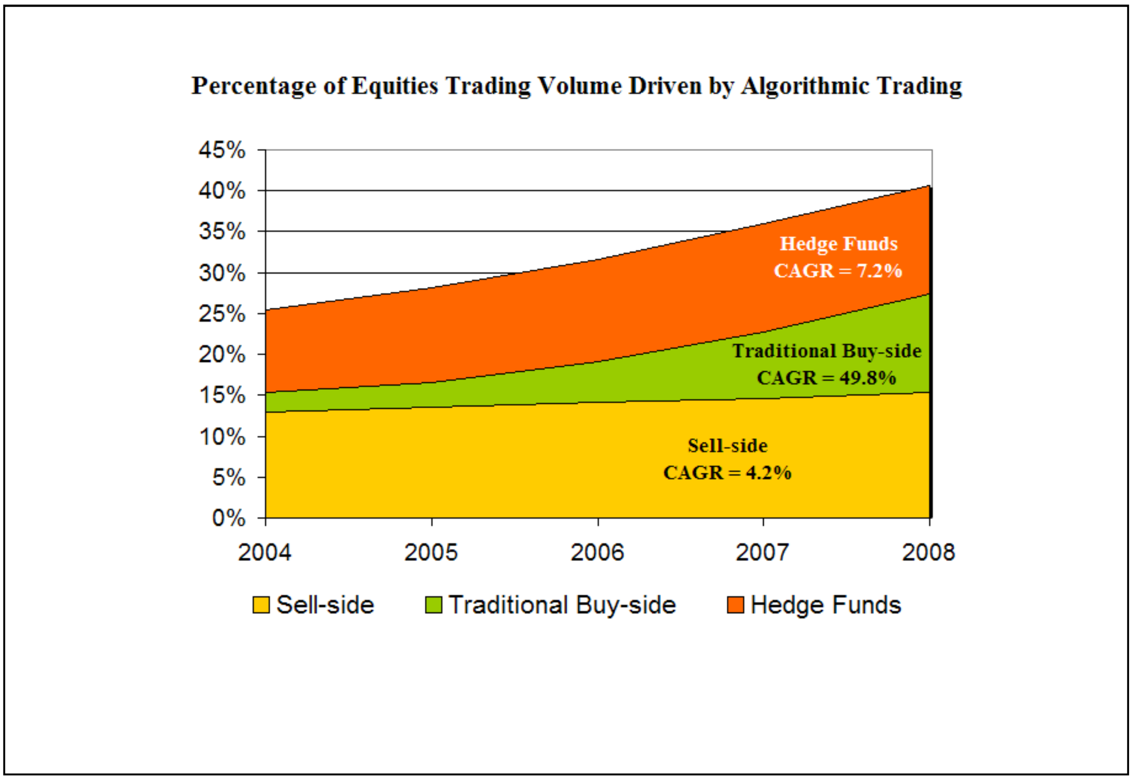


Figure 1: Percentage of algorithmic trading volume
Source: (www.aitegroup.com)

1.2. Technology requirements

What do we need to successfully design and implement automated trading system or strategy? This is the question concerning both hardware and software and the answer depends on the level of sophistication of the strategy and timeframe in which it will operate. High frequency trading algorithms operate in milliseconds and therefore require much more from the resources than daily or hourly algorithms. We will get back to the description of various timeframes and types of systems later on. Basically, to run an automated trading system one needs an adequate computer and execution/testing platform. We will not go into details about hardware configuration. A computer that is capable of running a testing platform is also very likely capable of execution. On the other hand, the faster the computer is the better execution can be done which leads to bigger profitability in some cases. We will take a closer look at testing and execution platforms. There is always a possibility of writing a homegrown testing platform. However, with a vast array of commercial solutions available for a reasonable price, we don't feel the need to do so yet. Later on, when commercial platforms are not enough to satisfy our requirements, we can hire a programmer or write testing platforms ourselves. This is going to happen sooner or later. Systems in this thesis were written in AFL programming language, which is used in AmiBroker testing software. The language is very similar to C++ and even the platform was written in C++. For execution we use Tradestation brokerage and their platform. Internet connection is also important part of the mix. Generally, higher speed is better. This is especially true in high frequency timeframes as every millisecond counts. Some may go to the lengths of locating their trading computers near exchanges to get information and send orders as fast as possible.

1.3. Data requirements

There are several options how to get the data for instruments we plan to trade. For backtesting we require years worth of historical data. Retail execution platforms usually come with real time data feeds that also have the possibility of historical data download. There is a lot of data available for free on the internet. One has to be careful with free sources but there are several relatively high quality providers. For some strategies, that used index data and trade mutual funds, we used www.finance.yahoo.com. For futures and equities data we used Tradestation data feeds and historical databases. There are some pitfalls associated with data and we will discuss them later on.

1.4. Knowledge and skills requirements

It is required to understand statistics and more advanced techniques for more sophisticated strategies. However, to test and implement basic systems there is a need to know the basics of programming, mathematics and statistics. The level depends on the strategies that one wants to implement.

1.5. Time-frame

The selection of time frame entirely depends on the system developer. Time frame is an interval at which the data is processed. Selection also depends on hardware capability. “Tick resolution will give us the most precise information but storing data at that resolution and accessing those uses a lot of resources. Large data samples lessen the dangers of curve-fitting, lead to more stable statistics, and increase the likelihood that predictive models will perform in the future as they have in the past. On the downside, the day trader working with a short time frame needs a real-time data feed, historical tick data, fast hardware containing abundant memory, specialized software, and a substantial amount of time to commit to actually trading.” (Jeffrey Owen Katz et al., 2000, p. 5) Then there is the issue of data costs. In general, EOD (end of day) data are less expensive to obtain than shorter time frames. The most common time frames used in trading are:

1. Yearly
2. Monthly
3. Weekly
4. Daily
5. Intraday
    a. Hourly (60 minutes)
    b. 15 minutes
    c. 5 minutes
    d. 1 minute
6. Tick

“A data source usually has a natural time frame. For example, when collecting intraday data, the natural time frame is the tick. The tick is an elastic time frame: Sometimes ticks come fast and furious, other times sporadically with long intervals between them. The day is the natural time frame for end-of-day pricing data. Although going from longer to shorter time frames is impossible (resolution that is not there cannot be created), conversions from shorter to longer can be readily achieved with appropriate processing.” (Jeffrey Owen Katz et al., 2000, p. 5)

The selection of timeframe is also important from the design point of view. When we select a specific time frame we reduce the degrees of freedom. The problem is that if we want a system to be robust, it should be able to work on many time frames, not just one. For example, we can have excellent results on 8 minutes bars however, when we switch to 16 minutes, the results are not acceptable or even downright bad. The fact of the matter is that we can code any system that runs on 60 minute bars to be able to run on 1 minute bars. By choosing a time frame that suits particular strategy we are doing optimization and to some extend curve-fitting. We will discuss these issues later in the thesis.

1.6. Trading vehicles

Instruments or trading vehicles that are used in this thesis include a variety of equities, mutual funds and futures. Below you can find a brief description of these trading vehicles as defined by www.investopedia.com:

- Equities: A stock or any other security representing an ownership interest.
- Futures: A financial contract obligating the buyer to purchase an asset (or the seller to sell an asset), such as a physical commodity or a financial instrument, at a predetermined future date and price. Futures contracts detail the quality and quantity of the underlying asset; they are standardized to facilitate trading on a futures exchange. Some futures contracts may call for physical delivery of the asset, while others are settled in cash. The futures markets are characterized by the ability to use very high leverage relative to stock markets.
- Mutual funds: An investment vehicle that is made up of a pool of funds collected from many investors for the purpose of investing in securities such as stocks, bonds, money market instruments and similar assets. Mutual funds are operated by money managers, who invest the fund's capital and attempt to produce capital gains and income for the fund's investors. A mutual fund's portfolio is structured and maintained to match the investment objectives stated in its prospectus. (www.investopedia.com)

1.7. Long or short

We are going to use a variety of long and short strategies in this thesis, both combined and alone. Most of the strategies are long and short to ensure that they are valid through bull and bear markets as well. Long or long position means “buying of a security such as a stock, commodity or currency, with the expectation that the asset will rise in value. Short position or selling short means selling of a security that the seller does not own, or any sale that is completed by the delivery of a security borrowed by the seller. Short sellers assume that they will be able to buy the stock at a lower amount than the price at which they sold short.” (www.invetopedia.com)

Some of the systems presented here will be market neutral or to be more precise dollar neutral, by which we mean that the dollar value of long positions minus dollar value of short positions will be zero or approximately zero. This is a dollar neutral strategy which doesn’t have to be market neutral. Vidyamurthy defines them as “strategies that are neutral to market returns, that is, the return from the strategy is uncorrelated with the market return. Regardless of whether the market goes up or down, in good times and bad the market neutral strategy performs in a steady manner, and results are typically achieved with a lower volatility. This desired outcome is achieved by trading market neutral portfolios. Let us therefore define what we mean by a market neutral portfolio. In the CAPM (Capital Asset Pricing Model) context, market neutral portfolios may be defined as portfolios whose beta is zero. Not surprisingly, zero beta portfolios are also sometimes referred to as dollar neutral portfolios.” (Ganapathy Vidyamurthy, 2004, p. 5)

1.8. Components

Automated trading system consists of several parts, depending on the view that we take. The main parts of every strategy are:

1. Signal generation
    a. Entry signals
        i. Buy (when to buy)
        ii. Sell Short (when to sell short)
    b. Exit signals
        i. Sell (exits long trades)
        ii. Buy to cover (exits short trades)
2. Position Sizing (how much to buy or sell short)

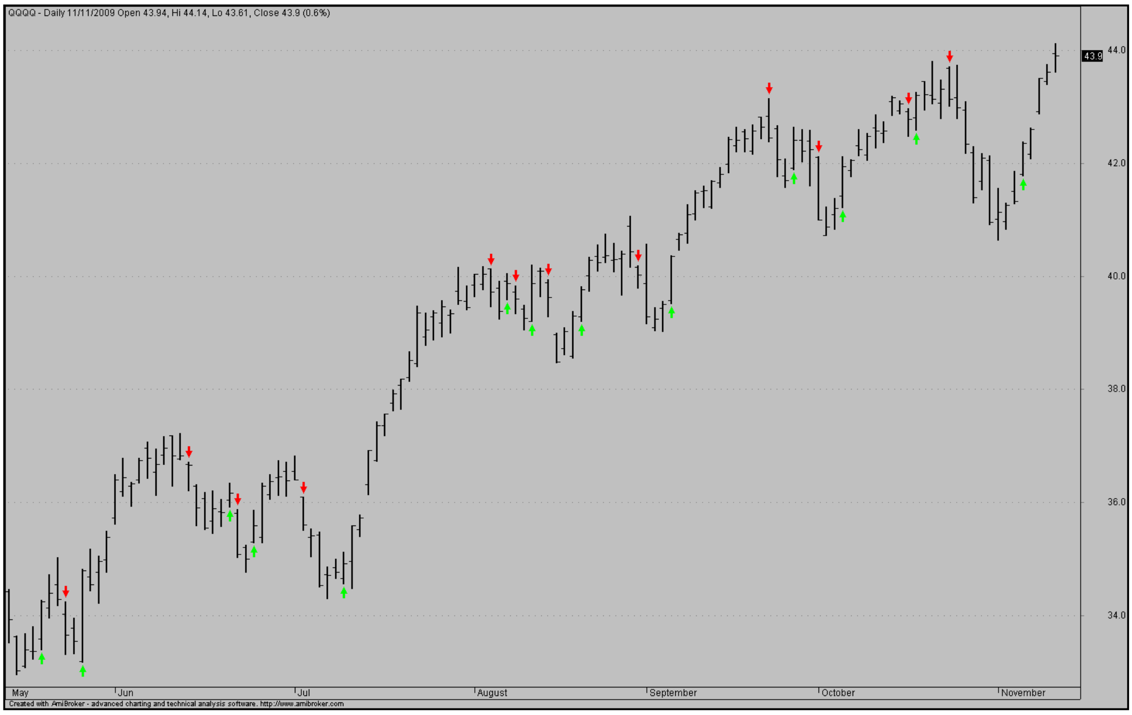


Figure 2: Buy and Sell signals on QQQQ chart with daily bars

Source: Own

Figure 2 shows Buy and Sell signals on QQQQ. Green arrows are Buy signals and red arrows are Sell signals. This is the first part of automated trading systems. Decisions about position sizing can’t be seen on this picture, but the algorithm is simple and allocates 1 % of equity on each trade. Same can be said for Short and Buy to cover signals.

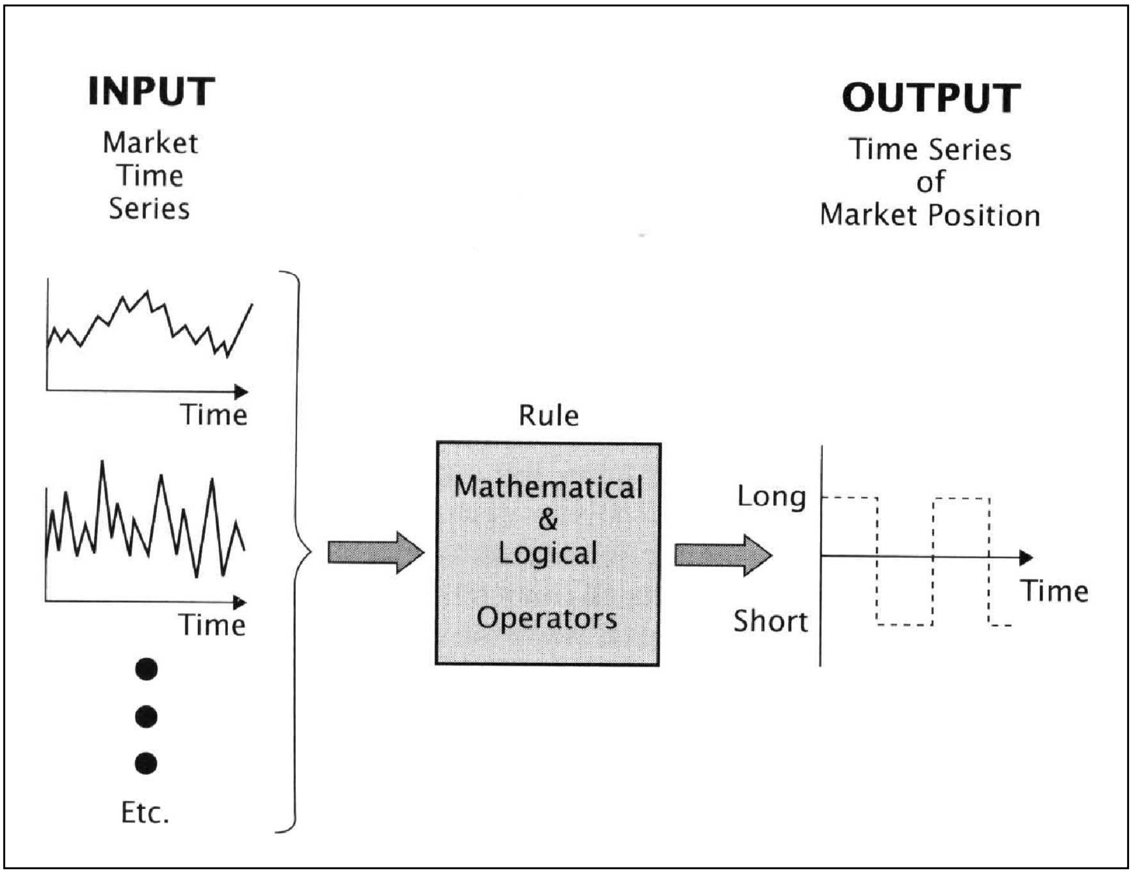


Figure 3: Rules transforms market time series to time series of signals

Source: (David Aronson, 2007, p.17)

Figure 3 shows how can we get trade signals using various mathematical and logical operators. Now we can think more about the trading signals and their necessity. We can get the same results as seen in the first two pictures just by specifying position size or amount of money to be invested at that time. As zero is also a measure of size, we can say, that for bars with no trades, we assign position size of 0. This way we won't buy or sell short at that particular point of time.

Time	1	2	3	4	5	6	7	8	9	10
Buy Signal	0	0	1	0	0	0	1	0	0	0
Sell Signal	0	0	0	0	1	0	0	0	0	1
Position Size	0 $	0 $	10000 $	10000 $	0 $	0 $	10000 $	10000 $	10000 $	0 $

Figure 4: Signal and Position Size time series

Source: Own

As we can see in Figure 4, it is possible to erase Buy and Sell signals and still get the same results just knowing the position size at each time increment. This is just a theoretical exercise and doesn't have a big impact on the system itself. It is here to show that even if we decide to stay out of the market, we are still making a decision to trade with a position size of zero.

2. Pitfalls

In his part we will take a closer look on various problems associated with designing of automated trading systems. We will identify several biases and try to propose a solution to overcome them.

A lot of problems come from the fact, that we simply have wrong assumptions at the beginning and when we take a system and implement it, it isn't working as planned. These may be trivial, like assuming that I can buy at open price of the day or sell at close or more difficult to spot, like assumptions about underlying distributions. Some problems occur even before we actually begin with designing a trading system. Data issues can transform a good system into bad and vice versa. To be concise we will start at the beginning with various data pitfalls.

2.1. Data integrity

Accuracy of the data on which are the systems tested is paramount to get realistic results. Common problems are bad ticks. These are trades that occur outside "normal" trading conditions. For example, shares of particular company are trading at 13.13 $ and suddenly a trade is printed at 15 $ before trading resumes at previous levels. If volatility of the stock is low, we can safely say that this was a "bad" trade and we can't base a system on this data. The good news is that most of these bad ticks are filtered directly by data providers and there is no need to adjust quality data.

Futures testing offer another pitfall in the form of a roll-over. When contract rolls from one period to another we need to adjust our position accordingly and backtesting process has to account for this. This also applies to continuous contracts which need to be constructed properly. It is beyond the scope of this thesis to show how it should be done.

