March 2012

INVESTMENT AND TRADING

Adrian Delphia
Worcester Polytechnic Institute

Brendan Hamm
Worcester Polytechnic Institute

Samuel Thomas Veilleux
Worcester Polytechnic Institute

Srinivas Vasudevan
Worcester Polytechnic Institute

INVESTMENT AND TRADING

An Interactive Qualifying Project Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the
Degree of Bachelor of Science
by:

Adrian C. Delphia

Brendan M. Hamm

Srinivas Vasudevan

Samuel T. Veilleux

Submitted Tuesday March 13th, 2012

Advised by Professor Hossein Hakim

Table of Contents

List of Figures

List of Tables

This page intentionally left blank

Abstract

The goal of this project was to create a successful trading strategy for use in the forex market and create a positive track record which could be used to launch a money management company. Several different trading strategies were considered, and were subsequently tested both through live trading and through programming automated trading robots. Additionally, other programs were created along the way, both to aid in manual trading and in data gathering and processing. With our performance history we then sought out possibilities for launching a money management company. Such strategies and examinations of the markets illuminate both the benefits and detriments of several potential trading philosophies, and provide a solid background for the beginning trader.

Chapter 1: Introduction

1.1 Intro to Markets

In the world of finance, a capital market is any place real or virtual in which financial instruments are traded between different parties. There exist many different kinds of markets in existence today, but they all operate by way of financial transactions of securities for currency. A variety of financial instruments are traded on a regular basis, including stocks, bonds, commodities, and even currency itself. All financial markets operate on the basic principles of supply and demand and thus the basis of investing is to create a portfolio which is expected to rise in value over the course of time (i.e. is expected to increase in demand). [43, 44, 45]

The most well-known of all the markets is the stock market, where partitions of companies are bought and sold. There are many different stock markets around the world, some of them physical trading rooms like the New York Stock Exchange (NYSE) while some only provide electronic transactions like the NASDAQ exchange. [45]

Stock exchanges trade partitions of companies in exchange for currency. When a company wishes to raise capital for various operating expenses or expansion projects they often seek to become publicly traded companies. In the U.S. to become a publicly traded company an institution must first file with the governing board, the Securities and Exchange Commission. Once this has been done the company must now be approved to be listed on an exchange, which often have criteria the company must meet to remain a listed stock. Once it has done this an Initial Public Offering (IPO) is announced within that exchange. [44]

The IPO is basically a one-time sale of a partition of the company (referred to as a stock) at a price set by that company. The company then gets the capital raised from this sale; however the purchaser of this stock now owns a portion of the company. Sometimes these shares are voting shares, which entitles the holder to vote upon business decisions made by the company, while other times they do not give the holder of the stock any voice in the operation of the company (non-voting shares). For this reason any entity controlling more than 50% of all voting shares of a company, holds the controlling share as they can win any vote held by the company.

Regardless of the type of share held by an entity these shares can now be resold within the market which they are listed under. These shares retain value based upon the principles of supply and demand. If a company is performing well, the stock of that company will rise in value as more investors seek shares of that company, whereas if it does not perform well the price falls as investors try and unload their shares on the market. [45]

An offshoot of the stock market is the mutual fund market. A mutual fund is a conglomeration of several different stocks managed by a mutual fund manager. By creating these funds the manager allows investors to purchase shares of an entire sector or a diversified fund. Since stocks are only sold in integer values, many high value stocks may not be feasible for some investors, but mutual funds allow for investors to hold partial shares of these same stocks. In addition mutual funds allow investors securities which are less volatile than stocks as the performance of an individual company is secondary to the performance of the fund, which in some cases is the entire sector. [46, 48]

The bond market is one in which currency is exchanged for a bond, a security which offers a specific interest rate of return over a given period of time. When this time expires (known as date of maturity) the bond can then be exchanged back to the issuer in exchange for the purchase price plus the interest accumulated over that time, very similar to a loan. The most common types of bonds in the U.S. are issued by the Treasury Department to provide liquidity to the U.S. economy. Many other nations also issue these bonds to raise capital for their economies. These types of investments are generally seen to be extremely low risk investments, but generally offer low rates of returns. Bonds are not always low-risk as some nations have bad track records for defaulting on loans (i.e. Greece), but the bonds of these riskier nations often interest investors by offering higher interest payments. Additionally in the U.S. many states, towns, and cities offer municipal bonds. These bonds are obviously riskier than Treasury Bonds, as they do not have the large economical backing provided by the taxpayers of the nation, but once again offer higher interest rates. [47, 48]

The commodities market is one in which physical goods are traded such as gold, oil, and agricultural products. Commodities can be traded either directly at the time of the transaction, but are often traded as futures. Futures are essentially transactions that take place today, with the promise of a deliverable product at a specified date. They provide price stability to large companies which rely upon receiving these goods at a future date for a price set today. In this manner airlines can ensure the price of a plane ticket several months into the future by ensuring that they will receive their jet fuel at a specific price set today. [50]

The foreign exchange market (forex) is the largest and most volatile of all the markets today. In the forex market a trader can exchange one form of currency for another. Such markets provide liquidity to entities which require foreign capital to conduct transactions abroad. Unlike other markets, the majority of transactions do not result in any deliverable product. In fact an entity must be registered to actually physically exchange these currencies. Despite this these non-receivable transactions still provide liquidity to the market, and help it operate more efficiently. [49]

1.2 Project Description

This report investigates basic ideas and information related to investments and trading. A variety of topics and ideas are discussed, ranging from a daily publication from a renowned investor about trading, to different fundamental and technical indicators used to predict security movement. Trading stations and trading systems are discussed and developed as well as current political and economic issues affecting global economies such as the Grecian default crisis going on now or Operation Twist occurring in the United States presently.

There is a heavy focus on indicators used to trade forex profitably, and the difference for each one is lengthily discussed. Many different common and more advanced methods for trading using the technical indicators are described, and the equation for each indicator is explained, showing where it came from and how it is plotted on the charts. Fundamental factors are discussed and whether investing based off of fundamental or technical indicators is generally a safer trade. Finally, programming in MQL4, a language developed to create automated trades on the Meta Trader 4 platform is reviewed and a relatively simple program is demonstrated from scratch to buy and sell under certain conditions based off of technical indicators.

This report is an introductory melting pot of many aspects of investing, with a primary focus and relation to foreign currency exchange. Many indicators are used to trade multiple securities such as stocks or commodities, but the described applications and methods used cater specifically to currency trading. This report dives into a variety of related investment topics, and how the world's current economic, political and social health interlaces with the currency market; Everything affects Everything! However, all topics discussed here only scratch the surface of each idea compared to the level of intricacy each topic can be studied in depth by professional investors.

Another heavy portion of this report is focused on individual and group trading portfolios. Each individual had an account valued at $100,000 leveraged 50:1 and trades were made according to their plan so that they were like our real capital being invested. With our portfolios, the possibility of starting a company was researched lightly.

Finally, programming became a major focus of this project, creating many useful tools for the trader. Programming is a useful way to make life simpler, faster, and more efficient in everyday tasks. When applied to the foreign currency exchange market it can be used to create powerful tools that can lead to ridiculous profits in some cases. Whether it is the newest indicator that nobody else is using, or the latest automated trading robot that makes 20% a week programming is effective in the market. Moreover it can be used to present the data in a different way which can be in turn used to see trends and make custom indicators.

Chapter 2: Background Research

2.1 Intro to Forex

The foreign exchange market (forex) is a market in which different currencies are exchanged between parties. Historically, the forex was instituted in order to provide banks and companies liquidity in various currencies rather than using gold or other precious metals for foreign transactions. [1] However, the use of computers today has reshaped the face of trading markets, allowing many smaller players to enter the market.

In today's forex market the majority of currency trades do not result in an actual physical transaction, but rather a position in the market to be resold to a third party at a later time. In fact, many of the online brokers which are used by traders never deliver any currencies. However, these players who act as middlemen provide liquidity to the market, which in turn allows it to operate for the purposes of its induction.

The forex market is the largest market with an average of $4 trillion USD (U.S. dollar) traded every day, based on 2010 information. [2] In 1998 the daily turnover was $1.5 trillion every day. [2] It is not hard to see that the growth rate in this market is astounding and continues to grow. Year after year the USD remains the highest traded currency; more than 40% of all forex deals include the USD as one of the currencies. [2]

In the forex market currencies are exchanged not sold, so if one wishes to purchase EUR (euro) they must buy it with another currency i.e. USD. In a spot transaction a buy (long position) order would be placed with a broker on EUR/USD at a specified price and when there is another party who places an order to sell EUR/USD, the broker completes the transaction. The latter party would be said to be selling (short position) the EUR/USD. The broker then makes profit based on the spread, which is the difference between the buy and sell price. The spread is typically between 1 and 5 pips of the base (first) currency, for the major pairs.

One of the features that make forex trading so popular today is the leverage. In the United States the forex market is limited to offering a 50:1 leverage, meaning for every 50 units of that currency a speculator only need to invest 1 unit of that currency to hold a position, while in other countries regulation may allow for up to a 500:1 leverage. An interesting consequence of this leverage system is that it becomes possible for an investor to lose more than the original amount invested. Thus, forex brokers hold the right to exercise a margin call. Margin calls occur when an investor no longer has enough free margin in their account to maintain a current position, in which case the transaction is closed immediately.

While spot transactions make up a large portion (37%) of the total volume of forex transactions that take place, foreign exchange swaps make up 44% of all forex trades. [2] A swap is a transaction in which one currency is exchanged for another for a specified amount of time, the time of maturity. At the time of maturity the currencies are then exchanged back to the original parties. This is a common type of investment for large corporations who wish to hedge different positions with differing maturity terms, to mitigate risk in the market.

2.2 Fundamental Analysis (FA)

2.2.1 Introduction to Fundamental Analysis

Fundamental Analysis in the world of securities is a methodology of determining the true value of a security, based on macro-economic issues, past data, and long-term trends. The basis of such a system is that in such a large market the short-term prices often differ from the "true" price of a security and that over a longer timeframe the "true" price will be approached. [32] In the Forex market there are several main contributors to the base price of a currency pair. Since currencies are always traded in pairs the fundamental analysis essentially comes down to a comparison of the current and expected future economic conditions of the countries in question (or region for EUR). Thus, FA is largely influenced by the central banks, their policies, and their leadership, as well as domestic factors such as GDP, interest rates, unemployment, and manufacturing. [32] Reports on those topics are released at predefined intervals governing bodies credited to determine them.

Given the importance of these numbers, speculations are made for their values before the official release. If the actual data released is greatly different than widely accepted speculative estimates, great volatility can be seen in the market. Economic reports, such as unemployment or GDP, are means by which a country's economic health is directly measured. These reports are released to the public at predictable, scheduled intervals. It has been proven again and again that if these reports indicate the presence of even a small deviation from the status-quo, it can cause huge changes in the market. These reports, along with speeches from important officials (such as "Chairman of the Federal Reserve Bank of USA, Secretary of the Treasury, President of the Federal Reserve Bank of San Francisco and so on"[7]) are watched carefully by forex traders.

Speeches by the aforementioned weight-carrying officials are watched closely enough that there are websites dedicated solely on notifying users about when speeches and reports are happening. This is used by some as an indicator that the market could become highly unpredictable during that time period, and dissuades them from trading. Other traders study how these reports change the market and, by utilize a real-time feed to the report or speech, 'trade the news' (anticipating market movement given what has been reported). That practice can be highly lucrative, but is viewed by some to be too risky to trust.

2.2.2 Macroeconomic Models

While no individual model is consistent in predicting the future of the forex market, there are several models in use today which attempt to quantize the market. Several of the models used in our current economic climate do a decent job at predicting and shaping exchange rates, but all of them have their own assumptions and flaws. "The Purchasing Power Parity Approach (PPP) holds that in the long run, exchange rates will adjust to equalize the relative purchasing power of currencies." [32] The essential basis of such a model is that goods are worth the same value in different countries and the difference in currency value stems from the overall purchasing power of a nation. Thus, PPP is largely based on tradable goods and services while neglecting many other issues which influence exchange rates. This model is decent at predicting long term prices and excels when modeling currencies experiencing high inflation, however is limited in its inability to make short to medium term predictions. [32]

John Williamson has put forth the concept of "fundamental equilibrium exchange rate" (FEER), in which a country is to develop a macroeconomic strategy to achieve an internal balance in terms of unemployment and inflation. [33] The problem with this concept is that there is little consensus on the proper way to measure the equilibrium. Using different models, different attempts to define this equilibrium price can have very different results. "John Williamson has suggested that FEER calculations could not realistically justify exchange rate bands narrower than plus or minus 10 percent." [32] The Monetary Approach to exchange rates is based upon the total supply and demand of a currency. Since central banks have monopolies on the supply of currency available, this system really boils down to the demand and central banking policies. This approach is limited in that it does not account for bonds or the supply and demand in the goods and services sectors. However, this model does account for the much greater short-term influence of policy while the prices of goods and services tend to lag behind. [32]

The Portfolio Balance Approach (PBA) is a theoretical improvement on the monetary approach in that it does not assume bonds and other financial assets to be perfectly substituted whether in foreign or domestic regions. However, there is no universal agreement as to a proper way to value all such instruments simultaneously. The PBA approach is often a much more complex model than other approaches and while using one system of relations may produce accurate results in one circumstance, it is unlikely to predict results in others. Due to this, many financial institutions prefer to use simpler models which have a more universal agreement. [32]

2.2.3 Central Banks

The Central Banks of a nation (or region) provide the basis for most any fundamental analysis, as they directly control the rate at which currency is produced. It is the duty of a central bank to attempt to keep their currency within a band or target zone around a FEER. [34] By increasing the production of a currency the central bank induces inflation and thus decreases its true value. In the United States the Treasury, in consultation with the Federal Reserve System, has the responsibility for setting U.S. exchange rate policy, while the Federal Reserve Bank New York is responsible for executing FX intervention. [35] The Federal Reserve is controlled by the Federal Open Market Committee(FOMC), which is a revolving board of the members of the reserve banks and treasury. The Federal Reserve is a system of 12 different regional banks controlled by one governing body. [34] The Federal Reserve differs from other central banks in that in addition to shaping the monetary policy of the nation, it is also tasked with unemployment policy. "Economic developments in this country [United States] have a major influence on production, employment, and prices beyond our border; at the same time developments abroad significantly affect our economy." [23]

2.2.4 Interest Rates

Interest rates, both long and short term, are largely influenced by the central banks as they set the rates at which private sector financial institutions borrow money. In addition, the influx of money (through the purchase of bonds) results in a decline in interest rates of both parties, as is common with the current U.S. China financial relationship. [36] One of the best predictors of economic climate is the yield curve, which is the spread between the interest rates of treasury notes with differing maturity time frames. The yield-curve method for forecasting markets has outperformed most other methods for long-term predictions. This is important due to the fact that policy decisions usually have a long-term impact. [37]

2.2.5 GDP and Debt

Historically, inflation has been the main concern of developed nations when it comes to monetary policy. However, in the majority of developed world economies, successful strategies for controlling inflation rates have been so successful that financial instability has become the major issue. [24] In recent years, with many nations facing financial crisis, the debt toGDP ratio has increasingly become the most important factor to currency evaluation.[22]

Gross Domestic Product (GDP) is the total value of all the goods and services produced internally and externally by a nation. GDP and inflation have a high correlation as rapid growth can induce inflation devaluing currency, raising prices on goods and services, and decreasing the purchasing power of a nation. [39] Despite this, GDP remains the best method for determining the financial stability of a nation.

A nation's debt can belong to either of two categories: domestic debt and foreign debt. Foreign debt occurs when one nation provides loans to another nation, while domestic debt is owned by private entities within the nation. The most typical form of such debts is bonds. While there are many different types of government bonds available domestic debt does not affect the value of currency nearly as much as foreign debt does.

Foreign debt arises largely from the unwillingness of the borrower to allow direct foreign investments into the infrastructure and private sector alike. Historically, Treasury bond yields have been below the growth rate of the economy, which is in part due to the security of the U.S. dollar. The national debt in the U.S. has been consistently growing faster than GDP since 2000 from approximately 38% to 53% in 2009. [40]

2.2.6 Inflation

According to Mishkin, Inflation policy involves a medium-term strategy for monetary policy which incorporates five main elements: "1) the public announcement of medium-term numerical targets; 2) an institutional commitment to price stability as the primary monetary policy; 3) an information inclusive strategy in which many variables, and not just monetary aggregates or exchange rate, are used for deciding the setting of policy instruments; 4) increased transparency of the monetary policy strategy through communication with the public and markets about the decisions of the monetary authority; and 5) an increased accountability of the central bank for attaining objectives." [36]

Inflation directly impacts the exports and imports which a nation experiences. When inflation occurs, the relative value of the currency drops making goods and services from that nation cheaper to foreign investors. When this happens, exports typically increase while imports decrease. "Over time, the depreciation in the home currency would lead to the determination of exchange rates growth in the nation's exports and a decline in its imports, and thus, to an improved trade balance and reversal of part of the original depreciation." [32]

While moderate inflation may be good in developed economies, it can have serious repercussions for smaller economies. Many of these economies rely much more heavily on imports and hyper-inflation will cause the prices of these imports to balloon. Such a circumstance can wreak havoc on the economies of smaller nations, as has been seen throughout much of the Asian market towards the end of the 20th century. [41]

2.2.7 Unemployment

Unemployment is greatly affected by such factors as exports, imports, and interest rates. It is this correlation that makes it an important indicator of the current financial stability of a nation. Unemployment can be directly linked to financial climates, as can readily be seen by the high levels of unemployment experienced in underdeveloped nations undergoing hyperinflation. In addition, unemployment can be the result of deflation in developed countries, as the strength of currency directly reduces the nation's ability to export goods.

Unemployment also acts beyond the broader scope of macroeconomics and begins to affect the psyche of a population. If a nation begins to experience higher levels of unemployment, the population risks going into an economic stasis by which they become much more conservative in their financial dealings. In such a case the liquidity in markets begins to dry up, as consumer spending drops and creates a so-called "snowball effect", which leads to recession, and perhaps depression. This has been readily seen in the U.S. since the collapse of the stock market in 2008.

2.2.8 Consumer Price Index (CPI)

The Consumer Price Index (CPI) gives a household's approximate cost for a 'basket' of goods and services, in comparison to the cost of that same 'basket' in 1982-1984. [42] The data is released monthly, in the middle of the month. Every year before the release of January's numbers, all the data gathered previously is considered and new indices are calculated. The new indicies are adjusted for seasonal factors, and replace the previous year's CPI data, along with that of the last 5 years. The CPI is used by the federal government to make adjustments to cash flow mechanisms such as pensions, Medicare, and adjustments to insurance policies. Resultantly, the CPI tends to have some effects on every investor in some way. Investors with a fixed-income should of course assure that their profits land them ahead of inflation, for having a yield which is not ahead of inflation leads to a loss of real wealth (despite a numerically apparent monetary gain!). “The U.S. index is based on the prices of goods and services, including food and beverages, housing, apparel, transportation, medical care, recreation, education, and communication, that people buy for day-to-day living in urban areas across the country.” [42]

Because the CPI can be considered a relatively direct measure of inflation, it is of great value to the forex trader. Because it marks the buying-power of the dollar, it can cause great movement in the stock market, as well as the Forex market. A CPI which indicates that the value of the dollar is sharply decreasing can be expected to drive investors to “change their preferences on the economy to invest.” [42] Resultantly, the value of the dollar will fall even farther, because that preferential change will tend toward an increased selling of the dollar. That in turn drives away more investors, and further decreases the value of the currency.

The government also keeps close watch on the CPI, so attempts can be made to keep the currency stable. Accordingly, the “state’s monetary policy is made with these statistics in mind” [42]. If indices point towards an inflating dollar, the monetary authorities are likely to increase interest rates in an effort to drive the dollar value back up. Similarly, if deflation is indicated, lowered interest rates are assumed with the goal of restoring the dollar’s former value. Additionally, values such as Social Security beneficiary payouts and food-stamp amounts are derived from the CPI so payouts can stay at a relatively constant economic value, even if the dollar amount has changed due to inflation.

2.2.9 Using Fundamental Analysis

Many of the preceding indicators are affected heavily by each other. For instance, Federal banks use Consumer Price Indices to make decisions based on their monetary policy. If a CPI points towards an inflating dollar, the monetary authorities are likely to increase interest rates in an effort to drive the dollar’s value back up. Thus, any single fundamental indicator should not be used blindly to forecast the market's movement. Instead, several fundamental indicators should be considered together, to determine if a common trend can be discerned.

Using several different fundamental indicators, a picture of the economic conditions driving the forex market can be painted. However, a completely different picture might be painted by the data in the recent price charts. There are several reasons for this, but what is important is in understanding how to use both fundamental indicators AND the technical indicators discussed in the next section. Since bothtechnical analysis and fundamental analysis each provide a unique view of the same picture, their use as complimentary methods can be quite successful.

2.3 Technical Analysis

2.3.1 Introduction to Technical Analysis

The following comprises a list of indicators that help one decide when and if they should enter a trade position on the foreign exchange market. They are different from fundamental indicators in that they treat all currencies the same. They are a pure mathematical formula applied to past data of a currency to help predict current and future trends. The theory behind technical analysis lies in the assumption that major economic influences tend to have little effect on short term fluctuations and thus are negligible. They rely on past values that the currency has traded at, and since they are purely statistical they are considered more reliable than trading the news to most investors.

2.3.2 Candlesticks

Candlestick charts are used widely in various markets today to graph the price fluctuations of currencies, stocks, and commodities. They were first introduced in the

18th century to chart the prices of rice in Japan, but have since been applied to everything which is traded in an open market. [3]

A candlestick chart uses a symbol commonly referred to as a bar, which conveys the price of the open, close, low, and high for the given time increment of a chart. Typically candlesticks will have one of two colors depending on whether the price has increased or decreased in that time interval. Typically if the close price is higher than the open price, a candlestick will be green and if the close price is lower than the open price it will be red. However, most programs contain the ability to change the colors of these charts, so one must be familiar with the program they use.

The candlesticks consist of two major parts, the wick and the body. The body is the wide part of a candlestick and contains the open and close information for that time period. The wick extends off of the main body and shows the high and low prices for that period. The main reason that candlestick charts are so widely used is that they only incorporate the exact prices which a currency pair has traded at, without any calculations. After all, price is the most important piece of information an investor can have.

Additionally, since the market is in constant oscillation often the market overreacts and a specific trend will eventually reverse itself. Candlesticks can be particularly useful in determining when this reversal will occur and there are several patterns which are indicative of such a case.

Classes of Candlesticks

Candlesticks take on a variety of different sizes and shapes (respective to the tail/body size and ratios of both). If a candlestick bar has no tail then we term it a Marubozu. If it has a small body and the tails are of equal length as the body, it is dubbed a spinning top. In both of the prior cases the candlestick by itself is usually pretty irrelevant when we try to determine a trend reversal. A similar candlestick is one which has a short body and no tails called a 4-price, which usually only happens during suspended trading periods. This is a pattern one would never wish to trade, as it is indicative of low volume.

A Doji is a candlestick which has very long wicks and a small body. A Doji is usually indicative of market-indecision and makes for a poor entry position as the price may continue in either direction. Similar to the Doji are the dragonfly and gravestone candlesticks. Both incorporate a small body, but the dragonfly has 1 long tail on the bottom and the gravestone has its long tail on the top. Both of these are single bar reversal patterns; the dragonfly is a bullish reversal whereas the gravestone is a bearish reversal. Strength of the dragonfly is increased by a previous bearish period, while the reverse is true for the gravestone. [3]

Candlestick Trend-Reversal Patterns

Dark Cloud Cover and the Piercing Line

Two of the most important reversal trends are the Dark Cloud Cover and the Piercing Line patterns. In a Dark Cloud Cover pattern a bullish trend must have occurred for four consecutive bars followed by a bar which has a close lower than the opening. In addition, this last bar must have opened above the high of the previous bar and closed below the midpoint of the previous bars' opening and close mark. The Dark Cloud Cover is indicative of a bearish reversal pattern. The Piercing Line is the opposite, a bullish reversal pattern such that there is a bearish trend for four bars followed by a positive bar whose close is above the midpoint of the prior bar. In both cases it is usually safe to wait for a "confirmation bar" in the form of a candlestick continuing in the direction of the reversal trend.

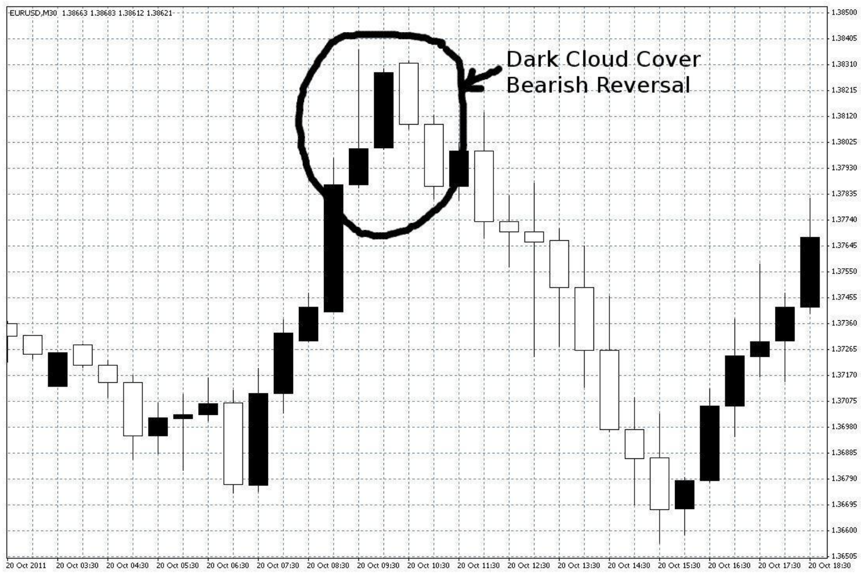


Figure 1: Dark Cloud Cover bearish reversal pattern on EURUSD 30 minute chart

Engulfing Reversals

There is also a class of pattern referred to as Engulfing Reversals. A Bearish Engulfing reversal takes place when there is a bullish trend followed by a bar whose body extends long enough to "engulf" the previous bar. Similarly a Bullish Engulfing reversal follows a bearish trend and "engulfs" the prior bar. Similar, but not as strong as these signals, is the Harami reversal pattern. The Harami bullish/bearish reversals are the same as those in their engulfing counterparts, however rather than the last bar engulfing the prior one, the prior bar engulfs the last bar.

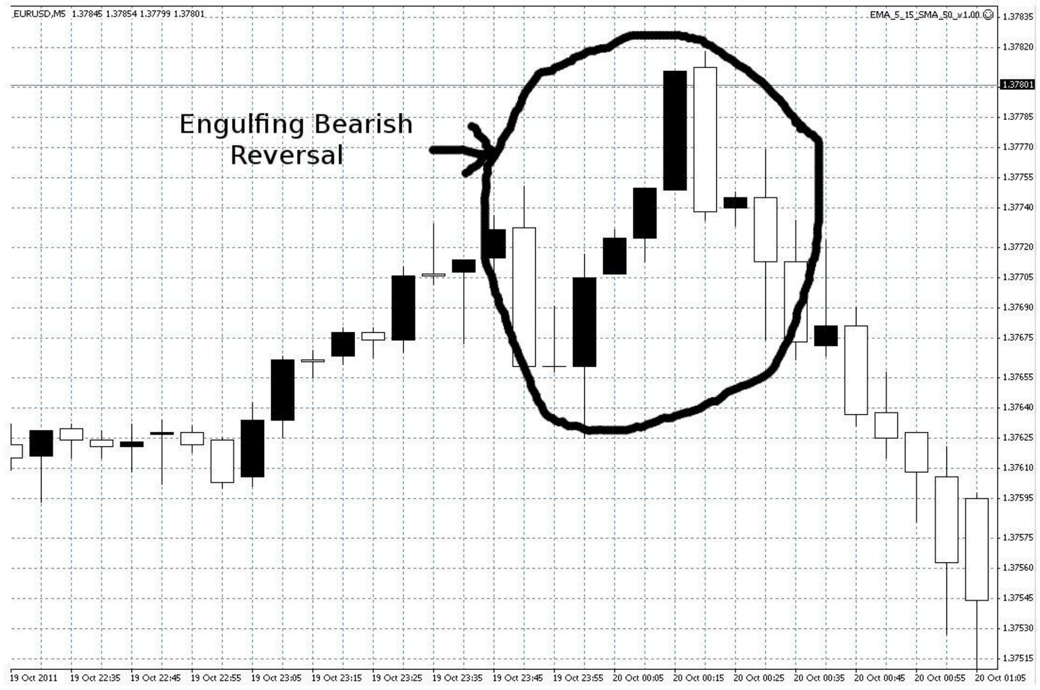


Figure 2: Engulfing Bearish Reversal on a 5 minute EURUSD chart

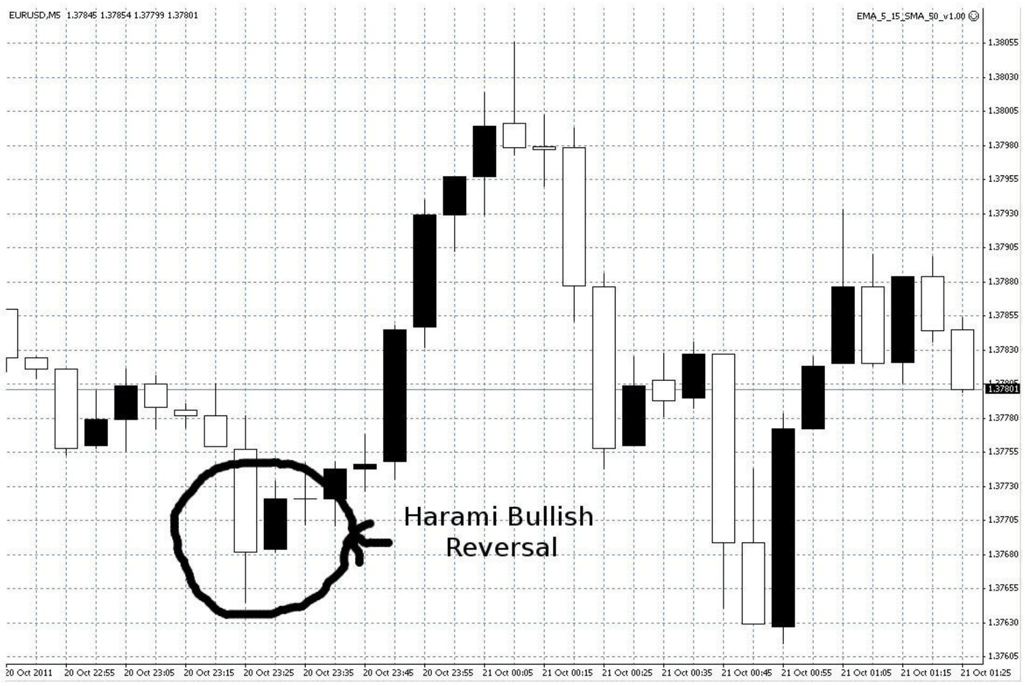


Figure 3: Hanging Man bearish reversal pattern in a 5 minute EURUSD chart

Hanging Man and Hammers

The hanging man is another important bearish reversal pattern. It is characterized by a bullish trend ending with a bar with a small body and a wick on the bottom which is longer than the body. Whether this last bar's close is higher than its open or not is irrelevant, only the wick to the bottom plays importance here. Likewise the hammer pattern is a bullish reversal pattern, characterized by a bearish trend followed by the same last bar as the hanging man. In addition, there is also a bullish reversal pattern called Inverted Hammer which is a very strong signal. An Inverted Hammer occurs after a bearish trend ending in a long bodied downward trending bar which is followed by an inverted (upside-down) hammer with a long wick up.

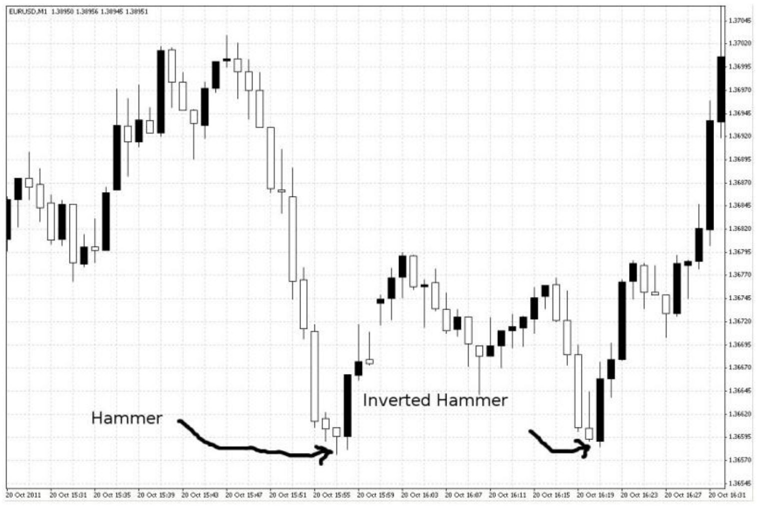


Figure 4: Hammer and Inverted Hammer on EURUSD 1 minute chart.

Shooting Stars

Another class of reversal patterns is the Shooting Star class. Following a bearish trend if the last downward bar has a large body and is followed by a spinning top whose body is completely below the body of the prior bar then we classify it a shooting star bullish reversal. Similarly, if there is a bullish trend, whose last upward bar has a long body and is followed by a spinning top, whose body is completely above the prior bar, we define it a shooting star bearish reversal.

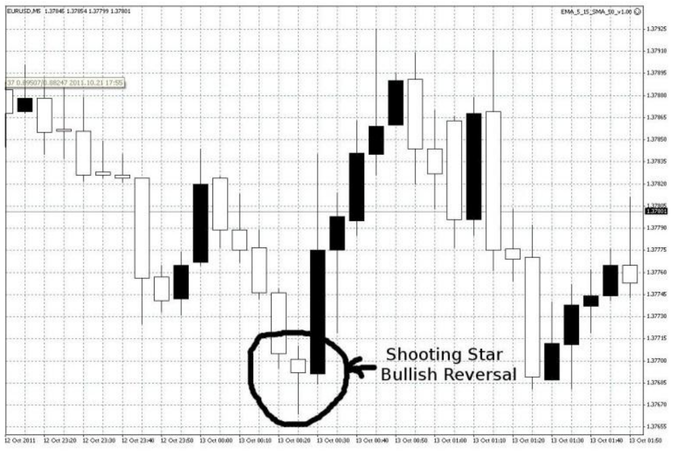


Figure 5: Shooting Star Bullish Reversal pattern on a 5 minute EURUSD chart

The opposite of the bullish shooting star is the bearish shooting star. It is shown in the figure below.

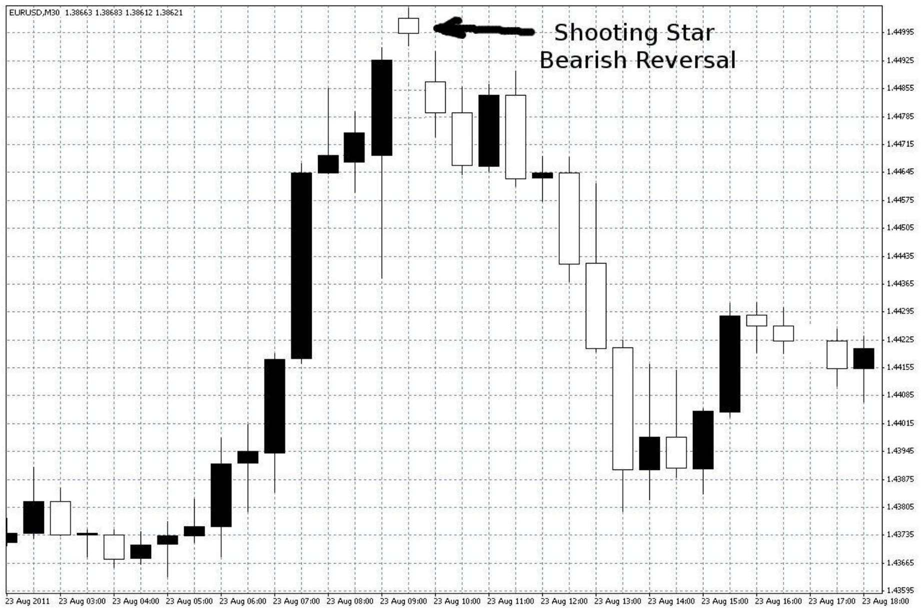


Figure 6: Shooting Star bearish reversal on a 5 minute EURUSD chart

Raindrop

The Raindrop/Star reversal pattern is one which is usually only seen in longer *n*-period candlestick charts. A bearish reversal trend begins with 3 upward trending bars have the pattern long body, then short body (usually a spinning top) above the large body, then long body below the short. After this last long bar if another spinning top follows with its body removed that of the previous bar we classify this as a star pattern. The same is true if the first three bars are bearish and the spinning tops are below the other bodies where the last has an upward trend. In the latter case this would be a raindrop reversal. However, these patterns themselves are not very strong or common and so it is wise to wait for a confirmation bar before trading. In fact they are so rare in the forex market, after hours of searching one could not be found for this paper.

Morning Star and Evening Star

Lastly, are the Morning Star and Evening Star patterns, which are two of the strongest pattern signals one can see. The Morning Star is a bullish reversal in which a longer than normal downward (usually 4 or 5 long bars down) trend has occurred, followed by a spinning top or doji with an upward trend whose body is completely removed from the previous bars' body. Finally, a last upward trending bar is needed to confirm the pattern; as such a large downswing warrants some caution. The Evening Star is the bearish reversal counterpart to the Morning Star and differs only in a bullish trend followed by the spinning top/doji above and removed from the last bar. Once again a final confirmation bar must follow before trading. They are very similar to the shooting star patterns and thus images have been omitted.

HeikenAshi

HeikenAshi is an offshoot of traditional candlestick chart which uses formulae for the open, close, high and low, rather than using the actual values for any timeframe. They are calculated as follows:

$$\begin{aligned} Close_i &= \frac{Open_i + High_i + Low_i + Close_i}{4} \\ Open_i &= \frac{Open_{i+1} + Closei + 1}{2} \\ High_i &= max\{High_i, Open_i, Close_i\} \\ Low_i &= min\{Low_i, Open_i, Close_i\} \end{aligned}$$

The HeikenAshi candlesticks are often used in trading because they make clear what the current direction of a trend is. This method mainly alters the size of the wicks and it can readily be seen that a bar with no upper wick is in a downward trend and a wick with no lower wick is generally in an upward trend.