2.2. Survivorship bias

"A historical database of stock prices that does not include stocks that have disappeared due to the bankruptcies, delistings, mergers or acquisitions suffer from the so-called survivorship bias because only "survivors" of those often unpleasant events remain in the database." (Ernest P. Chan, 2009, p. 24) Imagine having a database of Nasdaq 100 stocks constructed in 2009 which has data for 9 years. If you take a look 9 years back to 1999, you will find that only 76 companies out of the 100 today were in the index at that time. That means that 26 different companies were in the index before 2009 which are not included in the dataset. This bias is dangerous mainly in strategies that try to buy low and sell high. One of those strategies may be buying only a number of stocks with lowest return during previous year. There are filters that help to overcome this bias. Mainly price filters or range filters. It is important to note that survivorship bias-free data is expensive. In this thesis we used data that suffer from survivorship bias, but due to the nature of strategies presented here this bias doesn't affect them in a meaningful way. This bias is more dangerous in longer-term systems that hold trades for weeks and sometimes even months.

2.3. Data-Snooping Bias

This is subjectively the most dangerous bias that we encountered in the system development process. Other terminology that is often used is curve-fitting and reducing degrees of freedom. According to Katz: "Another issue found in trading system development is optimization, i.e., improving the performance of a system by adjusting its parameters until the system performs its best on what the developer hopes is a representative sample. When the system fails to hold up in the future (or on out-of-sample data), the optimization process is pejoratively called curve-fitting." (Jeffrey Owen Katz et al., 2000, p. 54)

This issue has to do with statistical significance to certain degree. We have two systems with same mean return and standard deviation of returns. System A has 10 trades and system B has 2000 trades. System B is said to be more statistically significant. There are measures on statistical significance. We will not dive in to them in this thesis. However we have to understand that "the larger the number of parameters being optimized, the larger the sample required. In the language of statistics, the parameters being estimated use up the available degrees of freedom." (Jeffrey Owen Katz et al., 2000, p. 11) This comes down to robustness and sensitivity analysis. We will examine this in detail in the chapter on optimization. For now, let's define robustness: "Practically speaking, robustness translates into getting the most realistic results by using the fewest parameters and the most data. Robustness is a term used to describe a system or method that works under many market conditions, one in which we have confidence; it is associated with a successful result using an arbitrary set of parameters for a system. This is most likely to occur when only a few parameters are tested over a long time period." (Perry J. Kaufman, 1998, p. 552) To overcome the problem of curve-fitting we should use as few parameters as possible or better yet use robust parameters. We will later try to measure robustness and perform Walk Forward Analysis – one of the methods to identify curve-fit systems.

2.4. Look-Ahead Bias

"Look-ahead bias also known as leakage of future information occur in the context of historical testing when information that was not truly available at a given point in time was assumed to be known." (David Aronson, 2007, p.29) There are several views that we can take on this subject. One is from the point of programmer. It is possible to be a victim of this bias just by careless programming. We can also assume that we have information about fundamental data that are released after some time, but we use this information to generate trading signals. Usually when a system looks too good to be true, it probably is, and we should start looking for a flaw in its design.

2.5. Assumptions about trading costs

When we test a trading system we are trying to replicate its behavior on a data set. This involves making assumptions about trading costs. There are several types of trading costs:

- Slippage
- Commissions
- Various fees

Slippage is the difference between assumed transaction price and actual transaction price. Slippage can occur from several different reasons:

- Liquidity: More liquid instruments have less slippage
- Spread: Bid-Ask spread is natural difference between Bid price and Ask price.
- Order Lag: There is a time difference between sending an order and pairing.

Slippage also varies with usage of different type of orders that can be executed and time at which they are executed

- Limit orders: There is no slippage because we are always receiving stated price or better. However we are making assumptions about fill probability. Incorrectly, we can assume that we will get filled, when the price touches our desired price. This is not the case in the real marketplace thought and such assumption can be dangerous. It is safer to assume that we will get filled when price gets below our desired price by at least one tick. Even this might not be true in real market but more often than not, we will get filled. We will therefore need to adjust slippage expectations and account for the facts that we might not get filled.
- Market orders: We will get filled on market orders but we will also have larger slippage. We might get positive or negative slippage. It is difficult to estimate whether on average it will be positive or negative. We should however assume that it will be negative and therefore have to guess how much slippage we could have. This also depends on the volatility and time of day. If we plan to use market orders near opening or closing time we should take into account higher volatility and slippage.

As far as commissions are concerned, this depends on individual brokers. Platforms fees and other fees like data fees are also individual. Trading costs assumptions also affect optimization. If we have high costs assumption, optimization algorithm will favor systems with fewer trades. If we specify low or zero costs, we might end up with systems with thousands of trades during short period of time. We have to bear in mind that there is nothing wrong with specifying zero transaction if we are able to replicate this in real trading. Some mutual funds have nearly zero transaction costs. However, as a retail trader this is highly unlikely. Institutional traders might have the benefit of zero transaction costs or even rebates for trades. To illustrate this problem in Figure 5, we can observe the same system with and without transaction costs. If we don’t include any costs the system has CAR of 63.55 % and CAR/MDD=4.4. This means that with a starting equity of 100,000 $ the Net Profit is 25,114,933 $. Now, if we introduce a transaction cost of 0.03 $ per share with a minimum of 1 $ we get the following stats: CAR=44.75 % with CAR/MDD=2.86 and Net Profit of 6,292,582 $. We do compound in this system.

Introducing transaction costs had a huge impact on the results. The difference is around 19,000,000 $ in lost profits. Such a huge difference is also due to the compounding effect. There were total of 3,725 trades made.

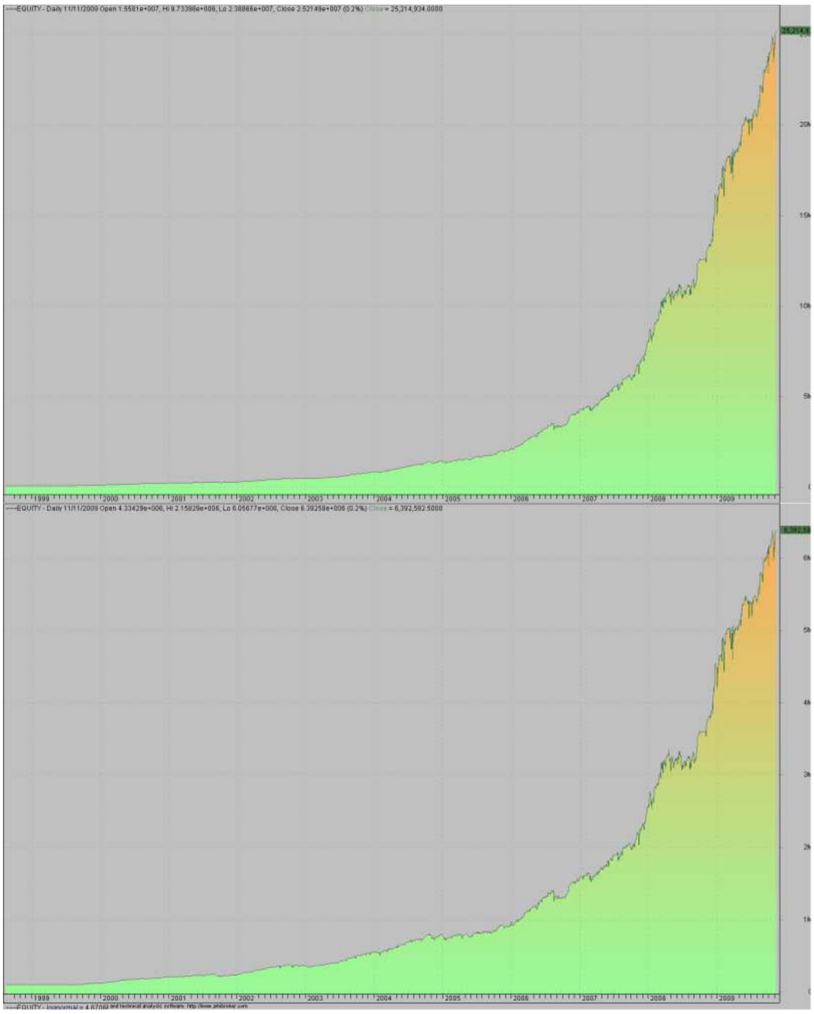


Figure 5: Without and with transaction costs

Source: Own

We examined pitfalls associated with development process. They are connected with the design phase. There are also other dangers connected with implementation. These are mainly hardware or software malfunction. Once we have a running strategy disruptions will cost money. It is possible to implement these disruptions in the development phase. Random shocks can be programmed. However, it is better to evade them altogether. Potential threats are:

- Internet connection failures
- Software crashes
- Hardware malfunctions
- Electricity outage

3. Performance measurement

Let’s start with a question. How do we compare two systems? And how do we decide that the system that we are testing is good enough to begin trading it? These are the questions that can be answered by system performance measurements. We will try to summarize common performance measurements used in investment business and point out their weaknesses by examples. After that we will try to come up with our own measurements by altering the existing ones.

3.1. Net Profit

Net profit is very much self explanatory. It is the profit that strategy generated through backtest during the whole time.

First disadvantage is that Net Profit doesn’t take into account drawdown. Drawdown is a distance from the peak of equity to subsequent lowest point in equity. In other word, if a system suffers a 20 % drawdown it went from the peak of let’s say 100000 $ to 80000 $. Drawdowns are important when evaluating a trading system. Most investors abandon strategies and withdraw money after drawdowns. We might take that as a cost of doing business in quantitative trading world.

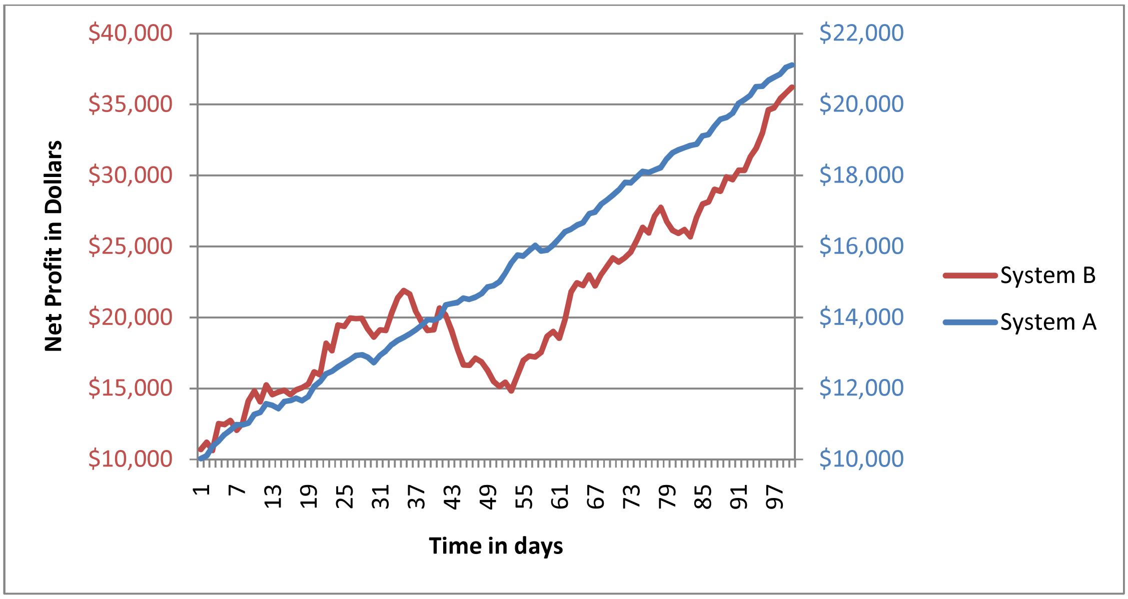


Figure 6: Equity curve of two systems

Source: Own

In Figure 6 we compare two different systems: System A has a Net Profit around 21000 $ and System B around 37000 $. But looking at the equity curves of both systems together a majority of investors would probably choose System A as it has lower drawdown. These ECs were created using Box-Muller method of creating standard normal distribution from uniform distribution and random number generator in Excel.

Another disadvantage is that we can't compare systems backtested over different periods of time. The system with longer data periods will have bigger Net Profit given that its expectations are positive. The solution for this would be adjusting for the length of backtest and computing Annual Net Profit.

We also do not know the capital required to achieve such Net Profit. In other words we might choose a system that has the highest Net Profit but annual returns are not sufficient.

3.2. % of Winning Trades

$$\% \, Winners = \frac{Number\ of\ Winnners}{Number\ of\ all\ trades}$$

[1]

This measure is very popular and used by sellers of trading systems. The reason for that probably is that it is widely recognized by general public. When somebody states that he has a system with 90 % winning trades, general public usually responds well to such statements. This is however not at all a valid measurement for a system for the following reasons.

It doesn't take into account drawdowns. In this respect is similar to Net Profit. We can compare systems with different length of backtest but it is very dangerous. One can simply take a period with high winning % and omit the rest. Also, we do not know anything about return on our investment. % of winning trades can be easily manipulated. Consider the following: We can have a system that buys randomly and waits for the first profit to sell. This system will have 100 % of winning trades but will also have massive drawdowns.

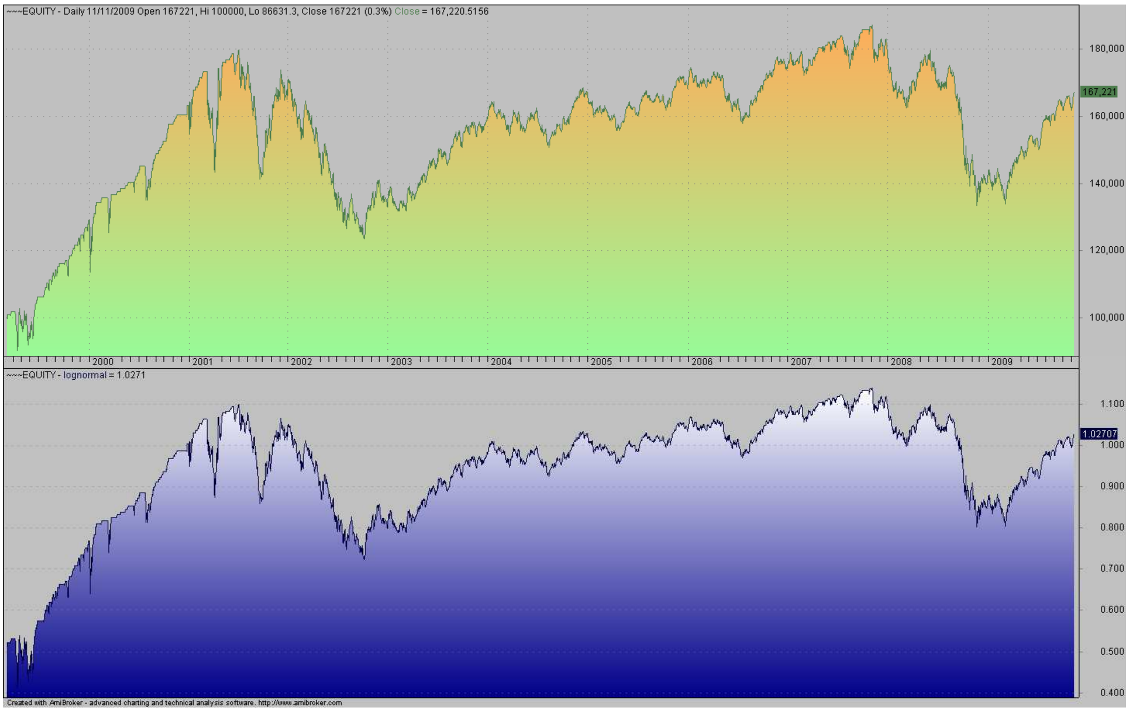


Figure 7: System with 100 % of winning trades

Source: Own

In Figure 7 we can see automated trading system that over the course of 9 years made 80 trades on QQQQ and all were profitable. While it might look tempting, we doubt that someone would like to trade this system, if he was presented with this equity curve. The picture shows a log scaled Equity Curve at the bottom.

3.3. Profit factor

$$Profit\ Factor = \frac{\sum Winners}{\sum abs(Losers)}$$

[2]

Profit Factor is equal to sum of profit of winners divided by sum of loss of losers. It is a number between zero and infinity. There are several disadvantages with using this measure.

First, it doesn't tell us anything about returns. A system might have relatively high profit factor and yet return just several percents per year. Second, if we do optimizations with Profit Factor as a target, we might get results with just one winning trade and infinitely high Profit Factor. However, we can compare systems using PF, even those applied to different data lengths. But it won't tell us anything about the profitability or whether those systems have one trade or hundreds which would get an indication of statistical significance. All in all, this measure is not suitable for decisions about system implementation.

3.4. Exposure %

$$\% \, Exposure = \frac{\sum \frac{\sum Open \ positions \ in \ a \ bar}{Portfolio \ Equity}}{\# \ of \ bars}$$

[3]

This metric gives us market exposure of trading systems. For example, we have a trading system that is fully invested over the whole period of time. Market exposure of such system would be 100 %. Now, we have a system that invests 50 % of our equity for the whole periods. Exposure would be 50 %. For a system that is invested 100 % for half of a time, market exposure would be 50 %.

This metric alone can't tell us a lot about the performance of a trading system. Only combination of this metric with others, like Net Profit, could give us some insight to a system. Generally, if we have systems with the same Net Profit we would choose a system with less exposure to the market. Logic behind is that if we have less exposure we also have less risk. According to some theories, like Capital Asset Pricing Model, investors want more return with a less risk. Given a choice of two investments with the same return, they choose the one with low risk. Exposure is to some a extend measure of risk. In the original CAPM the measure of risk is variance.

3.5. Compounded Annual Return (CAR)

CAR is also referred to as Compounded Annual Growth Rate (CAGR). This is the first measure that we can with some adjustments use for comparison of different systems and also to judge systems on their own. "CAGR is the year-over-year growth rate of an investment over a specified period of time. The compound annual growth rate is calculated by taking the n^{th} root of the total percentage growth rate, where n is the number of years in the period being considered." (www.investopedia.com)

$$CAR = \left(\frac{Ending\ Equity}{Beginning\ Equity}\right)^{\frac{1}{\#\ of\ Years}} - 1$$

[4]

Let’s have a system that starts with 100000 $ and ends after 5 years with 200000 $. CAR for this system would be:

200000/100000=2

2^0.2=1.149

1.149-1=0.149

So, if we took 100000 $ and had a 14.9 % yearly return, after 5 years we would end up with 200000 $. We can use CAR to compare two systems with different periods because the returns are annualized.