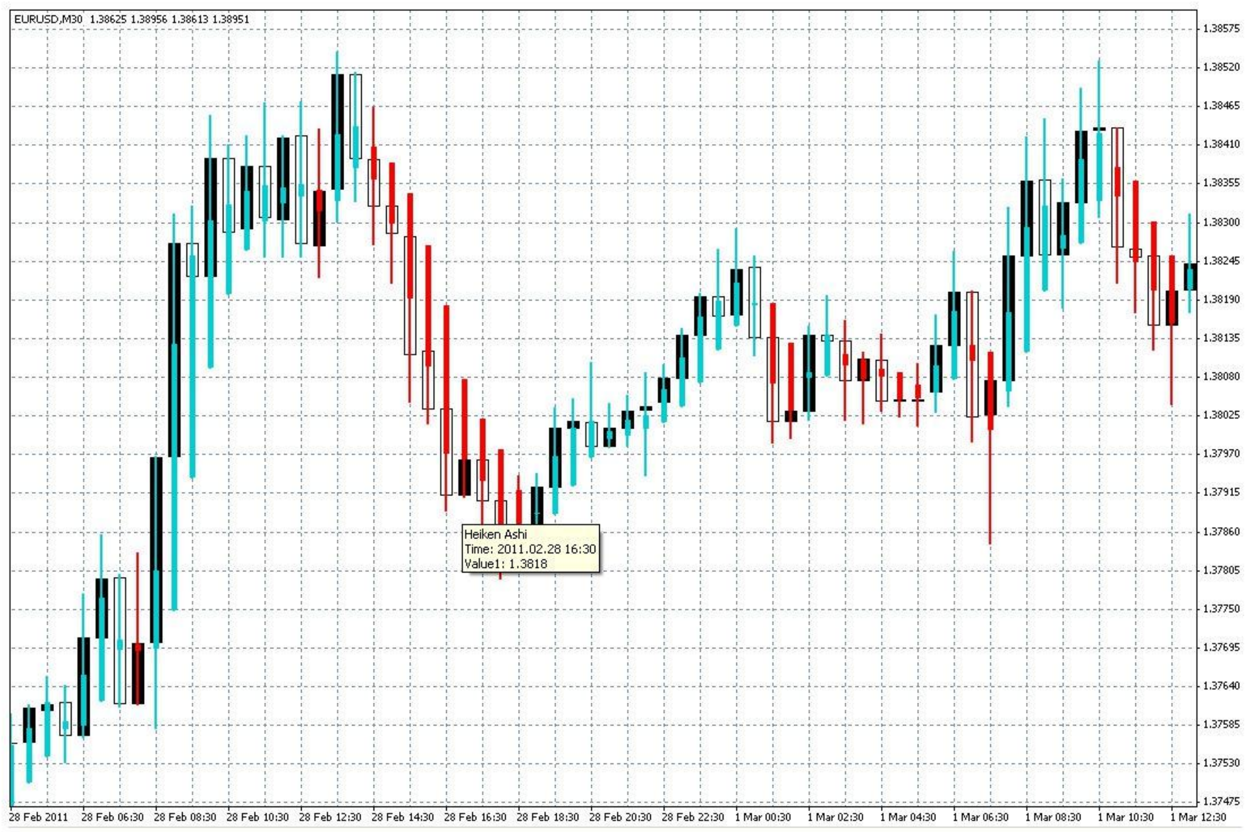


Figure 7: A HeikenAshi candlestick chart overlying a regular candlestick chart

2.3.3 Support and Resistance Lines

Support and resistance lines are the next most popular indicator used by traders. A support line is a line drawn horizontally on a chart at a specific price that has not been crossed below in a period of time. Likewise, a resistance line is one in which the price has not crossed above in a time period. Usually several different support and resistance lines are plotted in conjunction.

The basic premise of these lines is that they indicate prices, that when reached, experience high levels of long or short positions respectively. Support and Resistance lines are used in the market as indicators of entry and exit positions. When the price approaches either, one can be more confident in their ability to "bounce" back as past pressures around these levels have been experienced. This "bounce" is usually caused by larger institutions placing orders at these levels. If support or resistance lines are broken then it is a sign that previous sentiments around those prices has been broken and thus is indicative of a trend continuing in the respective direction.

2.3.4 Moving Averages (Simple and Exponential)

Moving averages are a great indicator for identifying trend direction and strength and some of the most commonly known and used technical indicators. When the slope is positive it indicates an uptrend, and when it’s negative a downtrend. The steepness of the slope is proportional to that trend’s strength [4]. It is possible to plot more than one MA at a time which can combine to indicate buy and sell points. One common method is plotting a fast moving average against a slower one. For example, the faster one follows the trend better so the period is smaller, say 50 and the slower one would have a period of more than 50, say 100. If the faster one cuts under the slower one it is a “bearish” crossover and it indicates a sell point. If the fast one cuts above the slower one it is a “bullish” crossover and one might consider buying. There are two common types of moving averages, exponential and simple which are explained below.

Simple Moving Average (SMA)

The simplest form of the moving average plots the average value of the past *x* candle sticks, where *x* can be chosen by the user. Say you set *x* to be 10 in MT4. For any given point in time on the SMA plot, that value was determined by the average of the last 10. Depending on your trading station what it plots can be customized such as plotting the past ten opening values instead of the past ten closing values or whatever the user prefers.

$$SMA = \frac{x_1 + x_2 + \cdots + x_n}{n} \; where\ n\ is\ the\ number\ of\ past\ data\ points\ used$$

In the above equation, x_n represents the nth most recent data point of the user selected value. For instance, if the user wanted a simple moving average of the past 25 days using opening prices, x_1 would be yesterday's closing price and x_{25} would be the closing price 25 days ago. All of these are summed together and then divided by the number of points (in this case 25) to plot one data point.

Exponential Moving Average

Very similar to the SMA but it places more weight on recent data. So for an EMA with a period of 10 minutes, data points from the most recent minute would weigh more than the 2nd most recent minute and so on. Traders tend to use exponential averages more because more recent prices are considered more relevant than older ones, which the simple moving average does not consider. This makes logical sense over moving averages with huge periods, since data 200 days ago could be out of date to predict today's price if there were other global factors affecting data back then [4].

$$EMA_{current} = EMA_{most\ recent} + \alpha * (Price_{current} - EMA_{most\ recent})$$

In the above equation, $EMA_{current}$ is the most recent data point that is being calculated, $EMA_{most\ recent}$is the most recent data point collected. So for instance, if you were doing a 10 day moving average, EMA$_{mostrecent}$would be yesterday's calculated EMA$_{current}$. Price$_{current}$isexactly what it sounds like, the current trading price and α is generally expressed as a factor of the period, shown in the following equation.

$$\alpha = \frac{2}{N+1} \; where\ N\ is\ the\ period\ size$$

N can be days, or minutes, or any time frame chosen by the user. Alpha isn't always calculated this way for all exponential moving averages, but this is the most common way it is found. Alpha is just how much less each subsequent EMA data point will be worth. It is always between 0 and 1 where a higher alpha means that older data is discounted faster.

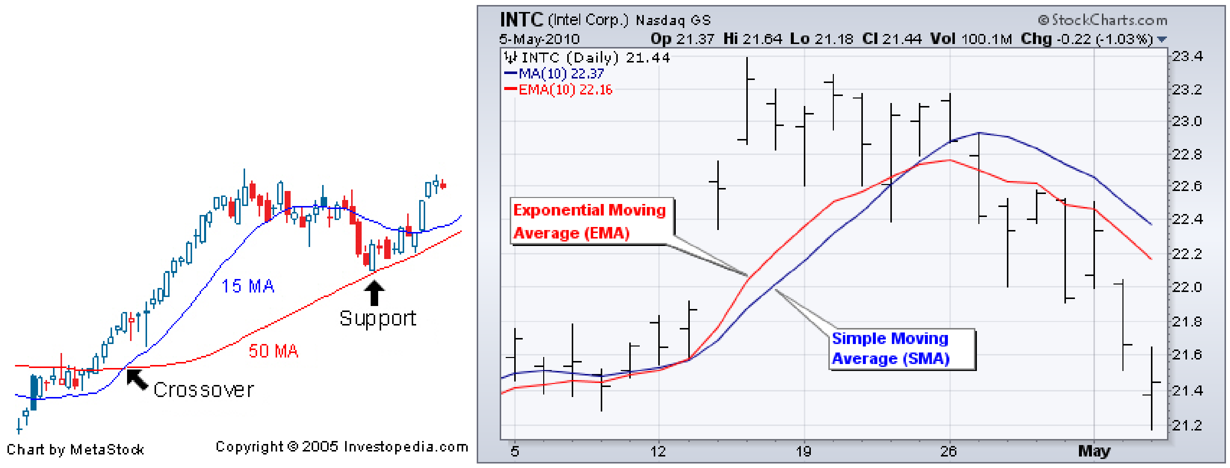


Figure 8: The figure on the left shows two different simple moving averages and the figure on the right shows a simple moving average plotted with an exponential moving average. [51]

One can notice how the moving average with a smaller period of 50 follows the prices much more closely than the one with a period of 50. That makes sense because when the period is 15 only the past 15 data points are used which were all closer to the current price whereas the past 50 data points can contain values very far from the current price affecting the whole line. These figures also show the crossover lines between a fast and a slow moving average indicating buy and sell points respectively. It is hard to tell from the figure on the right but the exponential moving average relies more heavily on recent data to plot the indicator versus the simple moving average which treats every data point the same [4]. This is why the two lines have the same period but different shape.

2.3.5 Moving Average Convergence/Divergence (MACD)

This is a very common indicator that can be utilized for trend and divergence recognition. It essentially plots two different exponential moving averages. When the MACD cuts through its horizontal line in the positive direction it generally indicates a bull market, and vice versa a bear market when it cuts down through the ‘trigger’ line. The vertical bars are the difference between two customizable exponential moving averages, so when one EMA crosses the other, the bars switch from one side to the other on the horizontal axis [5].

$$MACD = EMA_{minor} - EMA_{major}$$

In the above equation, $EMA_{minor}$is the exponential moving average with the smaller period and the other term EMA_{major} is the moving average with the larger period.

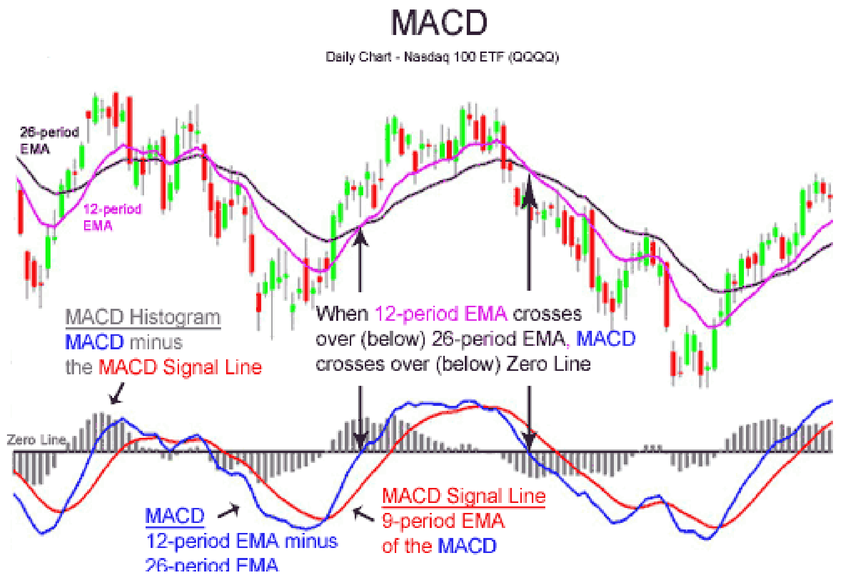


Figure 9: A detailed picture of the MACD indicator. In the top portion of the figure the two EMA are plotted against the candlesticks and the bottom portion shows the actual indicator. The height of the histogram bars are the difference between the two EMAs. In this case the major EMA is the 26-period EMA and the minor is the 12-period EMA shown above.

When the MACD is indicating a trend one way and the currency pair is going the opposite way negative divergence is indicated. The author at forexindicator.org states that this commonly leads to a downside movement, whereas during positive divergence an increasing movement in price is usually noted by this author [5]. There are several different trading strategies which use the MACD indicator as the basis.

The MACD crossover initiates a buy signal when the MACD rises above the signal line and initiates a sell order when it falls below. It is also popular to buy/sell when the MACD crosses zero. The MACD indicator is very insightful and can be even more reliable when combined with other technical indicators described further into this section. [18]

2.3.6 Bollinger Bands

Bollinger bands, when first added to a currency pair seem intimidating due to the fact three lines appear across your screen, but mathematically speaking, it is a simple indicator. The lowest plot on your screen will be a simple moving average minus 2 standard deviations [6]. The middle one is just the moving average and the upper on is the moving average plus 2 standard deviations. The period of the moving average and number of moving averages can all be customized in MT4. The general equation for Bollinger bands actually consist of three simple equations, one for each line. The middle line is a simple moving average and the equation has been shown previously. The other two equations making up the outside two lines are shown below.

$$Bollinger\ Bands = SMA \pm K\sigma$$

Where SMA is a moving average with a chosen period N. Note this can also be done with an exponential moving average as well. K is a user chosen constant, usually around 2, and σ is the standard deviation of the moving average. Essentially K is how many standard deviations from the moving average you want each band to be [7]. While the equation used for the standard deviation is very similar to its probabilistic definition, it should be noted that such stochastic processes, as exist in the market, have no "regular" distribution. For this reason standard deviation in this sense is merely a representation of such a concept in a continuous environment.

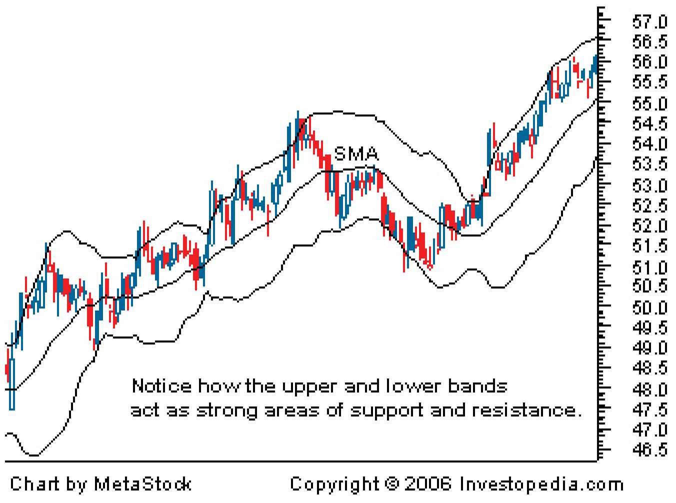


Figure 10: A graph of the Bollinger bands plotted. At any point each outside band should be equidistant from the moving average middle band. The outer bands can be used for levels of support and resistance respectively as shown in the figure.[51]

Bollinger bands can be used in a variety of ways. The lower and upper bands often indicate support and resistance respectively. Moreover, the width between the upper and lower bands indicates market volatility. When they are very narrow the market is in a period of consolidation with not much movement happening and when they spread wide there is a large price movement. They are also commonly used to predict trend reversals, which is when the trend's slope direction switches polarities [8]. When the price dips below the lower band for some time it could be an indicator to go long. This is great when the bands are very narrow and they suddenly swing out wide; if you go with the trend you have a good chance to make some quick pips.

2.3.7 Stochastic Indicator

This indicator is an oscillator that is generally used to measure momentum. It consists of

Two horizontal lines generally set at the ‘80’ and ‘20’ levels, and two signal lines, %K and %D. There are two main flavors of the stochastic, namely fast and slow [8]. The fast one is more sensitive to the market which causes more false market indications, whereas the slow one is less sensitive but indicates a position after the market has already moved.

$$\%K = 100 * \frac{Price_{close} - Low}{High - Low}$$

The terms *Low* and *High* are the lowest and highest price respectively over the past *n* periods, chosen by the user. $\text{Price}_{\text{close}}$ is the most recent closing price. The ‘%K’ line is plotted alongside of the ‘%D’ line which is simply an exponential moving average of %K with N=3.

When the signals move above the ‘80’ line, the market is considered to be overbought, and oversold when they move below the ‘20’ line. If they stay above or below these limits this can be an indicator of a strong trend in that direction. Additionally, this oscillator is a great tool for entry and exit levels. If the %K line crosses the %D line and moves out of the 20 or 80 level range it is time to go long or short depending on which crossed what [8] To go long would be to wait for the %K to go above the %D and cross the 20 mark.

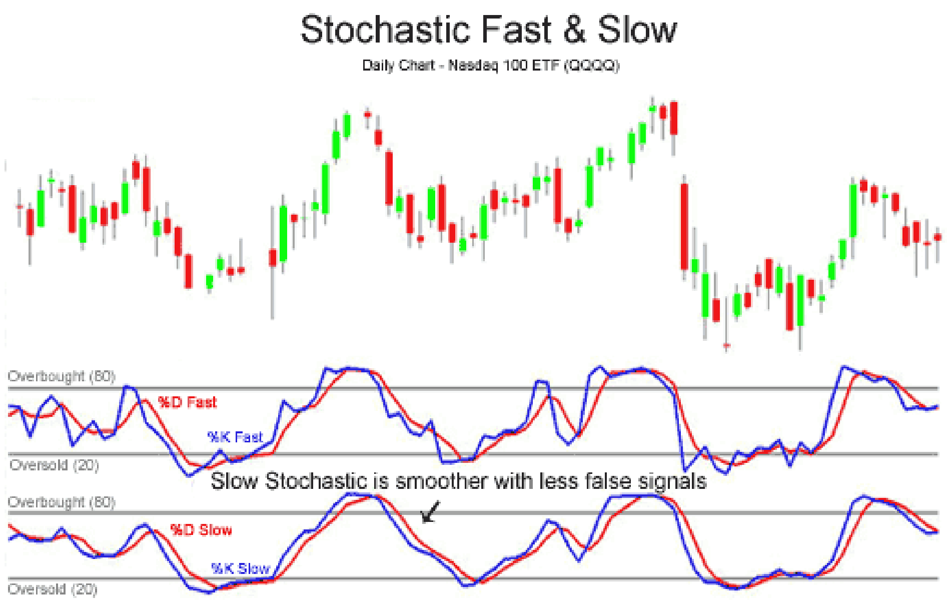


Figure 11: Two versions of the Stochastic plotted. The top one is the fast version and the bottom is the slow. One can see the general pattern of the currency pair with the indicator, as when they remain above the ‘80’ line the market seems to be overbought, and oversold when below the ‘20’ line.

2.3.8 Commodity Channel Index (CCI)

The CCI index is generally used as an entry and exit signal. It comes equipped with two levels, ±100 and ±200 and if the index passes the +100 it signals a good time to enter a long position, and exit once it moves back to or below the +100. One can infer the sell conditions easily from this, using the -100 and -200 levels instead. Essentially, this indicator shows a currency pair’s variation from its mean value [9].

$$\frac{1}{.015} * \frac{Typical\ Price - SMA(Typical\ Price)}{\sigma(Typical\ Price)}, where\ Typical\ Price = \frac{H + C + L}{3}$$

The formula looks complicated at first, but it just contains many functions. The outside scalar of 66.67 is just to make the final result a neater number to display on a graph and is not always used. The typical price is the average of the sum of the high, closing, and low prices. The

CCI is found by subtracting the SMA of the typical price from the typical price, and dividing that by the standard deviation of the typical price.

Another use of the CCI is using it as a reversal indicator. If you draw a trend line with the signal, using either the peaks or the valleys, and that line is majorly broken that could indicate a reversal. It also has positive and negative divergence to show trend reversal, whether the pair is going down and the signal is doing the opposite or vice versa. It usually signifies the market is changing, but may take some time [9]. Finally, if the index passes the +200 mark and eventually makes its way back down to the +100 level, go short on the trade, and vice versa for the negative levels. Like all indicators, this is not a 100% guarantee; otherwise everyone would be doing it!

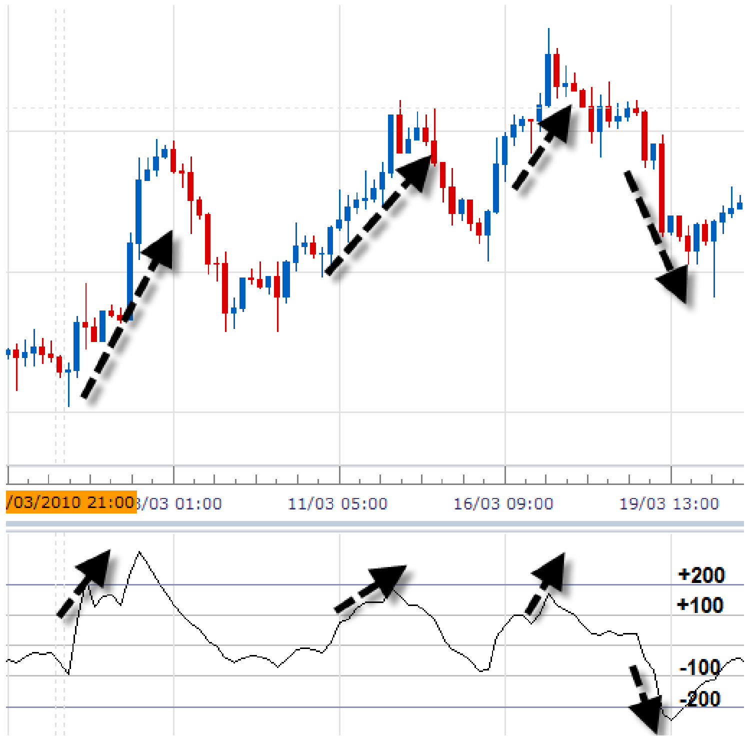


Figure 12: An example where the Commodity Channel Index predicted a bull market successfully; noted by the first 3 upward sloping arrows. The currency pair increased in value as the CCI passed the 100 level each time. [52]

This is one use of the CCI where it should be bought after passing the +100 level and closed once it comes back down and breaks through the +100 level going in the opposite direction. This is a very diverse indicator good for many trade predictions.

2.3.9 Relative Strength Index (RSI)

Much like the stochastic indicator, the RSI is another great way to find out if a market is overbought or oversold. The RSI can range from 0 to 100 and oversold and overbought conditions are marked at the magnitudes of 30 and 70 respectively.

$$RSI = 100 - \frac{100}{1 + RS}, RS = \frac{EMA(U, n)}{EMA(D, n)}$$

The RSI is found by converting the relative strength of a signal (defined above as RS) to an index between 0 and 100 as shown above. The relative strength is found by taking the ratio of the average U and D values (represented by an EMA) which are the upward and downward changes respectively. U is characterized by the difference between the current closing price and the previous closing price. The difference between the previous closing price and the current closing price is ‘D’ [13].

If the price approaches a resistance level and the RSI goes above the 70 line,this indicates a bearish market, and vice versa for a bullish market. The RSI can also be a trend confirmer simply by comparing it to the 50 line. If it’s above the 50 mark it’s an uptrend and a downtrend if below the 50 mark. Moreover, this indicator can be used as a divergence spotter, noting when the market increases while the indicator shows a decreasing downtrend. This alludes to the likelihood of the market switching from up to down.

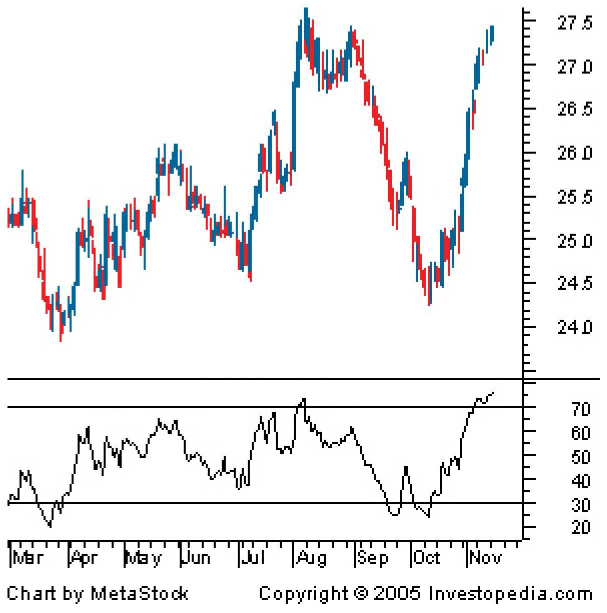


Figure 13: An RSI graph plotted with a currency pair. This shows how the RSI can chart the current strength or weakness of a security.[51]

As seen by the figure from investopedia.com, the market is overbought when the 70 level is breached and oversold when the 30 level is crossed. However this also can have positive and negative divergence so one must be careful to watch for this!

2.3.10 Fibonacci Retracements

Based off of ratios created by the famous Fibonacci sequence, this indicator is excellent for predicting support and resistance levels. The most common ratios are .382 and .618, derived from the famous sequence. If a trend is bearish and reaches the .382 line it is likely to bounce back up according to the Fibonacci retracement [14].

$$F_{61.8\%} = \left(\frac{1 + sqrt(5)}{2}\right)^{-1} = 0.6180 = \frac{n}{n+1} \; where\; n\; is\; a\; fibonacci\; term$$

The previous equation shows one specific retracement level. There is no real formula for this indicator; rather there is a group of ratios found using sequential Fibonacci terms. For example, dividing any nth term of the sequence with the nth+1 term will yield a value close to .6180 for all n. The larger the n, the more approximate it will be. That is just to find one level. To find another, say the 38.2% level, dividing any nth term in the series with the nth+2 term will approach .3819 for all n.

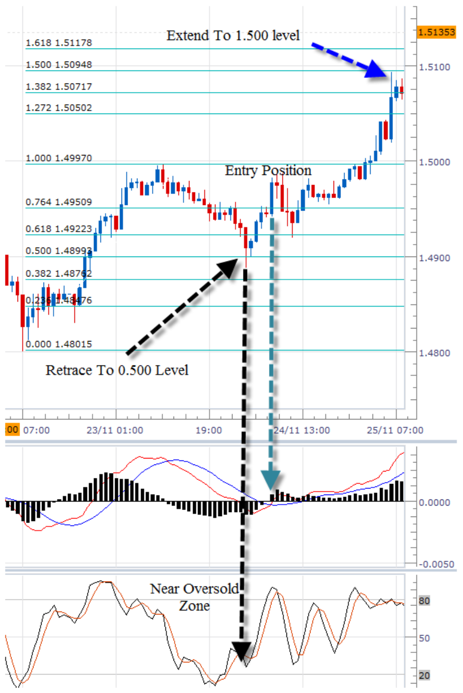


Figure 14: An example of how the market obeys the common Fibonacci levels as support and resistance levels. Moreover, it shows how trading with this alongside the previously discussed MACD indicator can show entry positions into the market. [52]

In this figure, the trader is using the MACD with the stochastic oscillator coupled with the Fibonacci levels to spot support and resistance levels. Notice how they form at the ratios from the Fibonacci series. The way the forex market and people trade is truly mysterious.

The basic idea behind these retracement lines is the tendency for a security which has undergone a large price swing to undergo a period of correction thereafter. In such a circumstance often these support and resistance levels are indeed the Fibonacci Retracement lines, and thus are often used as levels for entry and exit positions. Usually one would expect a trend to reverse to at least the 50 \% line if the trend is truly broken, however failure to do this is indicative of a continuation of the current trend. One of the key advantages of Fibo Retracement lines is that when you enter the market at one of these support lines, you can set a very tight stop loss and thus minimize risk. [17]

This indicator is arguably used best with other indicators such as the MACD (describe in point VI). If you see a currency pair retracing to a level and the MACD has flipped axis in a way that agrees with the Fibonacci bounce direction this would be a good entry level. Additionally, one can use the stochastic indicator as well to line up the Fibonacci retracement levels with oversold indications from the stochastic to create buy entry levels [14].

2.3.11 Average True Range(ATR)

The Average True Range is a tool developed by Welles Wilder to measure volatility. The ATR uses a measure called the true range and then plots an n-period moving average of the values obtained. The true range is defined as the maximum of:

- Current High - Current Low
- | Current High - Previous Close |
- | Current Low - Previous Close |

2.3.12 Directional Movement Indicator

The Directional Movement Indicator in conjunction with the ATR forms the basis of the Wilder trading system. The directional movement index is split into two components the DM$^-$ (minus DM) and the DM$^+$ (plus DM). They are defined as follows:

$$\begin{aligned}
\delta_i^h &= High_i - High_{i+1} \\
\delta_i^l &= Low_{i+1} - Low_i \\
DM_i^+ &= \begin{cases} \delta_i^h & \text{if } \delta_i^h > \delta_i^l \text{ and } \delta_i^h > 0 \\ 0 & \text{else} \end{cases} \\
DM_i^- &= \begin{cases} \delta_i^l & \text{if } \delta_i^l > \delta_i^h \text{ and } \delta_i^l > 0 \\ 0 & \text{else} \end{cases}
\end{aligned}$$

The superscripts here are indices, whileidenotes the current bar and i+1 is the previous bar.

This is another unique indicator because it does not indicate price movement. Instead it measures volatility of a pair. It takes 3 differences, between the recent high, low and previous close and bases it off the largest value. Your trading station will then plot it automatically and one can use the ATR to verify market strength to form decisions on your next position. Rather than having the range being the difference between the day's high and low, the true range takes into account the previous day's closing price if it happened to be outside of the current day's traditional range [10].

The ATR is simply the largest value of the minimum subtracted from the maximum. To find the maximum it is the larger of the two values that include the previous closing price or the high price of today, whichever is larger. Similarly, the minimum is found by taking the smaller value of previous closing price, or the low of the recent day, whichever is smaller.

The smaller the ATR is displaying the more likely the market is or will be consolidating or waiting to enter a breakout, that is a huge up or down trend after consolidation. If the value is large it means the trend is strong or the breakout is actually happening and consolidation is over.

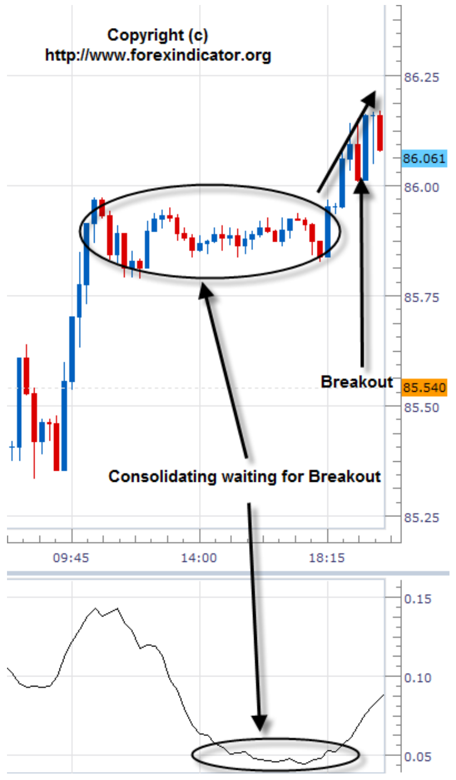


Figure 15: From this figure one can see how increasing ranges support the idea that traders will keep selling or buying a stock throughout the day, or let it consolidate if the range is decreasing. [52]

As seen on the graph provided by forexindicator.org, the value of the ATR is small while the market isn't getting beyond support and resistance levels. Once the indicator increases the trend breaks out and the resistance for this particular example was broken [10].

2.3.13 Average Directional Movement Indicator (ADX)

The Average Directional Index is an indicator developed by Welles Wilder which incorporates all of his previous indicators. The ADX is a more complex method than MAs in measuring the current trend of a currency pair. The basis for such an indicator is the directional index (DI). The DI always assumes a positive number, however this value is assigned to one of two variables, the plus-DI denoted DI^+ or the minus-DI denoted DI^-. Which variable takes on the value is determined by the following equation:

$$\begin{aligned} ADM_i^+ &= EMA_i(DM_i^+) \\ ADM_i^- &= EMA_i(DM_i^-) \end{aligned}$$

$$\begin{aligned} DI_i^+ &= 100\tfrac{ADM_i^+}{ATR_i} \\ DI_i^- &= 100\tfrac{ADM_i^-}{ATR_i} \end{aligned}$$

The Directional Movement Index (DX) is defined as:

$$DX_i = 100\frac{|DI_i^+ - DI_i^-|}{DI_i^+ + DI_i^-}$$

Then the ADX_i is simply the exponential moving average of DX:

$$ADX_i = EMA_i(DX_i)$$

In MT4 the default is a 14 period EMA.

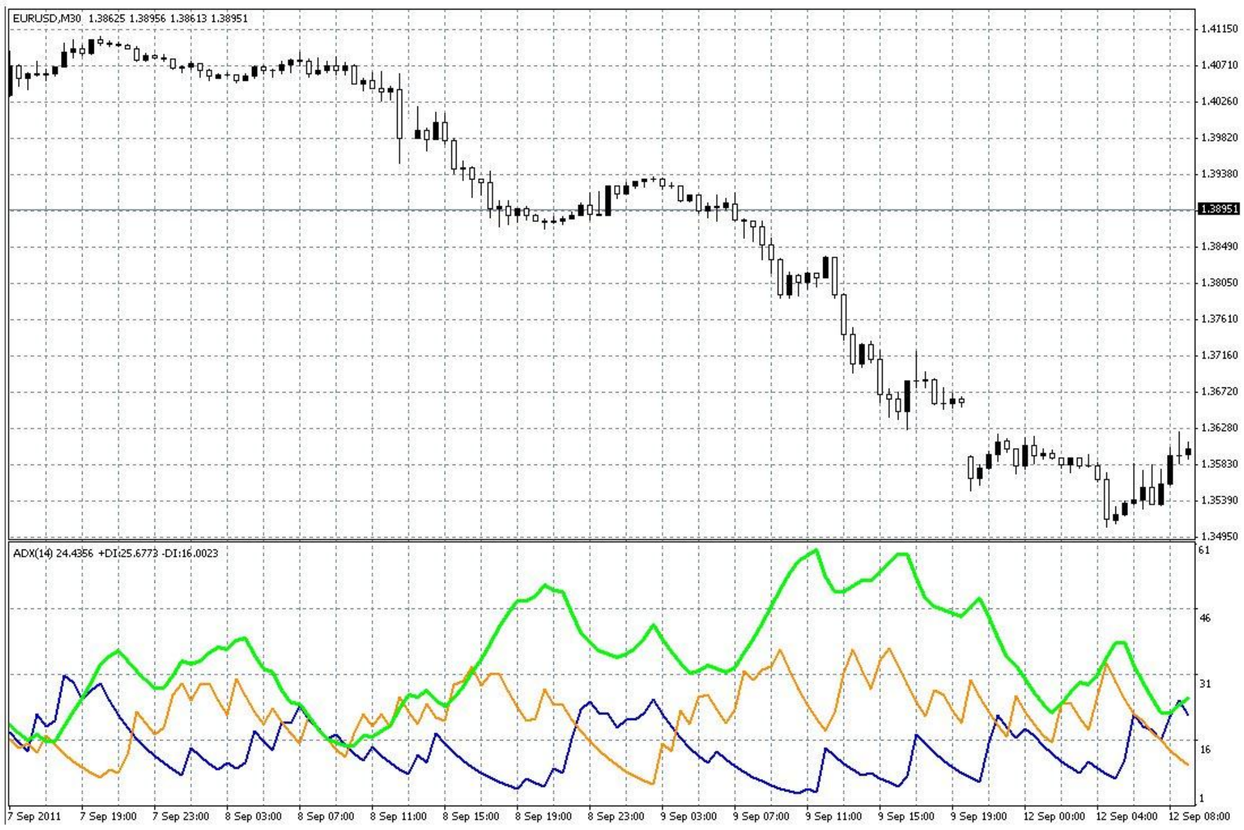


Figure 16: The ADX Indicator with DI^+ in Blue, DI^- in Orange, and ADX in lime green

When the ADX is sloping downwards or below the 25 mark it usually means the markets are consolidating. If it trends upwards above 25 this is usually a good time to buy [11]. Like the CCI this also has divergence to present a possible trend reversal. If the ADX says the market is consolidating and the pair is trending upward this could potentially be a good time to go short.

2.3.14 Parabolic Stop and Reverse (SAR)

The Parabolic SAR is yet another tool developed by Welles Wilder. It is similar to the moving average; however it gives a slightly more obvious indication of trend and actually calculates an expectation for the next bar into the future. The value of the SAR appears above the price if it is a downward trend and below the price if it is an upward trend.

$$SAR_i = SAR_{i+1} + \alpha \cdot (Price_{i+1} - SAR_{i+1})$$

Where α is the acceleration factor. MT4 sets by default α = 0.02 and it doubles every bar the current trend continues in that direction and returns to its default during a trend change. The $Price_{i+1}$ is defined as the low of the previous bar if it was a downward trend and the high of the previous bar if it was an upward trend.

The SAR is commonly used by many traders as an indicator for exit strategy, completing their trades when the SAR changes its trend. In many cases it is also used as a trailing stop. [20]

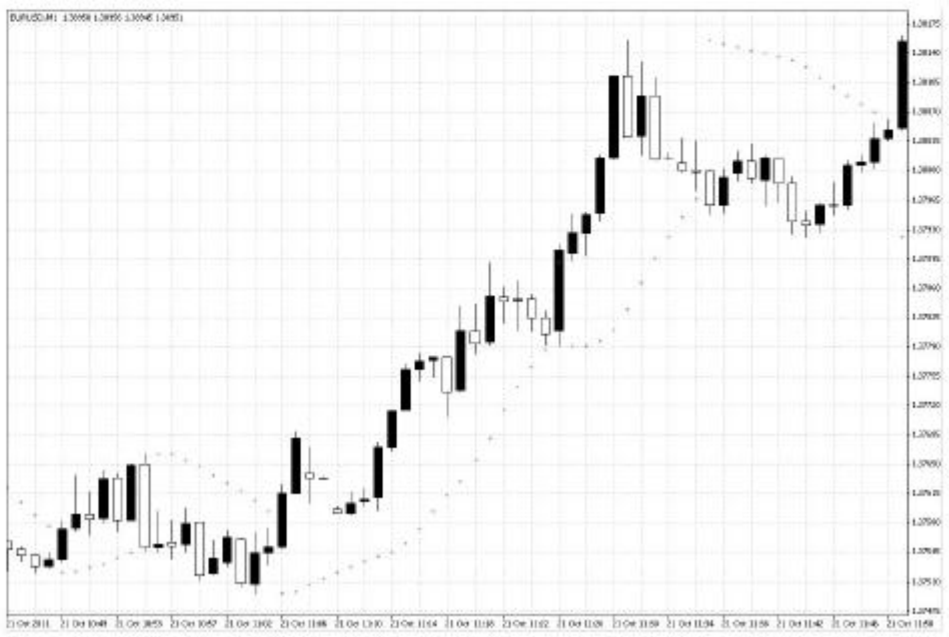


Figure 17: Parabolic SAR

The 'Stop and Reverse' is a popular indicator used mainly to identify the direction of price movement. When the dots are underneath the 'candlesticks' it indicates a time to buy and indicates a sell when the signal is above the 'candlesticks' [12].