However there are several problems with this metric. First it doesn’t say anything about drawdown. It is the same as with Net Profit. We could have a system with terrific CAR but 50 % drawdown or two systems with same CAR and different drawdowns. This is hardly acceptable. Second issue is more subtle. As the Annual Return is a geometric average of the return for the whole period, we might get a following problem:

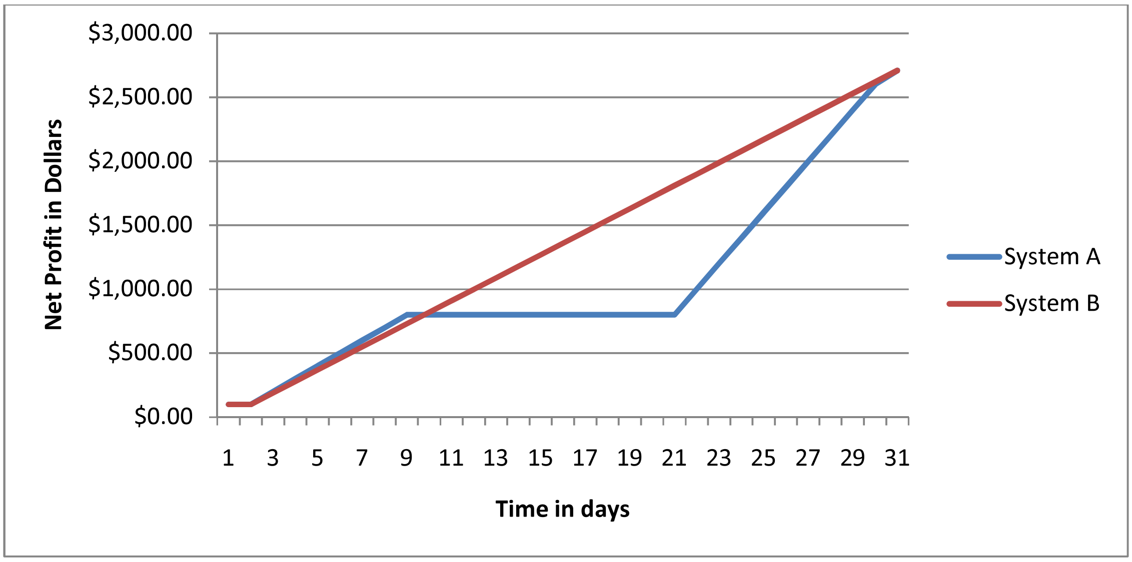


Figure 8: Two systems with same CAR

Source: Own

Let’s assume that these are ECs for two different systems. Both will have the same CAR because they are run over the same period and have the same starting and ending balance. However, by looking at the ECs we can clearly spot the difference. It is consistency that is different. While B consistently made money during the period, A made some at the beginning and then caught up with B in the end. There is no reason to prefer A over B just by looking at the CAR as we can’t see the difference. However by looking at the EC’s we would probably prefer B over A. Note that this has nothing to do with the drawdown issue, as they both have zero drawdowns.

3.6. Sharpe Ratio

Sharpe Ratio is probably the most used measurement in the investment industry. It is the most common metric stated in the performance reports of money managers. The following was taken from (Sharpe, 1994):

Let R_{Ft} be the return on the fund in period t, R_{Bt} the return on the benchmark portfolio or security in period t, and D_t the differential return in period t:

$$D_t \equiv R_{Ft} - R_{Bt}$$

[5]

Let D-bar be the average value of D_t over the historic period from t=1 through T:

$$\overline{D} \equiv \frac{1}{T} \times \sum_{t=1}^{T} D_t$$

[6]

And sigma_D be the standard deviation over the period:

$$\sigma_D \equiv \sqrt{\frac{\sum_{t=1}^{T}(D_t - \overline{D})^2}{T-1}}$$

[7]

The ex post or historic Sharpe Ratio (S_h) is:

$$S_h \equiv \frac{\overline{D}}{\sigma_D}$$

[8]

Basically, this metric calculates excess average return and then divides it by its standard deviation. It is important to note that these have to be annualized by multiplying them by ratio (Number of Bars per Year)/(Average Number of Bars Per Trade). A simplified equation follows:

$$SR = \frac{Average\ Annualized\ Return - Benchmark\ return}{Standard\ deviation\ of\ AAR}$$

[9]

Benchmark return is our threshold return. It can be market return for that period of time. It also can be risk free rate. This depends on the trader. If the BR is bigger than AAR Sharpe Ratio will be negative.

While this metric is widely used and recognized, it also has some drawbacks. First it doesn't address drawdowns.

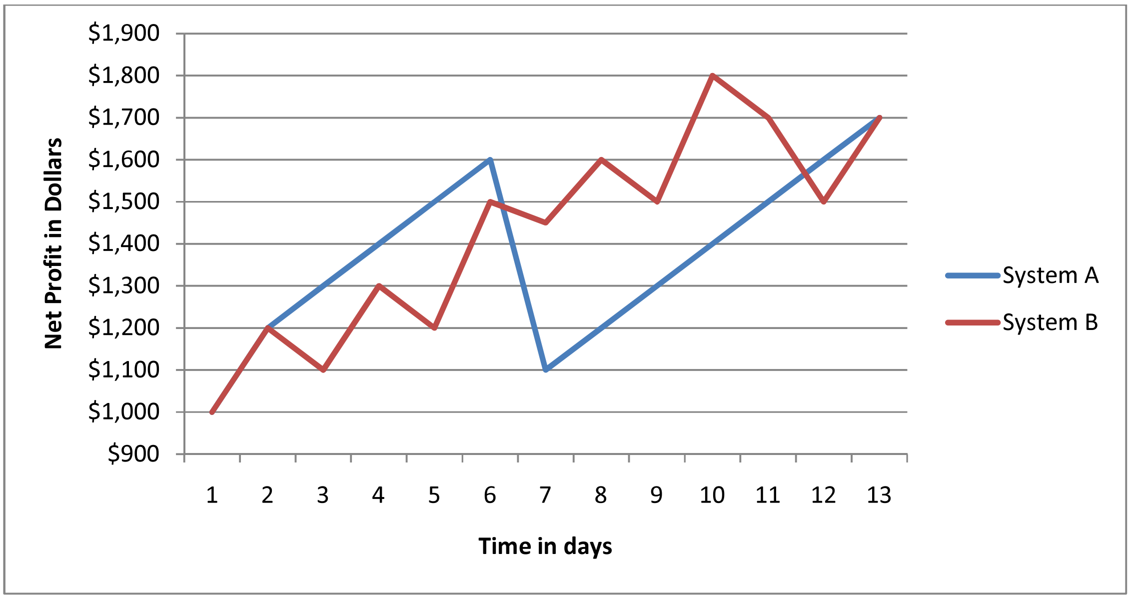


Figure 9: Two systems with nearly the same Sharpe Ratio

Source: Own

In Figure 9 we can see two systems with very similar Sharpe Ratio around 0.3 with benchmark return set to 0 for simplicity. Actually system A has a slightly higher Sharpe Ratio then B. Given the choice between the two, we doubt that someone would choose A over B. Other disadvantage is that this metric also penalizes upward volatility. Figure 10 shows ECs of two systems. A has Sharpe of 1.94 and B has Sharpe of 3.6. Again we set the benchmark returns to 0. This is just a proof of concept. Because of the penalization of upward volatility the system with higher returns has lower Sharpe Ratio. Note that both systems have zero drawdown.

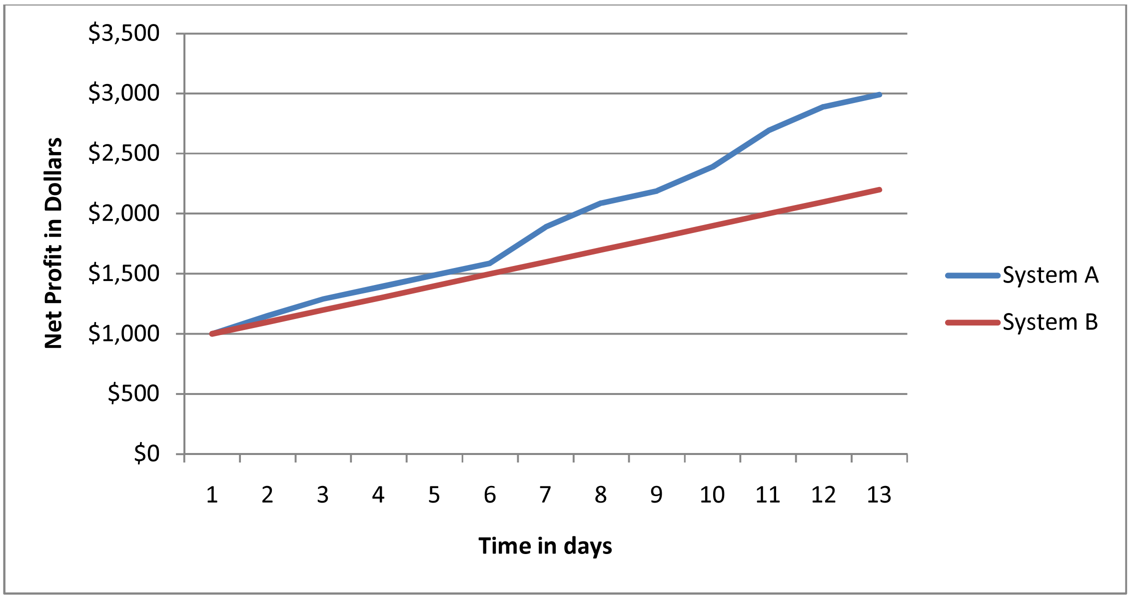


Figure 10: Sharpe A = 1.94 Sharpe B = 3.6

Source: Own

3.7. Ulcer Index and Ulcer Performance Index

Ulcer Index was specifically designed to address problems associated with Sharpe Ratio, mainly the penalization of upward volatility. “An indicator developed by Peter G. Martin and Byron B. McCann that is used to measure the riskiness of investments such as securities, commodities, indexes or mutual funds. It is created by factoring in the depth and duration of drawdowns from recent peaks.” (www.investopedia.com)

$$UI = \sqrt{\frac{\sum Drawdown^2}{N}}$$

[10]

We can see from the equation that we only take into account drawdowns and not upward volatility. Ulcer Performance Index uses Ulcer Index to penalize systems with big or lengthy drawdowns.

$$UPI = \frac{Annualized\ Return - Benchmark\ Return}{UI}$$

[11]

The equation is similar to Sharpe Ratio. Benchmark return can be again any required return we select. This metric doesn't have any serious disadvantages. It can be used to compare different systems backtested under different period lengths. It also takes into account drawdown. However, it measures only average squared drawdown. This means that it can favor systems with one big and a lot of smaller drawdowns over systems with medium sized drawdowns. We can see this on Figure 11. Notice the bigger drawdown on System A. It is subtle and even thought this is one of the metrics that are suitable for system development and implementation, it is important to understand its limits. Ultimately, it is upon developers' judgment to choose which one to use. Second issue is in the interpretation, which is hard for UPI and not as straightforward as by others. Again the ECs in this example are fictional and were created using Box-Muller method.

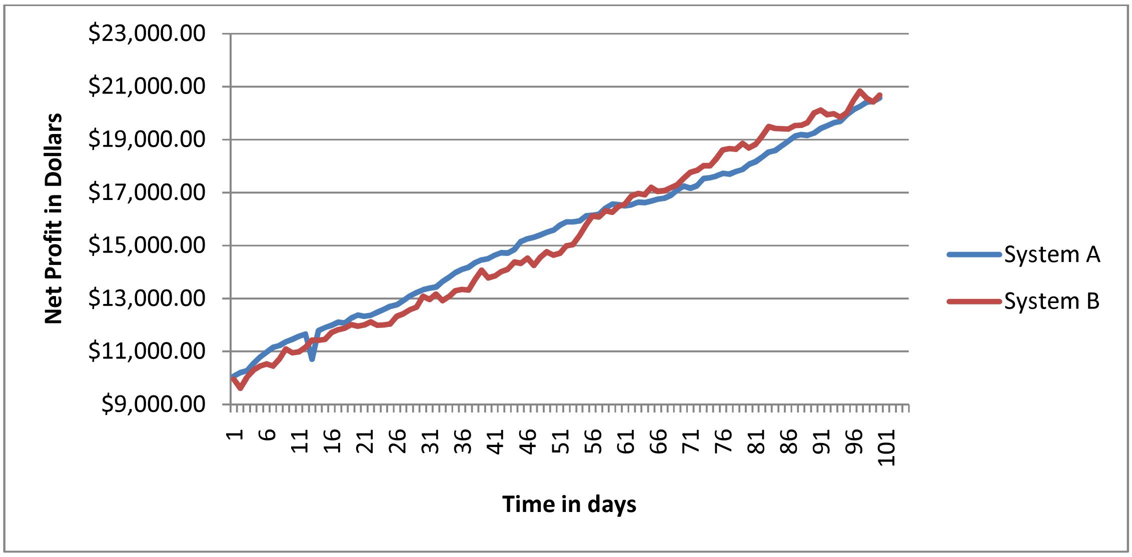


Figure 11: System A has higher UPI than System B

Source: Own

3.8. Recovery factor

This is the first that fully incorporates maximum drawdown. It is due to this fact that we think is suitable for system assessment, but only after some minor adjustments.

$$RF = \frac{Net\ Profit\ in\ Dollars}{Maximum\ Drawdown\ in\ Dollars}$$

[12]

First, Net Profit is not annualized. This means that it is difficult to compare systems tested over periods of different length. One way to solve this is to use Annualized Net Profit. Second issue is with Maximum Drawdown in Dollars. Therefore this metric will favor systems with zero drawdown even if they have only a few trades. If we have zero drawdown, our RF is infinitely big. One way to solve the problem is to add a minimum required return in dollars.

3.9. Modified Recovery factor

So our modified Recovery Factor will look like this:

$$MRF = \frac{Annualized\ Net\ Profit\ in\ Dollars - Annual\ Required\ Profit\ in\ Dollars}{Maximum\ Drawdown\ in\ Dollars}$$

[13]

This is our Modified Recovery Factor. Apart from Ulcer Performance Index, which has quite difficult interpretation, our MRF has the following straightforward interpretation. Let's say that we have a system with MRF of 2 and Required Profit of 10000 $. It is therefore safe to say that this system made on average each year twice the maximum drawdown during the whole period, on top of the required 10000$. The Annual Required Profit is there mainly for optimization purposes. We will get to this in the chapter on optimization, but it is important to note, that due to the required return this metric will no longer favor systems or parameters with fewer trades and miniscule maximum drawdowns. Therefore, it will first satisfy the need for Annual Required Return and then search for the parameters that offer the best Profit for lowest Maximum Drawdown.

Next, we will introduce the issue of compounding. Let's have two systems. These systems are very simple and produce a sequence of trades with returns of +1 % and – 0.5 % repeatedly over 500 periods. The difference is that one of them compounds the profits and losses and the other doesn't.

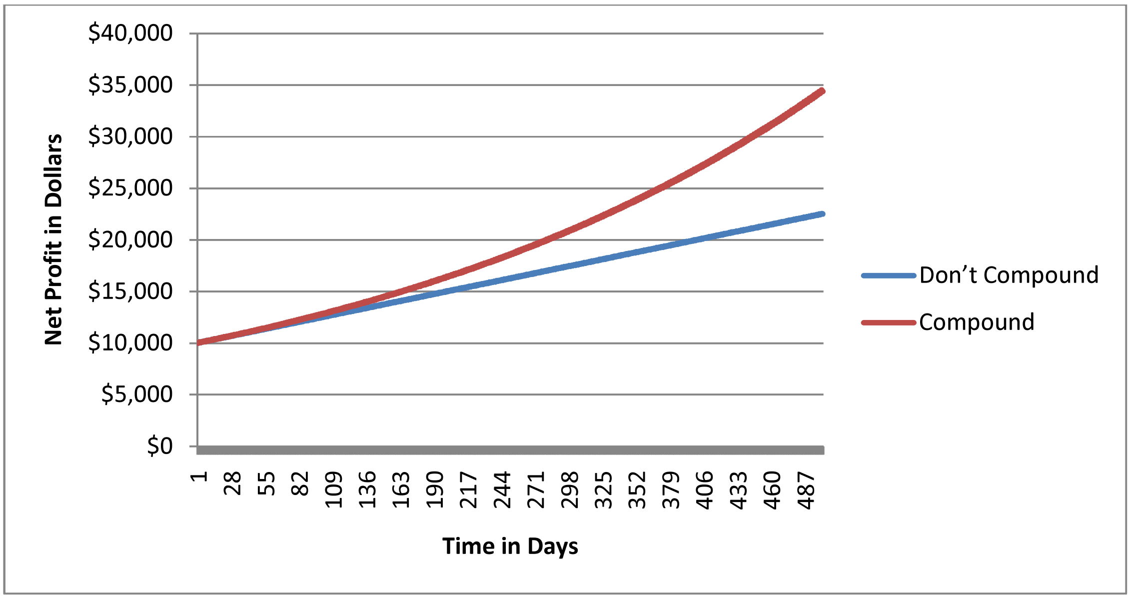


Figure 12: Systems with and without compounding

Source: Own

In Figure 12 we can see that after some time the effect of compounding becomes obvious. We will talk about the compounding more in the chapter about money management. What we want to point out here is that if we have a losing trade of 0.5 %, we will lose 50 $ in the blue system but more in the red system. Depending on the period in which the losing trade will occur, we will lose somewhere between 50.5 $ and 173 $. This will have an effect if we use MRF or RF as our performance measures. We will still have the same drawdown, percentagewise of 0.5 %, but the dollar amount will vary and therefore also the MRF and RF will vary for these two systems and vary considerably. RF for red and blue system is 141 and 250 respectively. What we are trying to say is that we shouldn't use these metrics for systems that compound. For example, if we do optimizations on a system that compounds and we want to maximize RF, we will end up favoring parameters that produce small drawdowns at the end of backtesting period when the dollar amounts are higher and allow bigger drawdowns at the beginning as measured by percentage. Another drawback is that we don't readily know the percent return on our investment. Other than that, we can safely use MRF to compare different systems, given the fact that we won't compound. We can compare systems with different backtest lengths. It is hardly possible to manipulate it as winning % can be. Consistency can be a problem. We will try to introduce a fix to avoid this problem in the next metric.