This is a simple indicator that is easy to read in a simple glance. It works best with other indicators to reaffirm trends. This indicator is not very helpful alone, especially if there is rapid price movement. Another use of this indicator is setting a stop loss level. If you have decided to buy, have your stop loss trail the previous Parabolic SAR “dot”. That way your profits will be maximized and your losses will be minimized.

As shown on the figure previously, the SAR is simple to use. If the dots are above the currency pair go short and if below go long. The steeper the slope from one dot to the next shows trend strength.

2.3.15 Ichimoku Kinko Hyo

The Ichimoku Kinko Hyo system was developed by a Japanese newspaper man in conjunction with several students whom he hired to do back testing. The base of the system uses 5 different charting lines; it is a hyper visually based system and its name directly translates to "Equilibrium chart at a glance". The lines are as follows:

- Tenkan Sen ”turning line” = (Highest High +Lowest Low)/2 for the past 9 periods
- Kijun Sen ”standard line” = (Highest High +Lowest Low)/2 for the past 26 periods
- Chikou Span ”lagging line” = Current Closing Price shifted (into the past) 26 periods
- Senkou Span A ”1st leading line” = (Tenkan Sen + Kijun Sen)/2 shifted (into the future) 26 periods
- Senkou Span B ”2nd leading line” = (Highest High + Lowest Low)/2 for 52 periods shifted (into the future) 26 periods

Together the Senkou Span A and B form what is called the IchimokuKumo which translates to cloud and is the area between these two lines. The Kumo is the shaded area in the Ichimoku indicator and serves as the equilibrium range for the currency pair in question. A trader utilizing this strategy will long pairs when the price breaks above the cloud and short pairs when it breaks below the cloud.

The size of the Kumo is an indication of the current market volatility, where a wider Kumo is a more volatile market. When the Senkou Span A is above the Senkou Span B this is seen as a bullish signal akin to a MA-crossover strategy and Senkou A is similar to the fast-MA. Likewise when the B is above the A it is a bearish sentiment.

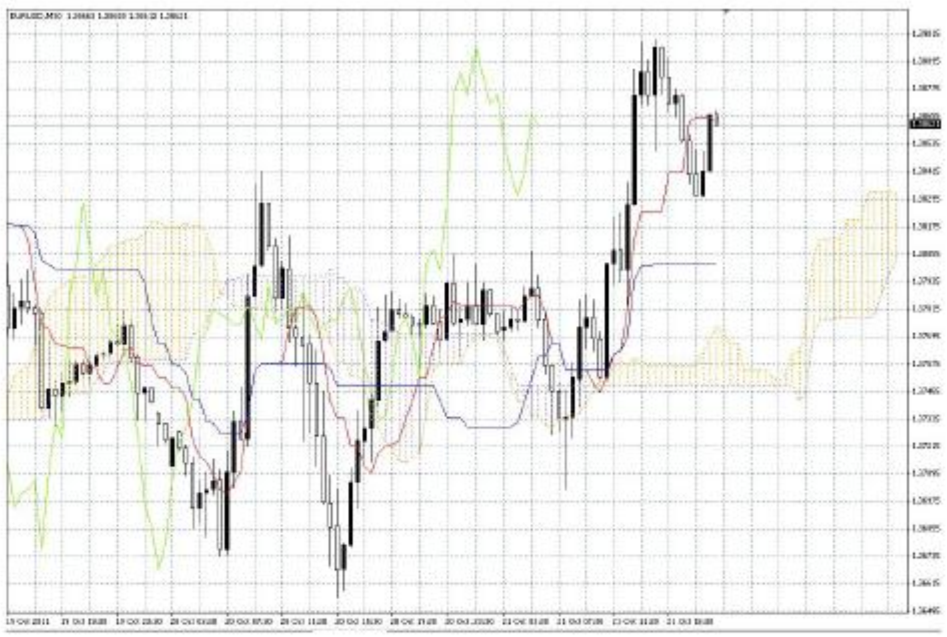


Figure 18: Ichimoku Kinko Hyo with Tenkan-sen = Red, Kijun-sen = Blue, Chinkou Span = Lawn Green, Senkou Span A = Orange, and Senkou Span B = Purple. The hatched areas are the Kumo. An orange Kumo is a bullish signal and a purple Kumo is a bearish signal. Notice the Kumo is shifted forward and thus is an indicator of future sentiments.

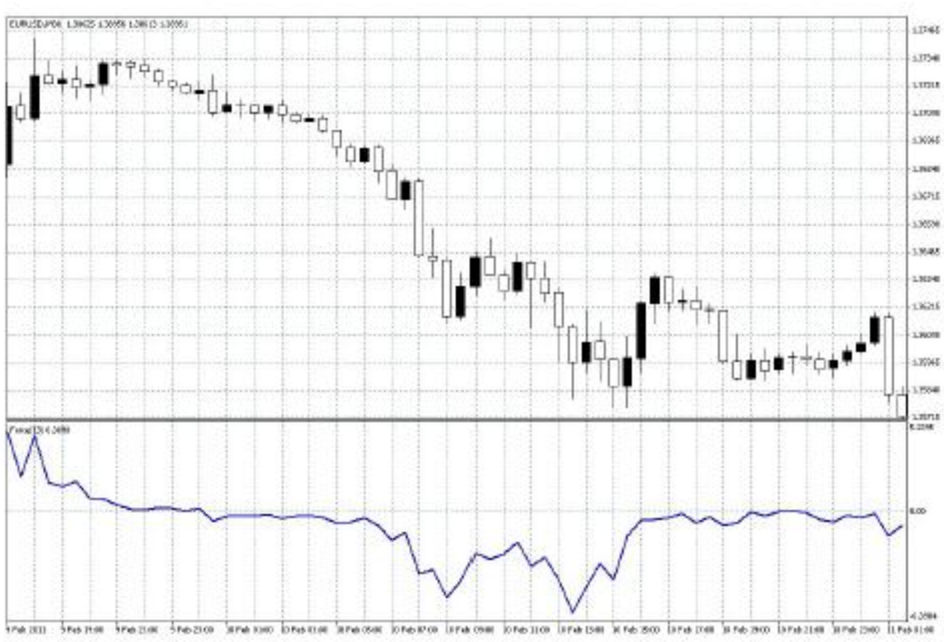


Figure 19: Forces Index - Notice how long periods of small movement has little impact on

2.3.16 Forces Index-FI

The Forces Index is another trend tool which is removed from the regular chart window. It is essentially a tool used to determine the current direction and strength of a trend. The calculation for FI is:

$$FI_i = Volume_i \cdot (MA_i - MA_{i+1})$$

Hereirefers to the current period, i+1 the previous, and MA anyn-period moving average. The index in conjunction with a short moving average (2-period) is well suited to finding exit and entrance positions. For higher order moving averages (13-period) the index shows the trends and their changes. [24] Since this indicator incorporates volume it is a good indicator of where the buy/sell pressure lies.

A typical FI trading strategy would involve buying when the FI becomes negative during increasing tendency and selling when it becomes positive during decreasing tendency. When a new trough/peak is reached in the FI indicator a continuation of the trend in that direction can be expected. [24]

Chapter 3: Methodology

3.1 Trade Methodology

3.1.1 Personal Trade Strategy – Adrian Delphia

After the first seven weeks of experimental trading and learning about the forex market and different indicators, the next third of the course was dedicated to a more structured trading simulation; this took place for the past recent seven weeks of the course. A new demo account valued at $100,000 leveraged 50:1 was setup and trades were made according to my plan so that they were like my real capital for our company was invested. This account was less experimental than the one in the first part of the course, and many things were learned now that I was trading according to a well-defined plan. It was a dynamic plan that was modified along the way to increase profitability and reduce risk. The changes made to the system and the reasons why are discussed in this section.

Trading these past few weeks taught me a lot of things about the forex market. I took trades much more seriously this term and traded only to make a profit. In the first part of the course I made many trades without even thinking just to see outcomes and try to gamble with the system. When I was curious about a trade this term I would trade just a micro-lot just to see if the trend went the way I was thinking but did not have enough confidence to put my 'real' money into it (trade a standard lot or more). For this reason I will only comment on the real trades I made for the purpose of furthering my account profit which consists of trades where more than one standard lot was bought or sold.

One of the main things I learned from trading was the unpredictability and wildness of the market. I did not think currencies could swing hundreds of pips against each other in every way! Sometimes it even happened multiple times a day. It really shows how volatile the market can be, and that there is definitely money to be made for the right trades. I also learned the importance of fundamental news in the market, as many different things happening in Greece involving the debt crisis immediately affected currencies everywhere. I never really thought it could have such a dramatic effect as some did such as the currency strengthening significantly when Germany finally agreed to a bail out voting in December for Greece.

My trading methodology had major modifications to it, but it mostly was simplified. In the first third of the class my method was too confusing to follow or abide by. There were many conditions and too many indicators involved to produce a clear method. I tried simplifying it for ease of use, but it didn't add anything to my system, just made it less profitable. I have since simplified my system enough that I can strictly trade by only its rules.

My current strategy consists of just trading off of support and resistance lines. At the start of my trading I would just 'eyeball' it and go look there's resistance so I'm going to sell or buy if there's report. This was the baseline for my strategy. However, as you will soon read, I have certain conditions in which I am allowed to enter now.

Eventually, to make it more specific I needed to define exactly when I would buy or sell with support and resistance. This is difficult to do when you don't know exactly how your mind is thinking but I tried to narrow it down each week when I noticed another pattern or what was working and what was not. My current entry position guidelines for my support and resistance system are as follows:

"**Support or resistance lines need to be obeyed at least more than 1 time unless it is a global maximum or minimum in a 15 minute window or larger period"**

This is my golden rule, and every case of when I can and cannot enter a trade is outlined using the following figure. I have labeled 8 points accordingly and describe whether or not I could enter a trade and why.

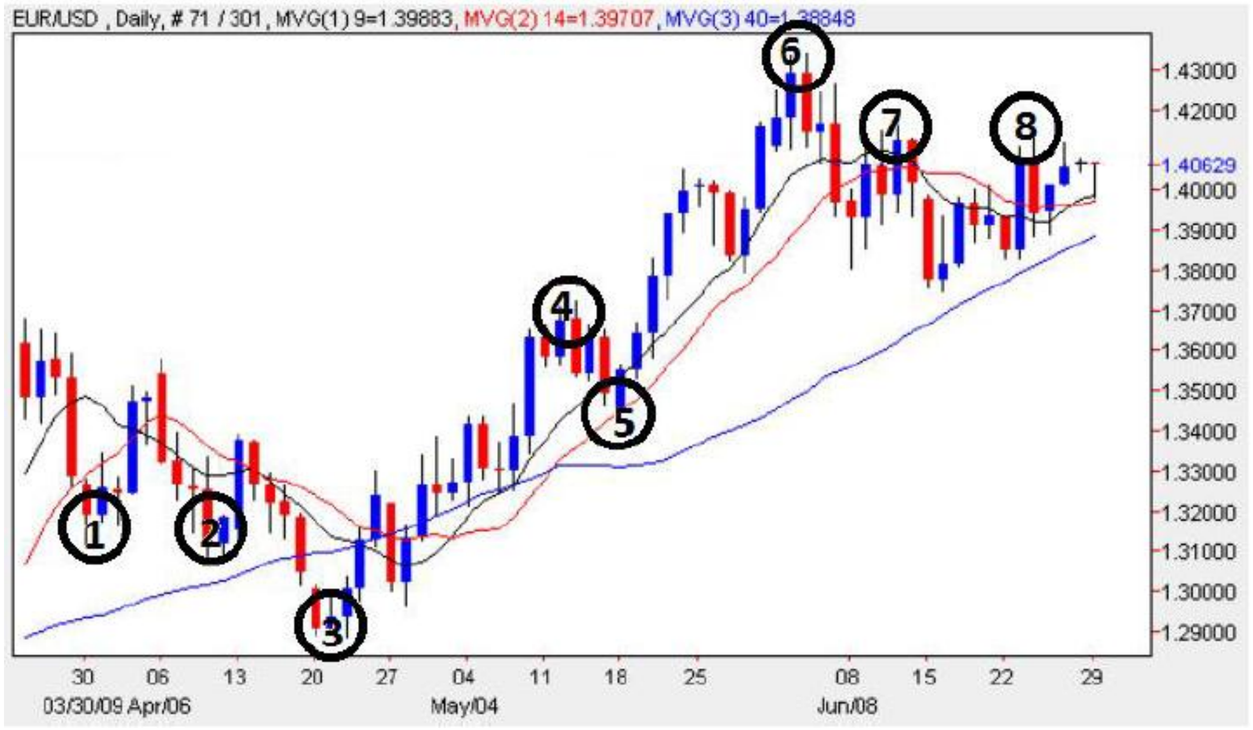


Figure 20: A sample graph showing when it is allowed and not allowed to trade according to my system. [53]

For all below remarks; if it says "allowed to trade" that means if that line is tested again in the future a position can be entered.

- Point 1: **Okay** to trade that support line because it is obeyed more than once (like at point 2)
- Point 2: **Okay** to trade that support line (See point 1)
- Point 3: **Okay** to trade that support line because it is a global minimum
- Point 4: **Not okay** to trade, local maxima obeyed only once
- Point 5: **Not okay** to trade, local minima obeyed only once
- Point 6: **Okay** to trade that support line because it is a global maximum
- Point 7: **Okay** to trade that resistance line because it is obeyed more than once (like at point 8)
- Point 8: **Okay** to trade that resistance line (See point 7)

In order for two peaks or two values have the same resistance or support lines, the difference in height between the peaks in questions needs to bewithin the difference of the minor tick marks . For the example figure used previously, the y axis ticks happen every 100 pips, sothe two peaks would need to be within 100 pips of each other. This scales nicely for all window time frames you are using. Smaller time scale means a smaller y axis range means the values need to be closer together to form a support or resistance line which makes sense.

My exit position is a trailing stop loss between 10 and 20 pips. I used to not use trailing stop losses and either set a discrete stop loss, or monitor the trade until I decided to exit for seemingly no reason. Using the trailing stop loss every time was a great improvement to my plan since B term because it really maximizes profits while minimizing losses. There are edge cases where a currency could theoretically drop 21 pips before skyrocketing 200 pips making me “lose” all that profit however this is rare and not a safe practice in trading and would affect my pull-down negatively.

These changes were necessary to make so I could figure out what was working and what was not. Before if I was losing a trade I wouldn’t know what to do differently because I didn’t really know what I did. If I just ‘eyeballed’ a trade, and it didn’t work, how could I know what to change to make a better trade next time? Picking specific conditions and trading on them allowed me to see what worked and what didn’t. I then could change a specific part that didn’t work and keep trading and evolving my system. For example, in an earlier version of my system I could trade at any local minima or maxima. I quickly figured out this led to many negative trades due to natural and common market noise and volatility so I only trade on local minima and maxima if they are tested at least two times. This greatly improved my profitable trade percentages.

Implementing the rule that a local peaks and valleys need to be hit more than once helped avoid trading on false support and resistance due to normal market fluctuation and noise. I again modified this rule a little more to exclude global minima and maxima. The reason for this is because if there is one maximum or minimum over a huge period say 4 hours or something it is unlikely it will be busted through so it is okay to have the option to trade on global peaks and valleys, especially if it is a line that hasn't been reached in days and you think the market will continue obeying it.

My trade sizes have more or less stayed the same throughout my trading system this term. The vast majority of my trades are one standard lot. Under certain conditions I do trade with two standard lots but never more than that. My method for when to trade 2 standard lots is when the line has been tested more than two times, and when viewed from a different time window, the line holds true still. A line obeyed three times or more, especially across multiple time windows is more likely to hold true and be profitable that it is worth risking more capital, 2 standard lots instead of 1.

My risk management also changed minimally through the past seven weeks. With being a new trader, it is important to keep risk as low as possible to build and condition oneself to good habits. Low risk is always better than high profit if one had to choose between the two. The backbone of my risk management is always setting a stop loss. Whether it is a trailing stop loss or a fixed stop loss, I must always place one to minimize risk. On short term trades it will be between 10 and 20 pips, my most common trades. For long term trades it can be larger, around 100 or 200 pips, but I will never risk above 15% of my capital for my long term trades.

Trading an average of 1 standard lot on my initial investment of 100 thousand dollars means I never risk more than 2% of my capital. Setting a stop loss to ten pips ensures I don't risk more than 5% of my leveraged trade (assuming 1 pip is about 10 dollars). I don't like to enter trades often, but rather wait for just the right trade to come along when I have almost no doubts. This will keep the risk low in my account and lead to consistent trades.

Finally to reduce risk even more, once 2% of my capital is lost in a day trading will cease until tomorrow. Additionally if I have made 3 consecutive good trades it is also time to stop. These two rules will maximize profit while reducing risk at the same time. No reason to keep trading after a good day, quit and do it again tomorrow rather than ending on a loss because I feel I'm on a 'winning streak' with three positive trades in a row. Combining control with discipline is the backbone of my strategy with the most important part being to strictly follow it, as I have designed it in a way where this can be done.

Here I will detail a sample trade made throughout B term and annotate it. You can see how well it matches up to my methodology. To see a review of all trades made in B term refer to the **Trade Portfolio** section of the report.

Out of all the trades I made this term I am going to explain the one that demonstrates my trading methodology the best. It is swing trading according to my plan with the EURUSD currency pair.

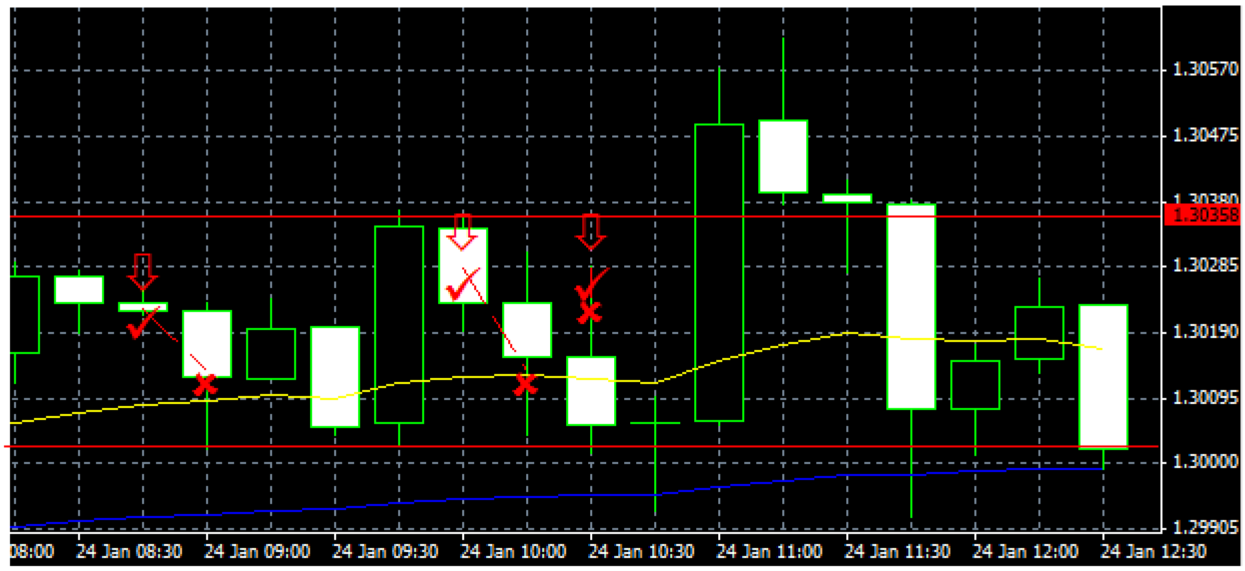
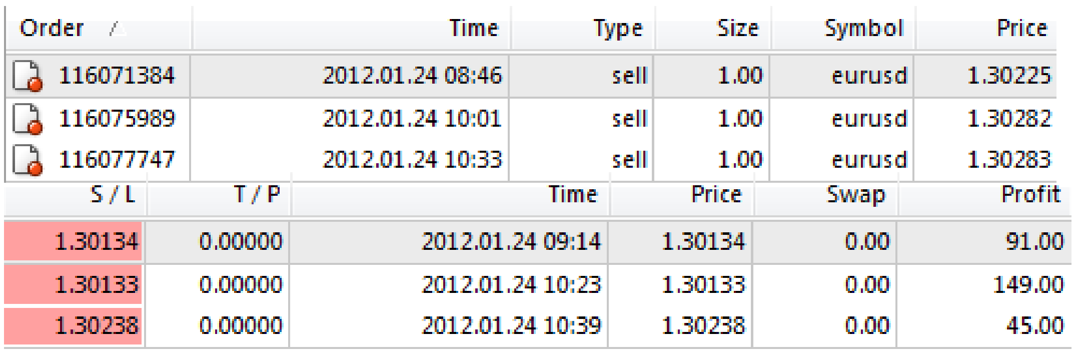


Figure 21: EURUSD Scalping.

First, deciding if these similar peaks and valleys were support and resistance lines according to my methodology was done. So the difference each minor tick value is 9.5 pips means that I can place a support or resistance line if two peaks or valleys occur within 9.5 pips from one another. The support line shown is easy to see that it follows my rule, and the resistance line follows the rule too but the other peaks for the line appear off to the left cropped out of the picture to more accurately show the trades made.

The left-most arrow was my first trade where it seemed to decline after resistance was nearly met. I sold one lot and set a trailing stop of 20 pips and I closed the trade after a profit of 9 pips. I continued watching this market data and again where the middle arrow indicates, the currency hit the resistance line and started to go down. Sure enough the pair went down again and my trailing stop activated profiting me 15 pips. I still further watched the market and after the currency bounced right back up to the resistance line and started to decline I sold another lot. This one had a lot of jumping around in it so I got out early happy with a profit of 4.5 pips. Looking forward in the future if I didn't go to bed I could've employed this strategy many more times during the consolidation to have made even more pips!



Order	Time	Type	Size	Symbol	Price
116071384	2012.01.24 08:46	sell	1.00	eurusd	1.30225
116075989	2012.01.24 10:01	sell	1.00	eurusd	1.30282
116077747	2012.01.24 10:33	sell	1.00	eurusd	1.30283

S / L	T / P	Time	Price	Swap	Profit
1.30134	0.00000	2012.01.24 09:14	1.30134	0.00	91.00
1.30133	0.00000	2012.01.24 10:23	1.30133	0.00	149.00
1.30238	0.00000	2012.01.24 10:39	1.30238	0.00	45.00

Figure 22: Summary of three previous trades made.

The total profit was 28.5 pips all in 90 minutes of watching the forex market. What I have learned is that consolidation can be a good thing if you're resistance and support lines are holding on. Moreover I should've considering reversing my position when I bought the currency back at the lows, that is buying the currency when I close my position. Then sell it when I open my next position. I could've in theory made twice the above amount! Finally, according to my rules I was allowed to sell two standard lots on the last trade I made because the line was tested two or more times. I decided not to which is fine anyways because I reached my performance for the week that da anyways, just good to recognize for future trades.

3.1.2 Personal Trade Strategy – Brendan Hamm

As a member of Empowerment Capital, we have been each given a $100,000 foreign exchange account which we have been tasked with trading. As a member of this organization we were to develop a trading strategy with which we were to adhere to in an attempt to create profit. As is expected for a novice trader such as myself, such a trading strategy is not necessarily concrete, but rather evolves over time. Over the course of the past couple of months my personal strategy has evolved in many facets for a multitude of reasons.

Any trading strategy must match the goals of the investor. As part of Empowerment Capital, we have created the goal of achieving a system which can generate 11% per year. With this relatively high level of profit we can charge a management fee of 2% of our clients' average annual account as well as a 20% performance fee. In part with this goal, minimizing drawdown is of utmost importance, as no investor wants to experience large swings. For the purposes of this project, I would like to set a maximum drawdown of 5% with the aim of achieving my goals in only 2%.

My trading strategy began outlined as follows:

1. Before beginning a trading session, making sure I am up to date on any current news will be of utmost importance. I am not allowed to trade until we have read the Gartman Letter for that day. In addition, I must check on ForexFactory.com for any news times which are scheduled. For any currency pairs involved in scheduled news, I will not allow any open orders during a one-hour block centered at the scheduled time. In addition, I will only trade major pairs which at least one belongs to the current trading session's region.
2. Multiple time frame charts will be brought up for the currency pairs in question. If any pair exhibits similar trends on the 5-min, 15-min, 30-min, and hour timeframes, they will be considered for trades in the following steps.
3. Every pair which has made it to this stage will be charted on the 5-minute time scale in conjunction with two exponential moving averages of periods 5 and 15 respectively. In addition

a 50-period simple moving average will be used as a signal line. Only buy orders will be placed if the ask price lies above the signal line and only sell orders if the bid price lies below the signal line. The only exception to this SMA rule will be when a large movement in price takes place followed by a retracement of over 50% in which case the signal line will be a 20-period moving average. This adjustment will make trading possible during large oscillation cycles, which while inducing greater risk also allow for greater returns. Due to this, all trades made using the 20-period SMA will be subjected simply to the base lot size.

4. Since I am trading only pairs which have a general trend our next step is to create support/resistance lines for the past 12 hours of trading. Emphasis will be placed on the most recent support/resistance lines through the use of color (yellow) and lines which have been broken during this period will be colored a dark blue.
5. I will then begin looking for entry positions both based on our EMA cross and signal lines, as well as the resistance lines and candlestick patterns. I will not sell close to the lowest support line and will not buy near the highest resistance level, unless they have been broken.
6. If a currency pair meets the entry position criteria then determining lot size will be the next step. The base lot size will be 1 standard lot. If the chart incorporates candlestick patterns (i.e. reversal/continuation patterns) favorable to my intended position this base lot size will be doubled.
7. For long positions a stop loss will be placed below the next support line which lies below the entry position. For short positions this s/l order will be placed above the next resistance line which is above our entry position. The actual numerical value will be calculate/approximated as 5% the distance between that line and the next, below for long positions and above for short. Similarly the take profit orders will be placed in a similar fashion using a distancing factor of 10%. These t/p values will lie above the next resistance line for long positions and below the next support line for shorts.

8. If I am in a long position and that price crosses the resistance line which has t/p 10% above it, I will update the s/l and take profit orders. This will be done by using the same 5% and 10% calculations and adjusting them up to the next pivot point. In a similar fashion we will do this with short positions. If a second update is made to any of the positions I will repeat all of the previous analysis steps. If the pair still meets all of our criteria an additional lot may be added using cognitive discretion. This extra lot will be treated in the same manner as all other trades.
9. If at any point there is indication (i.e. candlestick reversal) that a currency pair will not continue in the direction of the trade it will be exited immediately.

My original strategy revolved heavily with the use of moving averages. However, as time progressed I have reconsidered the heavy importance which the above trading strategy places upon these averages. My original thought process in using them in the first place began with the notion that in capturing small term price movements, a large volume of smaller trades could potentially see the returns I sought with a low drawdown. However, just using moving average crosses seems to initiate a lot of trades during non-movement periods and thus has a large number of small losses due to these “false” signals.

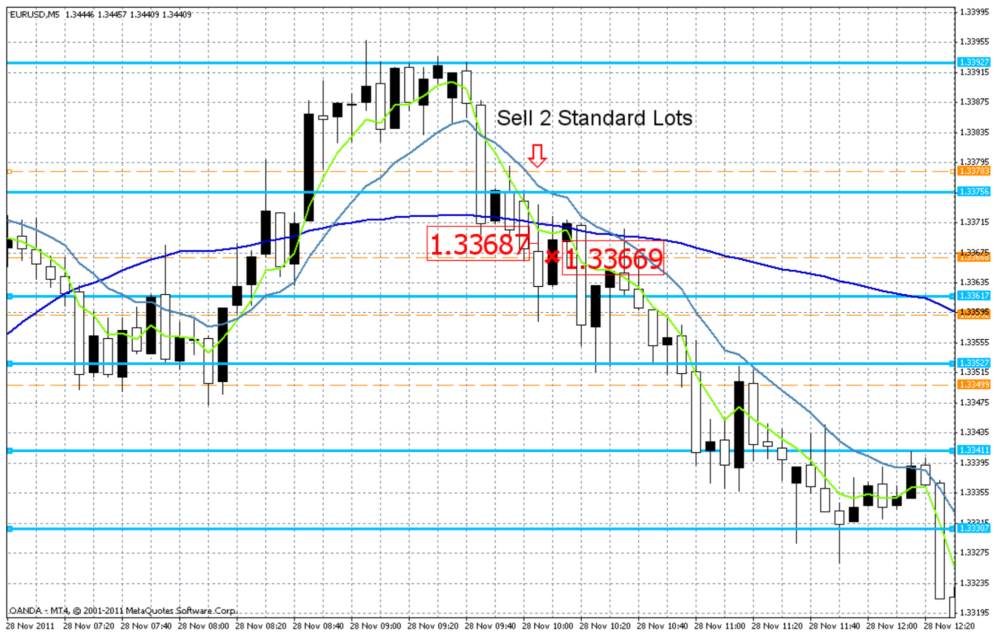


Figure 23: A Sample trade for the above trading strategy

For the reason of too many signals I had incorporated candlestick patterns to try and reduce the amount of signals. It seemed a natural thing to do at the time, but for a novice trader with a myriad of patterns, it is easy to confuse these candlestick patterns even though I keep graphics of them in front of my computer at all times. In addition, this may have narrowed trade numbers down much more than I anticipated and I find that in the majority of my sessions I did not trade.

For the reasons outlined above, I have taken a more pattern recognition approach to trading, while negating the use of moving averages in my strategy. Throughout the course of this project we have been utilizing WorldFXIQ.com to learn many basics of strategies for trading. Incorporated in this were several lessons regarding patterns in price graphs. These patterns include Double Top/Bottom, Double Pullbacks, Round Bottoms, Head & Shoulders, Gaps, Wedges, and Reversals. My final trading strategy rested solely on the presence of these patterns.

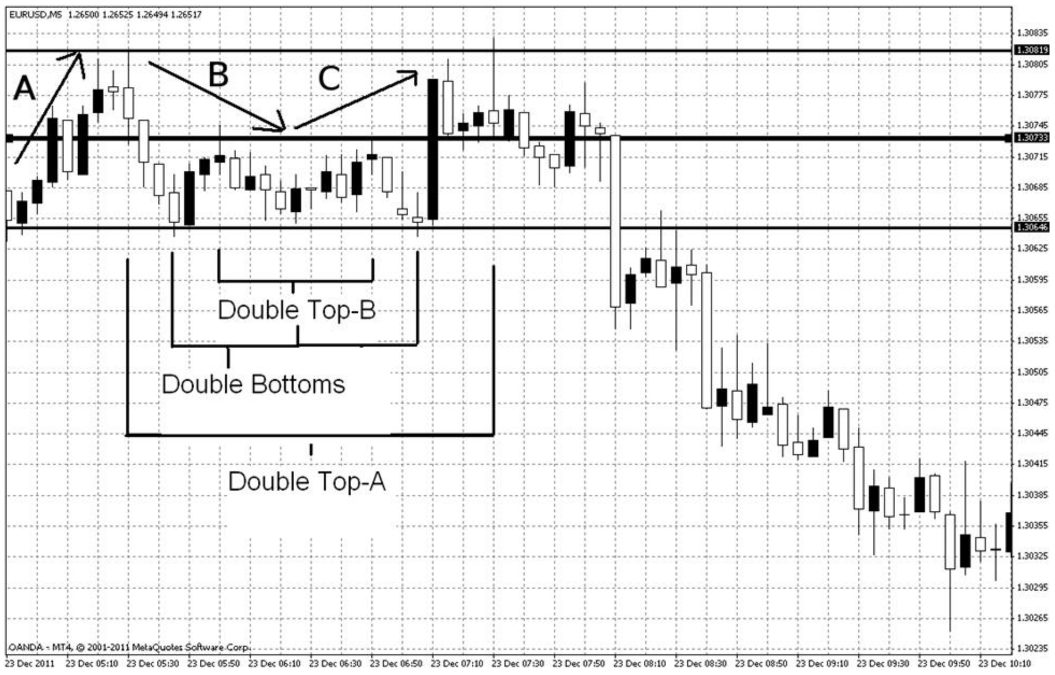


Figure 24: An example of a Double Top pattern with a minor nested Double Top pattern and two minor nested Double Bottom patterns.

A Double Top/Bottom pattern is one in which a currency pair experiences a reversal at a specified level, referred to as a support for a bearish reversal or a resistance for a bullish reversal. The pair must then reverse back again towards this support/resistance line. It is this point which becomes crucial to this pattern as it will wither progress into this pattern or revert to be a Double Pullback. As can be seen in**(Figure 1)**, a Double Top occurs at the culmination of trend line C. At this point the price bounces off of the resistance line. In this image the downward trend experiences a short period of sideways trading, but as is often the case with this pattern in the EURUSD the final impact is tremendous, offering a trader the possibility of a very profitable trade.

Similarly, there exist two nested Double Bottom patterns within this same trend pattern. During these Double Bottom Patterns the bottom trend line acts as a support line and the price tends to rise after reaching this level. A Double Pullback is also displayed in this same image as the support line which the Double Bottom trend relies upon is broken at the far right of the chart.

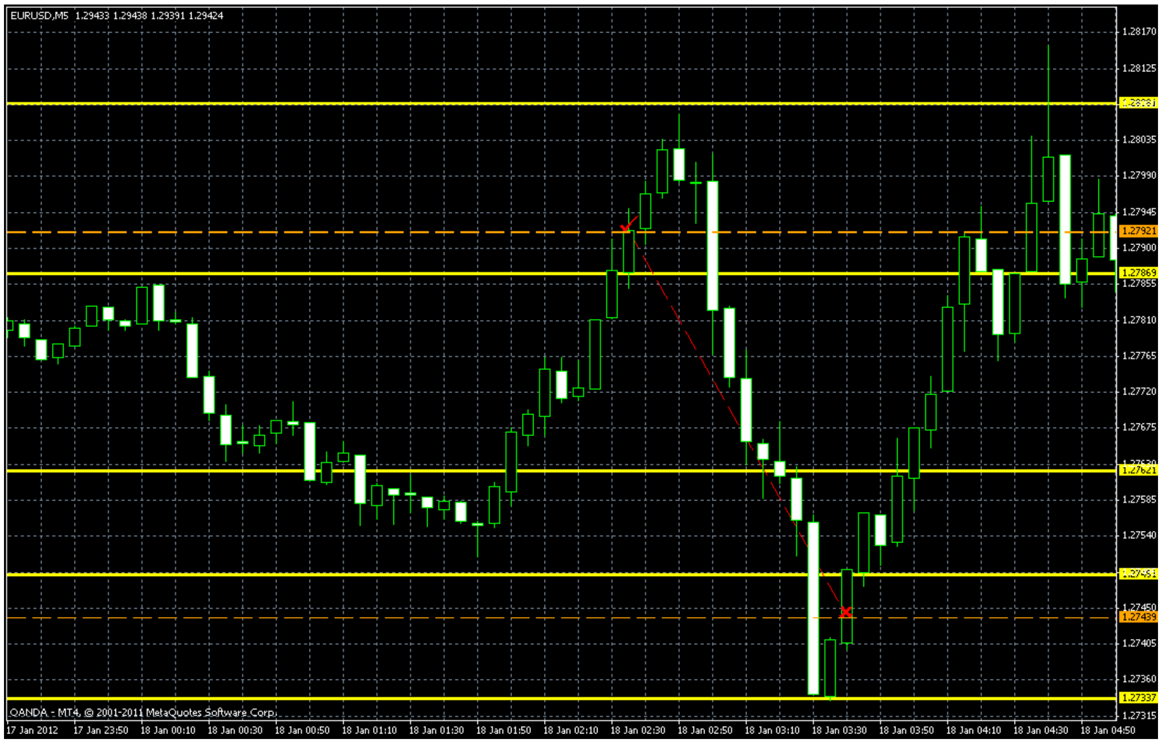


Figure 25: A sample trade from my final strategy. In this trade a sell stop was placed with sufficient room to an un-shown resistance line indicated by the top yellow line drawn by our program pivots_v1.07. Our trailing stop program then takes over the trade, ensuring a good exit point.

Of the patterns described above, I have found the Double Top/Bottom pattern to be the most useful and profitable trading pattern. However, this pattern alone can take a long time to develop, so with limited time it becomes difficult to catch. For this reason I wrote a program which will automatically recognize this pattern and trade accordingly. I have had some success in my beta testing for this program but only on short position trades. I feel that redefining the way it operates can further improve upon it, specifically for long position.

Another thing I found necessary to change was my lot size. I began trading only single and double lots as outlined in the strategy above. I found that the fluctuation in my account was much less than what I would want if it ever were to end on the positive side of the fence. As I moved into November the majority of my trades netted profits and losses of around $100, or 0.1% of the total equity. Around this time my highest drawdown was less than 1.5, which was within the allowable drawdown of my strategy. For this reason, I adopted a system of trading 2 standard lots as my base size, while expanding up to 4 standard lots for optimal positions.

Finally, my stop loss and take profit levels regarding the pivots had changed as well. Before, it was stated that I would set targets at 5% and 10% respectively from the pivot lines. However, in reality this tended to be too close, so I wrote a program to determine the average bar size over the course of the past hour. I then use this average to place my stop loss, at a factor of 1.5 average bars (on the 5 minute level), from the pivot in the stop direction. Utilizing this formula I believe my trading became more isolated from random noise, while still allowing me to exit if the initial assessment is proved wrong. Regarding my Take Profit Levels I have wrote an expert which cycles through my open orders and sets a Trailing Stop using this same formula. My Stop Loss feature only enables if the current stop loss is further away than the newly calculated level. These are two things which I find would've been cumbersome to do manually.

Finally, at the end of B-term I decided to switch the program I was using to execute my trades from TradeStation 9.0 to Meta Trader 4. Within TradeStation I found that I made many mistakes simply due to a lack of knowledge of that trading platform. I felt much more comfortable with the MT4 platform and have been spending much of the project writing scripts to aid in my trading.

My risk strategy has been set such that I never have more than 2% of my account on the line at any given time. Using the previous strategy, this actual level has never been surpassed or even approached. With the stop loss system I was using before my total equity at risk never exceeded 1% and was generally less than 0.5%. Utilizing these new levels for trade size and stop loss my exposure to the market has yet to exceed 1%, but remains generally above the 0.5% level.

3.1.3 Trade Strategy for Srinivas Vasudevan

As members of Empowerment Capital, LLC, we given a $100,000 foreign exchange account that we are assigned to effectively trade. We are responsible for developing a trading strategy that delivers returns consistent returns with a low drawdown. As a first step towards achieving this goal, we familiarized ourselves with common risk-management strategies such as employing stop losses and take profits and learned about various fundamental and technical indicators. Then, we developed our own indicators, such as support and resistance lines, entry and exit point calculators, and other tools. During this time, as novice indicators, we also familiarized ourselves with existing indicators such as the MACD, Bollinger Bands, simple moving averages, exponential moving averages, and the Parabolic SAR. Since we were unfamiliar with trading, our strategies tended to evolve over time and we eventually were able to achieve concrete trading goals and measure our performance against our objectives.

My trading strategy is to make one or two trades a week when all my trading conditions are satisfied and to endeavor towards an 80% success rate for my trades. Since I make very few trades that are generally short-term (within 30 minute), I trade with a lot size of 3 standard lots (6% of initial amount of $100,000). I use a trend following trading strategy, meaning that I buy when the market is moving upwards and sell when the market moves downwards, making trades only when my confidence in the trend is very strong. My goal is to achieve an average profit of 10 pips on all trades (not just profitable trades) with a stop loss of 15 pips Using these goals, I am able to project a yearly projected profit would be around 12%. Thus far, during this project, I have made a profit of $2835, which would yield a yearly profit of $11,340 so my achieved profit is very close to the profits estimated by my trading goal.

$$(15\text{ pips} * 0.75 - 15\text{ pip loss} * 0.25) * \frac{\$10/\text{pip}}{\text{standard lot}} * 3\ \frac{\text{standard lots}}{\text{trade}} * \frac{2\text{ trades}}{\text{week}} * 52 = \$11700/\text{year}$$

My trade strategy is outlined below:

1. Each day that I trade, I make sure that I am up-to-date with all fundamental issues that may affect the value of the currencies that I plan to invest; I read the Gartman Letter, Forex Factory, and also read the world news on Bing™ news, reading the most important world events as well as searching the currencies that I plan on trading, usually the US Dollar, the Euro, the Australian Dollar, and the Japanese Yen. The Gartman letter helps to provide me with some intuition on what may be influencing the chart trends. When there are very important news events that may cause higher-than-average market volatility, I avoid trading as my trading strategy is based upon technical rather than fundamental analysis.
2. The trading plan that I use is based upon a setup and a trigger. The setup is a particular set of conditions on the chart under which I may consider to make a trade on a given currency pair. The trigger is a particular event upon seeing which I execute the trade.
   a. When I first open MQL4, I look up the currencies that I am considering trading given the fundamental information that I have gathered. These pairs include EURUSD, AUDUSD, USDJPY, and CADUSD. The spread on these pairs is quite low, which is very important given that I generally plan to close my trade within 30 minutes.
   b. The setup conditions for a 'buy' is when a price falls to a support line and increases with two or three consecutive bars, the MACD is positive or it has a very high derivative and the simple moving average shows a positive trend. The setup conditions for a 'sell' is when a price increases to a resistance line two or three times, the MACD is negative or has a very high derivative, and the simple moving average shows a negative trend. If it moves in the opposite direction for a bar or two, I also look at the Parabolic SAR. Other specific patterns that I try to identify are the double top and the double bottom patterns which signal that the price may fall through or rise above the 'base' value.

c. Before trading, I look at the plots of the prices as well as support and resistance lines over using the 1-minute chart, the 5-minute chart, the 15-minute chart and the 30-minute chart for the last day and see important support and resistance points and try to identify support and resistance lines as well as the general overarching trend. I will also determine whether the currency pair follows trends or goes against the trend. If there is no coherent trend that I can follow, I avoid trading; if the prices generally seem to be trend following, then I consider the trade further.

d. When I identify the setup that is good for trading, I then look for the trigger at which I decide to actually execute the trade. The trigger is usually a crossover between the 9-period and 21-period exponential moving average identified by the MACD indicator, the double-top or double-bottom pattern, or a very strong trend that I determine will continue to follow and will generate the target profit of 15 pips.

e. The exit strategy is based on support-and-resistance lines as well as stop-losses and take-profits. When we are buying and we hit a resistance line that is shown by our support-and-resistance indicator and we also notice that the price movement is slowing or reversing, we exit the trade. Similarly, if we are selling and we hit a support line that is strong and the price movement slows or reverses, we exit the trade. If there is consolidation for an extended period of time and very limited price movement occurs, we try to make sufficient pips to overcome the loss incurred by the spread and then close the trade. This is because under periods of consolidation, we are highly unsure about whether the support or resistance will eventually prevail. Finally, if a stop-loss of 15 pips is incurred, then the trade is closed. I do not explicitly specify take-profit although if I lose around 30% or more of the profit on a trade, I close the trade. This exit strategy works well to maximize gains and minimize losses.

When I began trading, I started with lot sizes of one or two lots. However, towards the end of the project, I started buying or selling three standard lots of any given currency pair. The larger lot sizes enable me to focus more on identifying optimal setup conditions rather than requiring me to make many more trades in order to achieve my profit goals. However, if fewer than three of the setup conditions have been met, I plan to trade with two lots, though this has not yet happened since I simply do not trade under such conditions. If a trend is extremely profitable, meaning that it returns more than 25 pips of profit and if I can see the trend continuing, then I may add two more lots of the same pair, though under no circumstances do I plan to trade more than 5 standard lots. Because my trades are very short-term and I plan to exit within 30 minutes, I also do not trade multiple currency pairs at the same time, allowing me to devote my entire undivided attention to the current pair I am trading.

The results of my trading strategy seem to be quite solid; as a result of continued refinement of my plan, I eventually increased my stop-loss from 10 pips to 15 pips so I can catch some of the larger price movements in the chart. This strategy worked quite well, helping me achieve a profit of 47.3 pips, which over 3 lots is well over $1000 and can offset several negative trades at the stop loss. Trailing stop losses and buy-limits and sell-limits are also an important addition to my plan that I plan to implement if I continue trading at the professional level. A limit order is an order to buy a security at no more than a specific price, or to sell a security at no less than a specific price; this can be useful if you are anticipating the currency to rise above or fall below a certain value but do not wish to stay on the terminal waiting for the conditions to occur. The disadvantage is that it is possible that these conditions may never occur, but if they do, the profits may be very high. When these are combined with stop losses, trailing stop, or variable stops, we have higher flexibility while still enforcing robust risk management.

Below we have included a sample trade where I executed my trading plan, buying 3 standard lots of AUDUSD on 2011/11/17 at 05:35 for 1.00898 and selling on 2011/11/17 at 05:41 for 1.01073, achieving a profit of 17.5 pips or $525.

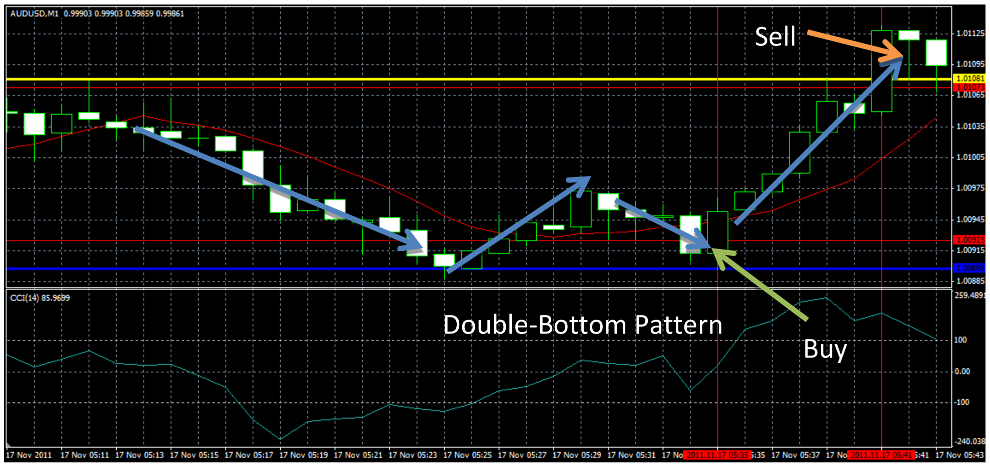


Figure 26 – Srinivas's Second Trade

The decision to buy was based on the MACD, support and resistance lines, and the simple moving average. The setup of the buy was a double-bottom pattern going downward with the price trending back upward. We also noticed the price hitting a distinct support line. We executed the market order when we saw that the price moved upwards for two consecutive bars and that a MACD crossing between the 9-period and 25-period EMA occurred. The 9-period simple moving average, denoted by the red line in the chart, also moved upward. We bought 3 standard lots with a stop loss of 10 pips. The trade made a profit of 17.5 pips or $525 dollars, which was quite good. We ultimately decided to sell noting a resistance line at the top. The price continued to increase further upward even after the resistance line although it later rebounded; nonetheless, we made a good profit and there was no way to be sure that the price would not rebound downwards strongly.

3.1.4 Trade Strategy for Samuel Veilleux

Over the last term, my trade strategy has evolved to only make trades which follow support and resistance lines in the market. In their purest form, these 'lines' represent prices at which currency pairs tend to reverse their movement. These can be caused by (among other things) a large financial institution's decision that a wise place to 'buy' sits at fixed (low) value, and they do so every time the currency pair reaches it. Resultantly, when the exchange rate falls to that value, the available liquidity at that price is exhausted by the large buying power of the financial institution. The price accordingly moves back upward. This may happen several times at the same price. A vigilant trader who notices that pattern of upward trend after the specific low value is reached can enter a long position the next time it happens, and keep the trade active while the trend continues.

Of course, this pattern cannot continue indefinitely, as even the funds of a large financial institution are not unlimited. Resultantly, a trade strategy which yielded profit on several iterations of the same support line will eventually fail. This 'failure' is due to a breakout – which could have been caused by any number of reasons, and is usually indicatory that the reversal line will no longer be useful in predicting trend reversals.

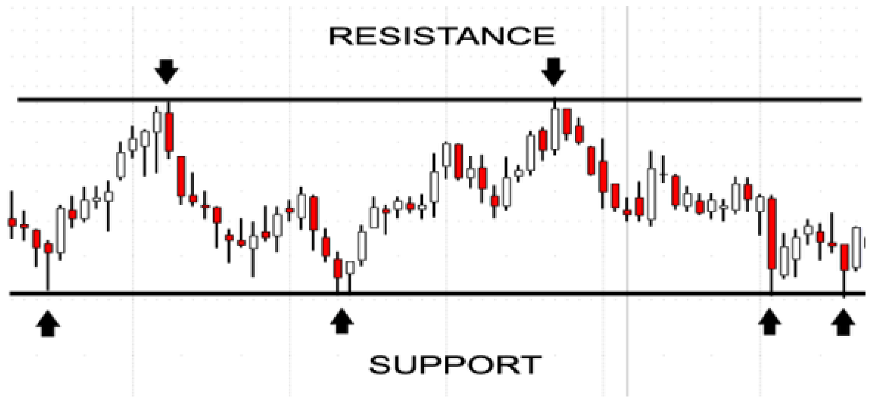


Figure 27: Support and Resistance Lines [54]

Sometimes more than one line can be observed in a chart, in the form of an upper and lower boundary. A support line is one which 'supports' the price of a currency pair from the low – creating local minima in the chart. Conversely, a resistance line 'resists' upward movement, and will lead to local maxima in the chart. These boundaries can be formed for the same or different reasons, and one may be ‘stronger’ than the other. The absolute strength of a support or resistance line is representative of the likeliness of a currency pair to ‘bounce’ off from it, as a result of the presence of people waiting to buy (or sell) at that line.

Unfortunately, the absolute strength of any line cannot be determined without insider trading knowledge! Thus, the apparent strength of a particular line correlated to how many times a currency pair has already ‘bounced’ off from it. Further, the apparent strength of a line is also correlated to how tightly the ‘bounces’ fall to the extreme-point. For instance, at a line of psychological importance (such as 1.00000 in USDCAD or 1.30000 in EURUSD) may see bounces executed by programs, causing the price to hit exactly 1.00000 or fall within a few tenths of a pip from there. Conversely, a weaker support line may look just as strong from a distant zoom level, but close examination may show that the variance among the local minima is as great as 5 pip.

A historically strong support line, once broken, can become a resistance line. This occurs because during the time that the line was acting as support, sellers waiting for the price to fall BELOW the line had to wait. Thus, after the breakthrough, sellers who had been waiting can now sell. Some lines of support and resistance can be followed tightly for a few days, then broken cleanly (by a high-volume purchase which exceeds the absolute strength of the line during that time), only to return a few days later and be obeyed again. Anomalies such as these can obscure a trader’s judgment of the strength of a given line, and can also fool robots programmed to identify these lines.

The methods which I have adopted to govern the way I make trades based off of support and resistance make up my trade strategy. There are several modules to this strategy, some of which can be used independently of the others. I first lay down some rules for any and all market-entrances. Following that I outline different strategies for making trades based on support and resistance lines, including break-outs from boxes and reversal prediction.

Firstly, no matter which strategy I am using, I always wait for the price movement to indicate to me that my prediction is being followed. Using the example of a strong support line I expect to continue being obeyed: if I watch the price fall down to it, I do not enter a long position immediately. It is my strategy to wait until I can see the price beginning to reverse – rather than entering too quickly only to find that it continues falling. Of course this method is not fool-proof, and will not lead to %100 profitable trades. One way to error-check this method is to look back at other times that the line in question has been met, and observe the noisy signal on a narrow candle-width chart as it turns around. This gives me a rough idea of what to expect a similar turn-around to look like.

Similarly, if I'm planning to enter a trade as soon as a notable line is broken, I will not do so until the break-out has at least exceeded the noise of the previous reversals. Often I will include a window of confidence for these decisions – a vertical 'distance' I expect the price to move by after meeting my set-up conditions to give me assurance that my prediction is coming to fruition. However, even with well tested care-measures in that respect, the market cannot be predicted. A large price movement can happen at any time as the result of some event somewhere in the world, and if this movement is not in my favor it could bring a hard hit to my account. Accordingly, I include another safety measure (namely the placement of a stop-loss) in all my trades. The value of my stop-loss varies with which method I am using, as is outlines below.

The simplest application of support and resistance lines is the identification of a single line which has been obeyed *n*or more times (for my personal trading I've found *n*=3 sufficient), and trade on the next time it is obeyed.   To do this, as outlined above, I look at the previous bounces and get an idea of how much noise to expect during the bounce, and look at the range which contains the absolute extrema. Considering the example of a support line (characterized by similar local minima in recent history), when the minima are very tightly grouped (i.e. within one pip) this method can be implemented by way of a simple buy-limit. Although my trade strategy dictates the requirement that I see the trend following my prediction before I enter the market, earlier (more simplistic) spins proved this to be a relatively safe way to earn some pips.

In the example of a strong support line with very tightly grouped minima, a breakthrough will come with the very quick warning of the price moving through the line identified by the previous reversal points. If every preceding reversal fell within 1 pip of a certain line, any slight breech in this line quickly raises a flag to the trader. Because that 'breech' can be identified after a two-pip drop below the support line, a stop-loss can be placed extremely close to the buy price – because noise is not expected to cause the price to fall that low during the reversal. Given that the maximum loss incurred by placing a buy-limit at the support line with a 4 pip stop-loss is very small, it is reasonable to let the simple, built-in pending-order functions of my trading platform execute the trade.

My exit strategy for this method has changed a lot this term. Initially, it was my decision to place a take-profit at a level determined by looking at the previous 'bounces' from the support line, and estimating where I could expect the price to reach with reasonably certainty (under the assumption that the bounce-pattern would continue.) However, I noticed major flaws in this (simple) methodology, because I expected the trades to be completed – both entry and exit - without my supervision. Every 'bounce' is new and unique, and subject to a very large number of unpredictable factors. Sometimes I would see that the price moved up into a profitable region, but not far enough up, only to fall back down to the stop-loss. Other times, I would see that the trade was profitable, but that the take-profit level fell in the middle of a very obvious up-trend, and more money could have easily been made.

Accordingly, I implemented a sliding stop-loss. This helped catch the first failure, because after a reasonably movement into profitability the trade is guaranteed to yield profit (even if small, still far more welcome than the stop-loss lurking below). Similarly, a trailing stop-loss continues to stay below a winning trade as it goes farther and farther in the preferred direction, eliminating the dismay brought by the premature action of a take-profit. Choosing a value of this trailing stop-loss is not something I have quantified yet, but my most successful tries have resulted from looking at previous ‘bounces’ from the same reversal-line and getting an idea of what ranges would have encompassed the noise in the signal.

This method has the value of permitting the trader to recognize a pattern in any time-frame, and place the pending-order on his or her own time. Every aspect of this trade can be chosen and designed according to any specific risk-loss management plan, usually without any rush, long before the trade goes is made. It should be noted that most (if not all) platforms allow the user to create an expiration date for pending orders such as the buy-limit, which means that the order is deleted and therefore never placed if the condition is not met after a certain period of time. It should be further noted that, while some brokers have this time-period defaulted to infinity, others do not. This was one important lesson learned by my group when we found that a pending order expired and prevented the placement of what was to be a very profitable trade.

An example of a trade I made this term following this strategy is shown in annotated screenshots from ForeX Capital Markets (FXCM), below.

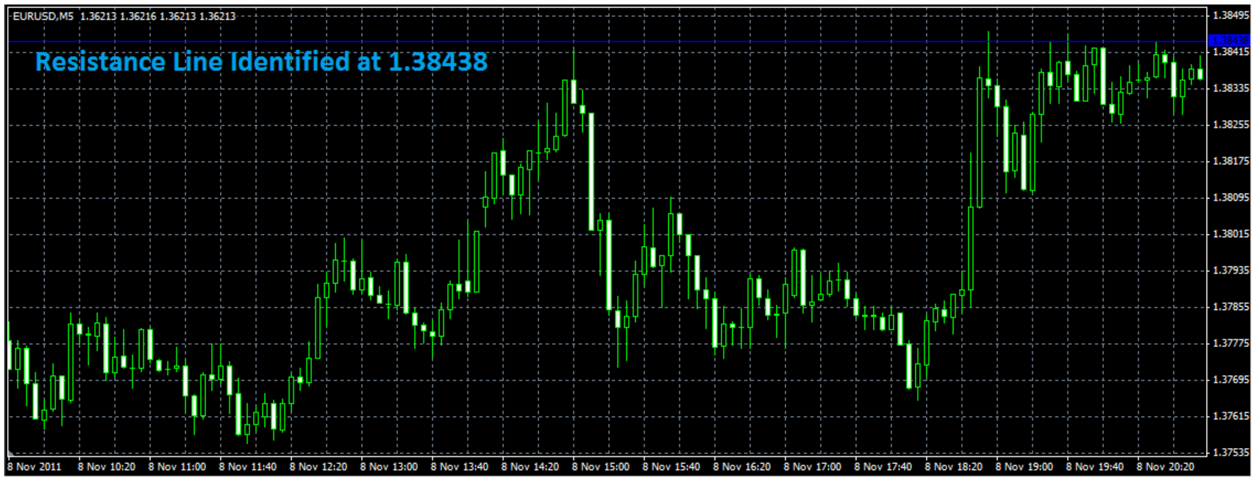


Figure 28: Setup for a Trade Based on One Resistance Line

As shown in the figure above, a resistance line was identified at 1.38438 (top, blue). Because it was tested 5 times and stayed strong, I decided to set a sell-limit and to graphically determine a stop-loss and a take-profit based on previous market activity. As outlined in my trade strategy, I configured every aspect of this trade long before it actually initiated. The next figure shows the execution of this trade on a tighter zoom-level.

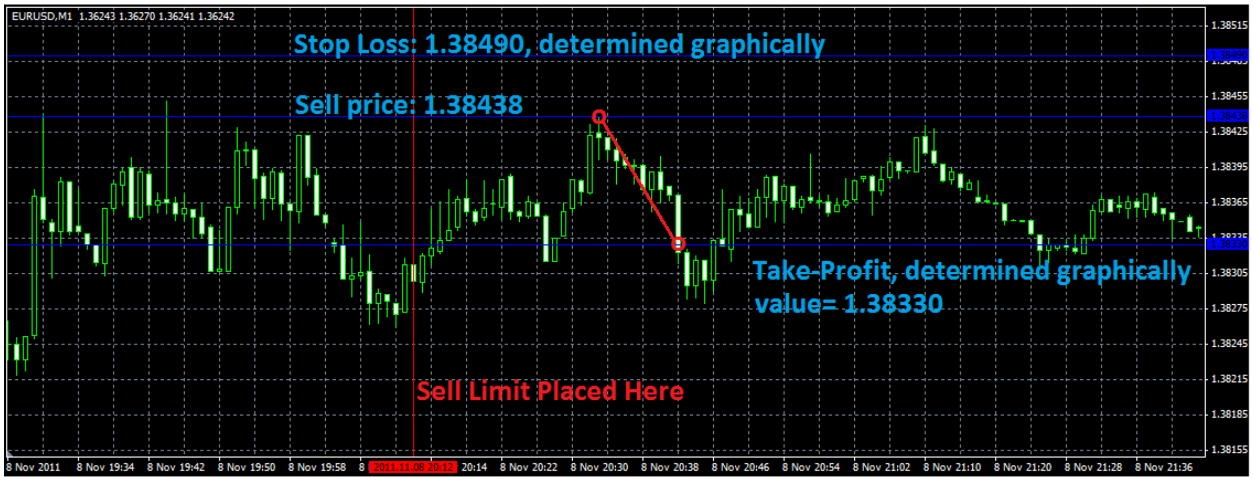


Figure 29: Execution of a Trade Based on One Resistance Line

The sell-limit was met 26 minutes after the placement of the pending-order, while I was in class. After the sale was made, the price rebounded as expected from the resistance line. It met the take-profit line after 8 minutes. The profit from this sale (which was in the volume of two standard lots) was $214.00 (10.7 pip). It can be seen on the chart that the rebound continued past my take-profit (lowest blue line), but not by much. In this particular case, the graphically determined take-profit demonstrated its value – however in my other trades I have found it to be an overall unreliable method, as discussed on the previous page.

An addition to my support-and-resistance trade strategy was developed to implement a ‘box’ of buy and sell stops around the critical bounce levels (both above and below, around sideways market movement). It operates in anticipation that when the limits are broken (and exceeded) by a certain specific amount, the market will continue moving in that direction. It is my stipulation that this method only be implemented if and only if both lines are identified and tested at least twice each. Buy-stop and sell-stop levels are selected at 25% outside of the support-resistance range. **Figure 30** shows the equations which govern the calculation of these values, along with the stop-loss levels.

$$\begin{aligned}&S = \text{Support Price}\\&R = \text{Resistance Price}\\&BuyStop = R + \tfrac{R-S}{4}\\&SellStop = S - \tfrac{R-S}{4}\\&StopLoss_{Initial} = \tfrac{S+R}{2}\\&StopLoss_{Trailing} = \tfrac{R-S}{2}\end{aligned}$$

Figure 30: Choosing stop-levels and exit strategy for market moves based on support and resistance levels

The constants *2* and *4* in these equations were selected graphically when I first decided to outline this method, and have proven profitable over the little hand-testing which I have implemented. The math which governs this strategy is simple. Sincethis strategy dictates the creation of pending orders, rather than a split-second decision and market-entry, its implementation can be achieved by hand or by a program. The only difficult part doing so is determining the values of support and resistance lines effectively. Doing this by hand can be tedious and requires the user to check the market frequently. However, a program which can effectively determine these levels would be very complex and does not yet exist (or at least, is not freely available to the public). One project by our group this semester was the creation of one such program. Its usefulness varied over different candle-widths and the program did not convince its creators that it could be a substitute for by-hand analysis.

A trade exemplifying this trade-strategy can be found on the following page. Our group's Support and Resistance Plotter analyzed the AUDUSD and identified a resistance line at 1.00385 (yellow) and support at the psychologically important 1.00000 (blue). Buy-stop and sell-stop levels are identified by dotted green lines.

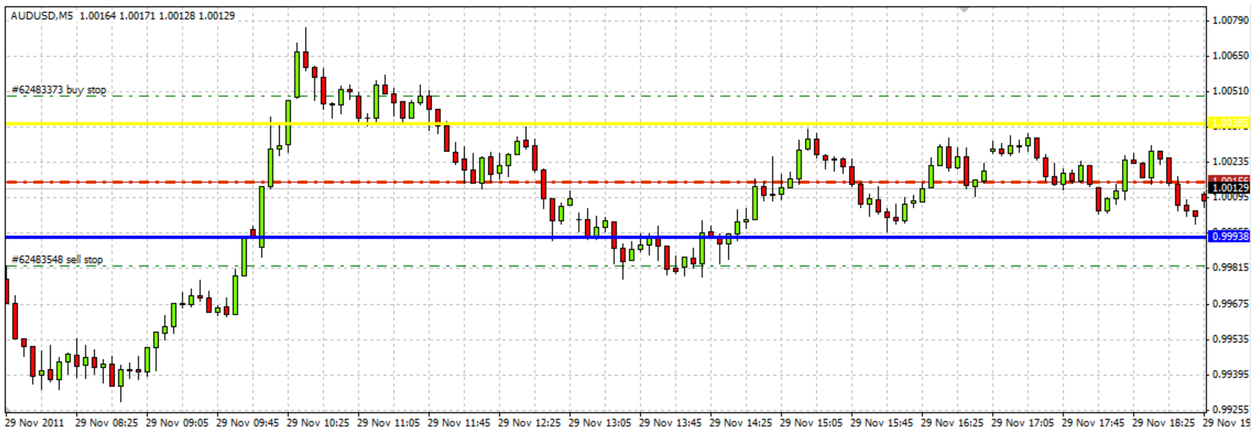


Figure 31: Setup for a Trade Based on a One Support and One Resistance Line

The stop-loss entered for the pending order was set to half way between two levels calculated using my updated trade strategy. The trades were then selected to have a trailing stop after the market order was made, whose value was equal to half the range between the support and resistance.

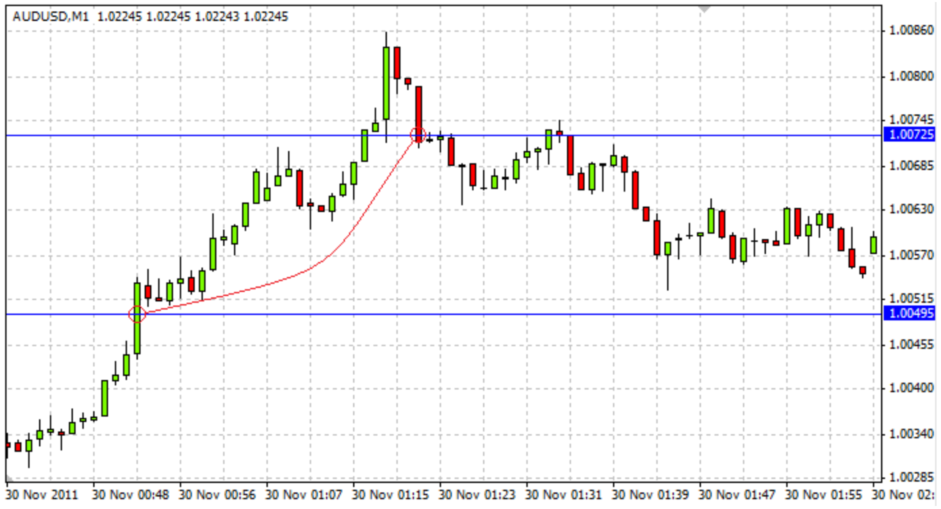


Figure 32: Execution of a Trade Based on a One Support and One Resistance Line

When the resistance line was broken, it was soon followed by the breaking of my buy-stop line. This event is marked in the figure above by the first red circle over the blue horizontal line. My trailing stop of 11.2 pips caused the trade to close at 1.00725 – a profit of 22.9 pips. Because this was a trade I made for my account while also in control of the group account, the lot-size was 4 standard lots. The profit was therefore $916.

3.2 Business Plan

We are collectively managing $800,000.00 USD for Empowerment Capital, LLC; as this is a very large sum, we work hard to make thoughtful trades that align with our risk management strategy and profit goals. We employ the highest ethical standards and leverage our technical skills and discipline to attract and satisfy our clients.

3.2.1 Profit Allocation and Customer Relations

- Overall goal of generating 11% profit per year
- Use stop losses to avoid large losses, preserving investor confidence
- Make reasonable profits for Empowerment Capital by charging the clients a 2% management fee per annum, and a 20% performance fee per month
- Empower society by donating 5% of our proceeds to charity
- Provide clients with the other 75% of our profit
- Provide investors with weekly performance reports on their investment, detailing all the specific trades as well as the overall profit
- Give customers security by facilitating transparency in our operation, allowing money to be withdrawn in part or in full at any time

3.2.2 Trade Logistics and Money/Risk Management for our Group

Our account will be leveraged 50:1, and our average trade will be for 2% of the total account capitol. Additionally, at any time there will be at most four trades open. Thereby, we never sink more than 8% of our total assets into investments. With the average trade being 2% of our initial investment, trades in our personal accounts will be in the volume of one standard lot. In the group account, individual trade size is 4 standard lots. Thus, to reach our group's goal of 11% annual yield, we need only to average (with a 250 day trading year estimate) 0.044% per day.

$$\frac{11\%}{year} * \frac{1\ year}{250\ trading\ days} = \frac{0.044\%}{trading\ day}$$

Trading 2% of the account at a time, with a leverage of 50:1, our group's account should show a profit

$$\frac{0.044\%}{trading\ day} * \frac{1}{0.02\ invested} * \frac{1}{50\ leverage} = \frac{4.4 * 10^{-4}}{trading\ day} = \frac{4.4\ pip}{trading}$$

It should be noted that the calculation above does not include the effects of compounding, as it assumes that the trade size will be 2% of the initial capital through the entire year of trading.

If any day's trading activity nets more than three losses, our day's trades will cease. We will stay up-to-date on fundamental developments in the world's economy, and how they are likely to affect the foreign-exchange market. To do this we will read regularly The Gartman Letters, and keep updated on some of the forex-news sites which we have come to trust. Although fundamental analysis is important, particularly in terms of the impact that news can have on the immediate future of the market, the force which drives our group's trades is that of technical analysis.

We typically use three different technical indicators to analyze the market. EMA's and the MACD are used heavily, as well as support and resistance level identification. We never make a trade without a stop-loss. The two currency pairs which we trade most frequently are EURUSD and AUDUSD mostly because of their high refresh-rate and low spreads. Specific details of our group members' individual trading strategies can be found in **Section 3.1** of this document.

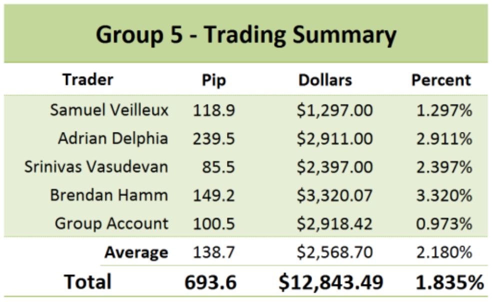
Over the past two terms we have built strong familiarity with the use of the MQL4 programming language in MT4. Happy with the processing power and response-time of our PCs' trade abilities in comparison to our own, we intend to continue improving our abilities with the language and adding functionalities to the programs we have already written. Details of programming-related projects completed by our group can be found in **Appendix B**.

Chapter 4: Trade Portfolio

4.1 Overall Performance Summary

Each team-member's personal account of $100,000 USD, and the shared group account of $300,000 USD, has been active since November 1st 2011. All trades were been documented each week for record keeping and progress presentations. A summary of all account activity can be found in the table below. The overall return on the virtual $700,000 investment was 1.835% ($12,843).

Table 1: Overall Trade Summary for Group 5



Group 5 - Trading Summary

Trader	Pip	Dollars	Percent
Samuel Veilleux	118.9	$1,297.00	1.297%
Adrian Delphia	239.5	$2,911.00	2.911%
Srinivas Vasudevan	85.5	$2,397.00	2.397%
Brendan Hamm	149.2	$3,320.07	3.320%
Group Account	100.5	$2,918.42	0.973%
Average	138.7	$2,568.70	2.180%
Total	**693.6**	**$12,843.49**	**1.835%**

A performance chart showing every trade made by the group, including both individual trades and trades made with the group account, is shown below. The y-axis is normalized over the $700,000 of total capital available to these accounts. The slope of the linear trend-line added to the plot indicates that the progression of the group capitol averaged 0.0145% per day, which can be extrapolated to 5.296% per year. While this investment return is less than our group's goal, it is still respectable and it is our belief that it is sustainable. The Sharpe Ratio for this performance is 24.75.

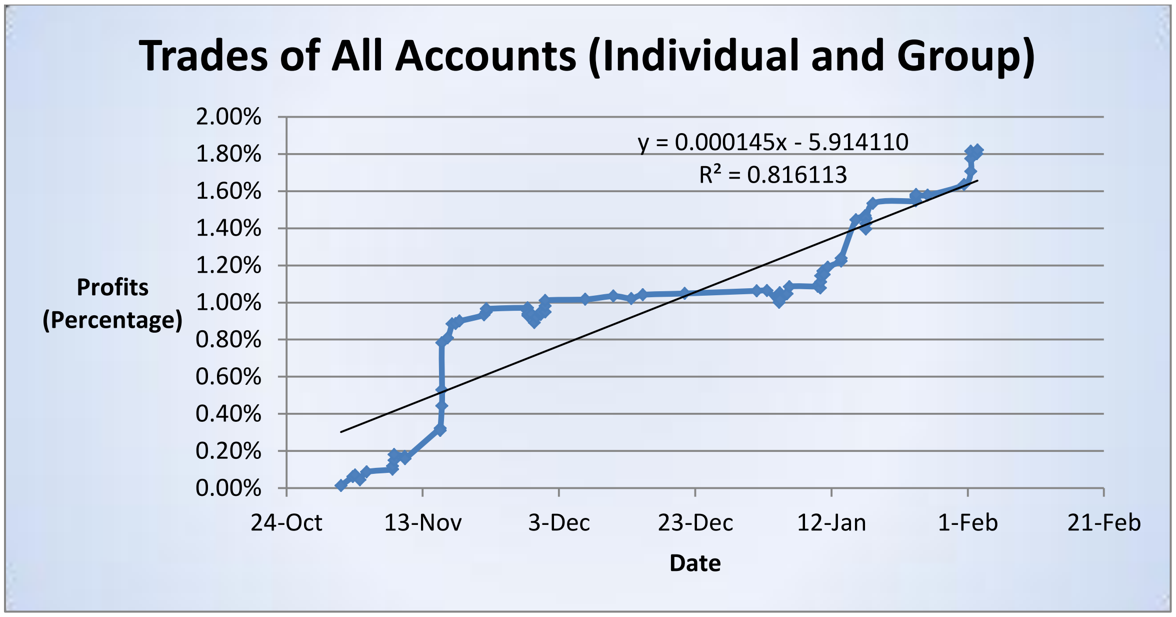


Figure 33: Performance Chart for All Accounts

4.2 The Group Account

Trades made in the group account totaled a net profit of $2,918.42 USD (0.973% of initial balance). In terms of basis points, 100.5 pips were gained in account's favor. The average trade size was 2.17 standard lots. Trades in the group account were usually made when all group members could be present, and a trade could be agreed upon before market-entry. Occasionally, when scheduling such a meeting was too difficult, control of the group account was transferred to one group member who was given the authority to make trades in the absence of the others. Resultantly, trades in the group account sometimes mirror trades of individual members who made trades in both accounts simultaneously. Trades of the group account are outlined in full below.

Table 2: Group Account Trades

The Trades of The Group Account

Trade	Open Time	Type	Size	Item	Price	SL	TP	Close Time	Price	Profit (PIP)	Profit ($USD)
1	11/2/11 17:47	buy	3	AUDUSD	1.03283	1.02900	NA	11/2/11 18:38	1.03394	11.1	$333.00
2	11/3/11 19:31	buy	3	EURUSD	1.37528	1.37428	1.37628	11/3/11 21:20	1.37469	(5.9)	($177.00)
3	11/4/11 18:13	buy	3	AUDUSD	1.03910	Trailing	1.04244	11/4/11 18:24	1.04009	9.9	$297.00
4	11/15/11 20:06	sell	3	AUDUSD	1.01990	1.02090	1.01395	11/16/11 0:14	1.01395	59.5	$1,785.00
5	11/22/11 8:46	sell	1	EURUSD	1.35369	Trailing	1.34367	11/22/11 9:20	1.35253	11.6	$116.00
6	1/4/12 11:07	sell	1	EURUSD	1.29141	1.29262	1.29020	1/4/12 11:15	1.29263	(12.2)	($122.00)
7	1/10/12 9:55	sell	2	EURAUD	1.23788	1.23850	1.23680	1/10/12 10:28	1.23678	11.0	$227.42
8	1/9/12 23:45	sell	1	EURUSD	1.27749	1.27799	1.00000	1/3/11 0:03	1.27743	5.0	$50.00
12	1/16/12 22:55	sell	2	EURUSD	1.27288	1.27375	1.27230	1/17/12 0:23	1.27250	3.8	$76.00
13	1/17/12 1:07	sell	2	AUDUSD	1.03918	1.04018	1.03775	1/17/12 1:07	1.04018	(10.0)	($200.00)
14	1/24/12 10:33	sell	1	EURUSD	1.30283	Trailing	NA	1/24/12 10:33	1.30238	4.5	$45.00
15	2/1/12 11:02	sell	4	EURUSD	1.32053	manual	1.31853	2/1/12 11:22	1.31931	12.2	$488.00
Average			2.17							8.4	$243.20
Total										**100.5**	**$2,918.42**

The figure below shows a performance chart outlining the balance of this account over time, with its y-axis normalized as percentages of the $300,000 initial balance.