3.10. CAR/Maximum Drawdown

This measure is very similar to Recovery Factor. Instead of Net Profit in Dollars we use Compounded Annual Return. In the denominator we use Maximum Drawdown in percentage terms. After some adjustments this measure can be safely used to judge systems. These adjustments will be very similar to the ones presented in MRF. We annualize return already; therefore we only introduce Annual Required Return in to the formula. We will call it MCARDD.

3.11. Modified CAR/Maximum Drawdown

$$MCARDD = \frac{CAR - Annual\ Required\ Return}{Max\ DD\ in\ \%}$$

[14]

This measure can be used when we do compound. Other than that it has the same advantages and drawbacks as MRF. Now we can try to address the issue of consistency. Recall, that inconsistent systems can produce inconsistent return in different periods. Sometimes they excel, sometimes they go flat. If we take a measure such as MCARDD, it won't tell us that a particular system is inconsistent. For a typical example see Figure 8. What we can do, is to take a rolling one year MCARDD or MRF and judge systems on the basis of the median rolling MCARDD over a period of time.

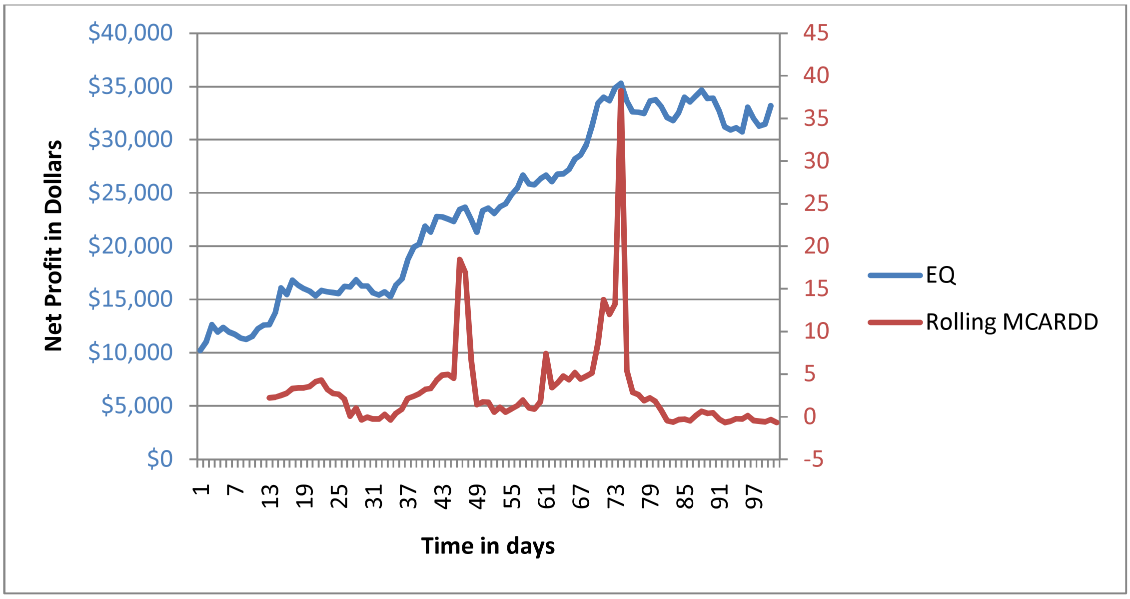


Figure 13: Rolling MCARDD

Source: Own

In Figure 13 we can see rolling 12 month MCARDD. Notice that in some periods it hits almost 40 but what counts is the median value and that is 2.01. It says that median value of all the years that we can look at during backtest is 2.01. To go a little further we can also develop confidence intervals and find out what would be 90 % confidence interval for this metric. However we will have to assume, that future returns will be the same as past returns. We can safely say that this will not be the case and this assumption is wrong and dangerous. Even thought we present it in Figure 14. The best treatment would be to apply this method to out-of-sample results. More on this is in the section about optimization. Required return is 0 for simplicity. Required Return comes to play mainly during optimizations. Every set of parameters, that didn't fulfill the Required Return criterion will have results below zero.

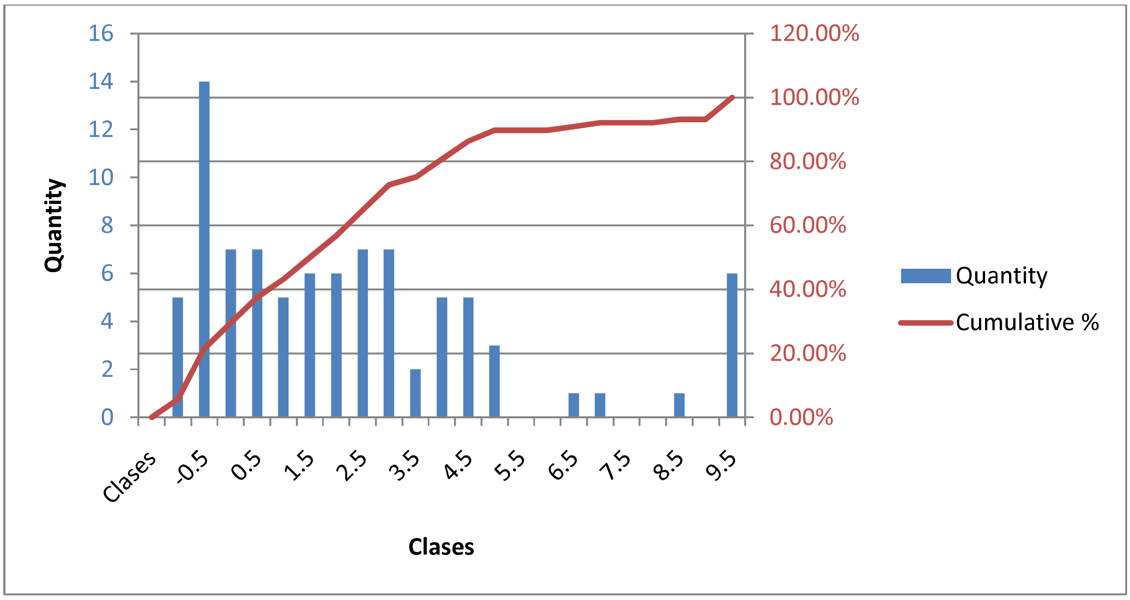


Figure 14: Rolling MCARDD histogram

Source: Own

Ultimately, the decision about performance measures is on the individual developer. We presented several measures and pointed out their advantages, as well as their flaws. We bring forward our modified versions of common metrics, MRF and MCARDD. We also put a twist on them in the form of rolling MRF and rolling MCARDD to get a relatively more robust metric that takes into account the consistency of system performance. Even though these metrics also have their limitations, we can safely use them for system development and performance measurement. The main contribution is the well defined and sound performance measures that can be used for judging system performance both alone and in comparison to other systems. These measures are adjusted so that they mitigate most of the disadvantages of commonly used measures.

4. Money Management

In this chapter, we will discuss the issue of position sizing and money management. We will start with definitions and general introduction to the topic, which is regarded by many as the one most important part of trading system. In general, there is more focus on entries and exit than money management, at least from the perspective of a retail trader.

Everyone applies money management to some extent. Even if one doesn't trade, he or she makes a decision about money management. It is simply a decision to allocate zero of my funds to trading. There are tons of different definitions for money management and same for position sizing. Jones defines money management as: “Money management, as defined here, is limited to how much of your account equity will be at risk on the next trade. It looks at the whole of the account, applies proper mathematical formulas, and lets you know how much of the account you should risk on the next trade.” (Ryan Jones, 1999, p.4) Broader definition that is not as useful for trading but encompasses a larger view is offered here: “The process of budgeting, saving, investing, spending or otherwise in overseeing the cash usage of an individual or group. The predominant use of the phrase in financial markets is that of an investment professional making investment decisions for large pools of funds, such as mutual funds or pension plans.” (www.investopedia.com)

Those are just two from many definitions of money management. Our definition, that will be used in this chapter follows: Money management or position sizing is determining the number of units to place on the next trade. Money management and position sizing are synonyms.

4.1. Fixed equity position sizing

If we plan to use fixed equity position sizing, we assume that our trading account doesn't change over time. This means that whether we have a loss or a win, we still have the same amount of dollars in the account. In other words, we are not compounding. This technique doesn't depend on the amount of dollars that we have. The algorithms doesn't take that into consideration and don't derive the position size from equity in any way whatsoever. These assumptions are not met in reality. However, for big accounts compared to allocations, it may get close.

4.1.1. Constant units

Constant unit is a type of fixed equity position sizing. It means that we will allocate on each next trade the same number of units.

$$Position\ Size = X\ units$$

[15]

Units can be shares, futures contracts, options etc. To examine the effects of various position sizing techniques, we are going to apply them to the same system. Everything else will stay the same apart from position sizing technique. We are assuming transactional costs of 0.03 $ per share and minimum of 1 $ per trade.

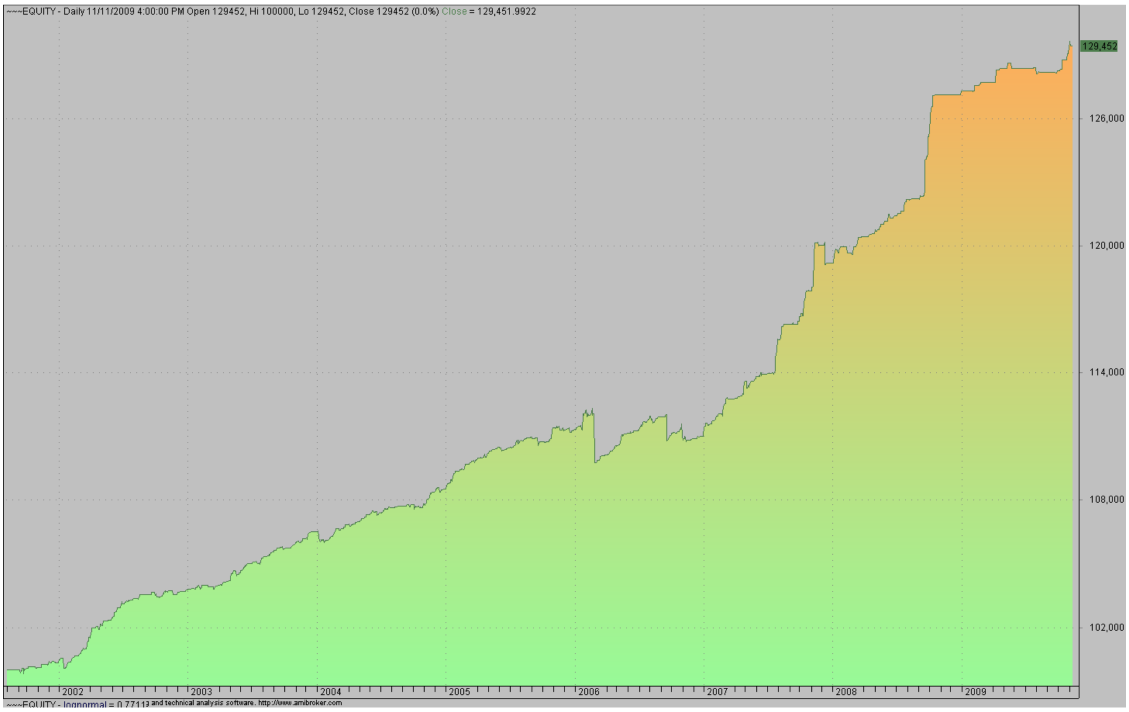


Figure 15: Constant units position sizing

Source: Own

In Figure 15 we can see that constant unit produces some volatility in our account. As we can see the progress was relatively smooth until 2006. Then it was disrupted by changes in volatility. This produced overexposure and wild swings in EC. The drawbacks are pretty obvious. We will get overexposure at certain times when volatility is high. This is not necessarily bad if the larger volatility is on the upside but that might not be the case every time. Another drawback of this technique is that we do not compound and won't experience the benefits that it brings. Everything happens linearly. Some may consider this as an advantage. It is much easier to implement as other techniques. As everything happens linearly we can see how the system performs more clearly. The implementation is also pretty straightforward. Another disadvantage is that the periods with higher volatility will impact our results more. For example, if we do optimizations and use constant unit, optimizer will favor parameters that will have good results in those periods of high volatility.

4.1.2. Constant Value

$$Position\ Size = \frac{Value\ in\ \$}{Price}$$

[16]

Constant value will place a specific constant value in dollars on each trade. This technique will place a certain value in dollars in each trade. For example, if we have stock trading at 50 that we would like to purchase with 10000 $, we would buy 200 shares.

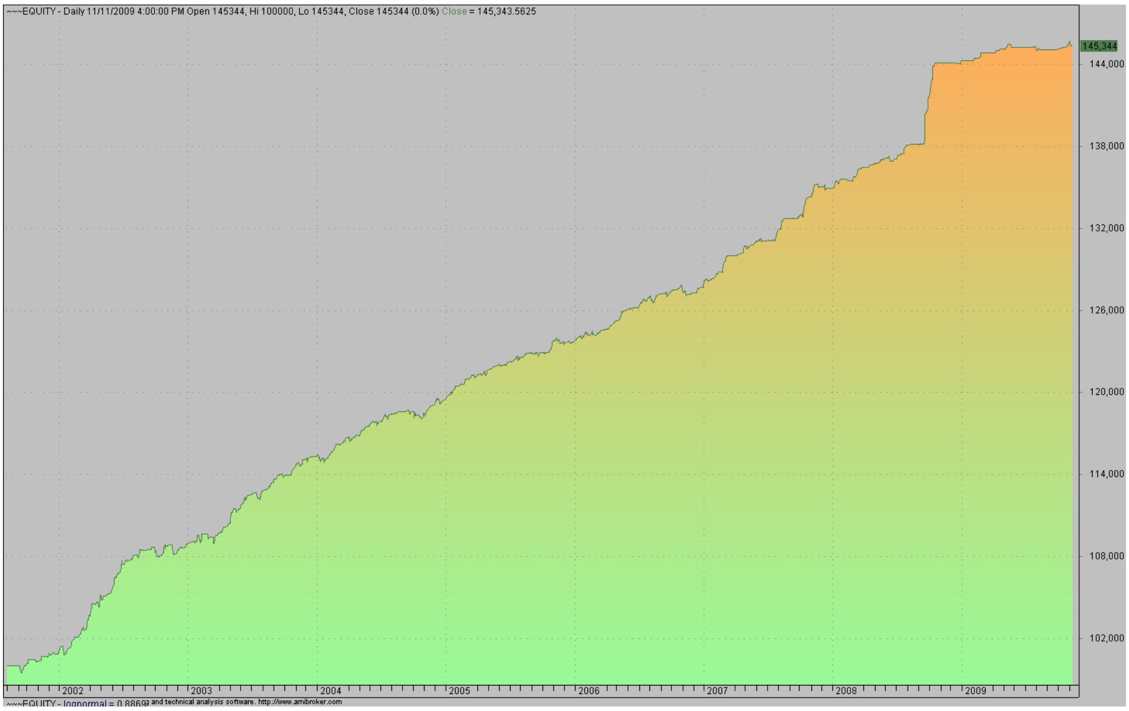


Figure 16: Constant value position sizing

Source: Own

In Figure 16 we can see constant value technique on the same system. The volatility lower than before and we obtained a much smoother equity curve. This is also because the volatility effect is somewhat accounted for but not fully. This technique is also quite simple to implement. This is true of most constant equity position sizing techniques. Everything happens linearly. Again, we don't get the benefits of compounding.

4.1.3. Constant Risk

This technique assumes that we know our worst case loss in advance. This is true of some systems. However, there are a lot of systems out there that don't have this attribute.

$$Position\ Size = \frac{Value\ Risked\ in\ \$}{Biggest\ loss\ in\ \$}$$

[17]

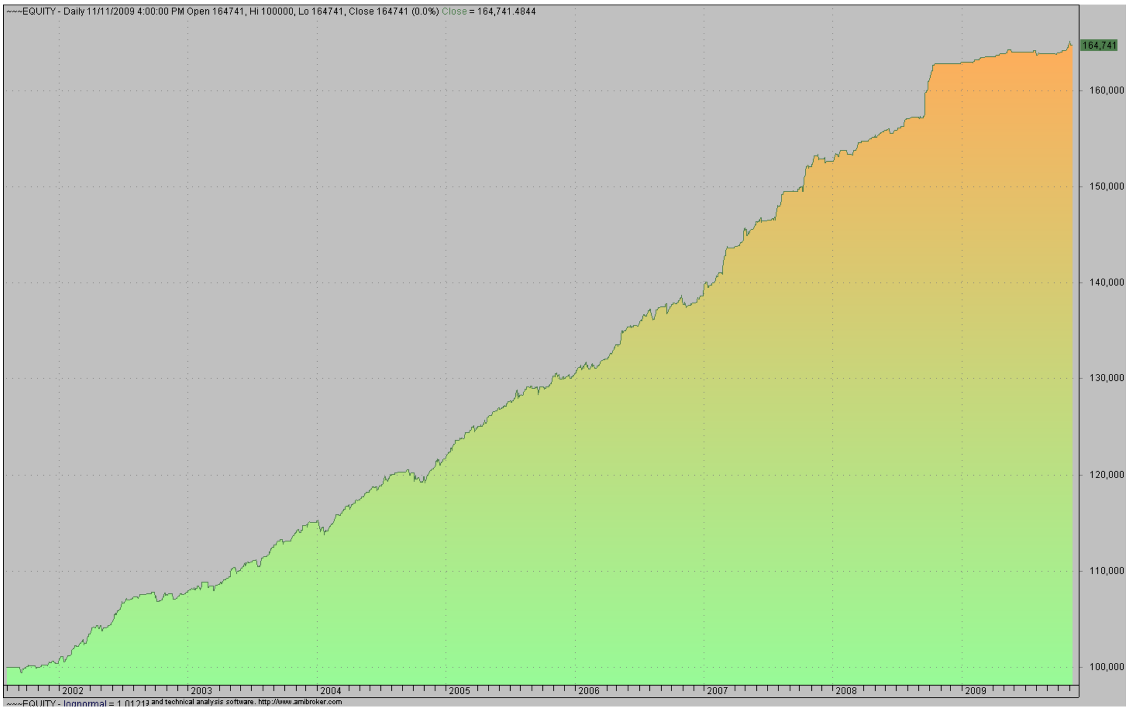


Figure 17: Constant risk position sizing

Source: Own

In Figure 17 we can see the effects of constant risk position sizing on our system. The equity curve is much smoother than before and volatility is relatively evenly distributed. It is slightly more difficult to implement and we have to know our biggest loss in advance. By biggest loss we mean largest loss that we will accept from the design point of view. This is not the biggest loss that will occur in case of some hardware or platform malfunction or sudden loss of liquidity.