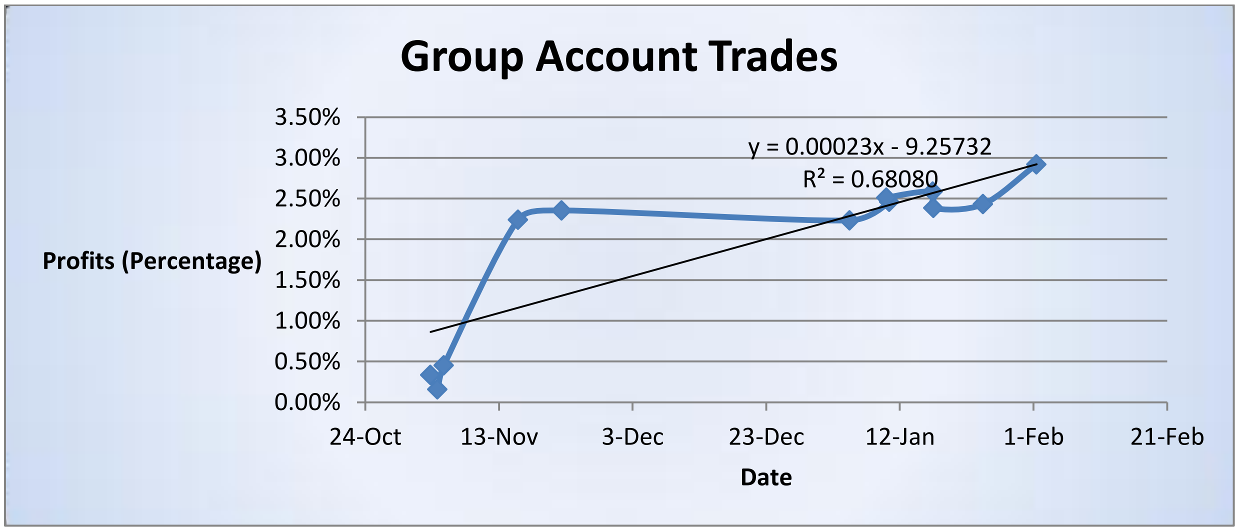


Figure 34: Performance Chart for the Group Account

4.3 Adrian Delphia's Account

Trades made in the Adrian Delphia's account totaled a net profit of $2,911.00 USD (2.911% of initial balance). An unabridged list of trades since November 1st 2011 is shown in the table below.

Table 3: Adrian Delphia's Trades

The Trades of Adrian Delphia											
Trade	Open Time	Type	Size	Item	Price	SL	TP	Close Time	Price	Profit (PIP)	Profit ($USD)
1	11/3/11 2:15	sell	1	AUDUSD	1.02455	1.02585	1.02255	11/3/11 2:30	1.02400	5.5	$55.00
2	11/8/11 14:15	buy	1	EURUSD	1.38252	1.38100	1.38300	11/8/11 14:30	1.38352	10.0	$100.00
3	11/8/11 14:22	buy	1	EURUSD	1.38224	1.38100	1.38300	11/8/11 14:30	1.38352	12.8	$128.00
4	11/10/11 10:15	sell	2	EURUSD	1.35610	Trailing	1.35550	11/10/11 10:21	1.35620	(1.0)	($20.00)
5	11/15/11 13:56	sell	1	EURUSD	1.01991	1.02009	NA	11/15/11 23:30	1.00843	114.8	$1,148.00
6	11/15/11 19:36	sell	2	AUDUSD	1.02150	Trailing	1.01395	11/16/11 0:14	1.01685	46.5	$930.00
7	11/29/11 0:28	sell	1	AUDUSD	0.99702	0.99800	0.99600	11/29/11 1:07	0.99721	(1.9)	($19.00)
8	11/30/11 8:02	buy	1	EURUSD	1.34499	1.34399	1.34599	11/30/11 8:10	1.34599	10.0	$100.00
9	12/6/11 21:02	buy	1	AUDUSD	1.02548	Trailing	NA	12/6/11 22:10	1.02588	4.0	$40.00
10	12/11/11 0:02	sell	2	AUDUSD	1.01065	Trailing	NA	12/11/11 8:10	1.01004	6.1	$122.00
11	12/15/11 8:02	sell	1	AUDUSD	1.00847	Trailing	1.00727	12/15/11 8:10	1.00707	14.0	$140.00
12	1/24/12 8:46	sell	1	EURUSD	1.30225	Trailing	NA	1/24/12 9:14	1.30134	9.1	$91.00
13	1/24/12 10:01	sell	1	EURUSD	1.30282	Trailing	NA	1/24/12 10:23	1.30133	14.9	$149.00
14	1/24/12 10:33	sell	1	EURUSD	1.30283	Trailing	NA	1/24/12 10:33	1.30238	4.5	$45.00
15	2/2/12 9:01	sell	1	EURUSD	1.30772	Trailing	NA	2/2/12 9:02	1.30870	(9.8)	($98.00)
Average			1.20							16.0	$194.07
Total										239.5	**$2,911.00**

A performance chart indicating the profits in Adrian’s account over time is shown in the figure below. The trend line on the plot indicates that profits averaged 2.8%% per day.

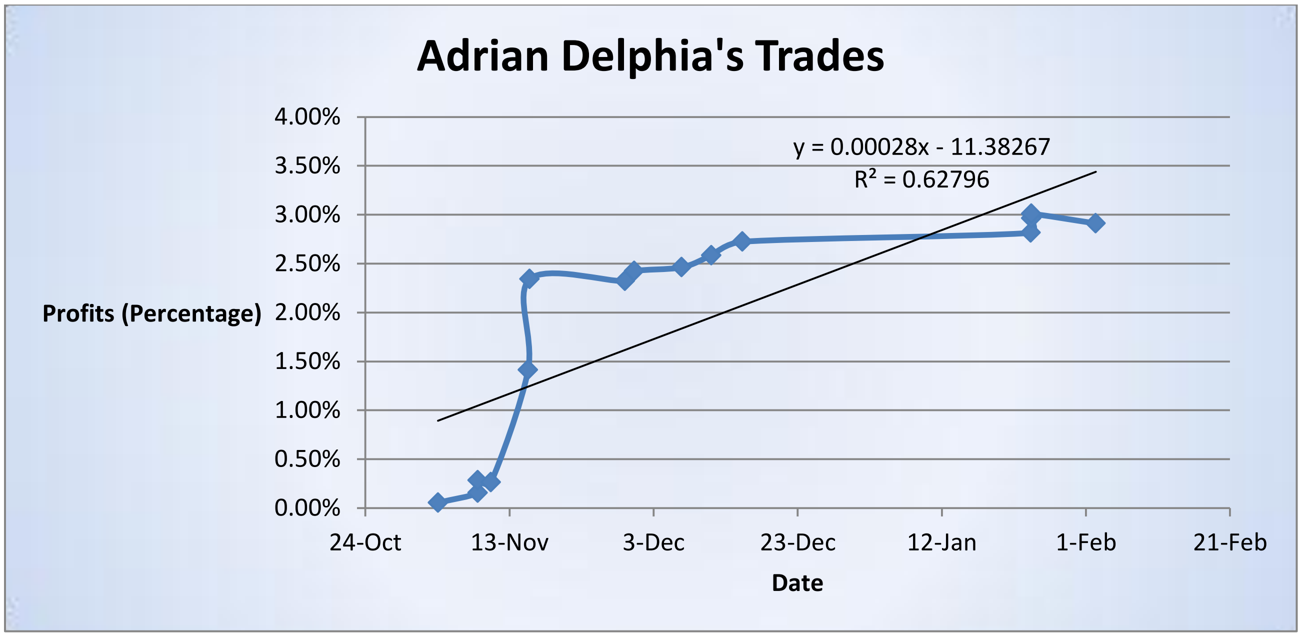
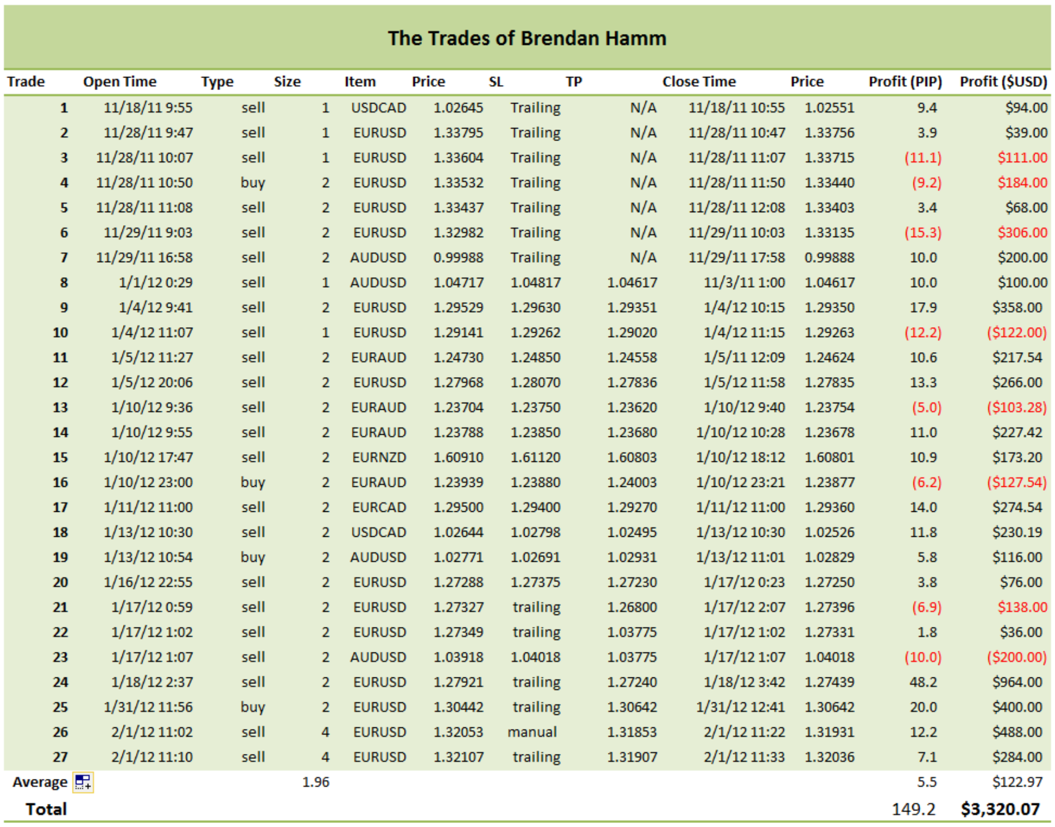


Figure 35: Performance Chart for Adrian Delphia

4.4 Brendan Hamm’s Account

Trades made in the Brendan Hamm’s account totaled a net profit of $3,320.07 USD (3.320% of initial balance). An unabridged list of trades since November 1st 2011 is shown in the table below.

Table 4: Brendan Hamm's Trades



The Trades of Brendan Hamm

Trade	Open Time	Type	Size	Item	Price	SL	TP	Close Time	Price	Profit (PIP)	Profit ($USD)
1	11/18/11 9:55	sell	1	USDCAD	1.02645	Trailing	N/A	11/18/11 10:55	1.02551	9.4	$94.00
2	11/28/11 9:47	sell	1	EURUSD	1.33795	Trailing	N/A	11/28/11 10:47	1.33756	3.9	$39.00
3	11/28/11 10:07	sell	1	EURUSD	1.33604	Trailing	N/A	11/28/11 11:07	1.33715	(11.1)	$111.00
4	11/28/11 10:50	buy	2	EURUSD	1.33532	Trailing	N/A	11/28/11 11:50	1.33440	(9.2)	$184.00
5	11/28/11 11:08	sell	2	EURUSD	1.33437	Trailing	N/A	11/28/11 12:08	1.33403	3.4	$68.00
6	11/29/11 9:03	sell	2	EURUSD	1.32982	Trailing	N/A	11/29/11 10:03	1.33135	(15.3)	$306.00
7	11/29/11 16:58	sell	2	AUDUSD	0.99988	Trailing	N/A	11/29/11 17:58	0.99888	10.0	$200.00
8	1/1/12 0:29	sell	1	AUDUSD	1.04717	1.04817	1.04617	11/3/11 1:00	1.04617	10.0	$100.00
9	1/4/12 9:41	sell	2	EURUSD	1.29529	1.29630	1.29351	1/4/12 10:15	1.29350	17.9	$358.00
10	1/4/12 11:07	sell	1	EURUSD	1.29141	1.29262	1.29020	1/4/12 11:15	1.29263	(12.2)	($122.00)
11	1/5/12 11:27	sell	2	EURAUD	1.24730	1.24850	1.24558	1/5/11 12:09	1.24624	10.6	$217.54
12	1/5/12 20:06	sell	2	EURUSD	1.27968	1.28070	1.27836	1/5/12 11:58	1.27835	13.3	$266.00
13	1/10/12 9:36	sell	2	EURAUD	1.23704	1.23750	1.23620	1/10/12 9:40	1.23754	(5.0)	($103.28)
14	1/10/12 9:55	sell	2	EURAUD	1.23788	1.23850	1.23680	1/10/12 10:28	1.23678	11.0	$227.42
15	1/10/12 17:47	sell	2	EURNZD	1.60910	1.61120	1.60803	1/10/12 18:12	1.60801	10.9	$173.20
16	1/10/12 23:00	buy	2	EURAUD	1.23939	1.23880	1.24003	1/10/12 23:21	1.23877	(6.2)	($127.54)
17	1/11/12 11:00	sell	2	EURCAD	1.29500	1.29400	1.29270	1/11/12 11:00	1.29360	14.0	$274.54
18	1/13/12 10:30	sell	2	USDCAD	1.02644	1.02798	1.02495	1/13/12 10:30	1.02526	11.8	$230.19
19	1/13/12 10:54	buy	2	AUDUSD	1.02771	1.02691	1.02931	1/13/12 11:01	1.02829	5.8	$116.00
20	1/16/12 22:55	sell	2	EURUSD	1.27288	1.27375	1.27230	1/17/12 0:23	1.27250	3.8	$76.00
21	1/17/12 0:59	sell	2	EURUSD	1.27327	trailing	1.26800	1/17/12 2:07	1.27396	(6.9)	$138.00
22	1/17/12 1:02	sell	2	EURUSD	1.27349	trailing	1.03775	1/17/12 1:02	1.27331	1.8	$36.00
23	1/17/12 1:07	sell	2	AUDUSD	1.03918	1.04018	1.03775	1/17/12 1:07	1.04018	(10.0)	($200.00)
24	1/18/12 2:37	sell	2	EURUSD	1.27921	trailing	1.27240	1/18/12 3:42	1.27439	48.2	$964.00
25	1/31/12 11:56	buy	2	EURUSD	1.30442	trailing	1.30642	1/31/12 12:41	1.30642	20.0	$400.00
26	2/1/12 11:02	sell	4	EURUSD	1.32053	manual	1.31853	2/1/12 11:22	1.31931	12.2	$488.00
27	2/1/12 11:10	sell	4	EURUSD	1.32107	trailing	1.31907	2/1/12 11:33	1.32036	7.1	$284.00
Average			1.96							5.5	$122.97
Total										149.2	**$3,320.07**

A performance chart indicating the profits in Brendan’s account over time is shown in the figure below. The trend line on the plot indicates that profits averaged 3.5%% per day.

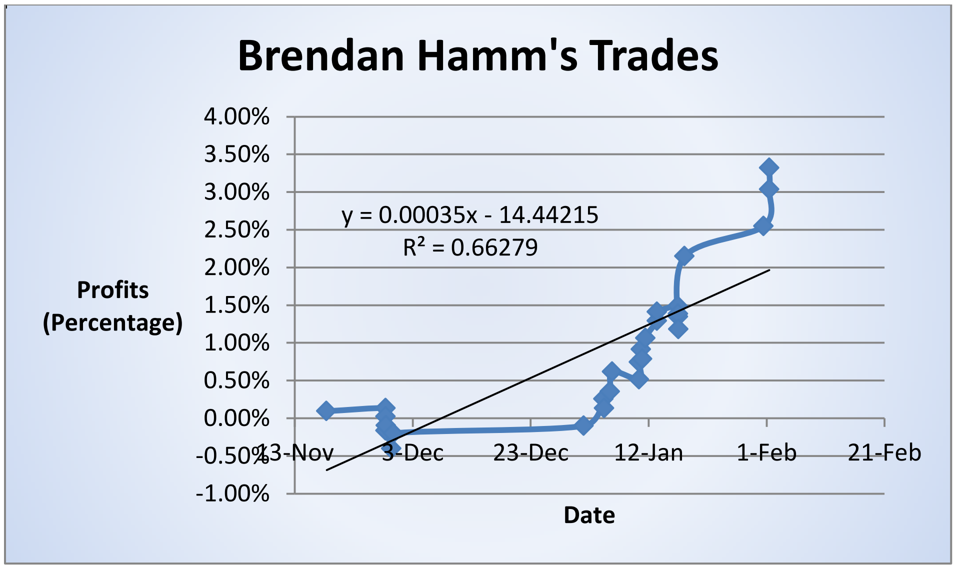


Figure 36: Brendan Hamm's Trades

4.5 Srinivas Vasudevan's Account

Trades made in the Srinivas Vasudevan's account totaled a net profit of $2,397.00 USD (2.397% of initial balance). An unabridged list of trades since November 1st 2011 is shown in the table below.

Table 5:Srinivas Vasudevan's Trades

The Trades of Srinivas Vasudevan											
Trade	Open Time	Type	Size	Item	Price	SL	TP	Close Time	Price	Profit (PIP)	Profit ($USD)
1	11/1/11 0:29	sell	1	AUDUSD	1.04717	1.04817	1.04617	11/3/11 1:00	1.04617	10.0	$100.00
2	11/8/11 20:12	sell	2	EURUSD	1.38438	1.38490	1.38330	11/8/11 20:38	1.38330	10.8	$216.00
3	11/10/11 8:51	sell	1	EURUSD	1.35650	1.35750	1.35550	11/10/11 8:53	1.35720	(7.0)	($70.00)
4	11/16/11 15:49	buy	3	AUDUSD	1.00841	1.00741	N/A	11/16/11 4:01	1.00900	5.9	$177.00
5	11/17/11 8:15	buy	3	AUDUSD	1.00898	1.00798	N/A	11/17/11 8:17	1.01073	17.5	$525.00
6	11/17/11 20:10	sell	3	EURUSD	1.34454	1.34354	N/A	11/17/11 20:15	1.34449	0.5	$15.00
7	11/30/11 23:24	sell	3	AUDUSD	1.02576	1.02676	N/A	11/30/11 23:35	1.02577	(0.1)	($3.00)
8	11/30/11 23:49	buy	3	AUDUSD	1.02531	1.02431	N/A	12/1/11 0:10	1.02607	7.6	$228.00
9	12/1/11 0:18	sell	3	EURUSD	1.34479	1.34579	N/A	12/1/11 0:39	1.34409	7.0	$210.00
10	1/4/12 8:06	buy	3	AUDUSD	1.30613	1.30463	N/A	1/4/12 8:52	1.30463	(15.0)	($450.00)
11	1/15/12 13:53	buy	3	EURUSD	1.27459	1.34579	N/A	1/15/12 14:29	1.27942	48.3	$1,449.00
Average			2.55							7.8	$217.91
Total										85.5	**$2,397.00**

A performance chart indicating the profits in Srinivas's account over time is shown in the figure below. The trend line on the plot indicates that profits averaged 2.3%% per day.

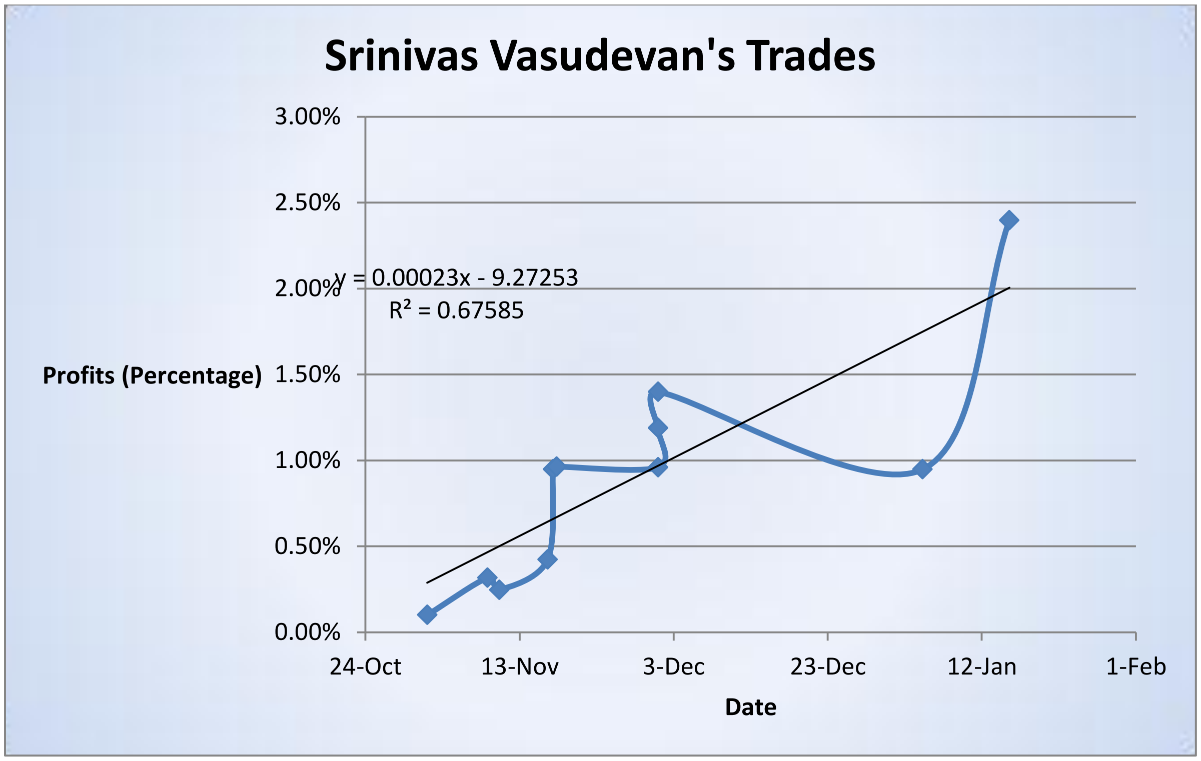


Figure 37: Srinivas Vasudevan's Trades

4.6 Samuel Veilleux's Account

Trades made in the Samuel Veilleux's account totaled a net profit of $1,297.00 USD (1.297% of initial balance). An unabridged list of trades since November 1st 2011 is shown in the table below.

Table 6: Samuel Veilleux's Trades

The Trades of Sam Veilleux

Trade	Open Time	Type	Size	Item	Price	SL	TP	Close Time	Price	Profit (PIP)	Profit ($USD)
1	11/8/11 20:12	sell	2	EURUSD	1.38438	1.38490	1.38330	11/8/11 20:38	1.38330	10.8	$216.00
2	11/10/11 8:51	sell	1	EURUSD	1.35650	1.35750	1.35550	11/10/11 8:53	1.35720	(7.0)	($70.00)
3	11/15/11 14:02	buy	1	AUDUSD	1.35459	1.35359	1.35659	11/15/11 14:04	1.35377	(8.2)	($82.00)
4	11/15/11 20:06	sell	1	AUDUSD	1.01990	1.02090	1.01395	11/16/11 0:14	1.01395	59.5	$595.00
5	11/22/11 8:46	sell	1	EURUSD	1.35369	Trailing	1.34367	11/22/11 9:20	1.35253	11.6	$116.00
6	11/22/11 0:52	buy	1	AUDUSD	1.00496	1.00645	2.00000	11/30/11 1:18	1.00725	22.9	$229.00
7	11/30/11 8:02	buy	1	EURUSD	1.34500	1.34402	2.00000	11/30/11 8:11	1.34599	9.9	$99.00
8	12/13/11 15:15	buy	1	EURUSD	1.31622	1.31532	2.00000	12/13/11 15:16	1.31532	(9.0)	($90.00)
9	12/21/11 11:00	sell	1	EURUSD	1.31015	1.31532	1.00000	12/21/11 11:02	1.30965	5.0	$50.00
10	1/2/12 12:50	sell	1	EURUSD	1.29364	1.29384	1.00000	1/2/12 12:52	1.29354	1.0	$10.00
11	1/9/12 23:45	sell	1	EURUSD	1.27749	1.27799	1.00000	1/3/11 0:03	1.27743	0.6	$6.00
12	1/26/12 2:59	buy	1	GBPUSD	1.56610	1.56590	0.00000	1/26/011 02:59	1.56590	(2.0)	($20.00)
13	2/2/12 10:33	sell	1	EURUSD	1.31897	1.31947	1.00000	1/3/11 0:03	1.31759	13.8	$138.00
Average			1.07							8.5	$92.64
Total										118.9	**$1,297.00**

A performance chart indicating the profits in Samuel's account over time is shown in the figure below.

The trend line on the plot indicates that profits averaged 1.0%% per day.

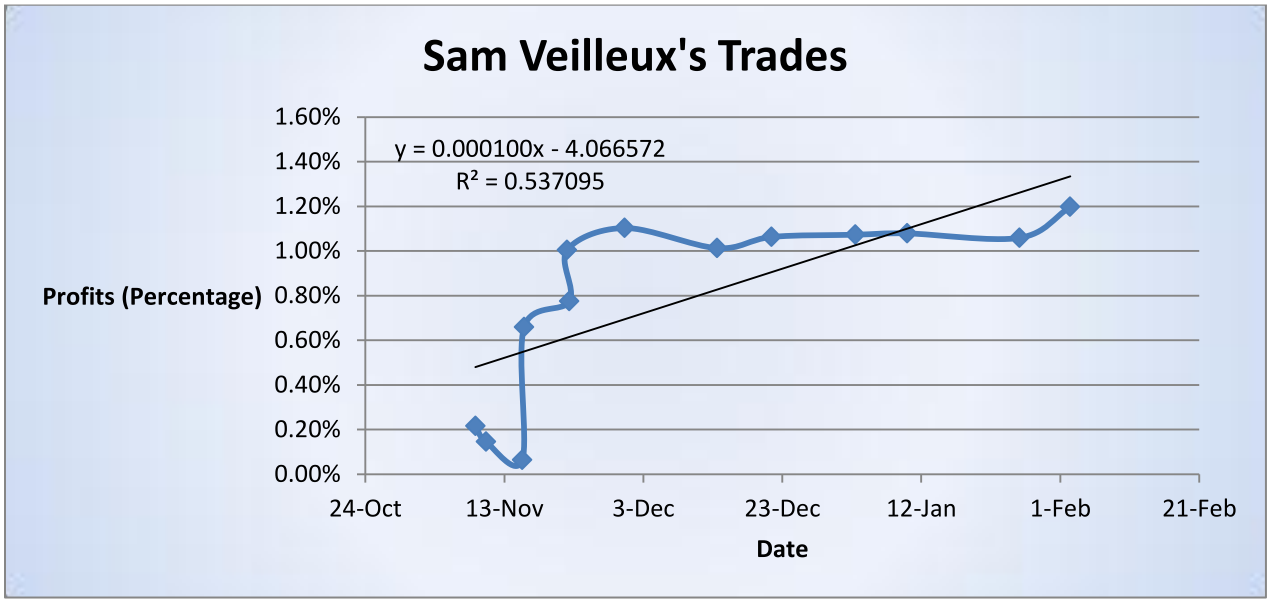


Figure 38: Samuel Veilleux's Trades

Chapter 5: Parameters for Launching Our Company

5.1 Company Structure

Our team plans to launch a foreign exchange money management firm Oakwood Investments. Oakwood Investments will be a hedge fund that will provide investors with an expected return of 20 percent per year. Our strategy will employ complex risk management strategies and will leverage the algorithmic and trading experience of our experienced team. The firm will be structured as a limited liability corporation. We plan to run effective marketing campaigns as well as recruiting wealthy investors during the next several years. Additionally, we plan to establish a legal plan to protect our company and its assets.

The LLC structure has many advantages, which make it desirable for our trading firm. We will have the liability protection of a corporation meaning that the partners in our corporation cannot be held personally liable for debts unless they have signed a personal guarantee. Furthermore, our leaders have the liability protection of a corporation. An LLC exists as a separate entity much like a corporation; members cannot be held personally liable for debts unless they have signed a personal guarantee. Unlike a partnership, we are free to select various forms of distributing profits and are not bound to distribute returns evenly. This way, our system can reward those who earn the highest returns, effectively creating an incentive plan to attract adept traders. We also plan to file articles of organization with the Secretary of State and hire a law firm to design an operating agreement.

The company will be funded by performance fees as well as management fees from investors. Oakwood Investments will charge investors a flat 2% management fee per year to cover the operating costs of the company as well as a flat performance fee of 20% for eligible accounts. We plan to employ several trading strategies based on statistical signal processing, advanced entry/exit triggers, and algorithm selection. Utilizing these several trading strategies we can offer our clients several different packages with varying risks and returns. In this manner we can attract investors with varying levels of risk management.

Our corporation will launch an extensive marketing campaign to seek private investments by venture capitalists and wealthy investors who can extend our name and brand recognition. We seek to manage several million dollars within the next few years, with intentions of expanding once we establish a positive reputation. We also have start-up costs and need to pay outside consultants, legal fees, and salaries to our employees; this means that for our first few years we plan on reinvesting the majority of our profits to help attain positive growth.

5.2 Findings Regarding Effective Marketing

In order to achieve and maintain profitability, it is imperative that we launch an effective marketing campaign. There are several restrictions on hedge fund advertising and marketing. Due to the broad mandates and relatively lenient registration and disclosure rules, hedge funds in the United States are only allowed to accept investments from accredited investors and institutions. Therefore, to gain visibility, we plan to use search engine optimization to ensure that our hedge fund is returned near the top of prominent search engines. One of the ways in which we can market the company is by speaking at conferences and events within the industry.

Developing a brand image for Oakwood investments will be integral in earning the trust of investors. We operate onshore and thus incur some regulation by the United States government. We also plan to use external consultants where appropriate. These consultants will be experts within raising capital within a specific channel, creating marketing materials and promoting a marketing message. We plan to select carefully for consultants who we can trust and who are effective at marketing hedge funds.

To start our company, we need seed capital. This capital will come from high net worth individuals who are familiar with the ability of our team as well as our wide and comprehensive portfolio management experience. Our excellent trading records and our principles of integrity should allow us to attract many serious investors. Additionally, to startup we will attract investments from family and friends whom are accredited investors, as well as faculty and staff at Worcester Polytechnic Institute and other companies with which we have affiliations.

Works Cited

1. Brian Dolan and GalantMark.Currency Trading for Dummies.John Wiley and Sons, 2011.

2. Bank for International Settlements. Foreign Exchange and derivatives market activity in April 2010-

Preliminary Results, 2010.

3. Logan, Tina. *Getting Started in Candlestick Charting*. Hoboken, NJ: John Wiley & Sons, 2008.

4. "Moving Average." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011.

<http://en.wikipedia.org/wiki/Moving_average>.

5."MACD." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011. <http://en.wikipedia.org/wiki/MACD>.

6."Commodity Channel Index (CCI) - ChartSchool - StockCharts.com." *StockCharts.com - Simply the Web's Best Financial Charts*.Web. 27 Oct. 2011.

<http://stockcharts.com/education/IndicatorAnalysis/indic_CCI.html>.

7."Bollinger Bands." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011.

<http://en.wikipedia.org/wiki/Bollinger_Bands>.

8."Stochastic Indicator." *Forex Indicators Guide*.Web. 27 Oct. 2011. <http://forex-indicators.net/stochastic>.

9."Commodity Channel Index (CCI) - ChartSchool - StockCharts.com." *StockCharts.com - Simply the Web's Best Financial Charts*.Web. 27 Oct. 2011.

<http://stockcharts.com/education/IndicatorAnalysis/indic_CCI.html>.

10."Average True Range." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011.

<http://en.wikipedia.org/wiki/Average_True_Range>.

11."Average Directional Index." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011.

<http://en.wikipedia.org/wiki/Average_Directional_Index>.

12. "Parabolic SAR." *Wikipedia, the Free Encyclopedia*. Web. 27 Oct. 2011.

<http://en.wikipedia.org/wiki/Parabolic_SAR>.

13. "Relative Strength Index (RSI) - ChartSchool - StockCharts.com." *StockCharts.com - Simply the Web's Best Financial Charts*.Web. 27 Oct. 2011. <http://stockcharts.com/education/IndicatorAnalysis/indic_RSI.html>.

14."Forex Fibonacci Indicator Explained | Forex Indicator Guide." *Best Forex Indicator - Forex Indicators - Forex Training*. Web. 27 Oct. 2011. <http://www.forexindicator.org/forex-fibonacci-indicator-explained.html>.

15. Governing Board of the Swiss National Bank. Swiss National Bank Quarterly Bulletin, 29(Q3).

16. Dennis Gartman. September 7th. The Gartman Letter, 2011.

17.“How to trade the Fibonacci retracement indicator.” Indicator Forex LLC.Web.20 Oct 2011 <http://www.indicatorforex.com/content/how-trade-fibonacci-retracement-indicator>.2011.

18. “Moving average convergence divergence.” MetaQuotes Software.Web.22 Oct. 2011<http://codebase.mql4.com/258>, 2000.

19. "Is Social Security a Ponzi Scheme? - Room for Debate - NYTimes.com."*The New York Times - Breaking News, World News & Multimedia*. 09 Sept. 2011. Web. 27 Oct. 2011. <http://www.nytimes.com/roomfordebate/2011/09/09/is-social-security-a-ponzi-scheme>.

20. “Parabolic stop and reverse.” MetaQuotes Software. Web.22 Oct. 2011 <http://ta.mql4.com/indicators/trends/parabolic_sar>, 2000.

21. Censky, Annalyn. "Federal Reserve Launches Operation Twist - Sep. 21, 2011." CNNMoney - Business, Financial and Personal Finance News. 21 Sept. 2011. Web. 27 Oct. 2011. <http://money.cnn.com/2011/09/21/news/economy/federal_reserve_operation_twist/index.htm>.

22. Michael A. S. Joyce, Ana Lasaosa, Ibrahim Stevens, and Matthew Tong. The _nan-cial market impact of quantitative easing in the u.k. International Journal of Central Banking, 7(3), page = 113-161), 2011.

23. The federal reserve system: Purposes and functions. http://www.federalreserve.gov/pf/pdf/pf 4.pdf, 2005.

24. Force index indicator. MetaQuotes Software Web. 27 Oct. 2011. <http://codebase.mql4.com/299>, 2000.

25. Forex Factory. Web. 13 Jan. 2012. <http://www.forexfactory.com/news.php?do=news>.

26. Map of euro area 1999-2011. http://www.ecb.int/euro/intro/html/map.en.html, 2011.

27. Dennis Gartman. September 9th. The Gartman Letter, 2011.

28. Dennis Gartman. September 16th. The Gartman Letter, 2011.

29. William L. Watts. German lawmakers approve bigger efsf. http://www.marketwatch.com/story/german-lawmakers-approve-bigger-efsf-2011-09-29, 2011.

30. Alexandra Hudson. ESFS could make germany liable for 465 billion euros: Ifo, September 2011.

31. "Crisis in Europe: Prepare for Repercussions from Standard & Poor's Credit Rating Downgrades - Money Morning." Investment News: Money Morning - Only the News You Can Profit From. Web. 05 Jan. 2012. <http://moneymorning.com/2012/01/13/crisis-in-europe-prepare-for-repercussions-from-standard-poors-credit-rating-downgrades/>.

32. *All About...The Foreign Exchange Market in the United States*. Federal Reserve Bank of New York, 1998.

33. John Williamson. The exchnage rate system.In*Policy Analyses in International Economics5*, 1983.

34. *Guide to Financial Markets*. Exmouth House, 4 edition, 2006.

35. U.S. foreign exchange intervention. http://www.ny.frb.org/aboutthefed/fedpoint/fed44.html, 2007.

36. Frederic S. Mishkin. Inflation targeting in emerging market countries. National Journal of Economic Research, (7618), March 2000.

37. Arturo Estrella and Frederic S. Mishkin.The yield curve as a predictor of US. recessions. *Current Issues in Economics and Finance*, 2(7).

38. Q.F. Akram and O. Eitrheim.Flexible inflation targeting and financial stability. Journal of Banking and Finance, 32(7), page = 1242-54), 2008.

39. Chris McMahon. Forex fundamentals: moving the markets. Futures Magazine Group, January 2007.

40. *The Economic Consequences of Rising U.S. Government Debt: Priveleges at Risk*, Departmental Working Papers, Department of Economics, UC Santa Barbara, 2010.

41. Robert Wade. The asian debt-and-development crisis of 1997-?: Causes and consequences. *World Development*, 26(8):1535{1553, 1998.

42. "CPI - Consumer Price Index - Forex Trading | MetaTrader Indicators and Expert Advisors - Fundamental Analysis." *Forex Trading | MetaTrader Indicators and Expert Advisors - Home*. Web. 26 Oct. 2011. <http://www.fxfisherman.com/forex/fundamental-analysis/a135-cpi-consumer-price-index/>.

43. "An Introduction to the Financial Markets." *About.com US Economy*.Web. 01 Mar. 2012. <http://useconomy.about.com/od/themarkets/a/capital_markets.htm>.

44. "Tutorial: Stock Basics Tutorial." *Investopedia*.Web. 01 Feb. 2012. <http://www.investopedia.com/university/stocks/>.

45. Madura, Jeff. *Financial Markets and Institutions*. Mason, OH: South-Western/Thomson Learning, 2003.

46. "Mutual Funds." *About.com*. Web. 20 Feb. 2012. <http://mutualfunds.about.com/>.

47. "Bond Markets & Prices." *Bond Markets Defined*. Web. 19 Feb. 2012. <http://www.investinginbonds.com/marketataglance.asp?catid=31>.

48. Faerber, Esme, and EsmeFaerber. *All about Bonds and Bond Mutual Funds: The Easy Way to Get Started*. New York: McGraw-Hill, 2000.

49. Galant, Mark, and Brian Dolan. *Currency Trading for Dummies*. Hoboken, NJ: Wiley Pub., 2007.

50. Bouchentouf, Amine. *Commodities for Dummies*. Indianapolis, IN: Wiley Pub., 2007.

51. *Investopedia*. Web. 05 Nov. 2011. <http://www.investopedia.com/>.

52. "Forex Indicator Guide." *Best Forex Indicator*. Web. 05 Nov. 2011. <http://www.forexindicator.org/>.

53. "Forex | Forex Trading | Forex News | Currency Trading." *Forex*.Web. 05 Mar. 2012. <http://www.forexrazor.com/>.

54. "EDUCATION." *Education Courses from ALGOSYS*. Web. 05 Mar. 2012. <http://algosys8.com/index.php/education/algosys_education/technical_analysis>.

55. Malcolm, Andrew. "Keystone XL Pipeline And Jobs — Put Up Or Shut Up." *News.investors.com*. Investor's Business Daily. Web. 15 Feb. 2012. <http://news.investors.com/article/597853/201201131851/chamber-of-commerce-boosts-keystone-xl-pipeline.htm>

Appendix A: Recent Macroeconomic News

A.1 The Pegging of the Swiss Franc

Throughout the course of this year the Swiss National Bank (SNB), Switzerland's central bank, had been unable to control the ever growing value of the franc (CHF). The effect of an ever growing franc worked to deflate the currency, thus by proxy reduced the nation's ability to export goods and fueled foreign investment in the nation. Such a trend was beginning to increase the nation's unemployment. On September 6, 2011 the SNB announced that it would "peg" the CHF to the EUR in an attempt to, "reduce the deflationary development that spring from massive overvaluation of the Swiss franc." [15] The SNB set a target that the EUR/CHF should not be allowed to cross 1.200 and remarkably this target has not only held since its inception, but has been surpassed reaching highs of about 1.24.