4.1.4. Constant Volatility

We can use constant volatility even if we don't know our biggest loss in advance. Basically, we will normalize our position size based on benchmark volatility that we want to accept.

$$Position\ Size = \frac{Benchmark\ volatility}{Actual\ volatility} * Number\ of\ units$$

[18]

To measure actual volatility we can use standard deviation or any other suitable formula. This will have similar effects like constant risk position sizing. Main advantage is in the reduction of volatility of returns. The drawback is that it isn’t as easy to implement as other techniques, but it is worth exploring.

4.2. Variable equity position sizing

Variable equity techniques assume that our equity is changing from trade to trade. Our wins and losses impact our trading account and it grows or diminishes because of this. This is much closer to reality than fixed equity assumption. We are going to encounter a new phenomenon in the form of compounding. As our account grows we can grow our size in units or we can put fewer units on each trade. There are several considerations when trying to implement any kind of variable equity technique. First, our winning trades will accrue exponentially and our losing trades logarithmically. Second, we can turn a profitable system into an unprofitable one by applying this technique. We are going to explore this phenomenon in this chapter.

When applying these techniques, we have to decide, whether we want to grow our size as we have winners or should we rather put smaller size. This applies to losers as well. We will explore martingale and anti-martingale techniques next.

4.2.1. Martingale system

Martingale system is “a money management system of investing in which the dollar values of investments continually increase after losses, or the position size increases with lowering portfolio size.” (www.investopedia.com)

“The martingale category simply states that as the value of an account is decreasing, the size of following trades increases. The basic characteristic of the martingale is that as the account suffers losses, the ability to make up those losses either increases or stays the same.” (Ryan Jones, 1999, p.19)

Let’s have an example. We have a system that produces 50 % winners and 50 % losers with each win and loss equal to 1 $. This is basically a coin toss with zero expectation. By applying martingale system, we will change this expectation to 1 $. We will double our position after each loss.

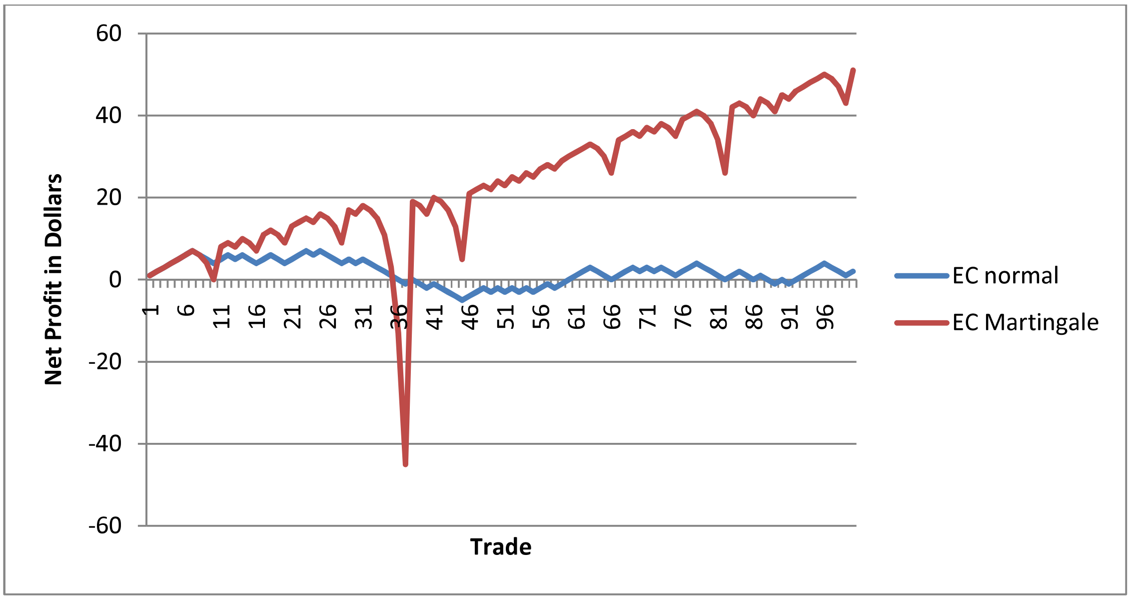


Figure 18: Martingale vs. Normal system

Source: Own

In Figure 18 we can see the comparison between martingale and normal system. This was done using random number generator in Excel to generate 100 random trades with our restrictions. We can see that with martingale we are going to make money with certainty. However, it applies only under one very important assumption. We will make money with certainty only if we have infinite wealth. Let's see the probabilities for runs of losing trades with different lengths.

# of losing trades in a row	Probability
1	0.5
2	0.25
4	0.0625
8	0.00390625
16	1.52588E-05
32	2.32831E-10
64	5.42101E-20
128	2.93874E-39

Figure 19: Probability of runs of losing trades

Source: Own

This assumption doesn't hold true for anyone. Therefore, if we are going to use martingale system, we are guaranteed to lose at some point in the future. This system is not viable for real world trading. There are some variations on this system. Increasing the size by 20 % of previous size can be one of them. The certainty of ruin is true for any martingale system.

4.2.2. Anti-martingale system

This system is the exact opposite of martingale. We increase our size as our equity increases. We put on more units after winning trades and fewer units after losing trades. The main advantage is that we will experience geometric growth. Drawback is that we can turn a profitable unit strategy into unprofitable strategy by using this system. It is because we are taking our losers with largest size.

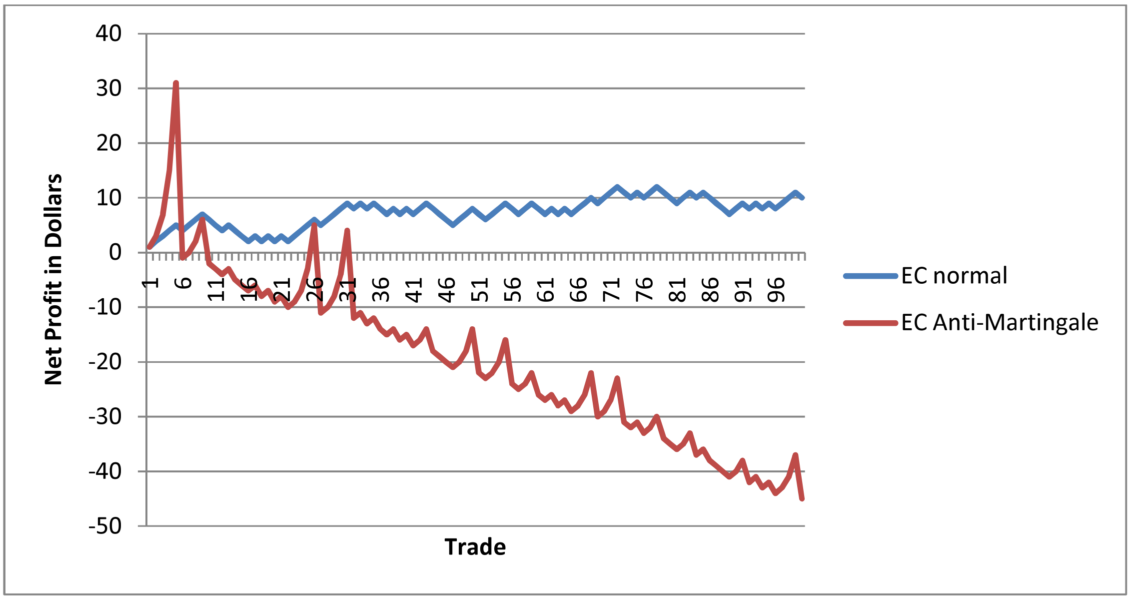


Figure 20: Anti-Martingale vs. Normal system

Source: Own

In Figure 20 we can see a rather extreme case of anti-martingale system that doubles after each win. There are other more conservative anti-martingale techniques that can be used and have relatively good results in real trading.

4.2.2.1. Fixed Fractional Position Sizing

This technique risks a constant fraction of equity on each trade. It is similar to constant risk. Advantages are geometric growth on the upside and logarithmic drawdowns on the downside. Drawbacks are that we can turn a strategy with positive expectations into losing strategy. Compared to constant risk we will have big variability of returns. Let's consider the following example. We have a strategy with positive expectations and would like to trade it risking 100 % of our equity on each trade. We will lose everything as soon as the first unprofitable trade comes. How much should we then risk when 100 % is clearly too much?

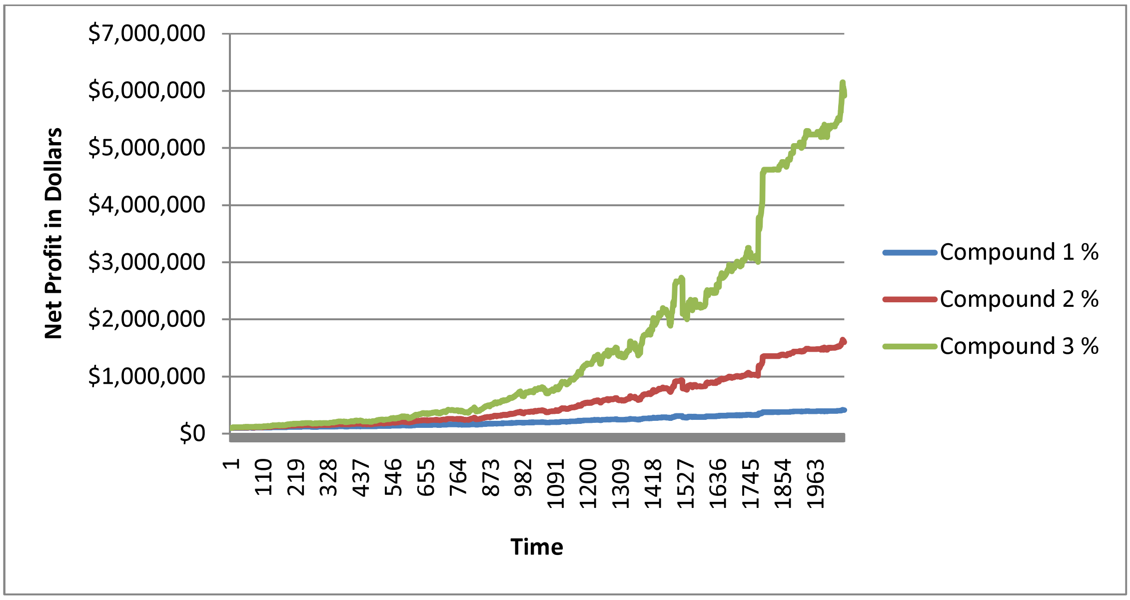


Figure 21: Compounding on 1, 2 and 3 %

Source: Own

In Figure 21 we can see the effects of compounding when using different fractions. What we can see is that we have higher net returns with higher fractions. We also have higher volatility. If we have a system with CAR/MDD of 1 under constant unit or constant risk position sizing, as soon as we apply fixed fractional with a fraction bigger or same as constant risk, our CAR/MDD will become smaller. On the other hand, net profit will rise substantially.

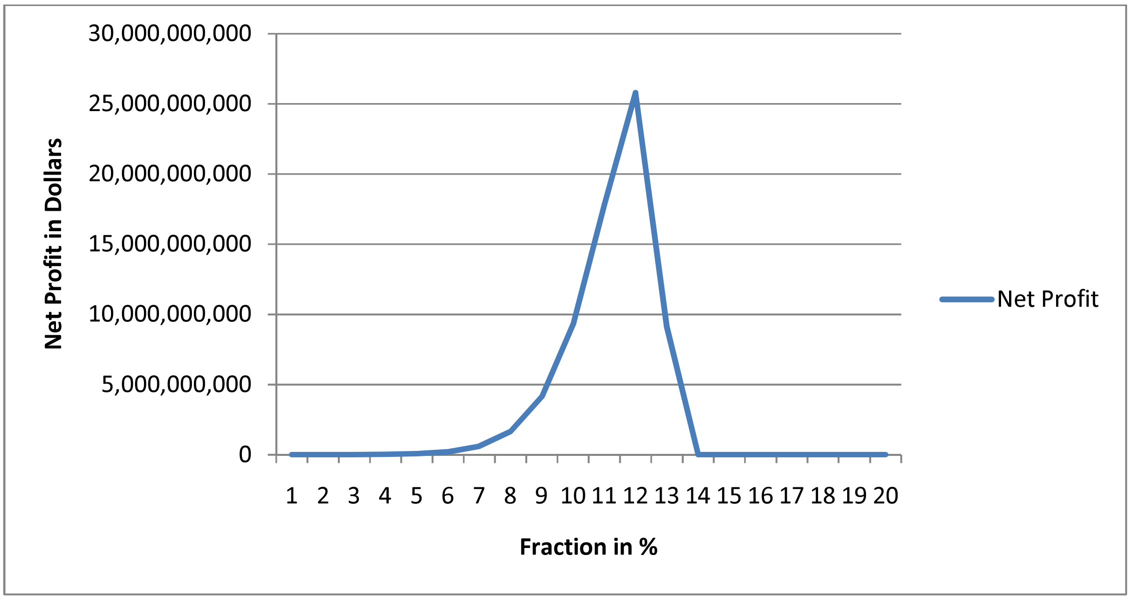


Figure 22: Net profit under different fractions in %

Source: Own

Figure 22 shows what happens to net profit when we increase fractions from 1 % to 20 %. The biggest gain is realized at 12 %. This point is called optimal F and every system has its optimal F. “Optimal f is defined as the fixed fraction that will yield more returns than any other fixed fraction applied to the same scenario.” (Ryan Jones, 1999, p.67) This is a mathematically optimal fraction to bet if we are looking at the net profit maximization. However there are several drawbacks of this method that render it very difficult to apply in real trading. First of all, the optimal F is calculated for particular situation and is different for every situation. It is also very sensitive to even short term changes in distribution of returns. Every system trading at optimal F will experience drawdowns at least as big as the fraction that is used. Better systems have higher optimal F and will also suffer bigger drawdowns. In conclusion, this method is mathematically optimal in producing the highest return in shortest period of time. However for real trading purposes is very difficult and dangerous to implement.

Now that we know the attributes of various techniques and know that there is a mathematically optimal solution to position sizing, the question of what fraction to use is still unanswered. It is possible and some books recommend using half of optimal F or more conservative approach. While it is better than nothing, this decision is completely arbitrary. We will now try to put forward a framework for position sizing based on our drawdown restrictions rather than on wealth maximization. While it's true that some investors are only looking for wealth maximization and this concept is still at the heart of many theories, it is also true that many investors have a pain threshold for drawdown. After this threshold is reached they abandon their systems or money managers. We will therefore try to construct a money management system starting with drawdown estimation. By no means are we claiming that this will be a new approach, as the techniques used here are relatively popular. However, we believe that this approach is not as common as those described above and deserves some exploration.

4.3. Drawdown Estimation

Every time we perform a backtest and get output in the form of returns, we are basically getting a sample from a population. Returns can be per trade or per period. They can be in percentage of equity or dollars. If we believe that this sample is a good representation of population, we can then try to obtain some characteristics of population by using sampling techniques. We are going to use bootstrapping as a technique to get a grasp on the distribution and try to estimate and get confidence intervals on drawdown. "Bootstrap is sampling with replacement from a sample. Bootstrap is sampling within a sample. The name may come from phrase "pull up by your own bootstraps" which mean 'rely on your own resources'. Bootstrap sampling relies on its own sample as often the only resources a researcher has. Bootstrap correspondence principle says that estimator of sub sampling (taken by bootstrap method) is equal to the estimate of sample."

- (Negative) Bootstrap method is not exact. For large sample, permutation test perform better than bootstrap.
- (Positive) Bootstrap requires very minimum assumption. Even if permutation test fail, bootstrap method still can do.
- (Positive) Though it can be used for parametric method (i.e. distribution is known), bootstrap method is most useful when the sample distribution is unknown (non-parametric). (Teknomo, Kardi. Bootstrap Sampling Tutorial. http://people.revoledu.com/kard/ tutorial/bootstrap/)

We are going to start with performing bootstrap sampling on the dollar returns of trades of optimized system. It is important to note that we can also perform bootstrapping on the period returns of some system (f. e. one day returns). It depends on the way the system works. Our system has been optimized and best parameters were chosen. Optimization goal was best Annualized RF or MRF with required return of 0. The best result was ARF of 4.43. We are not compounding and therefore choose this metric as relevant. Net Profit was 129,680 $ and maximum drawdown was 3,511 $. Now we can perform the sampling with 500 iterations and record net profit, maximum drawdown and ARF for each. What we end up with is a histogram with maximum drawdowns and ARF measures. From these we can make estimations about the kind of drawdown that we can see in the future.