Within an hour since the decision was announced by the SNB the EUR/CHF shot up approximately 1,000 pips. As Gartman puts it, "no where have we read of, been involved in, or even considered the notion that a strong currency regime will tie itself, willingly, to that of a lesser, weaker currency regime… never… ever." [16] This decision effectively ended the use of the CHF as a currency to be used in international transactions and acts to increase the usage of the other largest currencies: the British pound (GBP), the USD, and the Japanese Yen (JPY).

Pegging one currency to another is not an uncommon practice, in fact many nations around the world peg their currencies to the USD. However, all of the nations which do this have a weaker economy and partake in this practice as it alleviates the responsibilities of their central banks while still maintaining a specific monetary goal. This was not a true definition of "pegging" though since the Euro and Franc aren't exactly equal all the time. However, this essentially kept the euro and franc at the same value, due to the bank's buying power to force this price. They will buy it at a minimum of 1.20 but nothing less. Anything more is fine for them because they want the value of the CHF to decrease. This is very risky for Switzerland because they are buying a large volume of other foreign currencies, and they will likely sell them at a loss eventually. With the current crisis in the euro zone markets, only time will tell if such a move will have the long-term desired results.

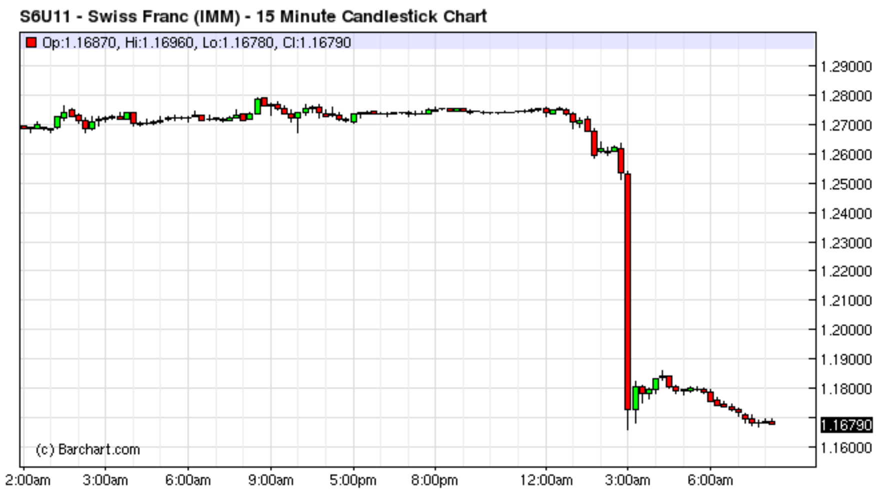


Figure 39: This is the CHFEUR value during the 'pegging'. Notice the massive drop instantaneously and how it isn't really a peg since the SNB agreed to buy it at any lower price, and it isn't following the Euro's value precisely.

Gartman believes that the president of the Swiss National Bank was trying to make up for his previous 15 billion dollars in damage to their economy previously, and to do that he is taking huge risks. An immediate result of devaluing the franc to the Euro caused the Euro to fall against most other currencies. The national bank of Switzerland is promising a global weakening of their currency and said they had to take extreme measures to do so, and was prepared to weaken it more if needed. The move does make some economic sense because their currency was so strong their exports increased in cost and lowered their tourism business.[15] The bank previously tried to weaken the currency by promising near-zero interest rates and increasing the supply of the franc, but this had no effect. The dramatic intervention was deemed necessary by the Swiss bank. [15]

Despite its global criticism, the Swiss is still trading near the euro and the bank is actually arrogantly spurring investors to buy the franc and sell the euro to break their relationship, but it holds strong. This is a macroeconomic issue because it affects the regional economy of all of Switzerland. This move put the risk of the entire Swiss economy crashing and the ruin of their financial industry. The market has more money than the Swiss Bank does, and as Gartman points out, they will not be able to keep buying all currencies to keep the relationship at their desired weakness level.[16]

A.2 The Largest Ponzi Scheme Ever Played

An issue that is brought and discussed more and more seriously as time progresses, social security and its inevitable collapse will eventually have to be dealt with for real. It has been no secret that there is not enough money to keep paying retirees and the funds are going dry. Older generations demand more from it, and younger generations refuse to put their hard earned wages into knowing they will never get it back. In Dennis Gartman's daily publication, the United States social security policy is alluded as a Ponzi scheme numerous times by a variety of people.[16] People still pay into it and that money goes to older people who paid into it first. Too much is being distributed that it has been deemed impossible to sustain later generations. It sounds exactly like a Ponzi scheme except one is forced to pay into it. Usually people enter Ponzi schemes on their own choice and free will, a position that both Gartman and Dr. Krugman, an extreme leftist agree with. [16]

President Obama temporarily cut social security spending around the middle of September of this year. This affected the entire economy of the United States because many older generations reacted nervously to the news and likely would be prompted to spend less to save their money in the fear of starvation by not being able to afford food. Gartman describes why this is a bad idea for a different reason because rather than just saving this money it should be invested so we can pay people what they deserve. [19]

Speaking of what people deserve, the first person to ever collect social security from the government was Ida Fuller. She paid less than $25 into the system and received a modest 92,000% return. That is no decimal point; she collected around $23,000 until she died, reaching triple digits in age. No reasonable person would think she deserved this. However, the policy was new and it was still working out the kinks and loop holes right? A more recent policy enables a young lady who has married an older gentleman to collect for the rest of her life if her spouse happens to perish. Is it reasonable that she is entitled for that money? Her husband worked his whole life for it while she's barely started. Governor Perry notices the idiocy of this concept and is one of the few emerging politicians who is speaking strongly and negatively about social security. This of course guarantees the loss of the votes from the older generation who fear not getting enough from the security, but strengthens his loyalties from the younger generation who don't want to pay in anymore. His upcoming election will be interesting for this split vote reasoning.[16]

Social Security has so many flaws, and the fact that people are forced to pay into it, and it has been acknowledged they likely won't be able to collect when they retire is robbery. Social security has left uneasy marks since its induction so many years ago. For instance, pre 1972 teach card was printed with a phrase that said how the card is not to be used as a form of identification, and it was never supposed to be. Suddenly that piece of information was taken off the card for 'design changes' and now it's a massive tool for identification purposes. Social Security is a very prominent macroeconomic issue in the United States that is here to stay. The closer and closer the funds get to being completely depleted, the more openly it will be discussed. It affects the entire US economy tremendously, by discouraging workers who fear they will never collect and scaring current collectors into not spending any money except on essentials. It is embedded into our economy, and will cause a lot more major problems as time continues and funds deplete.

A.3 United States Federal Bank Twisting the Yield Curve

This policy of asset purchases has come to be known as quantitative easing (QE). In general terms, QE is normally defined as a policy that expands the central bank's balance sheet in order to increase the level of central bank money in the economy. This is sometimes contrasted with a policy of changing the composition of the assets on the central bank's balance sheet (often referred to as credit easing); for example, by shifting between short and longer maturity government bonds or by shifting into riskier private assets, such as corporate bonds or equities. [22]

The yield curve is the relation between the interest rate of a loan, and the time to maturity or term of the debt. [21] To start this process, the Federal Bank sold off most off their bonds (hundreds of billions worth) and purchased long-term bonds with the money made. Most bonds they sold were close to maturing, most within 3 years. The point was to lower yields on long term bonds and simultaneously leaving the short-term rates unaffected. All of this was cleverly done without pumping more money into the economy, just reusing already existent revenue.

At the end of September the Federal Reserve announced that it would be buying long-term Treasury securities in an attempt to drive down these long-term rates. In effect, the decision was put in place to attempt to reduce mortgage rates in the U.S. and spur the otherwise declining sector of home sales.

Short-term interest rates, such as those on Treasury bills and commercial paper, are affected not only by the current level of the federal funds rate but also by expectations about the overnight federal funds rate over the duration of the short-term contract. As a result, short-term interest rates could decline if the Federal Reserve surprised market participants with a reduction in the federal funds rate, or if unfolding events convinced participants that the Federal Reserve was going to be holding the federal funds rate lower than had been anticipated. Similarly, short-term interest rates would increase if the Federal Reserve surprised market participants by announcing an increase in the federal funds rate, or if some event prompted market participants to believe that the Federal Reserve was going to be holding the federal funds rate at higher levels than had been anticipated. [23]

According to the Federal Reserve, "changes in long-term interest rates also affect stock prices, which can have a pronounced effect on household wealth. Investors try to keep their investment returns on stocks in line with the return on bonds, after allowing for the greater riskiness of stocks." [23] This is essentially what the Federal Reserve is attempting to do with the "twisting" of the yield curve by buying back long term treasuries. The Fed also notes that as the interest rates are raised the dollar will bid up in currency markets, precisely what happened following the announcement by the FOMC.

The major criticism of the decision by the Fed is that increasing the interest rates of short term bonds could adversely affect many of the smaller banks in the nation who rely on short term loans as the basis of their operations. However, when the FOMC announced the "twisting", the USD immediately saw a boom, swelling up over 200 pips vs. many other currencies.

A.4 The Grecian Default – The Euro and PIIGS

The euro (EUR) is the first currency created to be the base currency of a conglomerate of nations under control of one central bank known as the European Central Bank (ECB). Seventeen different nations from the European Union currently use the euro as their currency; they are: Belgium, Germany, Estonia, Ireland, Greece, Spain, France, Italy, Cyprus, Luxembourg, Malta, The Netherlands, Austria, Portugal, Slovenia, Slovakia, and Finland. [26]

Since this is historically the first time a group of nations has decided to use one currency under the control of a board overseen by the union there have been many unseen problems which have arisen. Several treaties have been enacted by the member nations of the euro block in an attempt to properly regulate the currency, including the treaties of Lisbon and Maastricht.

The treaty of Maastricht explicitly sets the levels of economic activity allowed by the ECB. Since the treaty's induction in 2000 the euro zone has breached the deficit and debt limit on 137 different occasions. [30] The failing of so many nations to adhere to the limits which they agreed to has created a partition of euro countries known as PIIGS, which are the nations which have been causing the greatest turmoil. Portugal, Italy, Ireland, Greece and Spain are the nations whose debt and policies has been causing so much turmoil in Europe.

Of all the nations of PIIGS, Greece perhaps is the most troublesome. Their debt to GDP ratio is by far the worst in the union at 157% [27] and every measure which they take to alleviate that debt seems to fail. Over the past few months government workers have been laid off in droves and still Greece cannot bring its debt under control.

Italy is next on this list with 129% debt to GDP ratio. [27] In addition to being plagued with a financial crisis in the EU, Prime Minister Berlusconi has been accused of handing out big government contracts in exchange for sexual favors, with as many as eight women in one night. [28]

As can be seen by the countries above treaties governing the financial policies of several of the euro countries have been broken repeatedly. The European System of Financial Supervisors (ESFS), was developed to help settle disputes and keep nations compliant with the outlined regulations.[28] However, the original inception of the ESFS saw little impact as its ability to enforce regulations was limited. Prompted by the need to do something, in September, the member nations of the ECB voted overwhelmingly to expand the powers of the ESFS in an attempt to stabilize the euro. [27]

The enhanced ESFF has been designed to head off a deeper crisis, by allowing it to provide support to euro-zone countries. It would also effectively take over the European Central Banks bond-buying duties, allowing the fund to take action in markets to hold down bond yields. This is an effort to further insulate vulnerable countries from being engulfed by the debt crisis. [29]

The heavy burden of the bailouts being issued has been place on the hands of the largest economy in the euro-zone, Germany. Despite much opposition from the German public, Angela Merkel, the Prime Minister of Germany, has approved many loans to the nations of the PIIGS. It has been estimated by the Institute for Economic Research (IFO), a German economic think tank, that the expected cost of further bailouts to Greece alone is 465 billion euro. [30]

It is widely speculated that despite efforts to inject money into Greece, their inability to implement a successful fiscal policy will inevitably lead to default. Greece has defaulted on loans on many occasions and given the current path it seems only a matter of time until it defaults again. The major fear is that if Greece defaults then so will Italy, Spain, and the rest will follow. If this trend continues only time will tell if the euro remains a currency in the coming years.

Continued political conferences and debates have been constantly going on about what to do in Europe. Other Eurozone countries have reported their own financial difficulties, brought on by the decreasing value of the Euro. The Euro has been steadily falling over the past year which has resulted in more than 1000 pip loss (relative to the US dollar). Clearly the world has put less faith into the Euro because of this crisis. As more and more people lose faith in the Euro its value perpetually decreases. People decide to invest their money in a stronger currency so they buy one, so for example the CHF. This then creates more demand for the CHF and less for the Euro. As more and more people do this with more and more euro currency its value keeps going down. The crisis in Greece is affecting every country which uses the Euro as the main currency, and it is hurting them all. Greece is replacing high ranking officials as to of no avail yet, but something needs to change to pull out of the dangerous crisis in Europe. [25]

A.5 Selling your Debt – Italy and Spain Auctions

A lot of currency in the form of Spanish debt was being auctioned off and fetched a surprisingly generous bid. This was the first bullish appeal the Euro has seen in quite some time. Investors realized that they had become too bearish of European debt, largely as a consequence of the enormous amount of liquidity of the European Central Bank, which lent nearly half a trillion dollars of three-year bonds with similar offers earlier. The Spanish treasury said it sold €5.6 billion ($7.3 billion), which is far greater than the €4.5 billion it had initially sought. [27] Investors demanded an interest rate of only 1.74 percent to lend the government 3-month money, down sharply from 5.1 percent in the last such auction onNov. 22. [31] The rate for the 6-month bills was 2.44 percent, down from 5.22 percent. [31] The buyers were mostly domestic banks and domestic individuals in Spain, along with the ECB. He fact that domestic buyers are supporting this shows how faith really has been restored in the Euro a little, because they are definitely well aware of the state that the Eurozone has been in recently.

During the days of January 12 and January 13, Italy also managed to achieve a successful bond auction, though its auction was not as successful as Spain's auction. Italy auctioned 12 billion euros on January 12 and another 4.5 billion euros on January 13 in treasury bills as borrowing costs plunged in the country's first debt sale of the year. This also restored confidence to Europe because Italy was second in debt leaders only to Greece. The treasuries of Greece, Ireland and Portugal sought bailouts because their bond yield rates were above seven percent, which is extremely high. Italy's success in lowering its yield rates and being able to attract investors to buy large quantities of debt shows that confidence in the Italian economy is improving.

The fundamental impacts of the debt sale was on a macroeconomic scale and mostly positive; stocks rose and the price of oil fell. The Euro strengthened against other currencies since faith was starting to be put back into the Euro. The euro added 0.9 percent after the sales and rose to most main currencies. The Dow Jones Industrial average also strengthened 21.57 points to 12,471.02. [31] This does not mean the crisis is over in any way though, in fact we are far from it This is just one of many first baby steps. Considering the S&P recently downgraded the government debt of France, Austria, Italy and Spain, problems clearly exist and such consecutive downgrades are a horrific thing for Europe. At least Germany's debt remains at AAA ranking to all three international debt-rating companies.

A.6 Oil Pipeline

A project proposal for the construction of an international oil pipeline has been a buzz around the world recently. Indeed, the Keystone XL project has found abundant support among some import and influential groups in the United States. Some include: the International Brotherhood of Electrical Workers, the International Union of Operating Engineers, the Teamsters, the Laborers' International Union, the Building and Construction Trades Department of the AFL-CIO and the United Association of

Plumbers and Pipe Fitters for the United States & Canada.[27] This makes sense because many jobs would be created for these groups. Furthermore, U.S. Chamber of Commerce President Tom Donohue has publicly endorsed the project to President Obama, prompted as a response to his recent announcement to award companies which bring jobs to the United States.

Like every coin, there are two sides to this project. Some adamant opponents of the project have environmental concerns, as the initial pipeline course brought oil over the Ogallala Aquifer in Nebraska. A secondary route was outlined by TransCanada to mitigate these concerns. However, the project is still not underway. President Obama has been criticized1 to be waiting for election time to receive the good public appeal which would come from pushing this project through. “At a press conference following Donohue's address, Bruce Josten, chamber vice president of government affairs, noted that Keystone XL had already cleared an extensive three-year review and that only presidential politics stood in the way. This is a classic example of a macroeconomic issue because it is affecting the entire world at the same time, making way for all these debates and can result in both good and bad things for many economies.” [55]

Appendix B: Programming in the Forex Market

B.1 Abstract

Programming is a useful way to make life simpler, faster, and more efficient in everyday tasks. When applied to the foreign currency exchange market it can be used to create powerful tools that can lead to high gains in some cases. Whether it is the newest indicator that nobody else is using, or the latest automated trading robot that makes 20% a week, programming can be useful. Moreover it can be used to present the data in a different way which can be in turn used to see trends and make custom indicators, such as extracting chart data to MATLAB and applying useful functions to it.

One project created a variety of programs over the course of the 3 term project. They range from adding MATLAB integration to MQL4 to exporting TradeStation data to a C++ environment. Some indicators and expert advisors are created such as a moving average (for practice) and a double top alerter respectively. Other things were programmed to just make trading easier, such as a program that automatically draws support and resistance lines to one that adds a trailing stop loss to all your trades automatically. The bottom line is that programming can be used everywhere in trading securities, and forex is certainly no exception.

B.2 Programming for MQL4

B.2.1 MATLAB Interface

Our newest script uses a DLL to interface MQL4 and Matlab. It gives users the ability to access variables from code in both programs, and to call Matlab functions from inside an MQL4-based script, indicator, or robot. Thus, advanced analytical tools can be used on the discrete-time signal that is the Forex market. Powerful numerical analysis of past data can be performed quickly and in a platform with which we are very familiar.

Physical Code

```
// MatlabMql.cpp : Defines the exported functions for the DLL application.
//

#include "stdafx.h"

Engine *__ep=NULL;
mxArray*T =NULL;

_DLLAPI int _stdcall initMatlabEngine()
{
        if(__ep==NULL){
                if(!(__ep=engOpen("\0"))){
                        fprintf(stderr, "\nCan't start MATLAB engine\n");
                        returnEXIT_FAILURE;
                }else{
                        returnEXIT_SUCCESS;
                }
        }
        returnEXIT_SUCCESS;
}

_DLLAPI void __stdcall evalExpression(char* text)
{
        engEvalString(__ep, text);
}


_DLLAPI int _stdcall exportDoubleVector(char*var_name, double* vector, int size){
        T =mxCreateDoubleMatrix(1, size, mxREAL);
        memcpy((void*)mxGetPr(T), (void*)vector, size);
        intretval=engPutVariable(__ep, var_name, T);
        mxDestroyArray(T);
        returnretval;
}

_DLLAPI int _stdcall exportDouble(char*var_name, double value){
        T =mxCreateDoubleScalar(value);
        intretval=engPutVariable(__ep, var_name, T);
        mxDestroyArray(T);
        returnretval;
}

_DLLAPI int _stdcall exportIntegerVector(char*var_name, int* vector, int size){
        T =mxCreateDoubleMatrix(1, size, mxREAL);

        double*double_vector=(double*)(malloc(size *sizeof(double)));

        for(inti=0;i<size;i++){
           double_vector[i]=(double) vector[i];
        }

        memcpy((void*)mxGetPr(T), (void*)vector, size);
        intretval=engPutVariable(__ep, var_name, T);
        mxDestroyArray(T);
        free(double_vector);
        returnretval;
}

_DLLAPI int _stdcall exportInteger(char*var_name, int value){
        T =mxCreateDoubleScalar((double) value);
        intretval=engPutVariable(__ep, var_name, T);
        mxDestroyArray(T);
        returnretval;
}


_DLLAPI int _stdcall exportString(char*var_name, char*str){
        T =mxCreateString(str);
        intretval=engPutVariable(__ep, var_name, T);
        mxDestroyArray(T);
        returnretval;
}

_DLLAPI void _stdcall closeEngine(){
   engClose(__ep);
}

_DLLAPI int __stdcall engineSetVisible(){
        returnengSetVisible(__ep, true);
```

```
}

_DLLAPI int __stdcall engineSetInvisible(){
        returnengSetVisible(__ep, false);
}
```

B.2.3 Trailing Stop For Live trade

During the course of this project we found that often time constraints forced us to close open positions before any signals occurred which warrant this action. For this reason it became necessary to create a program which would determine if conditions warranted such a close of our position, allowing us to leave orders open while we were not physically available to check for such conditions. For this reason we created a simple trailing stop program.

This program utilizes a user defined stop loss level (SL), but is set by default to 15 pips. At every incoming tick it cycles through all open orders of the current selected currency pair. During each of these cycles it checks to see if the current price at which the stop loss is set to trigger is further than the SL away from the current price (Ask or Bid depending on direction). If it is indeed further then the program updates the stop loss trigger level and then cycles on to the next open order. In this manner, every time a tick goes in favor of the current trade direction the stop loss is moved proportionally along with it, while if a tick goes against a trade it does nothing.

Physical Code

```
//+------------------------------------------------------------------+
//|                                          TrailStop4LiveTRade.mq4 |
//|                                                    Group 5       |
//|                                        http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5"
#property link      "http://www.metaquotes.net"

externdoubleTrailingStop     =100.0;//SL in pips
externboolFiveDigBroker      =True;//true if 5 dig; false if 4 dig broker

//+------------------------------------------------------------------+
//| expert initialization function                                   |
//+------------------------------------------------------------------+
intinit()
  {
//----

//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert deinitialization function                                 |
//+------------------------------------------------------------------+
intdeinit()
  {
//----

//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert start function                                            |
//+------------------------------------------------------------------+
int start()
  {
//----
//if (FiveDigBroker ==True) TrailingStop=TrailingStop*10.0; //adjusts the level of stops to a 5 dig broker

int total=OrdersTotal();//makes sure our loop runs for every open trade for respective currency pair
for(int count=0;count<total;count++)
         {
OrderSelect(count,SELECT_BY_POS,MODE_TRADES);//select by pos
Print("Order Selected");
if(OrderType()<=OP_SELL &&OrderSymbol()==Symbol())// OrderTypes are indexed: buy, sell, pend buy limit, p b stop, p s.l., p s.s.
            {
Print(Symbol());
if(OrderType()==OP_BUY)//work with long positions (as stops will be in opp direction as shorts)
               {
if(TrailingStop>0)//check to make sure we don\'t have -SL
                  {
if(OrderStopLoss()<Bid-Point*TrailingStop)//look to see if TS needs to be updated
                        {
Print("Should modify now?");
OrderModify(OrderTicket(),OrderOpenPrice(),Bid-Point*TrailingStop,OrderTakeProfit(),0,Green);//change SL
return(0);
                        }
                   }
```

```
                }
if(OrderType()==OP_SELL)//do same steps for short positions
                {
if(TrailingStop>0)
                  {
if(OrderOpenPrice()-Ask>Point*TrailingStop)
                    {
if(OrderStopLoss()>Ask+Point*TrailingStop)
                      {
OrderModify(OrderTicket(),OrderOpenPrice(),Ask+Point*TrailingStop,OrderTakeProfit(),0,Red);
return(0);
                      }
                     }
                   }
                }
              }

          }
//----
return(0);
  }
//+------------------------------------------------------------------+
```

B.2.4 Draw Entry Exit Program

Throughout the course of this project our groups had to make periodic presentations to the other members of the project and our professor. While these presentations contained many pieces of information they included the trades we made for the week. For this reason it became necessary to display the individual trades we made and the underlying reasons for making them. The process of gathering screenshots of all of the weekly trades could be quite a laborious one and thus we decided to write a program which would automatically draw directly on the chart all of the trades within a specified time frame. A check-mark is placed at the entry position and a cross at the close position with a connecting dotted red line. The prices are indicated at the dotted orange lines (which highlights price on right). Short trades are indicated by a downward arrow, while long positions are indicated by upward arrows.

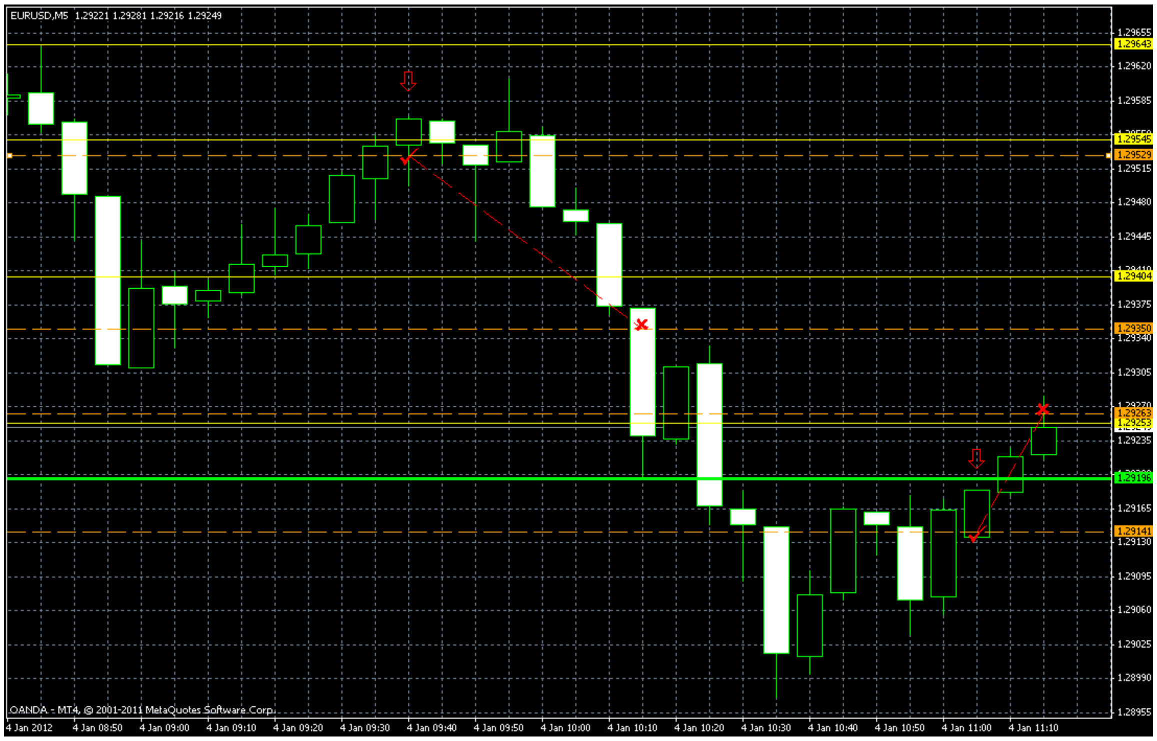


Figure 40: A Screenshot of the Draw entry exit Program. Two short position trades are shown in this chart. (Green and yellow lines are from a separate indicator).

Physical Code

```
//+------------------------------------------------------------------+
//|                                                DrawEntryExit.mq4 |
//|                                                          Group 5 |
//|                                           http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5"
#property link      "http://www.metaquotes.net"

#property indicator_chart_window
#property indicator_buffers 2
#property indicator_color1 Green
#property indicator_color2 Red
#property indicator_width1 2
#property indicator_width2 2
//---input parameters
extern color opencolor        =Orange;
extern color closecolor       =Orange;
externintopenwidth          =1;
externintclosewidth         =1;
externintopenstyle          =STYLE_DASH;
externintclosestyle         =STYLE_DASH;
externintorderbuysig        =SYMBOL_ARROWUP;
externintordersellsig       =SYMBOL_ARROWDOWN;
externintorderclosesig      =SYMBOL_STOPSIGN;
```

```
externintorderopensig       =SYMBOL_CHECKSIGN;
externintorderclosewidth    =3;
externintorderopenwidth     =3;
//--- buffers
double ExtMapBuffer1[];
double ExtMapBuffer2[];
//+------------------------------------------------------------------+
//| expert initialization function                                   |
//+------------------------------------------------------------------+
intinit()
  {
//----
SetIndexStyle(0,DRAW_LINE);
SetIndexBuffer(0,ExtMapBuffer1);
SetIndexStyle(1,DRAW_LINE);
SetIndexBuffer(1,ExtMapBuffer2);
//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert deinitialization function                                 |
//+------------------------------------------------------------------+
intdeinit()
  {
//----

//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert start function                                            |
//+------------------------------------------------------------------+
int start()
  {
//----

inttotalclosed   =OrdersHistoryTotal();
bool Opens         = FALSE;
bool Closes        = FALSE;
intCurrentOTime, CurrentCTime, CurrentOPos, CurrentCPos, run;
```

```
double rise, RiseOverRun;
stringhlineopen, hlineclose, openarrow, closearrow, opentime, closetime, dots;

for(int count=0;count<totalclosed; count++)
   {
OrderSelect(count, SELECT_BY_POS, MODE_HISTORY);
if(OrderSymbol()==Symbol())
      {
hlineopen  ="open "+OrderOpenTime()+count;
hlineclose="close "+OrderOpenTime()+count;
opentime   ="OpenTime "+OrderOpenTime()+count;
closetime  ="CloseTime "+OrderOpenTime()+count;
dots       ="dots"+OrderOpenTime()+count;
openarrow  ="oarrow"  +OrderOpenTime()+count;
CurrentOTime=OrderOpenTime();
CurrentCTime=OrderCloseTime();
ObjectCreate(hlineopen,OBJ_HLINE, 0, 0, OrderOpenPrice());
ObjectCreate(hlineclose,OBJ_HLINE, 0, 0, OrderClosePrice());
ObjectSet(hlineopen, OBJPROP_COLOR, opencolor);
ObjectSet(hlineclose, OBJPROP_COLOR, closecolor);
ObjectSet(hlineopen, OBJPROP_WIDTH, openwidth);
ObjectSet(hlineclose, OBJPROP_WIDTH, closewidth);
ObjectSet(hlineopen, OBJPROP_STYLE, openstyle);
ObjectSet(hlineclose, OBJPROP_STYLE, closestyle);
ObjectCreate(closetime, OBJ_ARROW,0,OrderCloseTime(), OrderClosePrice()+18*Point);
ObjectSet(closetime,OBJPROP_ARROWCODE, orderclosesig);
ObjectSet(closetime,OBJPROP_WIDTH, orderclosewidth);
ObjectCreate(openarrow,OBJ_ARROW, 0, OrderOpenTime(),OrderOpenPrice()+10*Point);
ObjectSet(openarrow,OBJPROP_ARROWCODE,orderopensig);
ObjectSet(openarrow,OBJPROP_WIDTH,3);
   ObjectCreate(dots,OBJ_TREND,0,OrderOpenTime(),OrderOpenPrice(),OrderCloseTime(),OrderClosePrice());
ObjectSet(dots,OBJPROP_STYLE,STYLE_DASH);
ObjectSet(dots,OBJPROP_RAY,0);
if(OrderType()==OP_BUY)
            {
       ObjectCreate(opentime, OBJ_ARROW, 0, OrderOpenTime(), OrderOpenPrice()-75.0*Point);
       ObjectSet(opentime,OBJPROP_ARROWCODE, orderbuysig);
       ObjectSet(opentime,OBJPROP_WIDTH, orderopenwidth);
       ObjectSet(dots,OBJPROP_COLOR,Green);
```

```
                 }
      if(OrderType()==OP_SELL)
                 {
      ObjectCreate(opentime, OBJ_ARROW, 0, OrderOpenTime(), OrderOpenPrice()+90.0*Point);
      ObjectSet(opentime,OBJPROP_ARROWCODE, ordersellsig);
      ObjectSet(opentime,OBJPROP_WIDTH, orderopenwidth);
      ObjectSet(dots,OBJPROP_COLOR,Red);
                 }
             }
        }
      //----
      return(0);
       }
      //+------------------------------------------------------------------+
```

B.2.5 Double Tops

The program DoubleTopsv3.mql4 is one which was created to recognize and trade the Double Tops pattern. The Double Tops pattern can be summarized into 3 parts.

A. An upward trending pattern

B. A reversal to a downward trend creating a resistance line

C. A second reversal back to the resistance line

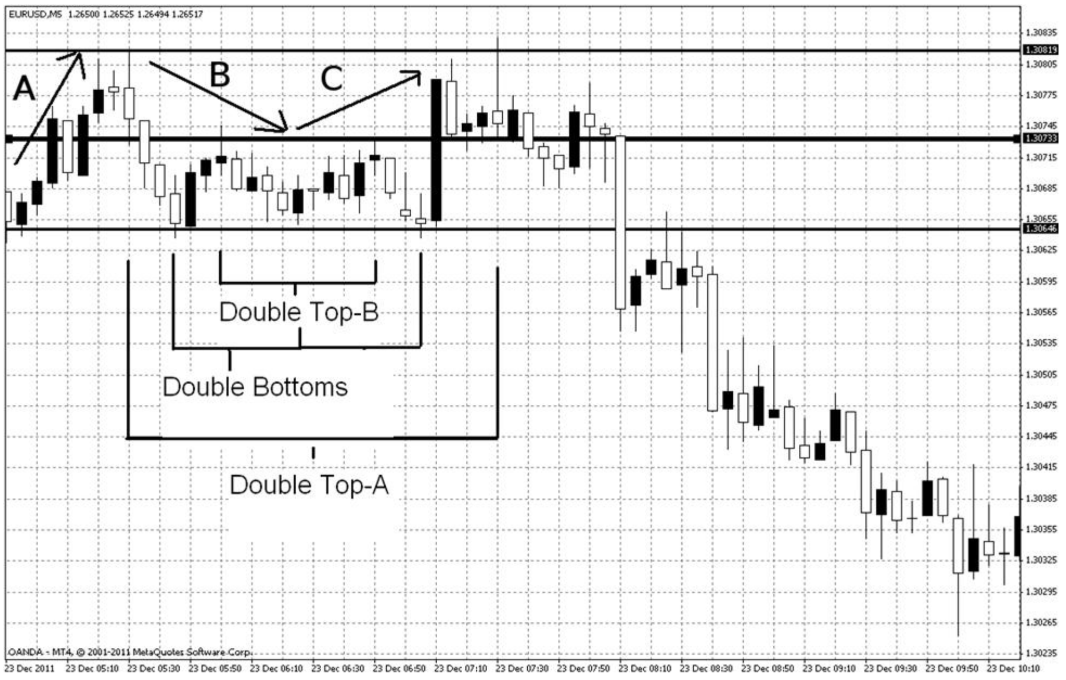
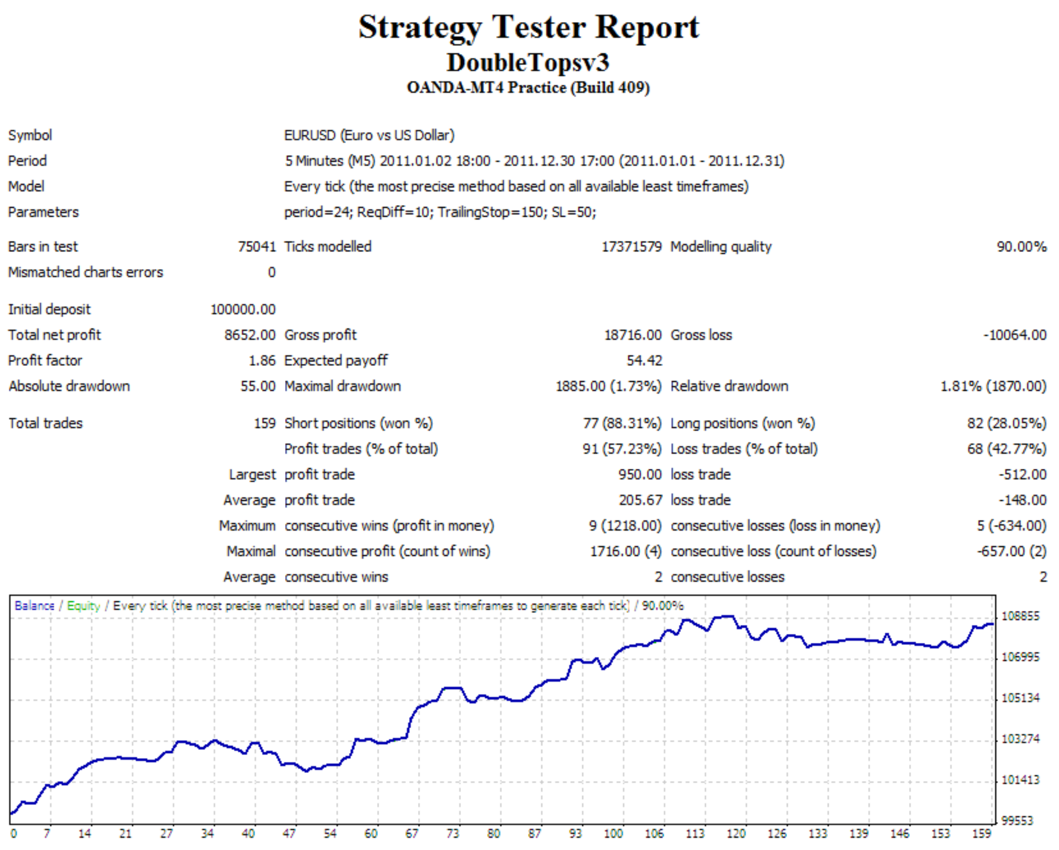


Figure 41: Double Top Pattern contains 3 trend lines denoted A, B, and C

After these three parts have been exhibited the price can be expected to reverse yet again off of the resistance line, else if it breaks this line it would be considered a Double Top Pullback. The Double Bottom pattern is just the inverted version of this, in which case the price would be bouncing off of a support line.

This program is set by default to look at the past 24 bars, and meant to be used on a 5-minute chart, thus utilizing only data for the past 2 hours. This value of 24 was determined by data mining past charts for occurrences of this pattern. It then stores both the highest and lowest points and waits until the price begins to return to one of these levels. If the price approaches within 10% (user-adjustable) of the difference between the high and low for this period then the program initiates a trade in the respective direction. Only one trade is allowed to be open at a time, and a user defined trailing stop is initialized (default of 15 pip).

This program was back tested using 5 minute data for the EURUSD downloaded from the MetaQuotes history center. This program achieved some of the best results we saw out of any of our expert advisors. It has both a 57% profit ratio, as well as an 8.652% total profit over the year of 2011, with a maximal drawdown under 1.8%. However, for reasons which remain unknown the program achieved much better results for short positions than it did for long ones.