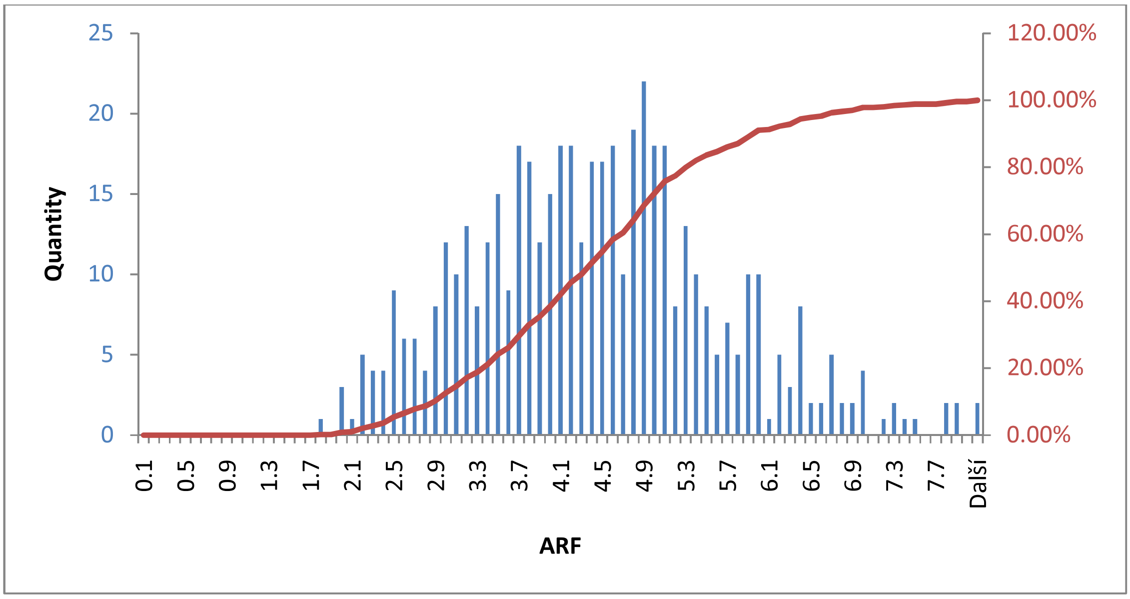


Figure 23: Histogram with ARF from sampling

Source: Own

Figure 23 shows a histogram of ARF measurements in samples that we created using bootstrapping. We can see that the median value is around 4.35, this is slightly lower as our optimized system. What we can do now is make a 90 % confidence interval. We can say that with 90 % probability the results can be expected to be above 2.9 for ARF. Figure 24 shows this kind of histogram for maximum drawdown and w can say that with 90 % we can expect the drawdown to be less than 5000 $. However, these results were produced by sampling the returns of optimized system now we are going to do the same process but with returns from walk forward analysis. We are talking about WFA in the chapter about optimization. For now we can say that results obtained using WFA are closer to the kind of results we can expect going forward. So the bootstrapping method will tell us more realistically what results we can expect in the future.

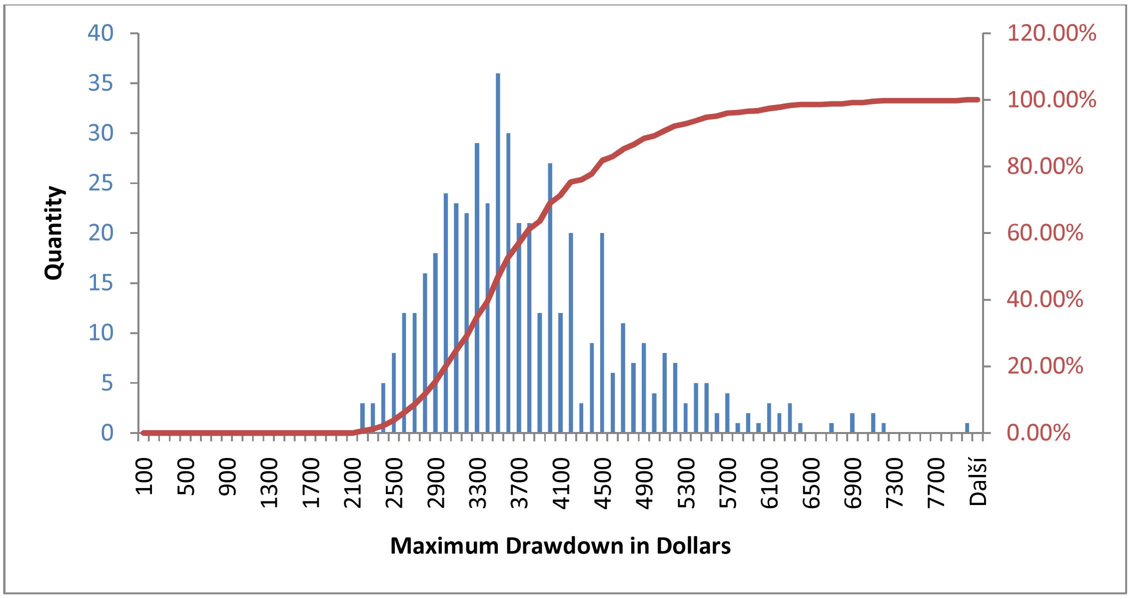


Figure 24: Maximum drawdown histogram from sampling

Source: Own

We performed a simple walk-forward analysis with one in-sample (IS) periods and one out-of-sample (OOS) periods with approximately the same length. We got the following result for OOS period. Our net profit was 69,080 $ and maximum drawdown was 4,298 $. Our ARF was 4.13. We now perform bootstrap sampling on the trades from OOS period. After that we got 90 % confidence interval for ARF of 3.9 and maximum drawdown of 4270 $. This was just a simple WFA with one IS periods and one OOS period. If we specify WFA with more realistic IS and OSS periods we would get more realistic results. One of the reasons for this sampling producing relatively good result is that the OOS periods produced exceptionally good result. The other reason for this is that the IS optimized result contained a string of losing trades that were not exactly replicated with sampling. Having said that, the results from bootstrapping are more conservative than the results from WFA alone. How do we then use these results? One way is to base position sizing on the maximum drawdown that we are capable to overcome. If we believe that we are not able to stomach drawdown of more than 2100 $, we can cut our positions to half so we get a 90 % interval of 2135 $. How does this relate to money management? As we said at the beginning we are trying to approach money management from the drawdown side. Each day we can rebalance and adjust our position size according to what our drawdown requirements are. This way we get the maximum possible return, given the drawdown constraint that we specified. This is not to say that we won't experience drawdown bigger than our estimations. If we trade long enough we can be certain that our drawdown in the future will be bigger than what we have seen in the past. However, by using this method we at least have some control and more realistic assumptions about possible returns and risks. In this section we explored various money management techniques and discussed their pros and cons. We introduced our own opinion on the subject of money management and approached it from a different point of view. We put forward a drawdown target and maximization of wealth given this restriction as opposed to just wealth maximization. We used sampling techniques to arrive at an estimation of future drawdown. Our recommendation would be to use these techniques on the out-of-sample results of systems to obtain the most realistic expectations going forward. The contribution is in the different view on the subject of money management with a focus on drawdowns instead of profit. We believe that this approach is more realistically reflecting thought processes of retail traders that have limited capital to play with; together with the method for future drawdown estimation we can get a more realistic view on what we could be facing and adjust our risk accordingly.

5. Optimization and robustness

In this chapter we talk about optimization and robustness. We will define optimization and try to explain robustness and how to measure it. We also dig deeper into various optimization algorithms, their advantages and drawbacks. “An optimizer is simply an entity or algorithm that attempts to find the best, possible solution to a problem; optimization is the search process by which that solution is discovered. Optimizers exist to find the best possible solution to a problem” (Jeffrey Owen Katz et al., 2000, p. 29)

We have to define the following:

- best possible solution
  - It is a solution to a problem according to our requirements. It is a solution that maximizes our expectations.
- problem
  - What parameters should we use for a trading system? What trading rules should we use for a trading system?

“The goodness of a solution or trading model, in terms of how well it performs when measured against some standard, is often called fitness. In practice, fitness is evaluated by a fitness function, a block of programming code that calculates a single number that reflects the relative desirability of any solution. A fitness function can be written to appraise fitness howsoever the trader desires.” (Jeffrey Owen Katz et al., 2000, p. 30)

We defined several fitness functions in the chapter about performance measure, RF, CAR/MDD, MRF, MCARDD can all be used as fitness functions. In other words, by optimizing, we are finding a set of parameters or/and rules that will lead to a best RF, CAR/MDD, MRF or MCARDD of a system.

Optimizers are usually already incorporated into the software that is used for testing. The sophistication of any particular optimizer depends on the algorithm that it uses. We recognize two main types of optimizers:

- exhaustive or brute-force algorithms
- “smart” algorithms

5.1. Exhaustive optimizers

We have a system that has two parameters, A and B. Each of these parameters can vary from 1 to 100 by increments of 1. By doing and exhaustive optimization, we will iterate through each step of A and B and get 100*100 results. The pseudo-code for this type of optimization would be:

```
for A=1 to 100 begin
        for B=1 to 100 begin
                //algorithm that we want to implement and get results for
        end;
end;
```

We will get 10,000 results of a system with different parameters. We might also get 10,000 different systems, depending on the definition of a system that we choose. If we rank these results according to desired fitness function, we can then pick the system or parameters with the best value. Advantage is that it will explore all possible combinations and can find a global optimum.

This type of optimization is sufficient for a majority of problems that we encounter. However, sometimes we might have such a number of parameters that this type of optimization would be difficult to implement. Drawback is that it takes a lot of time when we have big parameter space that we need to explore.

5.2. Smart optimizers

Let’s imagine that we want to optimize a system with 10 parameters. Each parameter can vary from 1 to 100 by increments of 1. We have to do 100^10 iterations to get to the best result. If we assume that every iteration takes approximately one second we will need approximately 3.17*10^12 years to get the job done. So clearly, this type of algorithm is not suitable for the problem we face. Therefore, we have to use smart optimization. We are not going to explain different algorithms in detail. However, general smart optimization process is described below (taken from AmiBroker 5.30 User's Guide):

1. The optimizer generates some (usually random) starting population of parameter sets
2. Backtest is performed for each parameter set from the population
3. The results of backtests are evaluated according to the logic of algorithm
4. New population is generated based on the evolution of results
5. If new best is found - save it and go to step 2 until stop criteria are met

Example stop criteria can include:

- Reaching specified maximum iterations
- Stop if the range of best objective values of last X generations is zero
- Stop if adding 0.1 standard deviation vector in any principal axis direction does not change the value of objective value
- Others

Smart algorithms includes but are not limited to: Standard Particle Swarm Optimizer, Adaptive Parameter-less Particle Swarm Optimizer, Covariance Matrix Adaptation Evolutionary Strategy optimizer.

The advantage of such algorithms is mainly speed. The disadvantage is that they best work on smooth continuous parameter spaces and objective functions. This means that binary parameters or "switches" affect these negatively. Also the problem with searching such a vast parameter space is that we might find local optimums rather than global optimums.

5.3. Robustness and Over-fitting

The problem when using optimizers is over-fitting or curve-fitting. "Data curve fitting occurs when system developers eliminate a portion of their historical data or intentionally reduce their historical data series at the study's outset to filter out losing trades. Parameter curve fitting can be defined as the practice of the system developer adapting trade criteria parameters to match or fit in-sample data.” (Richard L. Weissman, 2005, p. 124)

There are many definitions but for the practical implementation we want to ensure that the systems work in the future as they worked in the past or better. There is no guarantee that they will but this is what we should aim for. We defined data-snooping bias and robustness before. We are therefore going to focus on robustness measurement and identification of over-fitted systems. In general and simplified way, small samples and large parameter spaces are reasons for issues with robustness. In layman’s terms we have to aim for as simple systems as possible that work on as largest samples as possible, given that we want to find relatively robust systems.

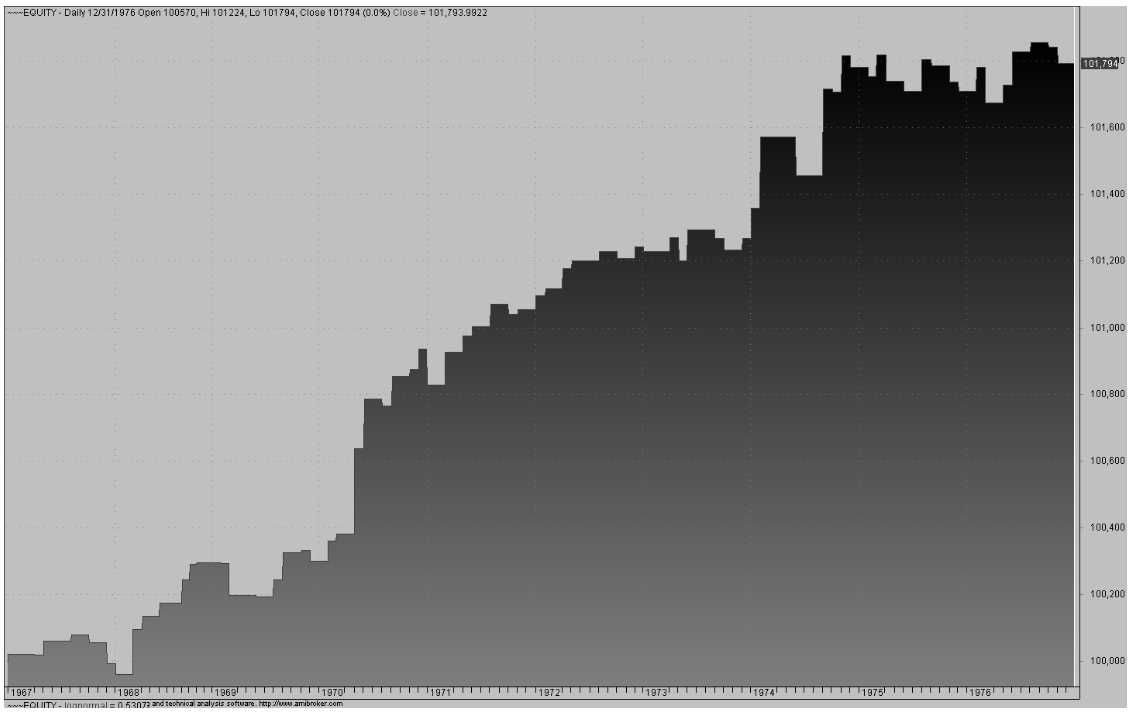


Figure 25: Optimized system with three parameters

Source: Own

In Figure 25 we can see a system with three parameters optimized over ten years worth of daily data. We then apply this optimized system on the next ten years of data and see what happens. In Figure 26 we can see the effect of trading the same system for the next ten years on unseen or OOS data. This system was clearly fitted to the data on which it was optimized and doesn’t work going forward. This is an example of over-fitted system. What is interesting is the fact that we didn’t need a lot of parameters to render the system useless in OOS. Even with only 3 parameters there is much room for over-fitting.

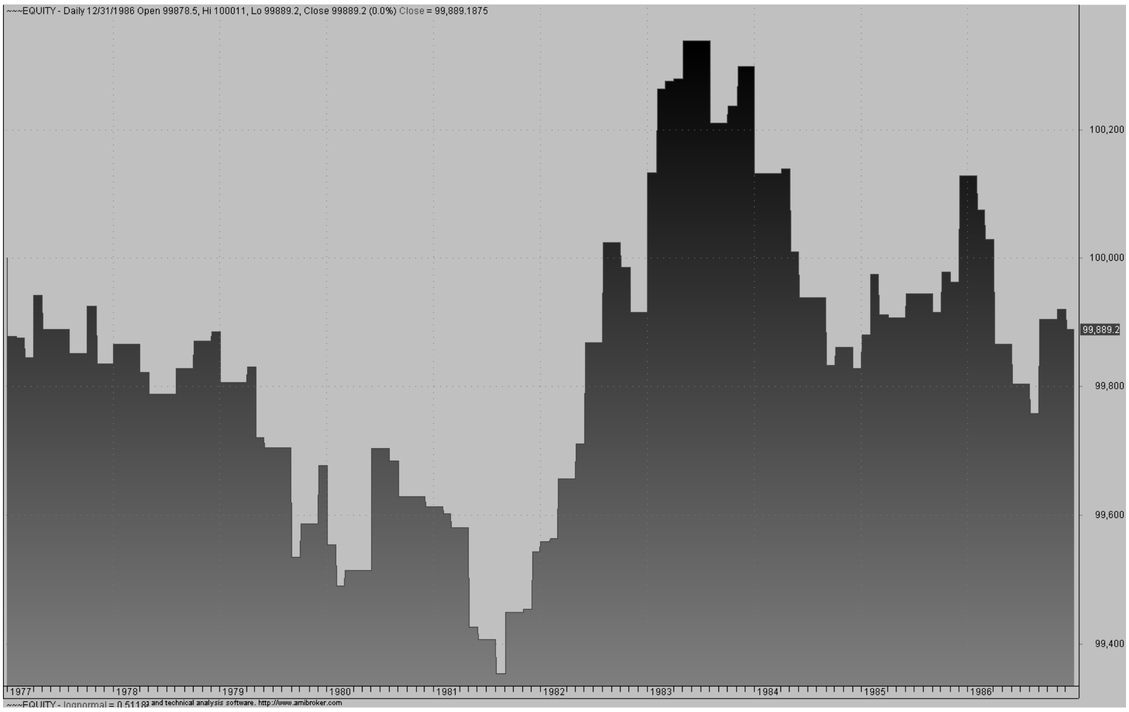


Figure 26: Next ten years of trading the same system

Source: Own

The question then is: how we can detect if particular system is likely to fail going forward and how can we measure and compare different systems for robustness? We will start with the first part of the question. To find out how the system is probably going to work in the future, we can simulate the process of optimization as it is going to be performed in the future. This is called Walk-Forward Analysis (WFA) or Walk-Forward Optimization (WFO).

5.3.1. Walk-Forward Analysis

The point of doing WFA is to simulate the optimization-implementation process on the data. We optimize on one part of data. This data is called in-sample (IS) data. We than use the parameters obtained on the subsequent part of data called out-of-sample (OOS) data. We can see the whole process in Figure 27. We optimize on IS data and then roll forward one period. We then stitch the OOS periods together and obtain OOS results. These results are what we are probably going to see if we implement the system going forward.

Periods	1	2	3	4	5	6	7	8	9	10	11	12
1st Step	IS				OOS							
2nd Step		IS				OOS						
3rd Step			IS				OOS					
4th Step				IS				OOS				
5th Step					IS				OOS			
6th Step						IS				OOS		
7th Step							IS				OOS	
8th Step								IS				OOS
OOS Results				OOS	OOS	OOS	OOS	OOS	OOS	OOS	OOS	OOS

Figure 27: Walk-Forward Analysis

Source: Own

There are several issues that need to be addressed before doing this type of analysis. First, we introduced two new parameters to our strategy. The length of IS and OOS periods. We have to decide upon the length of these periods. In general, it makes little sense to specify shorter IS and longer OOS periods. Usually, it is the other way around. One can experiment with different lengths and even optimize these for the best results. There are other ways of figuring out the appropriate IS and OOS periods but that is not in the scope of this thesis. We should note that it is also possible to optimize after number of trades executed or any other criterion. These are however more difficult to code. As a rule of thumb we can use the ratio of 10 to 1 in favor of IS data length. For 10 months IS we can use 1 month OOS. This is the more traditional approach. What we propose is to specify as short OOS period as possible. It can be one trade or one period after which we are able to do re-optimization. In the system performance reports at the end of this thesis we will generally use one day as OOS period and re-optimize after each day or sooner if possible. The rationale behind this is that we would like the system to be as up to date as possible and have the most recent information. Following this rationale it doesn't make sense to optimize every month but rather as soon as possible given the restrictions imposed upon us by optimization time and data availability. We can also decide to do an anchored WFA in which case we will not roll the periods but will use all the data that we have for optimization until the OOS period. In Figure 28 we can see a

WFA of a real world system with arbitrary selection of IS and OOS periods. The top EC is compounded and the bottom shows log scaled EC.