Strategy Tester Report
DoubleTopsv3
OANDA-MT4 Practice (Build 409)

Symbol	EURUSD (Euro vs US Dollar)				
Period	5 Minutes (M5) 2011.01.02 18:00 - 2011.12.30 17:00 (2011.01.01 - 2011.12.31)				
Model	Every tick (the most precise method based on all available least timeframes)				
Parameters	period=24; ReqDiff=10; TrailingStop=150; SL=50;				
Bars in test	75041	Ticks modelled	17371579	Modelling quality	90.00%
Mismatched charts errors	0				
Initial deposit	100000.00				
Total net profit	8652.00	Gross profit	18716.00	Gross loss	-10064.00
Profit factor	1.86	Expected payoff	54.42		
Absolute drawdown	55.00	Maximal drawdown	1885.00 (1.73%)	Relative drawdown	1.81% (1870.00)
Total trades	159	Short positions (won %)	77 (88.31%)	Long positions (won %)	82 (28.05%)
		Profit trades (% of total)	91 (57.23%)	Loss trades (% of total)	68 (42.77%)
	Largest	profit trade	950.00	loss trade	-512.00
	Average	profit trade	205.67	loss trade	-148.00
	Maximum	consecutive wins (profit in money)	9 (1218.00)	consecutive losses (loss in money)	5 (-634.00)
	Maximal	consecutive profit (count of wins)	1716.00 (4)	consecutive loss (count of losses)	-657.00 (2)
	Average	consecutive wins	2	consecutive losses	2

Balance / Equity / Every tick (the most precise method based on all available least timeframes to generate each tick) / 90.00%
108855
106995
105134
103274
101413
99553
0 7 14 21 27 34 40 47 54 60 67 73 80 87 93 100 106 113 120 126 133 139 146 153 159

Figure 42:Results of the Double Top/Bottom program "DoubleTopsv3"

```
//+------------------------------------------------------------------+
//|                                                 Double Tops.mq4 |
//|                                                         Group 5    |
//|                                        http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5"
#property link      "http://www.metaquotes.net"

externintperiod          =  24;
externdoubleReqDiff      =  10;
externdoubleTrailingStop=  150;
externdouble SL           =  50;
//+------------------------------------------------------------------+
//| expert initialization function                                   |
//+------------------------------------------------------------------+
intinit()
  {
//----

//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert deinitialization function                                 |
//+------------------------------------------------------------------+
intdeinit()
```

```
  {
//----

//----
return(0);
  }
//+------------------------------------------------------------------+
//| expert start function                                            |
//+------------------------------------------------------------------+
int start()
  {
//----
intticket,volume,slippage,magic;
double highs[2];
double lows[2];
inthighspos[2];
intlowspos[2];
doublecurrenthigh, currentlow, TP;
volume =1;
slippage =3;
magic =10101;
```

```
TP=15000;
int total=OrdersTotal();//makes sure our loop runs for every open trade for respective
    currency pair
currenthigh=High[0];
currentlow=Low[0];
for(inti=0;i<period;i++)
   {
if(High[i]>currenthigh)
         {
currenthigh=High[i];
highs[1]=highs[0];
highspos[1]=highspos[0];
highs[0]=currenthigh;
highspos[0]=i;
         }
if(Low[i]<currentlow)
         {
currentlow=Low[i];
lows[1]=lows[0];
lowspos[1]=lowspos[0];
lows[0]=currentlow;
```

```
lowspos[0]=i;
         }
    }
if(total<1)
{
if(highs[0]>highs[1]&& highs[0]-lows[0]>=ReqDiff*10.0*Point
&&highspos[0]>lowspos[0]&&highspos[1]<lowspos[0])
   {
if(High[0]>highs[0]-0.025*(highs[0]-lows[0]))
         {
            ticket=OrderSend(Symbol(),OP_SELL,volume,Bid,slippage,0,0,NULL,magic,0,Red);
OrderModify(ticket, OrderOpenPrice(),Ask+(SL*10*Point),Ask-TP*Point,0,Red);
         }
   }
if(lows[0]<lows[1]&& highs[0]-lows[0]>=ReqDiff*10*Point
&&lowspos[0]>highspos[0]&&lowspos[1]<highspos[0])
   {
if(Low[0]<lows[0]+0.025*(highs[0]-lows[0]))
         {
            ticket=OrderSend(Symbol(),OP_BUY,volume,Ask,slippage,0,0,NULL,magic,0,Green);
OrderModify(ticket, OrderOpenPrice(),Bid-(SL*10*Point),Bid+TP*Point,0,Green);
```

```
         }
   }
}
for(int count=0;count<total;count++)//basic formula for a "for" loop
         {
OrderSelect(count,SELECT_BY_POS,MODE_TRADES);//select by pos
if(OrderType()<=OP_SELL &&OrderSymbol()==Symbol())// OrderTypes are indexed: buy, sell, pend buy limit, p b stop, p s.l., p s.s.
            {
if(OrderType()==OP_BUY)//work with long positions (as stops will be in opp direction as shorts)
               {
if(TrailingStop>0)//check to make sure we don\'t have -SL
                  {
if(OrderStopLoss()<Bid-Point*TrailingStop)//look to see if TS needs to be updated
                               {
       OrderModify(OrderTicket(),OrderOpenPrice(),Bid-Point*TrailingStop,OrderTakeProfit(),0,Green);//change SL
       return(0);
                               }
                          }
               }
if(OrderType()==OP_SELL)//do same steps for short positions
               {
if(TrailingStop>0)
                  {
if(OrderOpenPrice()-Ask>Point*TrailingStop)
                     {
if(OrderStopLoss()>Ask+Point*TrailingStop)
                        {
OrderModify(OrderTicket(),OrderOpenPrice(),Ask+Point*TrailingStop,OrderTakeProfit(),0,Red);
return(0);
                        }
                      }
                   }
               }
            }
               }
//----
return(0);
  }
```

```
125.        //+------------------------------------------------------------------+
```

B.2.6 Support and Resistance Indicator: Identifying Critical Levels

First, a small fixed range is decided (ex: candles in window ÷ 10). In the example below there are 6 candles in this range. That range is now a window which is stepped back through the candles under analysis. Each time it is moved, the local extrema are determined. The value is discarded if it appears at either end of the window – because that can be indicatory of a trend, and the next window-location will pick it up if it is a true local extreme.

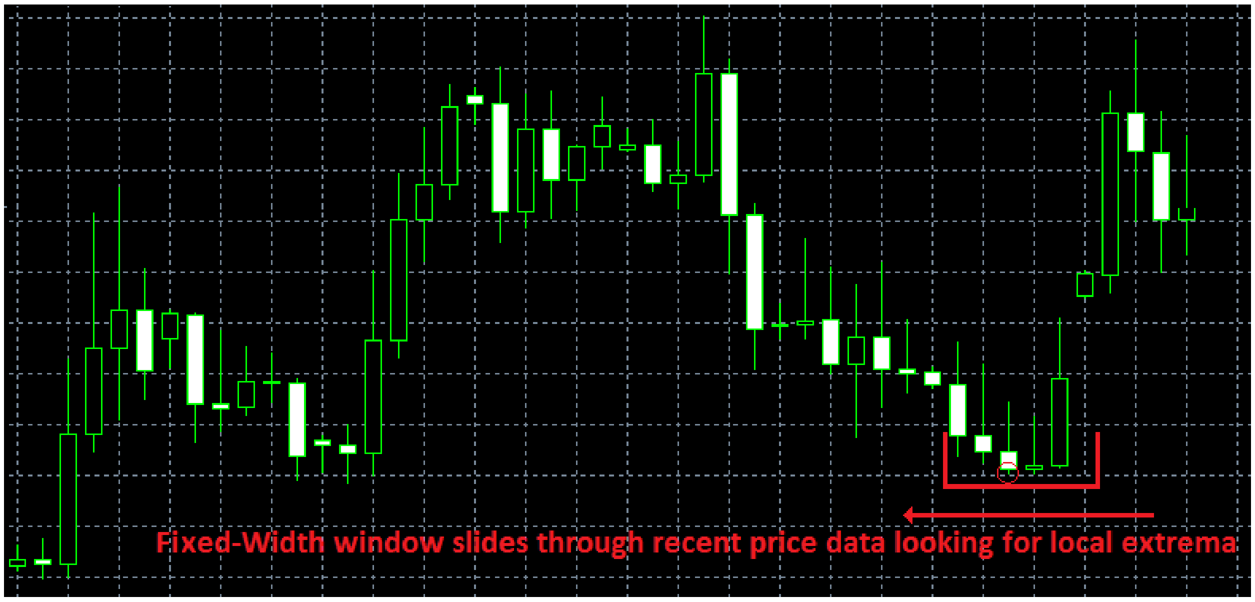


Figure 43: Sliding Window

After the slide is complete, all possibly (but not definitely) noteworthy local extrema have been identified and stored. The figure below shows only the minima, for simplicity.

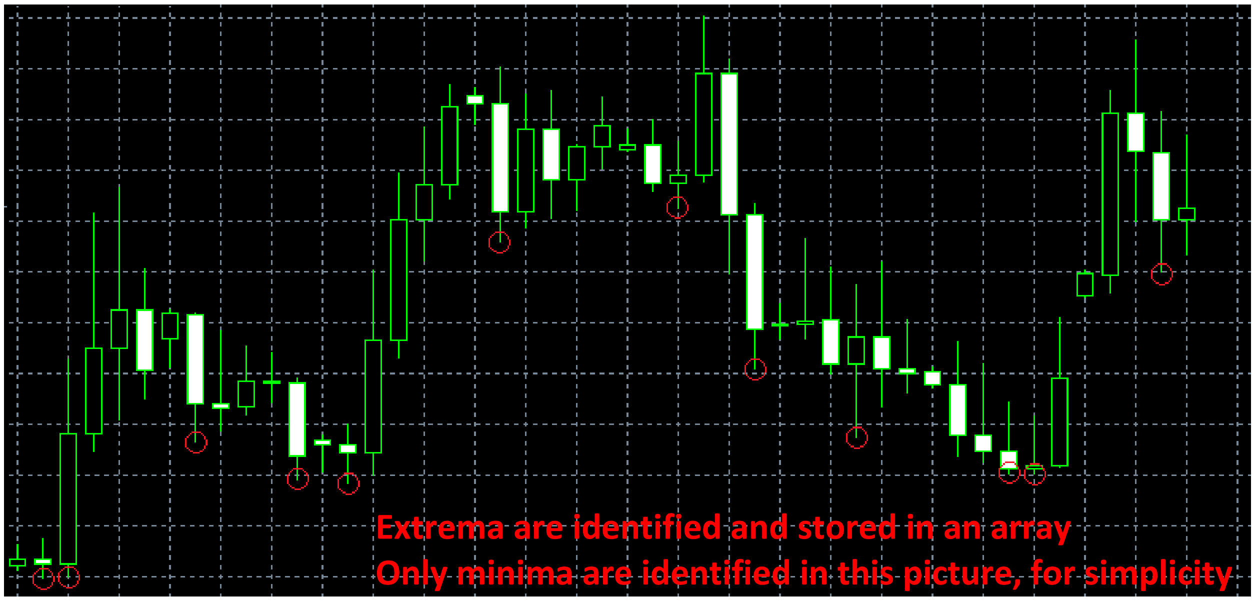


Figure 44: Identified Extrema

Next, the price-values of those minima are stored in an array and sorted. Graphically, this is equivalent to looking at all the vertical heights on one axis (as shown on the left side of the figure below)

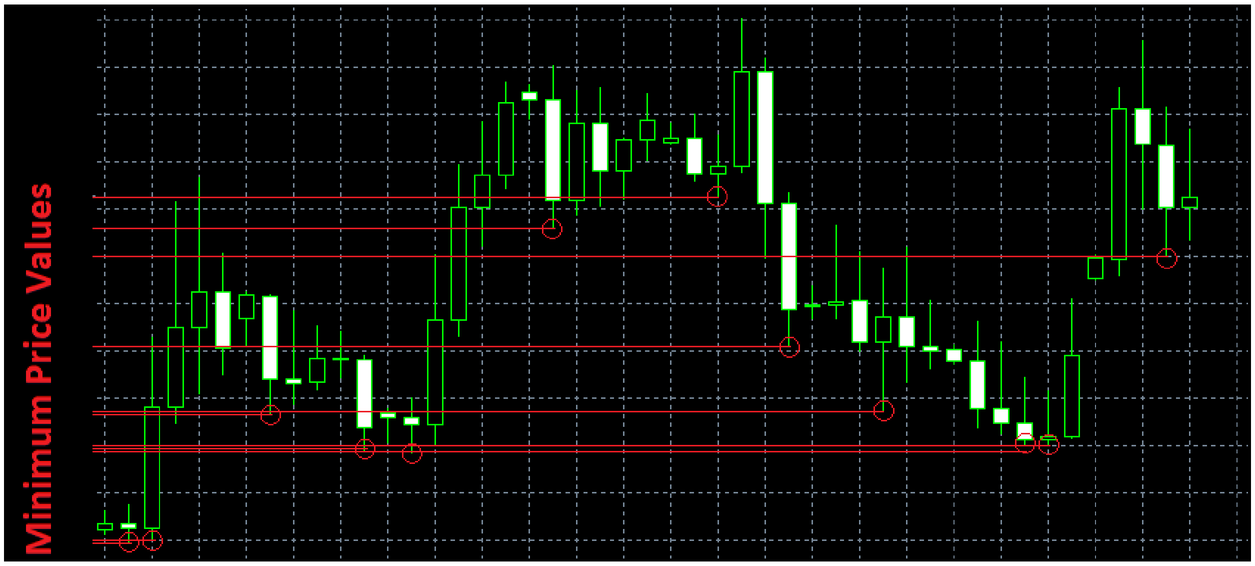


Figure 45: Extrema Sorted

Now an algorithm must look through the sorted array of local minima and decide which are important or noteworthy. Graphically, a group of lines close together (in vertical spacing) is more important than a group of lines which are farther apart.   A simple way to implement this is to break up the vertical scale into a series of ‘buckets’, and see which contain the most horizontal lines.

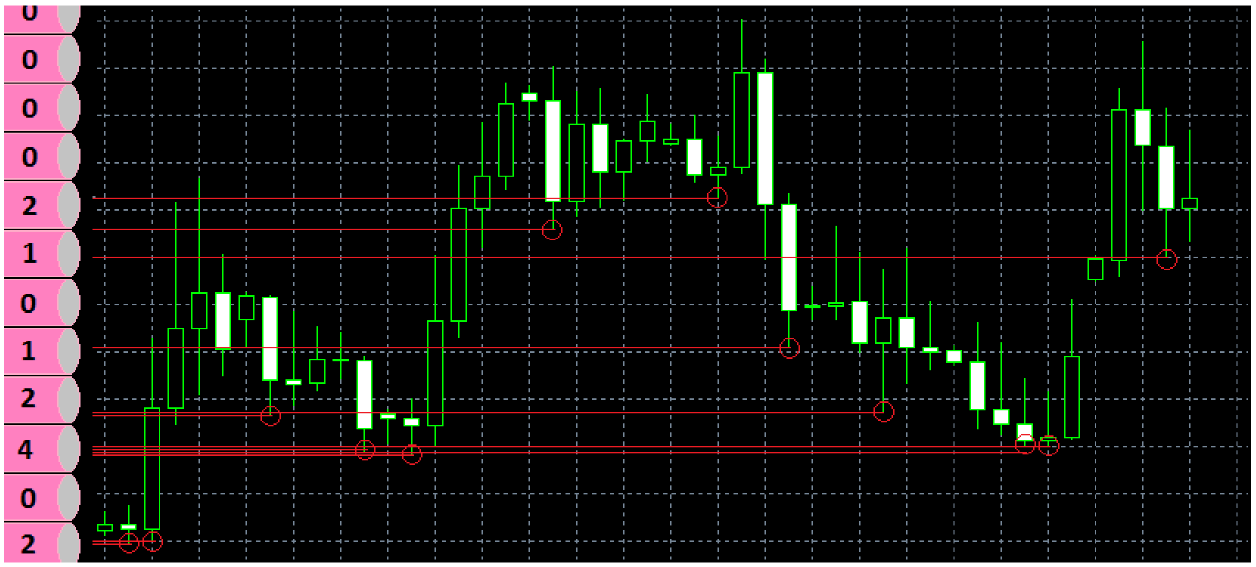


Figure 46: Applying Buckets

The second bucket from the bottom has the most minimums in it – therefore we associate it with a meaningful support line. The output of this function is a horizontal line located at the average of the minimums from within that bucket. This is shown in the next figure.

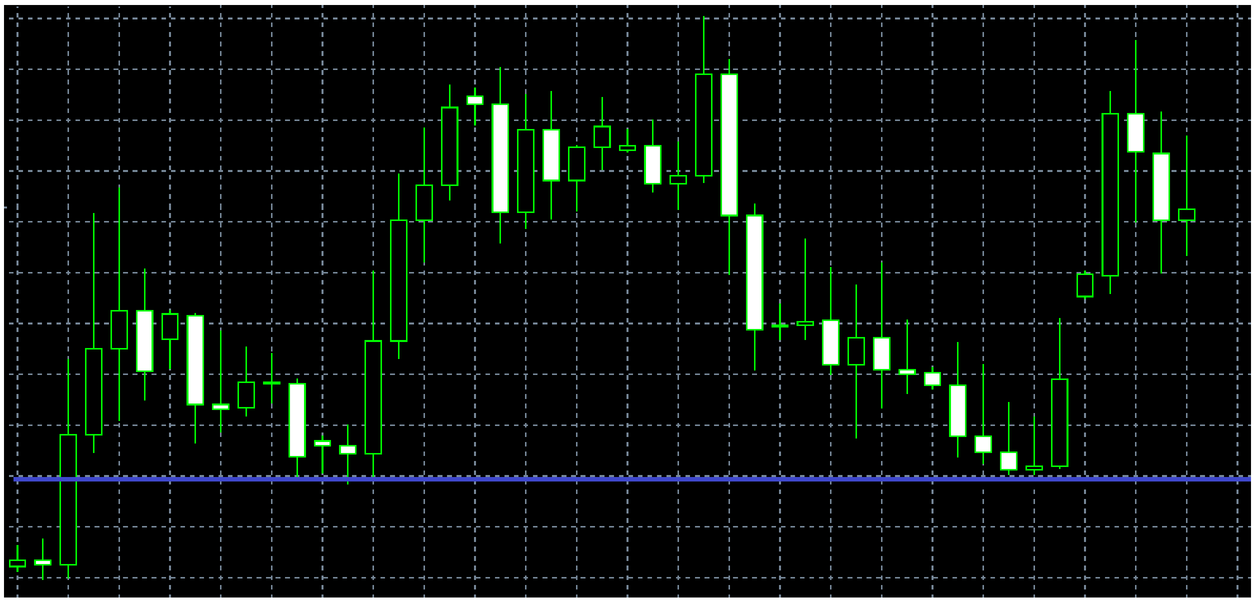


Figure 47: Support Output

While the previously described process is occurring, very similar lines of code are applying the same processing technique to local maximum values. Buckets of similar nature are created, and local maxima are determined. Currently, out algorithm plots lines for all buckets with three or more values in them. However, this has the flaw of drawing a huge number of lines on the screen when a lot of candles are being processed. A more sophisticated output might include plotting the three ‘fullest’ buckets.

Demonstration of the Program’s Functionality

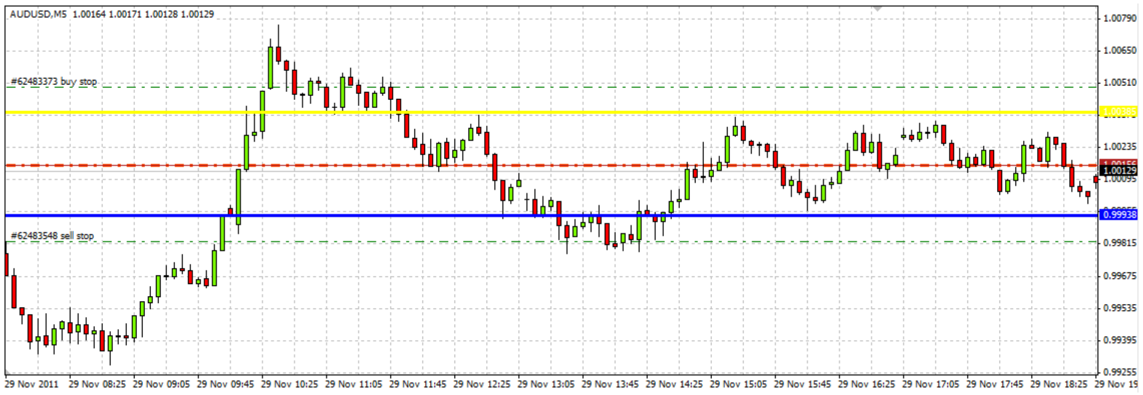


Figure 48: The support and resistance lines drawn automatically using our expert advisor.

Note the yellow line indicates a resistance line and the blue line a support level. One can see how the yellow line was placed due to catching so many peaks in its “bucket” described earlier, 4 on the graph. The same can be seen for the blue support line. The program updates with every tick so it is constantly drawing you the best 1, 2, or n number of support and resistance lines for that specific window of data, determined by the user.

Physical Code

```
//+------------------------------------------------------------------+
//|                                 Support and Resistance Lines.mq4 |
//|                                                          Group 5 |
//|                                        http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5”
#property link      "http://www.metaquotes.net"

#property indicator_chart_window
//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+

intminima_plot_index;
intmaxima_plot_index=0;

intinit()
  {


//---- indicators
//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator deinitialization function                       |
//+------------------------------------------------------------------+
intdeinit()
  {
//----
      // remove minima objects
      /*
      for(inti = 0; i<minima_plot_index; i++) {
         ObjectDelete("Minima" + i);
      }
      */

      // remove maxima objects

      for(int j =0; j <maxima_plot_index; j++){
         ObjectDelete("Maxima"+ j);
      }
//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator iteration function                              |
//+------------------------------------------------------------------+
int start()
  {

   intcounted_bars=IndicatorCounted();

   intbars_count   =WindowBarsPerChart();

   intsliding_window_size=bars_count/10;

   double min;
   double max;
   double range;

   intwindow_min_index=0, window_max_index=0;

   double minima[1000];
   double maxima[1000];
   doubleminima_copy[1000];
```

```
doublemaxima_copy[1000];

doubleminima_weight[1000];
doublemaxima_weight[1000];
doublesorted_maxima_weight[1000];
doublesorted_minima_weight[1000];

intminima_index=0;
intmaxima_index=0;

for(inti=0;i<bars_count-sliding_window_size;i++)
{
    min =1000000;
    max =-1000000;

    for(int j =0; j <sliding_window_size; j++)
    {
         if(min > Low[i+j]){
               min = Low[i+j];
               window_min_index= j;
         }

         if(max < High[i+j]){
               max = High[i+j];
               window_max_index= j;
         }
    }

    range = max - min;

    // if max and min are at boundaries, then we ignore them
```

```

      if((window_max_index!=sliding_window_size&&window_max_index!=0)&&((maxima_index==0)||(MathAbs(max - maxima[maxima_index-1]))>0.0001)){
            maxima[maxima_index]= High[i+window_max_index];
            maxima_weight[maxima_index]= range *5;
            maxima_index++;
          }

          if((window_min_index!=sliding_window_size&&window_min_index!=0)&&((minima_index==0)||(MathAbs(min - minima[minima_index-1]))>0.0001)){
            minima[minima_index]= min;
            minima_weight[minima_index]=10*MathSqrt(range);
            minima_index++;
          }

        }

        ArrayCopy(minima_copy, minima);
        ArrayCopy(maxima_copy, maxima);

        minima_plot_index=minima_index;

        /*
           for(int n = 0; n <minima_index; n++) {
              ObjectCreate("Minima" + n, OBJ_HLINE, 0, NULL, minima[n]);
              ObjectSet("Minima" + n, OBJPROP_COLOR, Blue);
              ObjectSet("Minima" + n, OBJPROP_STYLE, STYLE_SOLID);
              ObjectSet("Minima" + n, OBJPROP_WIDTH, 3);
           }
        */

        ArraySort(maxima, maxima_index, 0, MODE_ASCEND);
        ArraySort(minima, minima_index, 0, MODE_ASCEND);

        int w =0;

        for(w =0; w <maxima_index; w++){
           intsorted_maxima_index=ArrayBsearch(maxima, maxima_copy[w], maxima_index);
           sorted_maxima_weight[sorted_maxima_index]=maxima_weight[w];
        }

        for(w =0; w <minima_index; w++){
           intsorted_minima_index=ArrayBsearch(minima, minima_copy[w]);
           sorted_minima_weight[sorted_minima_index]=minima_weight[w];
        }



        doubleoutlier_adjusted_range=MathAbs(maxima[maxima_index-1]- minima[0]);

        intmaxima_plot_index=0;

        // if there exist maxima
        if(maxima_index>0)
        {
           // the index of the current maxima we are comparing against
           intcurrent_maxima_index=0;
           // the current maxima itself
           doublecurrent_maxima= maxima[0];
           // the average of the cluster of values
           doubleaverage_current_maxima=0;
           // the average weight of this cluster
```

```
doubleaverage_weight=0;

int k;

for(k =0; k <maxima_index; k++){
   Print("outlier_adjusted_range", "::", outlier_adjusted_range, "::", 10*(maxima[k]-current_maxima));
   if(10*MathAbs(maxima[k]-current_maxima)>outlier_adjusted_range){
      // determine whether values are significant and plot
      average_current_maxima=average_current_maxima/(k -current_maxima_index);
      // assume only one maxima with exact same value
      average_weight=average_weight/MathSqrt(MathSqrt(k -current_maxima_index));

      Print("average_weight: ", average_weight);
      if(average_weight>0.05){
            ObjectCreate("Maxima"+maxima_plot_index, OBJ_HLINE, 0, NULL, average_current_maxima);
            ObjectSet("Maxima"+maxima_plot_index, OBJPROP_COLOR, Yellow);
            ObjectSet("Maxima"+maxima_plot_index, OBJPROP_STYLE, STYLE_SOLID);
            ObjectSet("Maxima"+maxima_plot_index, OBJPROP_WIDTH, 3);
            maxima_plot_index++;
      }

      Print("Average weight ", average_weight);
      current_maxima= maxima[k];
      current_maxima_index= k;
      average_current_maxima= maxima[k];
   }else{
      // assume only one maxima with a specific value
      average_current_maxima=average_current_maxima+ maxima[k];
      average_weight=average_weight+sorted_maxima_weight[k];
```

```
                }
              }
            }

              // repeat code:
                    // determine whether values are significant and plot
                    Print("debug2 ", k, " ", current_maxima_index, " ", maxima_index);
                    average_current_maxima=average_current_maxima/(k -current_maxima_index);
                    // assume only one maxima with exact same value
                    average_weight=average_weight/(k -current_maxima_index);

                    Print("Data: ", average_current_maxima, ":", average_weight);

          return(0);
        }
```

B.2.7 Pivots

Similar to the Support Resistance program displayed earlier, the Pivots indicator also creates horizontal lines at key points in a chart. However, the support resistance program calculates those prices which are returned to several times whereas the pivot program is designed to create one line for each bucket. In this manner regardless of where the current price is this program can identify key points around that price. The program also displays 2 additional lines at the chart extrema displayed in a bold green line.

Additionally this program only considers the most recent bars whereas the support resistance program updates when the window is scrolled. It operates much in the same manner, looking at 6 bar increments for highs and lows and storing them. It continues in this manner until all bars in the window have been considered. The fact that this window does not update allows you to consider these same points as time progresses and was used heavily in the latter formulation of Brendan Hamm's trading strategy. The number of buckets to be looked at is user controlled, so if one feels that too many points are being considered, they need only reduce the buckets.

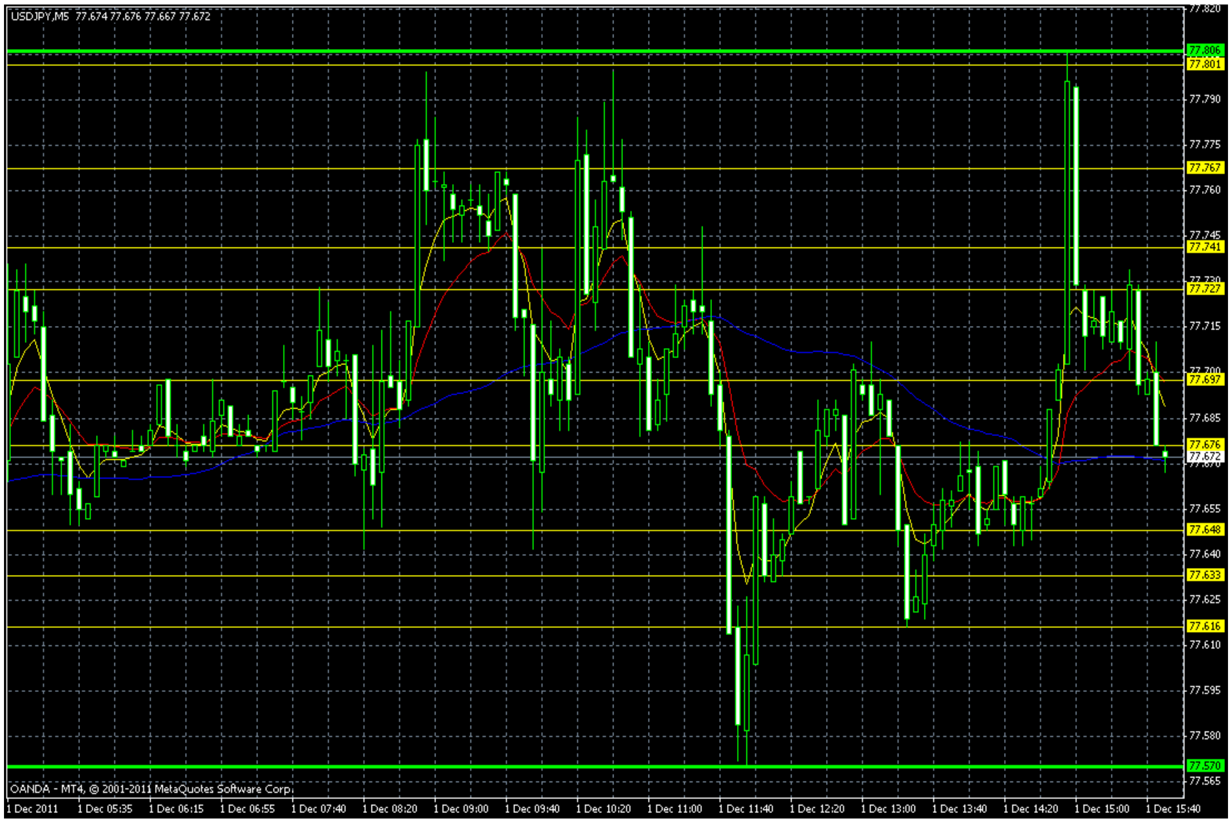


Figure 49: Pivots indicator set to find 10 pivot points (yellow lines). The extrema are shown as bolded green lines. In this instance the lowest point is the only point of interest for the bottom bucket, thus only 9 yellow lines appear.

Physical Code

```
//+------------------------------------------------------------------+
//|                                               Pivotsv1.07.mq4 |
//|                                                  Brendan Hamm |
//|                                       http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Brendan Hamm"
```

```
#property link      "http://www.metaquotes.net"

#property indicator_chart_window
//--- input parameters
externint       shift=6;
externint       num_of_buckets=10;
//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+
intinit()
  {
//---- indicators
//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator deinitialization function                       |
//+------------------------------------------------------------------+
intdeinit()
  {
//----

//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator iteration function                              |
//+------------------------------------------------------------------+
int start()
  {
   int    counted_bars=IndicatorCounted();
   int    pos=Bars-counted_bars;
   double high[];
   ArrayResize(high, shift);
   double low[];
   ArrayResize(low, shift);
   int    highpos,lowpos;
   int    bars_count=WindowBarsPerChart();
   int    first=bars_count-WindowFirstVisibleBar();

```

```
string hlinehigh, hlinelow;
string highs[];
ArrayResize(highs, bars_count/shift+1);
string lows[];
ArrayResize(lows, bars_count/shift+1);
string globalhigh="globalhigh";
string globallow  ="globallow";
doublemaxhigh, current1, current2;
doubleminlow=4;


for(int k=first; k<bars_count+first; k=k+shift)
   {
   for(inti=0;i<shift;i++)
      {
      high[i]=High[i+k];
      if(Low[i+k]!=0)
      {
      low[i]=Low[i+k];
      }
      }
      highpos=ArrayMaximum(high);
      lowpos=ArrayMinimum(low);
      hlinehigh="hlinehigh_"+(highpos+k);
      hlinelow="hlinelow_"+(lowpos+k);
      highs[k/shift]=hlinehigh;
      lows[k/shift]=hlinelow;
      ObjectCreate(hlinehigh,OBJ_HLINE,0,0,high[highpos]);
      ObjectCreate(hlinelow,OBJ_HLINE,0,0,low[lowpos]);
      ObjectSet(hlinelow, OBJPROP_COLOR, Blue);
   }
   minlow=ObjectGet(lows[0],1);
for(int h=0; h<bars_count/shift+1; h++)
   {
   current1=ObjectGet(highs[h],1);
   if(current1>maxhigh)maxhigh=current1;
   current2=ObjectGet(lows[h],1);

   if(current2<minlow&& current2!=0)  minlow=current2;
   }
```

```



      ObjectCreate(globalhigh,OBJ_HLINE,0,0,maxhigh);
      ObjectCreate(globallow,OBJ_HLINE,0,0,minlow);
      ObjectSet(globalhigh, OBJPROP_COLOR, Lime);
      ObjectSet(globallow, OBJPROP_COLOR, Lime);
      ObjectSet(globalhigh, OBJPROP_WIDTH, 3);
      ObjectSet(globallow, OBJPROP_WIDTH, 3);

      doublebucketsize=(maxhigh-minlow)/num_of_buckets;
      int count;
      doubleavgprices;
              string avg;

              for(int p=1;p<=num_of_buckets;p++)
                 {
                 avgprices=0;
                 count=0;
                 for(int q=0;q<bars_count/shift+1;q++)
                    {
                    current1=ObjectGet(highs[q],1);
                    current2=ObjectGet(lows[q],1);
                    if(current1<=minlow+(p*bucketsize)&& current1>minlow+((p-1)*bucketsize))
                    {
                    avgprices=avgprices+current1;
                    count=count+1;

                    }
                    if(current2<=minlow+(p*bucketsize)&& current2>minlow+((p-1)*bucketsize))
                    {
                    avgprices=avgprices+current2;
                    count=count+1;

                    }

                    }//end q loop
                    if(p==1)
                       {
                       avgprices=avgprices+minlow;
```

```
                         count=count+1;

                         }
                      if(avgprices>0)
                      {
                      avgprices=avgprices/count;
                      avg="avg_"+p;
                      ObjectCreate(avg,OBJ_HLINE,0,0,avgprices);
                      ObjectSet(avg, OBJPROP_COLOR, Yellow);
                      ObjectSet(avg, OBJPROP_WIDTH, 1);
                      }

                    }//end p loop
                      for(int r=0; r<bars_count/shift+1; r++)
                      {
                      ObjectDelete(highs[r]);
                      ObjectDelete(lows[r]);
                      }
                      WindowRedraw();


        //----

        //----
           return(0);
          }
        //+------------------------------------------------------------------+
```

B.2.8 Bars

After having experimented with several different forms of expert advisors we began to feel that the various variables which are held constant might better suit the intentions of a program if they were adjusted to the current market conditions. For instance, when determining the level at which to place a stop loss, a fixed value of 5 pips will not be a good value for a currency pair experiencing a period of high volatility, but may be suitable in a rather calm environment.   Thus to adjust such values to the current market conditions it can be very useful to have an idea of what the average bar size over a given period is. Thus, we have created a program which does exactly this and displays this information on the chart screen. While it may not be useful to most users who are manually trading, the functionality of this indicator can be combined with an expert advisor to enhance its capabilities.

Since this program was intended to be used for a myriad of purposes, we decided to calculate average bar sizes both from wick to wick, as well as from open to close. The open close measurements can be compared to the total bar size to give the user an idea of how much noise is being experienced during the time interval in question.

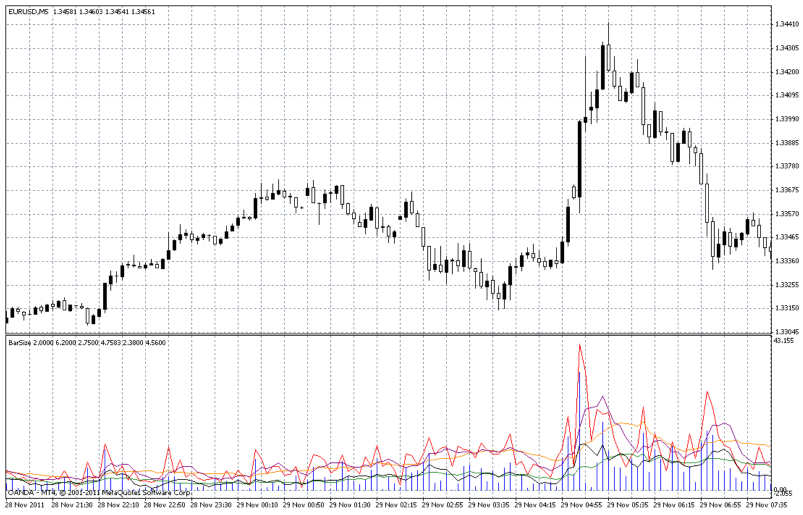

EURUSD,M5 1.34581 1.34603 1.34541 1.34561
BarSize 2.0000 6.2000 2.7500 4.7583 2.3800 4.5600
OANDA - MT4, © 2001-2011 MetaQuotes Software Corp.

Figure 50:The Bars Program is displayed in a separate window from the chart. The blue histogram displays the body size for each individual bar, while the red line displays the total bar size. The black and purple lines are for a 5-period moving average of the body and total size respectively. Similarly, the green and orange lines are for a 12-period moving average of the body and total size respectively.