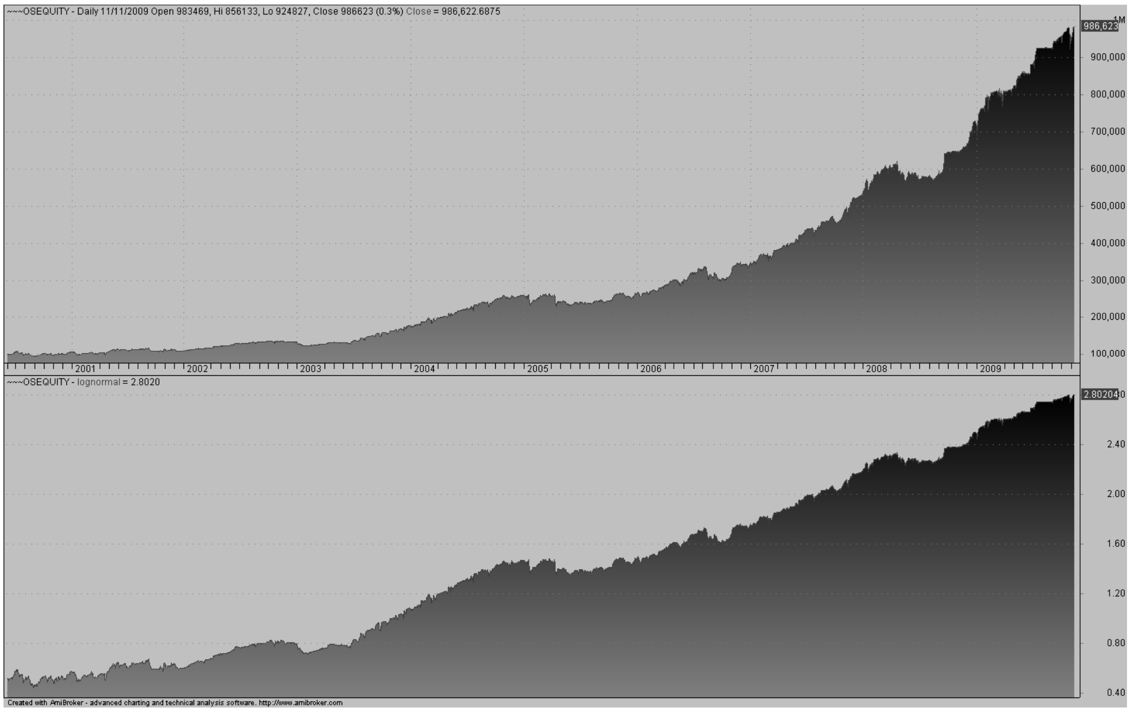


Figure 28: WFA OOS results with arbitrary 1 and 0.5 year IS and OOS periods

Source: Own

5.3.2. Visual inspection of robustness

We can use other methods to find out whether the system is going to perform well in the future. If we have a system that has two or less variables, we can plot the parameters and fitness function on the graph to see how it changes over the whole parameter space. There is one issue that we must address before we do this. For example, we want to do optimization on the system using moving average and to vary the length of moving average from 1 to 200 units, incrementing by one. First, we have made decision about the parameter length and then which is 200 units and then about the granularity which is 1 unit. This decision is arbitrary and we must take that into consideration. We might as well iterate by 5 from 20 to 100. These decisions will have an impact on sensitivity analysis and robustness measurements. We don’t have non-arbitrary solution for this at the moment. However, it makes sense to choose length of a parameter in such a way that it will include all reasonably possible options for this parameter that can be used in real world trading and then some. The same goes for granularity. If we choose a granularity of 50 in case of moving average for WFA or sensitivity analysis, we shouldn't then try to pick a length of 159 periods for real trading. The goal is to test as one wants to trade. After we chose parameters, their length and granularity we can perform exhaustive optimization on the whole dataset and then plot the parameters and fitness function. We will use 3D graph for two parameters and 2D for one parameter.

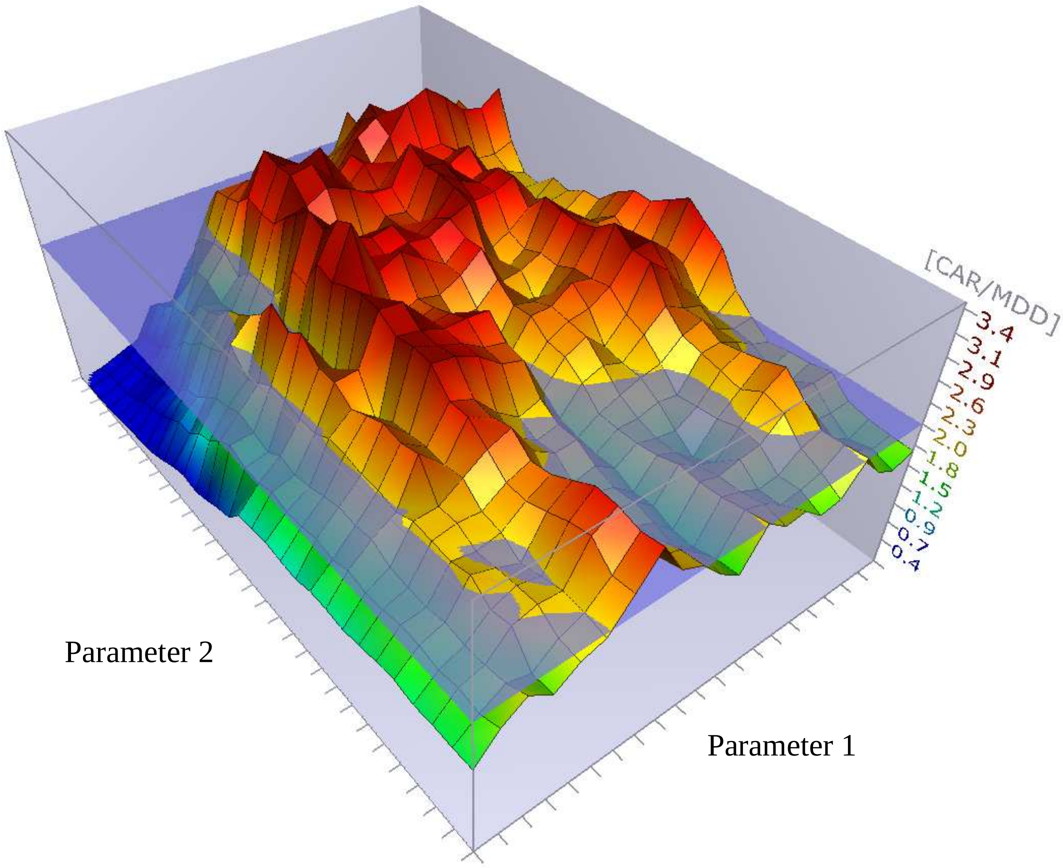


Figure 29: 3D graph with two parameters and CAR/MDD fitness function

Source: Own

In Figure 29 we can see how the different parameters influence our fitness function. We can also see that the surface is relatively smooth with majority of results above 2.0 watermark. It suggests that this is a relatively robust system and can be expected to perform well with majority of parameters. On the other hand in Figure 30 we can see the system that failed in our WFA. It is obvious that this system has a very "spiky" surface suggesting not very robust system.

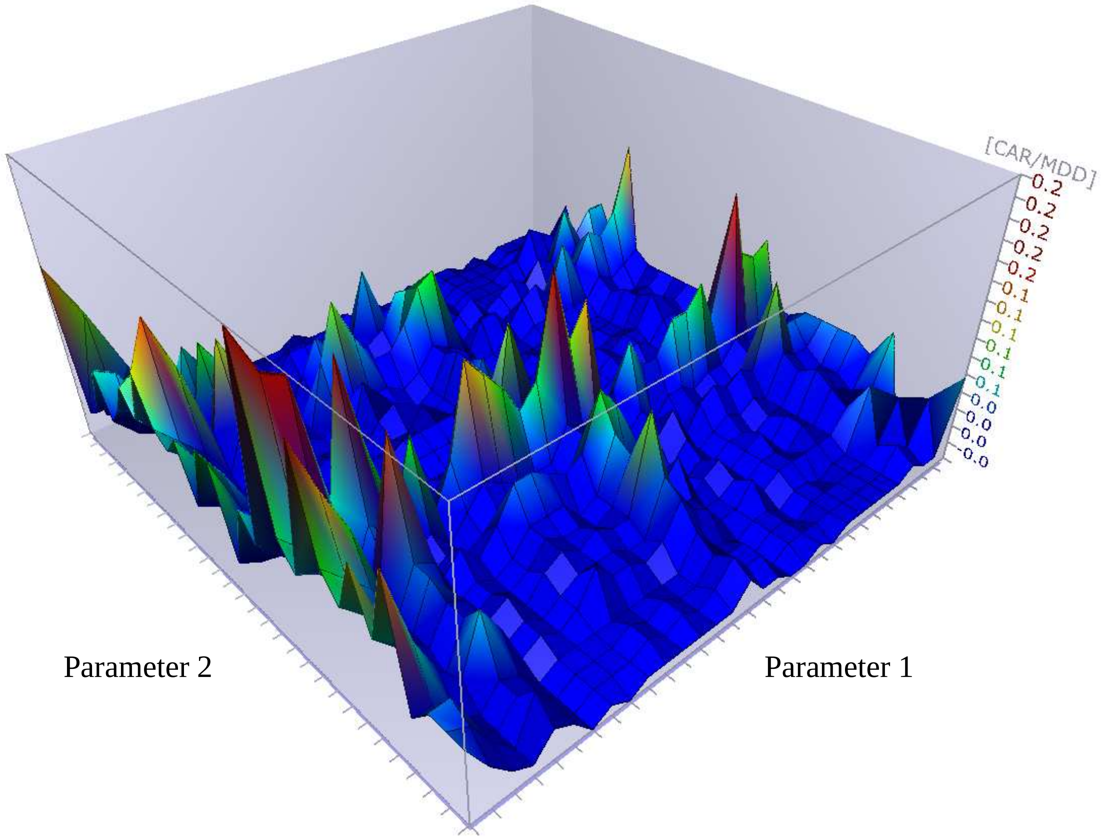


Figure 30: 3D graph with two parameters and CAR/MDD fitness function
Source: Own

We can inspect visually whether the system is robust or not, but how can we measure this robustness and arrive at number that can be used for comparison between different systems? How will we decide between two systems, given the fact that we can only choose one? We can use the process already described here: WFA and visual inspection of parameter space. First, this is not applicable on systems with more than two parameters and second, visual inspection can be subjective. We might clearly see the difference when it is obvious but assessing two similar systems in terms of parameter space can be tricky. We will therefore try to explore different ways of measuring robustness. We will start with very simple ones and move on to others. However, it is important to note that this process is very much dependent on parameter length and granularity specification and GIGO applies here (garbage in garbage out). If we specify lengths that we know are profitable, we will get bogus robustness measurements and the results are useless for decision-making.

	CAR/MDD	
Parameter	**System A**	**System B**
1	0.5	0.2
2	0.7	0.6
3	0.8	0.2
4	0.9	0.8
5	1.2	0.2
6	0.9	1.4
7	1.2	0.7
8	1.1	1.5
9	1	0.3
10	0.9	0.1
11	0.7	0.5
12	0.8	1.1
13	0.5	0.2
14	0.5	0.1
15	0.5	1
Average	0.81	0.59
Median	0.8	0.5

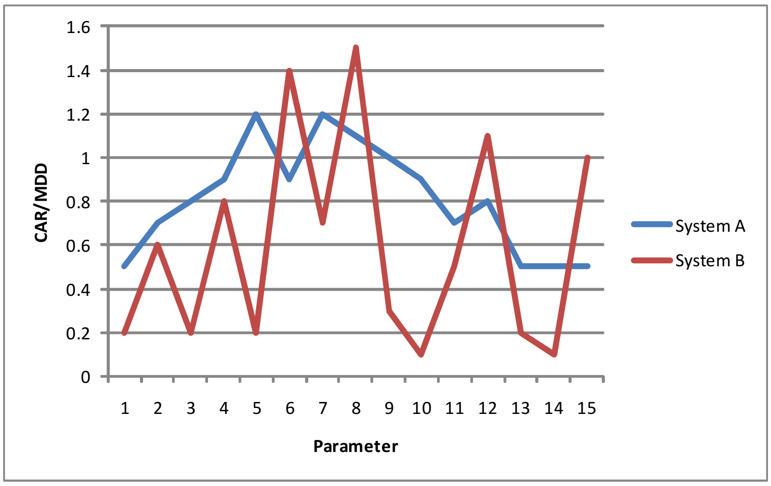


Figure 31: Two system with simple robustness measurements

Source: Own

5.3.3. Robustness measurements

We can start with the very simple robustness measurements. In Figure 31 we have two fictitious systems and their optimization results. System B has very inconsistent results but also some parameters produce relatively high CAR/MDD. System A is more consistent. We can compute the average result to get an idea about consistency. Since average can be influenced by outlier we reckon that median is a better measurement of consistency. We arrived at a number that will tell us something about the robustness of a system. Systems with higher median value of all the reasonable parameters can be labeled as more robust than those with lower median. To get even more confidence we can make a histogram of optimization results such as in Figure 32. From this we can take a 90 % confidence interval for lowest value and that is around 1.3 CAR/MDD. The following results are those of a real system.

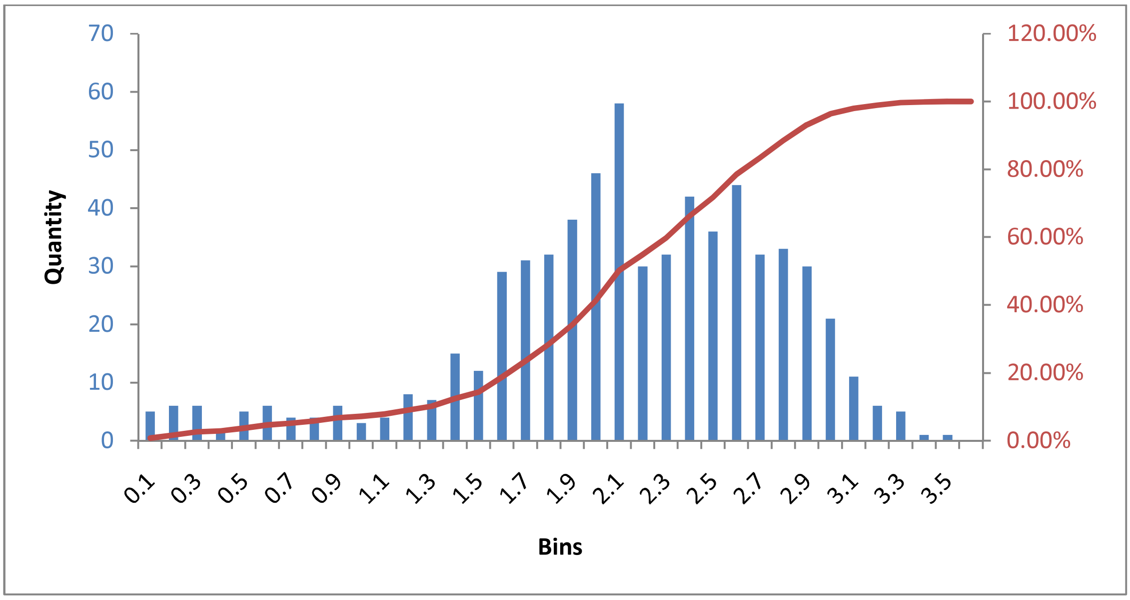


Figure 32: Histogram of optimization results

Source: Own

Another option is to measure the variability between the results. To arrive at measure of robustness we can compute the standard deviation of results and divide the median by standard deviation. This way we will obtain a robustness measure per unit of variability within the results:

$$Robustness = \frac{Median\ of\ results}{Standard\ Deviation\ of\ results}$$

[19]

For our example:

$$Robustness = \frac{2.1}{0.655} = 3.2$$

These measures take into account the profitability of the system. We can use them as standalone metrics if we want to but they are better put to use as comparison tools between several systems. This is applicable to any system with any number of parameters.

In this chapter we explored the issue of over-fitting. We described a method for systems testing that can reveal this problem. We introduced several measurements of robustness and explained how they work and how to use them. There are many others that can be derived from these. The condition is that they will capture differences between optimization results and express them as some interpretable number. They have the advantage over the subjective visual methods in being able to work with as many parameters as required and arriving at precise number that can be used for comparison of many systems. The introduction of several robustness measurement methods and reconciliation of subjective robustness assessment is the main contribution of this part.

6. Systems performance results

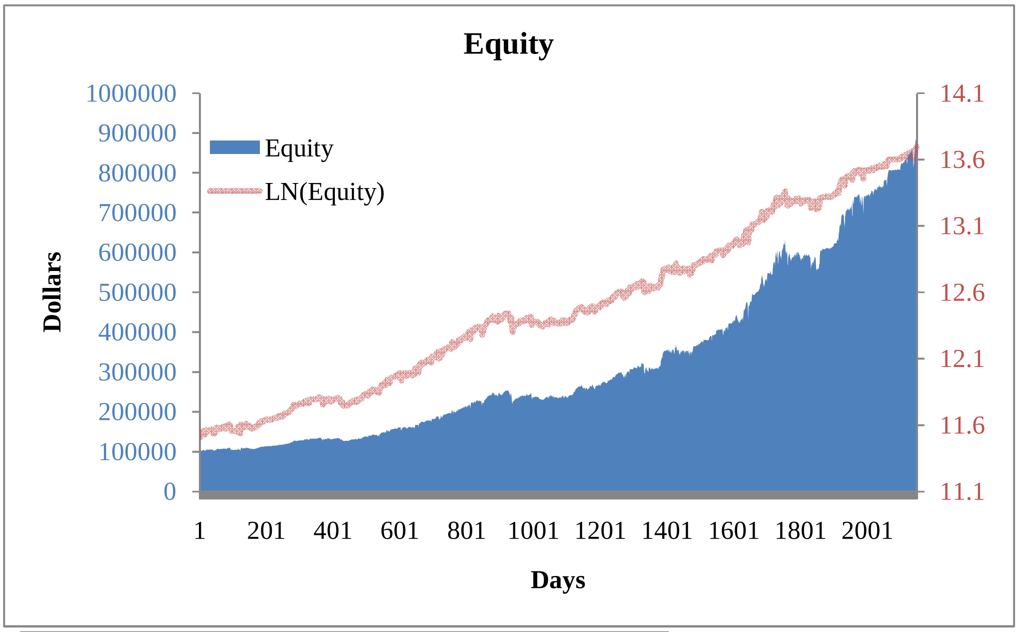
In this section we will present results for systems that we developed. We followed the methodology presented in this thesis during the development. All results are out-of-sample. We picked one length of in-sample period and used that length to generate more detailed backtesting report. We present equity curve, both normal and logarithmic, for the chosen IS period length. We also presented results for other in-sample lengths and calculated their median and average. We disclosed what data, type of orders or cost assumptions were used during backtesting.

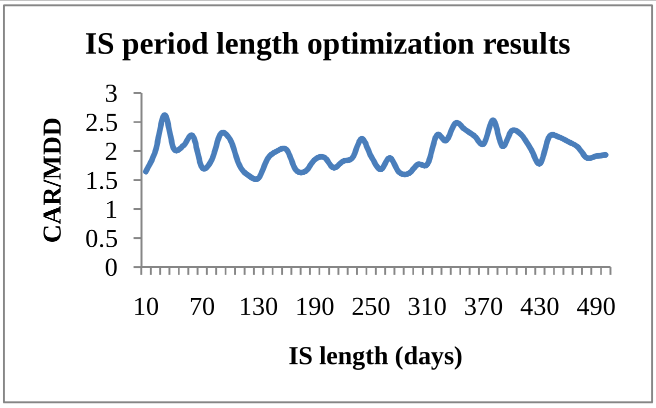
System A: Long/short system

All results are out-of-sample. In-sample period length of 240 days is chosen for detailed analysis. Other in-sample period lengths are also tested and results are presented below. For calculating Sharpe ratio, 3.5% risk free rate is used.

Data: 100 components of NASDAQ 100 index from 5/1/2001 to 11/11/2009

	All trades	Long trades	Short trades
Initial capital	100,000	100,000	100,000
Ending capital	886,899	695,246	291,653
Net profit	786,899	595,246	191,653
Annual Return (%)	29.13%	25.50%	13.36%
Max Drawdown (%)	-13.18%	-15.62%	-41.95%
CAR/MDD	2.21	1.63	0.32
Profit Factor	1.67	2.08	1.31
Sharpe Ratio	1.56	1.94	1.10





	CAR/MDD
Median:	1.96
Average:	1.99

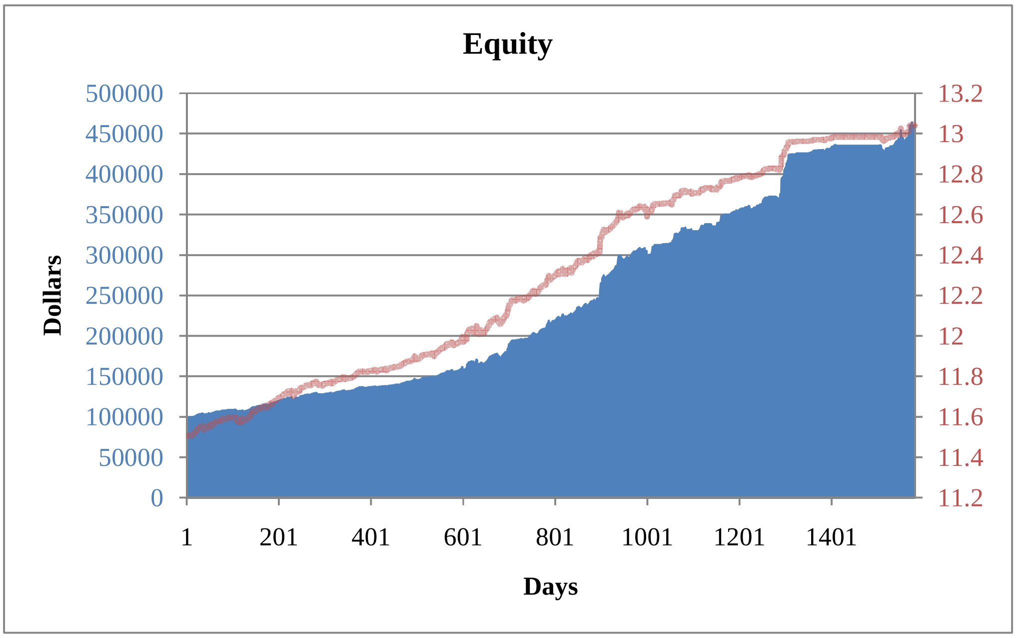
All results are out-of-sample. System uses market and limit orders. We assumed slippage of 0.02 $ per share and transactional costs of 0.01 $ per share with a minimum of 1 $.

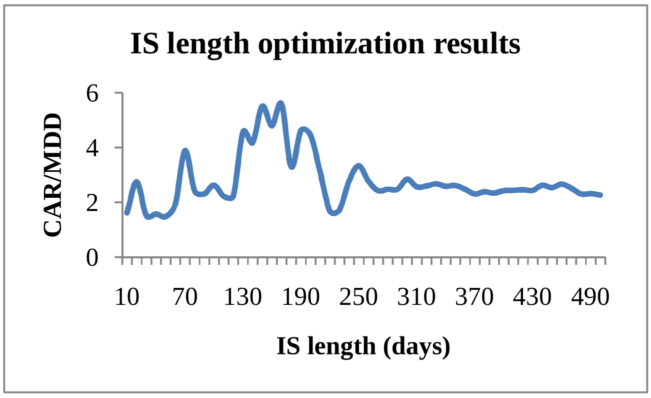
System B: Long system

All results are out-of-sample. In-sample period length of 160 days is chosen for detailed analysis. Other in-sample period lengths are also tested and results are presented below. For calculating Sharpe ratio, 3.5% risk free rate is used. This is intra-day system.

Data: 100 components of NASDAQ 100 index from 8/5/2003 to 11/11/2009

	All trades	Long trades
Initial capital	100,000	100,000
Ending capital	459,967	459,967
Net profit	359,967	359,967
Annual Return (%)	27.54%	27.54%
Max Drawdown (%)	-5.73%	-5.73%
CAR/MDD	4.80	4.80
Profit Factor	1.77	1.77
Sharpe Ratio	5.78	5.78





	CAR/MDD
Median:	2.50
Average:	2.78

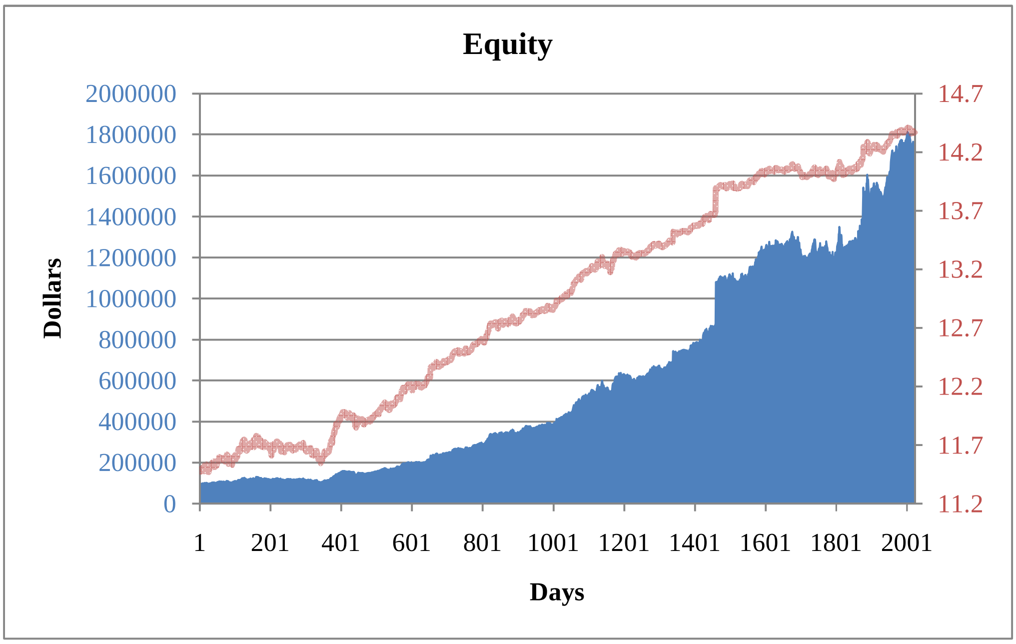
All results are out-of-sample. System uses market and limit orders. We assumed slippage of 0.02 $ per share and transactional costs of 0.01 $ per share with a minimum of 1 $.

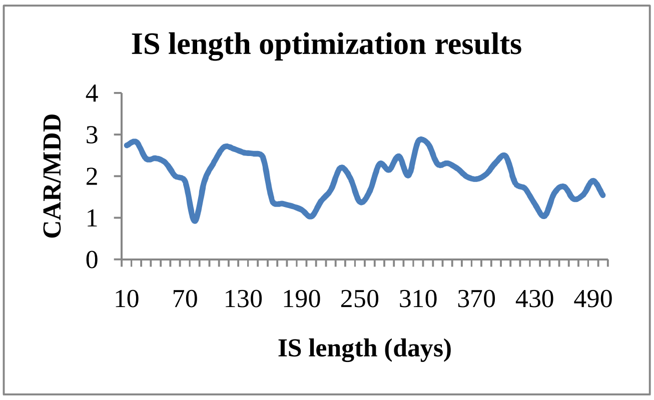
System C: Dollar neutral system

All results are out-of-sample. In-sample period length of 300 days is chosen for detailed analysis. Other in-sample period lengths are also tested and results are presented below. For calculating Sharpe ratio, 3.5% risk free rate is used. This is a longer-term hedged system.

Data: 100 components of NASDAQ 100 index from 11/1/2001 to 11/11/2009

	All trades	Long trades	Short trades
Initial capital	100,000	100,000	100,000
Ending capital	1,734,860	2,132,299	-297,438
Net profit	1,634,860	2,032,299	-397,438
Annual Return (%)	42.65%	46.36%	N/A
Max Drawdown (%)	-21.11%	-52.42%	-99.28%
CAR/MDD	2.02	0.88	N/A
Profit Factor	1.41	1.94	0.78
Sharpe Ratio	0.64	0.69	-0.49





	CAR/MDD
Median:	2.00
Average:	1.98

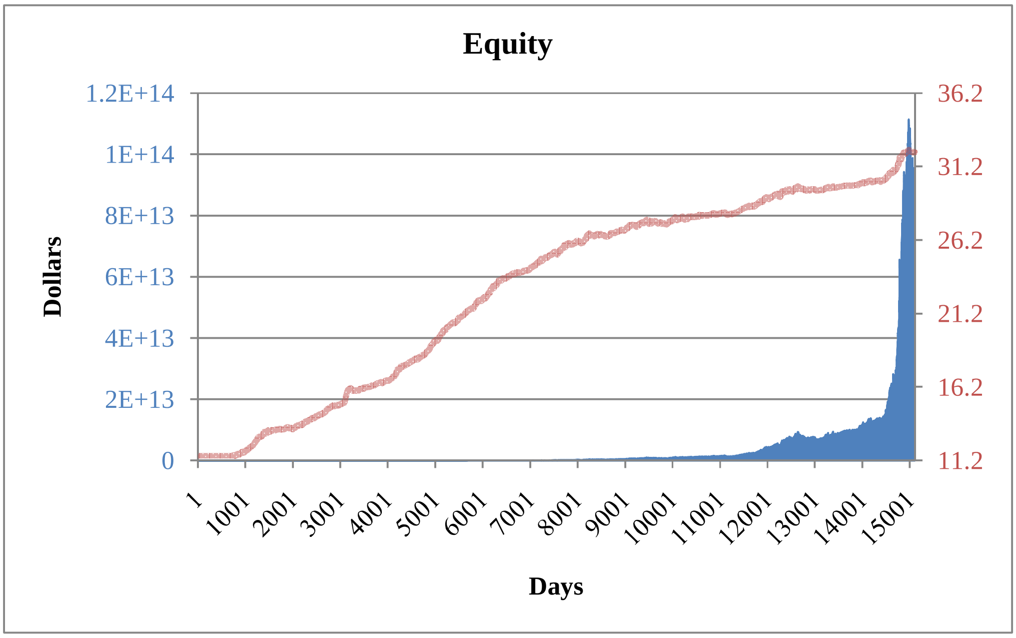
All results are out-of-sample. System uses market orders. We assumed slippage of 0.02 $ per share and transactional costs of 0.01 $ per share with a minimum of 1 $. Dividends not included.

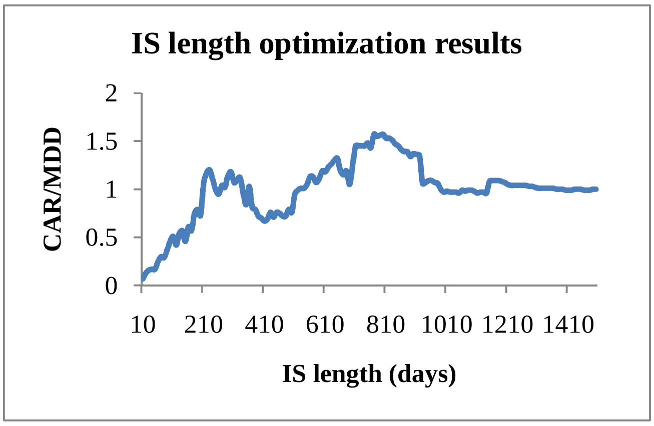
System D: Long/short mutual fund system

All results are out-of-sample. In-sample period length of 710 days is chosen for detailed analysis. Other in-sample period lengths are also tested and results are presented below.

Data: SP500 index data to emulate mutual funds data from 1/3/1950 to 1/12/2010

	All trades
Initial capital	100,000
Ending capital	9.49949E+13
Net profit	9.49949E+13
Annual Return (%)	41.08%
Max Drawdown (%)	-20.47%
CAR/MDD	1.45





	CAR/MDD
Median:	1.01
Average:	1.00

All results are out-of-sample. System uses market orders. There are no transaction costs assumed as mutual funds only have small expense ratio.

Resume

We put forward a framework for system design, development and evaluation in this thesis. We identified main issues pertaining to the subject of:

a) Performance measurement in Chapter 3
b) Money management in Chapter 4
c) Optimization and robustness in Chapter 5

We explored known solutions and offered some of our own. We also provided several system results that were developed using ideas and methods presented in this thesis. With this in mind we are convinced that we fulfilled our main and partial goals sufficiently as presented in foreword and together these findings can work as a basic framework for system design and development.

The contribution is in the development of framework, identification of several pitfalls associated with the process in Chapter 2, adjusting the existing metrics to better measure performance in Chapter 3, reconciliation of money management techniques and proposing a different take on this subject in Chapter 4, introduction of robustness measures and solving optimization and robustness pitfalls in Chapter 5.

We believe that there is a lot of work to be done in the areas of robustness and optimization. While money management and performance measurements are not regarded as closed subject, we think that they need to be tailored to the specific needs of the user. Therefore it is harder to seek an optimum solution.

The future research will probably be in self-optimizing or adaptive systems and refinements of the existing ones. There are several projects we would like to explore including application of signal processing methods and machine learning. There is a constant need for improvement in the investment business as it is one of the most competitive.

References

Aldridge, I. 2009. High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems. New Jersey : John Wiley & Sons, 2009. 0-470-56376-1.

Aronson, D. 2006. Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals. New Jersey : John Wiley & Sons, 2006. 0-470-00874-1.

Chan, E. 2009. Quantitative Trading: How To Build Your Own Algorithmic Trading Business. New Jersey : John Wiley & Sons, 2009. 978-0-470-28488-9.

Goldberg, R. 2008. The Battle for Wall Street: Behind the Lines in the Struggle That Pushed an Industry into Turmoil. New Jersey : John Wiley & Sons, 2008. 0-470-22279-4.

Investopedia. Investopedia. Investopedia. [Online] [Cited: May 20, 2010.] http://www.investopedia.com/.

Janeczko, T. 2010. AmiBroker 5.30 User's Guide. [Manual] 2010.

Jones, R. 1999. The Trading Game: Playing by the Numbers to Make Millions. New Jersey : John Wiley & Sons, 1999. 0-471-31698-9.

Katuscak, D. 2004. Ako pisat zaverecne a kvalifikacne prace. Nitra : Enigma, 2004. 80-89132-10-3.

Katz, J et al. 2000. The Encycklopedia of Trading Strategies. USA : McGraw-Hill, 2000. 0-07-058099-5.

Kaufmann, P. J. 2005. The New Trading Systems and Methods. New Jersey : John Wiley & Sons, 2005. 0-471-26847-X.

Selectorweb. Selectorweb. Selectorweb. [Online] [Cited: May 15, 2010.] http://www.selectorweb.com/algorithmic_trading.html.

Teknomo, K. Kardi Teknomo's Page. Kardi Teknomo's Page. [Online] [Cited: May 20, 2010.] http://people.revoledu.com/kardi/tutorial/.

The Sharpe Ratio. Sharpe, W. F. 1994. New York : Institutional Investor, 1994. 0095-4918.

Vidyamurthy, G. 2004. Pairs Trading: Quantitative Methods and Analysis. New Jersey : John Wiley & Sons, 2004. 0-471-46067-2.

Weissman, R. L. 2004. Mechanical Trading Systems: Pairing Trader Psychology with Technical Analysis. New Jersey : John Wiley & Sons, 2004. 0-471-65435-3.