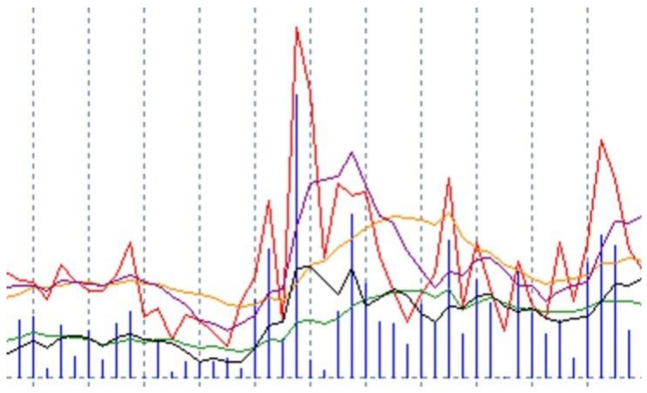


Figure 51: Zoom of the Bars Indicator using the same values as the previous figure.

Physical Code

```
//+------------------------------------------------------------------+
//|                                                      BarSize.mq4 |
//|                                                        Group5 |
//|                                         http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5"
#property link      "http://www.metaquotes.net"

#property indicator_separate_window
#property indicator_buffers 6
#property indicator_color1 Blue
#property indicator_color2 Red
#property indicator_color3 Chartreuse
#property indicator_color4 Orange
#property indicator_color5 Yellow
#property indicator_color6 Purple
//--- input parameters
externdouble N=12;
externdouble M=5;
//--- buffers
double ExtMapBuffer1[];
double ExtMapBuffer2[];
double ExtMapBuffer3[];
double ExtMapBuffer4[];
double ExtMapBuffer5[];
double ExtMapBuffer6[];

//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+
intinit()
  {
//---- indicators
   SetIndexStyle(0,DRAW_HISTOGRAM);
   SetIndexBuffer(0,ExtMapBuffer1);
   SetIndexStyle(1,DRAW_LINE);
   SetIndexBuffer(1,ExtMapBuffer2);
   SetIndexStyle(2,DRAW_LINE);
   SetIndexBuffer(2,ExtMapBuffer3);
   SetIndexStyle(3,DRAW_LINE);
   SetIndexBuffer(3,ExtMapBuffer4);
   SetIndexStyle(4,DRAW_LINE);
   SetIndexBuffer(4,ExtMapBuffer5);
   SetIndexStyle(5,DRAW_LINE);
   SetIndexBuffer(5,ExtMapBuffer6);
   stringshort_name="We got Bar Sizes!";
   //IndicatorShortName(short_name);
//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator deinitialization function                       |
//+------------------------------------------------------------------+
intdeinit()
  {
//----
```

```

//----
   return(0);
  }
//+------------------------------------------------------------------+
//| Custom indicator iteration function                              |
//+------------------------------------------------------------------+
int start()
  {
   int    counted_bars=IndicatorCounted();
   doubleBarSize;
   doubleMaxSize;
   doubleAvgMaxSize=0;
   doubleAvgBarSize=0;
   doubleOutputB;
   doubleOutputM;
   int j=0;//used to count to 12

//---- check for possible errors
```

```
   if(counted_bars<0)return(-1);
//---- last counted bar will be recounted
   if(counted_bars>0)counted_bars--;
//---- DEfine the position to look at
   intpos=Bars-counted_bars;

   while(pos>=0)
     {
     OutputB=0;
     OutputM=0;
     BarSize=MathAbs(Open[pos]-Close[pos])*10000;  //calculate the body size
     MaxSize=(High[pos]-Low[pos])*10000;  //Calculate the size from wick to wick

     ExtMapBuffer2[pos]=MaxSize;//send to buffer2 to plot
     ExtMapBuffer1[pos]=BarSize;//send it to buffer to plot

      for(int k=0; k<N; k++)
         {
         OutputB=OutputB+ExtMapBuffer1[pos+k];
```

```
          OutputM=OutputM+ExtMapBuffer2[pos+k];
          }
         ExtMapBuffer3[pos]=OutputB/N;
         ExtMapBuffer4[pos]=OutputM/N;
         OutputB=0;
                OutputM=0;
                  for(k=0; k<M; k++)
                 {
                 OutputB=OutputB+ExtMapBuffer1[pos+k];
                 OutputM=OutputM+ExtMapBuffer2[pos+k];
                 }
                ExtMapBuffer5[pos]=OutputB/M;
                ExtMapBuffer6[pos]=OutputM/M;


            pos--;
            }

      //----
      //----
         return(0);
        }
      //+------------------------------------------------------------------+
```

B.2.9 Order History Write

The program OrderHistoryWrite.mq4 was created to export trade data to a .csv file. This program is useful for when you need to export all data from your trades to create a spreadsheet of the progress of a portfolio. In the MT4 platform one can select the range of history they are interested in and run the script. As is the case with all scripts in MQL4it runs only once creating a file called tradetracker.csv located in ROOT/experts/files which contains the following information in each row: OrderOpenTime, Order Type, Order Lots, OrderOpenPrice, OrderStopLoss, OrderTakeProfit, OrderCloseTime, OrderClose Price, OrderPipProfitand,OrderProfit. The fields are defined as follows:

Order Open Time = yyyy.mm.dd hh.mm at which the transaction was executed (for some brokers this value may not be the same time zone as the users)

Order Type = buy/sell

Order Lots = integer value in standard lots

Order Open Price = Price at which the transaction was executed

Order Stop Loss = Last known value of stop loss applied to transaction (unfortunately any changes to a stop loss are notrecorded and thus unavailable)

Order Take Profit = Last known value of take profit level (also does not record changes during transaction)

Order Close Time = yyyy.mm.dd hh.mm at which the transaction was terminated (for some brokers this value may not be the same time zone as the users)

Order Close Price = Price at which transaction was terminated

Order Pip Profit = Number of total pips made, expressed as:

$$(\text{Order Open Price} - \text{Order Close Price}) * \text{Order Lots} * 1000$$

Order Profit = Total Profit of the trade (currently only available for USD based currencies)

For an example of the output, refer to any of the trading portfolio's for our group.

Physical Code

```
//+------------------------------------------------------------------+
//|                                           OrderHistoryWrite.mq4 |
//|                                                          Group 5|
//|        http://www.metaquotes.net |
//+------------------------------------------------------------------+
#property copyright "Group 5"
#property link      "http://www.metaquotes.net"


bool Comma = True;  // Some Spread Sheets want "," instead of "." for cent
string Doub2String (doubledoub, int digits)
```

```
   {
      string num, temp;
num=DoubleToStr(doub,digits);
      temp =num;
      if(Comma)
      {
doub=StringFind(num,".",0);

            if(doub!=1)
            {
               temp =StringSubstr(num,0,doub);
            }
      }
      return(temp);
   }
string Typ2String(int type)
   {
      string typ;
      if(type == OP_BUY)  typ="buy";
      if(type == OP_SELL)typ="sell";
      return(typ);
   }

voidWriteOrder(int handle)
   {
int type;
      doublepipprofit;
      type =OrderType();
      if(type==OP_BUY || type==OP_SELL)
         {
         if(type == OP_BUY)
            {
```

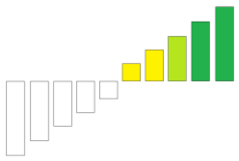
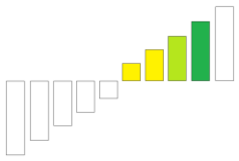
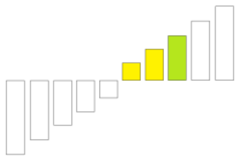
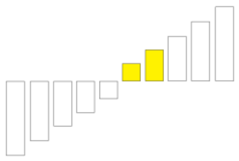
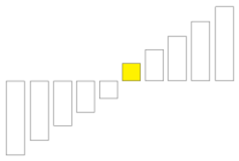
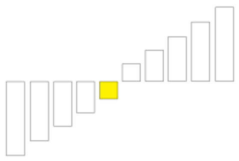
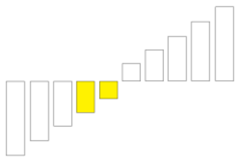
```
pipprofit=(OrderClosePrice()-OrderOpenPrice())*10000;
            }
         if(type == OP_SELL)
            {
pipprofit=(OrderOpenPrice()-OrderClosePrice())*10000;
            }
FileWrite(handle,TimeToStr(OrderOpenTime()), Typ2String(OrderType()),
                        Doub2String(OrderLots(),3), Doub2String(OrderOpenPrice(),5),
                        Doub2String(OrderStopLoss(),5),Doub2String(OrderTakeProfit(),5),
TimeToStr(OrderCloseTime()), Doub2String(OrderClosePrice(),5),
                        Doub2String(pipprofit,5), Doub2String(OrderProfit(),5));
         }
    }
//+------------------------------------------------------------------+
//| script program start function                                    |
//+------------------------------------------------------------------+
int start()
  {
//----
int handle, total, error;
total=OrdersHistoryTotal();
handle =FileOpen("TradeTracker.csv", FILE_CSV|FILE_WRITE,\',\');
FileWrite(handle, "OrderOpenTime", "OrderType", "OrderLots", "OrderOpenPrice",
"OrderStopLoss", "OrderTakeProfit", "OrderCloseTime", "OrderClosePrice", "OrderPipProfit",
"OrderProfit");
for(int count=0;count<total;count++)
{
   if(OrderSelect(count,SELECT_BY_POS, MODE_HISTORY)==true)
       {
WriteOrder(handle);
       }
       else
```

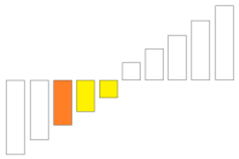
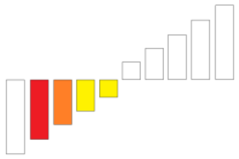
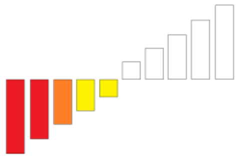
```
        {
        error=GetLastError();
        Print("Last Error was ",error);
        }
}
FileClose(handle);
//----
   return(0);
  }
//+------------------------------------------------------------------+
```

B.2.10 Currency Meter

General Overview

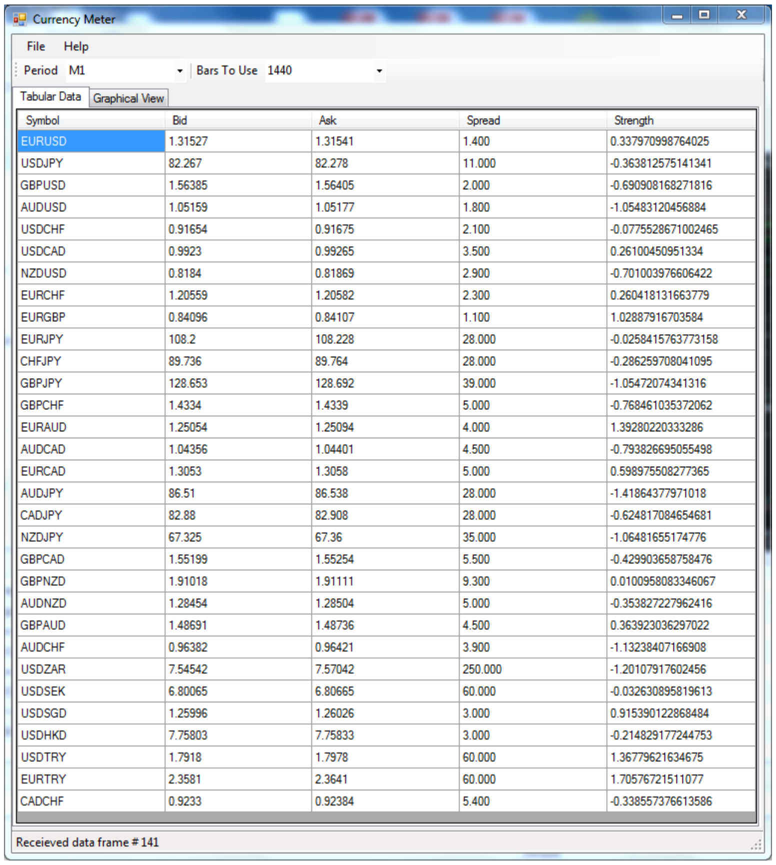
For this interactive qualifying project, our team has implemented a simple currency meter that determines and assigns strength to each currency using chart values for each currency and displays the calculated value graphically to the user. This software will help the user to decide which currency pair to trade as it makes sense to trade a strong currency against a weak currency. The user can specify approximately how many bars will be used in the calculation as well as the period of the bars (which can be 1 minute, 5 minutes, 15 minutes, etc.). To report the strength of the currency, the tool will calculate a numeric strength and quantize it an integer strength "index". Then a graphical icon denoting the strength index is displayed to the user in the form of a number of upward or downward facing bars, with upward facing bars indicating strength and downward facing bars indicating weakness.

Strength Description	Icon Description	Icon
Exceptionally strong	Five upward-facing bars	
Extremely Strong	Four upward-facing bars	
Very Strong	Three upward-facing bars	
Strong	Two upward-facing bars	
Slightly strong	One upward-facing bar	
Slightly weak	One downward-facing bars	
Weak	Two downward-facing bars	

Very weak	Three downward-facing bars	
Extremely weak	Four downward-facing bars	
Exceptionally weak	Five downward-facing bars	

The currency meter features tiles for all currencies that are supported by the individual broker and also features a display mode, in which all the currency pairs are shown which can be sorted by spread, bid price, ask price, and the spread. The currency meter shows the current state of connection with the MT4 client (i.e., whether it is connected or not and if it is connected, how many frames it has received).

Below is the screenshot of the software in tabular mode, where we can see each currency pair, the strength, buy and ask values, and the spread.



Period M1
Bars To Use 1440

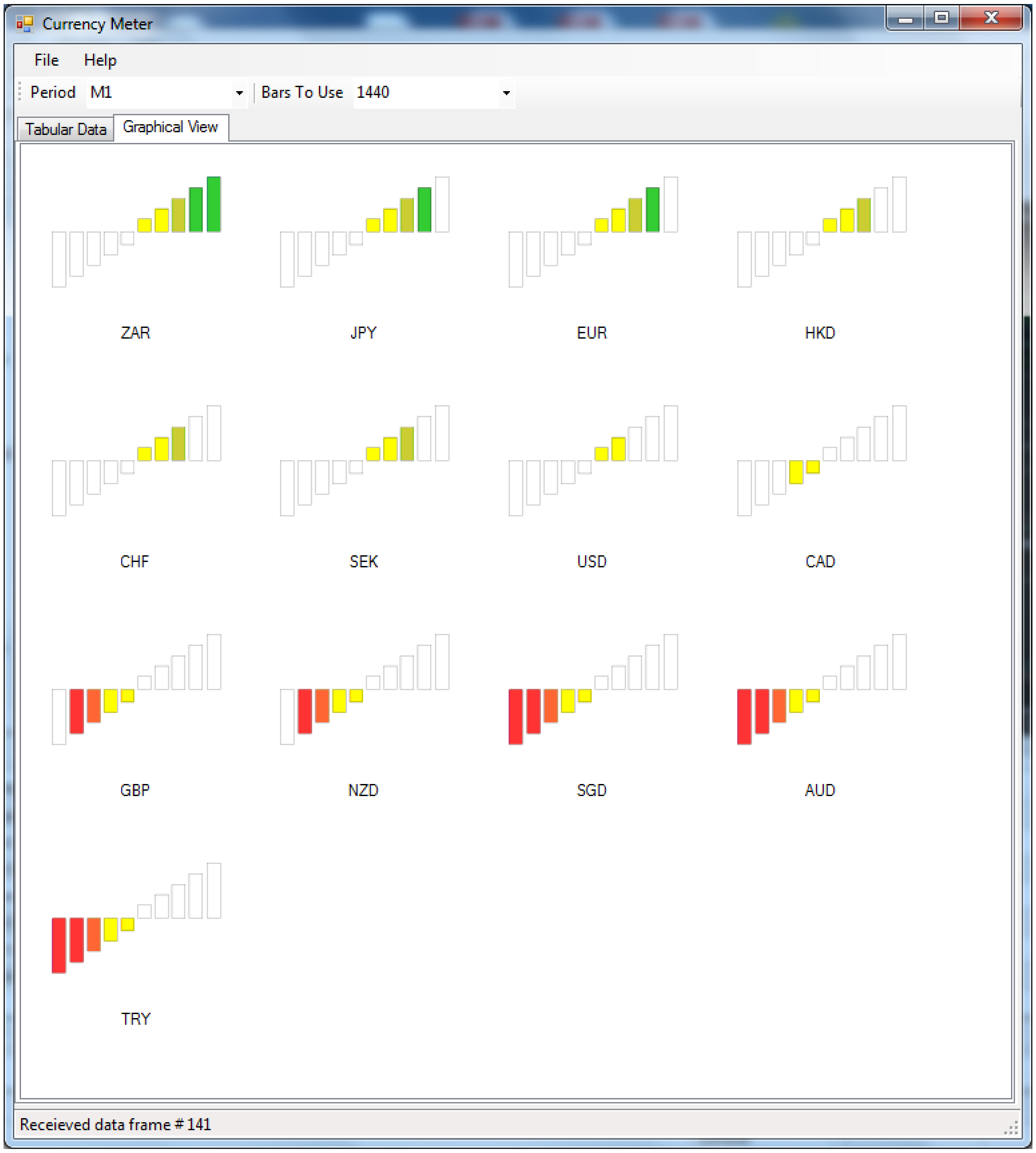
Symbol	Bid	Ask	Spread	Strength
EURUSD	1.31527	1.31541	1.400	0.337970998764025
USDJPY	82.267	82.278	11.000	-0.363812575141341
GBPUSD	1.56385	1.56405	2.000	-0.690908168271816
AUDUSD	1.05159	1.05177	1.800	-1.05483120456884
USDCHF	0.91654	0.91675	2.100	-0.0775528671002465
USDCAD	0.9923	0.99265	3.500	0.26100450951334
NZDUSD	0.8184	0.81869	2.900	-0.701003976606422
EURCHF	1.20559	1.20582	2.300	0.260418131663779
EURGBP	0.84096	0.84107	1.100	1.02887916703584
EURJPY	108.2	108.228	28.000	-0.0258415763773158
CHFJPY	89.736	89.764	28.000	-0.286259708041095
GBPJPY	128.653	128.692	39.000	-1.05472074341316
GBPCHF	1.4334	1.4339	5.000	-0.768461035372062
EURAUD	1.25054	1.25094	4.000	1.39280220333286
AUDCAD	1.04356	1.04401	4.500	-0.793826695055498
EURCAD	1.3053	1.3058	5.000	0.598975508277365
AUDJPY	86.51	86.538	28.000	-1.41864377971018
CADJPY	82.88	82.908	28.000	-0.624817084654681
NZDJPY	67.325	67.36	35.000	-1.06481655174776
GBPCAD	1.55199	1.55254	5.500	-0.429903658758476
GBPNZD	1.91018	1.91111	9.300	0.0100958083346067
AUDNZD	1.28454	1.28504	5.000	-0.353827227962416
GBPAUD	1.48691	1.48736	4.500	0.363923036297022
AUDCHF	0.96382	0.96421	3.900	-1.13238407166908
USDZAR	7.54542	7.57042	250.000	-1.20107917602456
USDSEK	6.80065	6.80665	60.000	-0.032630895819613
USDSGD	1.25996	1.26026	3.000	0.915390122868484
USDHKD	7.75803	7.75833	3.000	-0.214829177244753
USDTRY	1.7918	1.7978	60.000	1.36779621634675
EURTRY	2.3581	2.3641	60.000	1.70576721511077
CADCHF	0.9233	0.92384	5.400	-0.338557376613586

Receieved data frame # 141

Figure 52 - Currency Meter Running in Table Mode

Here, we have adjusted for the spread of currencies that include the Japanese Yen as well as rounding the spread values to the nearest third decimal place. The strength value is also shown and was used extensively for debugging purposes.

The software also runs in a graphical mode, in which we can see a strength indication for each currency. We can also change the number of bars used in the calculation as well as the period of the calculation (M1, M5, M15, H1, etc.). The software sends the new symbol period to the server and gets new base data, to which new bars are added.



Period M1

Bars To Use 1440

Receieved data frame # 141

Figure 53 - Currency Meter Running in Graphical View

Algorithmic Overview

The software uses a robust algorithm to determine the strength of each currency based on pairwise currency strength comparisons. We keep track of the last 1440 bars (which is equivalent to one day when each bar represents one minute). However, the user can select the number of bars that are actually used to determine the strength calculations that are presented to the user. When a new bar is added, an old bar is removed. This enables us to use a finite amount of memory and avoid leaks. The latest tick values are also included in the display interface though they are not stored due to memory constraints.

One of the core ideas behind this project is to define "strength". Currencies that are stronger tend to rise in terms of the other currencies. We determine a basic intrinsic "strength" of a currency as the weighted sum of its change over the entire period with respect to the US dollar. The weighting depends on how much data you wish to consider. To determine the exponential weighting, we determine the value α based on the bar N, where the Nth bar has ½ the weighting of the first bar. We select n as the number of bars the user wishes to use in the calculation. Mathematically, we would show α as follows.

$$\alpha = (0.5)^{1/N}$$

We then simulate exchange rates at the open and close times by dividing the target currency by another currency (this is repeated for all currencies). The close price is subtracted from the open price and divided by the close price quotient to normalize the values. Correlation is used to determine the similarity of movement of the open-to-close price and the high-to-low price for each currency; the two correlation values are averaged. The pairwise strength between currencies that are less correlated is weighted higher. We make the assumption that our process has an underlying normal distribution so we use Pearson product-moment correlation over every bar.

We determine a correlation matrix, which is symmetric due to the relation $\rho(x, y) = \rho(y, x)$. The correlation coefficient ρ can be calculated as below.

$$\rho(x, y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

We calculate the matrix of weightings P as follows, knowing that it is symmetric across the main diagonal, which are ones because each currency is perfectly correlated with itself. Perfect correlation has zero weighting as weighting is 1 minus the correlation value. We divide the raw value for a currency by the sum of the entries in its corresponding row.

$$P = \begin{bmatrix} 0 & \rho(X_1, X_2) & 1 - \rho(X_1, X_3) \\ 1 - \rho(X_2, X_1) & 0 & 1 - \rho(X_2, X_3) \\ 1 - \rho(X_3, X_1) & 1 - \rho(X_3, X_2) & 0 \end{bmatrix}$$

The intrinsic strength is calculated as the movement of every other currency with respect to the US dollar. Fortunately, every currency has a pair with the US dollar. If the other currency is the first in the pair with the dollar, then we invert the currency so that it is normalized to the dollar, which is the base currency. For the US dollar or a possibility of a currency without a pair against the US dollar, we normalize both currencies with respect to the Euro or Japanese Yen. We also normalize the currency with respect to itself in order to get a percentage gain so that weightage is based on the correlation alone. However, we use the alpha parameter to discriminate based on how long ago the bar occurred. The actual strength is calculated as follows:

$$final\_strength(curr1) = \sum_{curr2\, \epsilon\, CURRENCIES} (1 - \rho(\mathrm{curr1}, \mathrm{curr2})) * raw\_strength(curr1, curr2)$$

$$raw\_strength(curr1, curr2) = \sum_{n=0}^{N} \frac{\alpha^n * \left(\frac{open(curr1,n)}{open(curr2,n)} - \frac{close(curr1,n)}{close2(curr2,n)}\right)}{\frac{close(curr1,n)}{close2(curr2,n)}} / \sum_{n=0}^{N} \alpha^n$$

There is one caveat, which is to invert the currency if the US Dollar is the first term; for example, we would invert USDJPY to get the simulated value of JPYUSD. The resulting strengths are then quantized into icons displaying strength using quantization buckets that we determined by visualizing large sets of data.

To make the strengths more relevant, we can also add a weightage factor based on the trading period and integrating the auto-correlation$R_{x,y}(t,\tau)$, where x is the strength of the currency and y is the difference between open and close. This makes sense because if the currency strength as a whole is not correlated with the increase in the currency after τ units, then the statistic is less meaningful. If we know this information, we can also incorporate the spread into the calculation (its usefulness depends on trading duration).

Software Overview

The software is designed to operate with the MQL4 trading platform though it can be easily adapted to TradeStation or any other platform with C++ integration. Since we already figured out the task of C++/TradeStation integration, we could easily implement the client side module to interface with TradeStation, although for demonstration, the work that we have completed suffices.

The basic operation is that the trading software first sends the last 1440 bars of open, close, high, low, bid, and ask values for each currency pair over to the currency meter software. This is done using a named pipe, which is an inter-process communication mechanism in Windows. Each of these messages is sent in ASCII over a 64 kB frame. Then, for each bar and for each tick, new data is sent. If the data is sent for a bar, then it is added to the circular buffer and the first bar recorded is removed. If the data is sent for a tick, then the tick data is merely shown on the tabular display of the runtime.

The algorithm is calculated in each process and the tables as well as the graphical view are updated instantaneously to changes in the number of bars to use. The bar period is also incorporated into the system; this value is sent to MQL immediately after a new data frame has been received (for a tick or bar). If the period changed then the initialization data of the first bars is sent from MQL to the Currency Meter. There is code in the MQL4 end that sends new bars according to the period of the bar.

```cpp
// MatlabMql.cpp : Defines the exported functions for the DLL application.
//

#include "stdafx.h"
#include "GlobalPipeVars.h"
#include "PipeUtils.h"
#include <sstream>

using namespace std;

struct MqlStr {
        int     len;
        char    *string;
};

_DLLAPI int __stdcall initialSendDataToServer(int startIndex, int numBarsToDownload, int numSymbols, char* symbols, double* high, double* low, double* open, double* close);
_DLLAPI void __stdcall sendStringToServer(char* textToSend);
/*
_DLLAPI void __stdcall testSendDataToServer() {
        char** symbols = {"EURUSD", "AUDUSD", "CADUSD", "GBPUSD", "JPYUSD"};
        double transportCosts[25] = { 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5 };
    initialSendDataToServer(10, 5, 5, symbols, transportCosts, transportCosts, transportCosts, transportCosts);
        //sendStringToServer("BING");
}
*/

_DLLAPI int __stdcall initialSendDataToServer(int startIndex, int numBarsToDownload, int numSymbols, char* symbols, double* high, double* low, double* open, double* close) {
    DWORD cbWritten;

        stringstream ss (stringstream::in | stringstream::out);

        int code_1 = 100;
        ss << "                " << code_1 << " ";

        ss << startIndex << " ";
        ss << numBarsToDownload << " ";
        ss << numSymbols << " ";

        string str = symbols;
        ss << str << " ";

        for(int i = 0; i < numSymbols * numBarsToDownload; i++) {
                ss << high[i] << " ";
        }

        for(int i = 0; i < numSymbols * numBarsToDownload; i++) {
                ss << low[i] << " ";
        }

        for(int i = 0; i < numSymbols * numBarsToDownload; i++) {
                ss << open[i] << " ";
        }

        for(int i = 0; i < numSymbols * numBarsToDownload; i++) {
                ss << close[i] << " ";
        }

        string textToSendStr = ss.str();

        const char* textToSendArr = textToSendStr.c_str();
        int bytesTextToSend = textToSendStr.length();

    if (!WriteFile(
        hPipe,                          // Handle of the pipe
        textToSendArr,                  // Message to be written
        bytesTextToSend + 1,            // Number of bytes to write
        &cbWritten,                     // Number of bytes written
        NULL                            // Not overlapped
        ))
    {
        dwError = GetLastError();
        cleanUp();
                return 100;
    }

        return 90;
```

```cpp
}

_DLLAPI int __stdcall sendNewBarToServer(int numSymbols, double* high, double* low, double* open, double* close, double* currentBid, double* currentAsk) {
    DWORD cbWritten;

        stringstream ss (stringstream::in | stringstream::out);
        int code_1 = 1000;

        ss << "                " << code_1 << " ";

        ss << numSymbols << " ";

        for(int i = 0; i < numSymbols; i++) {
                ss << high[i] << " ";
        }

        for(int i = 0; i < numSymbols; i++) {
                ss << low[i] << " ";
        }

        for(int i = 0; i < numSymbols; i++) {
                ss << open[i] << " ";
        }

                for(int i = 0; i < numSymbols; i++) {
                        ss << close[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentBid[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentAsk[i] << " ";
                }

                string textToSendStr = ss.str();
                const char* textToSendArr = ss.str().c_str();
                int bytesTextToSend = textToSendStr.length();

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                textToSendArr,              // Message to be written
                bytesTextToSend + 1,        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
                        return 1;
            }

                return 0;
        }


        _DLLAPI int __stdcall sendNewTickToServer(int numSymbols, double* high, double* low, double* open, double* close, double* currentBid, double* currentAsk) {
            DWORD cbWritten;

                stringstream ss (stringstream::in | stringstream::out);
                int code_1 = 10000;

                ss << "                " << code_1 << " ";

                ss << numSymbols << " ";

                for(int i = 0; i < numSymbols; i++) {
                        ss << high[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << low[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << open[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << close[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentBid[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentAsk[i] << " ";
                }

                string textToSendStr = ss.str();
                const char* textToSendArr = ss.str().c_str();
                int bytesTextToSend = textToSendStr.length();

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                textToSendArr,              // Message to be written
                bytesTextToSend + 1,        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
                        return 1;
            }

                return 0;
        }

        _DLLAPI int __stdcall sendDataToServer(int numSymbols, int tickNumber, int barNumber, double* high, double* low, double* open, double* close, double* currentBid, double* currentAsk) {
            DWORD cbWritten;

                stringstream ss (stringstream::in | stringstream::out);

                int code_1 = 0x00000000;
                ss << code_1 << " ";
                ss << numSymbols << " ";
                ss << tickNumber << " ";
                ss << barNumber << " ";

                for(int i = 0; i < numSymbols; i++) {
                        ss << high[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << low[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << close[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentBid[i] << " ";
                }

                for(int i = 0; i < numSymbols; i++) {
                        ss << currentAsk[i] << " ";
                }

                string textToSendStr = ss.str();
                const char* textToSendArr = ss.str().c_str();
                int bytesTextToSend = textToSendStr.length();

            if (!WriteFile(
                hPipe,                          // Handle of the pipe
                textToSendArr,                  // Message to be written
                bytesTextToSend + 1,            // Number of bytes to write
                &cbWritten,                     // Number of bytes written
                NULL                            // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
```

```
                    return 90;
            }

                return 100;
        }

        _DLLAPI void __stdcall sendStringVectorToServer(char** textToSend, int numStrings) {
            DWORD cbWritten;

                char* last_str = textToSend[numStrings - 1];
                int lengthOfFinalStr = strlen(last_str);
                int bytesTextToSend = (&(textToSend[numStrings - 1])) - textToSend + 1;

            if (!WriteFile(
                hPipe,                          // Handle of the pipe
                textToSend,                     // Message to be written
                bytesTextToSend,                // Number of bytes to write
                &cbWritten,                     // Number of bytes written
                NULL                            // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
            }
        }

        _DLLAPI void __stdcall sendStringToServer(char* textToSend) {
            DWORD cbWritten;
            size_t bytesTextToSend = sizeof(textToSend);

            if (!WriteFile(
                hPipe,                          // Handle of the pipe
                textToSend,                     // Message to be written
                bytesTextToSend,                // Number of bytes to write
                &cbWritten,                     // Number of bytes written
                NULL                            // Not overlapped
                ))
            {
                dwError = GetLastError();

```

```
                    std::ofstream m;
                    m.open ("C:\\\\Users\\\\Srinivas Vasudevan\\\\example.txt");
                    m << dwError;
                    m.close();

                cleanUp();
            }

            std::ofstream m;
            m.open ("C:\\\\Users\\\\Srinivas Vasudevan\\\\example.txt");
            m << "success";
            m.close();
        }

        _DLLAPI void _stdcall sendDoubleVectorToServer(double* vector, int size) {
            DWORD cbWritten;
            size_t bytesTextToSend = sizeof(double) * size;

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                vector,                                         // Message to be written
                bytesTextToSend,                        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
            }
        }

        _DLLAPI void _stdcall sendDoubleToServer(double value) {
            DWORD cbWritten;
            size_t bytesTextToSend = sizeof(int);

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                &value,                                         // Message to be written
                bytesTextToSend,                        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
            }
        }

        _DLLAPI void _stdcall sendIntVectorToServer(int* vector, int size) {
            DWORD cbWritten;
            size_t bytesTextToSend = sizeof(int) * size;

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                vector,                                         // Message to be written
                bytesTextToSend,                        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
            }
        }

        _DLLAPI void _stdcall sendIntToServer(int value) {
            DWORD cbWritten;
            size_t bytesTextToSend = sizeof(int);

            if (!WriteFile(
                hPipe,                      // Handle of the pipe
                &value,                                         // Message to be written
                bytesTextToSend,                        // Number of bytes to write
                &cbWritten,                 // Number of bytes written
                NULL                        // Not overlapped
                ))
            {
                dwError = GetLastError();
                cleanUp();
            }
        }
```

```
// dllmain.cpp : Defines the entry point for the DLL application.
#include "stdafx.h"
#include "PipeUtils.h"
#include "GlobalPipeVars.h"

bool initialize();

BOOL APIENTRY DllMain( HMODULE hModule,
                       DWORD  ul_reason_for_call,
                       LPVOID lpReserved
                                         )
{
        switch (ul_reason_for_call)
        {
        case DLL_PROCESS_ATTACH:
                return initialize();
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
                break;
        }

        return true;
}

bool initialize() {
        //
        // Send a request from client to server
        //
        while (TRUE)
        {
                hPipe = CreateFile(
                        FULL_PIPE_NAME,                  // Pipe name
                        GENERIC_READ | GENERIC_WRITE,    // Read and write access
                        0,                               // No sharing
                        NULL,                            // Default security attributes
                        OPEN_EXISTING,                   // Opens existing pipe
                        0,                               // Default attributes
                        NULL                             // No template file
                        );

                // If the pipe handle is opened successfully ...
                if (hPipe != INVALID_HANDLE_VALUE)
                {
                        wprintf(L"The named pipe (%s) is connected.\\n", FULL_PIPE_NAME);
                        break;
                }

                dwError = GetLastError();

                // Exit if an error other than ERROR_PIPE_BUSY occurs.
                if (ERROR_PIPE_BUSY != dwError)
                {
                        wprintf(L"Unable to open named pipe w/err 0x%08lx\\n", dwError);
                        goto Cleanup;
                }

                // All pipe instances are busy, so wait for 5 seconds.
                if (!WaitNamedPipe(FULL_PIPE_NAME, 5000))
                {
                        dwError = GetLastError();
                        wprintf(L"Could not open pipe: 5 second wait timed out.");
                        goto Cleanup;
                }
        }

        return true;

        Cleanup:

        // Centralized cleanup for all allocated resources.
        if (hPipe != INVALID_HANDLE_VALUE)
        {
                CloseHandle(hPipe);
                hPipe = INVALID_HANDLE_VALUE;
        }

        return false;
}
```

```
#include "stdafx.h"
#include "mql4ipc.h"
#include "Form1.h"
#include "globals.h"

/****************************** Module Header ******************************\\
* Module Name:  CppNamedPipeServer.cpp
* Project:      CppNamedPipeServer
* Copyright (c) Microsoft Corporation.
*
* Named pipe is a mechanism for one-way or duplex inter-process communication
* between the pipe server and one or more pipe clients in the local machine
* or across the computers in the intranet:
*
* PIPE_ACCESS_INBOUND:
* Client (GENERIC_WRITE) ---> Server (GENERIC_READ)
*
* PIPE_ACCESS_OUTBOUND:
* Client (GENERIC_READ) <--- Server (GENERIC_WRITE)
*
* PIPE_ACCESS_DUPLEX:
* Client (GENERIC_READ or GENERIC_WRITE, or both) <-->
* Server (GENERIC_READ and GENERIC_WRITE)
*
* This code sample demonstrates calling CreateNamedPipe to create a pipe
* named "\\\\.\\pipe\\SamplePipe", which supports duplex connections so that both
* client and server can read from and write to the pipe. The security
* attributes of the pipe are customized to allow Authenticated Users read and
* write access to a pipe, and to allow the Administrators group full access
* to the pipe. When the pipe is connected by a client, the server attempts to
* read the client\'s message from the pipe by calling ReadFile, and write a
* response by calling WriteFile.
*
* This source is subject to the Microsoft Public License.
* See http://www.microsoft.com/opensource/licenses.mspx#Ms-PL.
* All other rights reserved.
*
* THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
* EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
\\***************************************************************************/

void cleanup(HANDLE hNamedPipe);
void calculate_strength(int n);

using namespace std;

vector<boost::circular_buffer<double> *> high;
vector<boost::circular_buffer<double> *> low;
vector<boost::circular_buffer<double> *> open;
vector<boost::circular_buffer<double> *> close;
vector<boost::circular_buffer<double> *> bid_bar;
vector<boost::circular_buffer<double> *> ask_bar;

string strarray2[100];

double latest_high[100];
double latest_low[100];
double latest_open[100];
double latest_close[100];
double latest_bid[100];
double latest_ask[100];

// used for computation
double data1[NUM_BARS_TO_USE];
double data2[NUM_BARS_TO_USE];
double data3[NUM_BARS_TO_USE];
double data4[NUM_BARS_TO_USE];

double processed_strength_vector[MAX_NUM_SYMBOLS];

```

```
std::map<std::string, std::vector<std::string>*> currency_pair_map;
std::set<std::string> currencies;
std::map<std::string, int> currency_pair_index_map;
std::map<std::string, double> processed_strength_vector_map;

vector<string> symbols_vector;

PCURRENCYDATA pData;
DWORD   dwThread;
HANDLE  hThread;

string textToUse = "Windows";

int code = 0, startIndex = 0, numBarsToDownload = 0, numSymbols = 0;

DWORD WINAPI getDataFromMql(LPVOID lpParam)
{
    DWORD dwError = ERROR_SUCCESS;
    HANDLE hNamedPipe = INVALID_HANDLE_VALUE;

        // Initialize array of Global strarray
        Globals::strarray = gcnew array<System::String^>(100);

    // Create the named pipe.
    hNamedPipe = CreateNamedPipe(
        FULL_PIPE_NAME,             // Pipe name.
        PIPE_ACCESS_DUPLEX,         // The pipe is duplex; both server and
                                    // client processes can read from and
                                            // write to the pipe
                PIPE_TYPE_MESSAGE |         // Message type pipe
                PIPE_READMODE_MESSAGE |     // Message-read mode
                PIPE_WAIT,                  // Blocking mode is enabled
                PIPE_UNLIMITED_INSTANCES,   // Max. instances
                BUFFER_SIZE,                // Output buffer size in bytes
                BUFFER_SIZE,                // Input buffer size in bytes
                NMPWAIT_USE_DEFAULT_WAIT,   // Time-out interval
                NULL                        // Security attributes
                );

            if (hNamedPipe == INVALID_HANDLE_VALUE)
            {
                dwError = GetLastError();
                wprintf(L"Unable to create named pipe w/err 0x%08lx\\n", dwError);
                cleanup(hNamedPipe);
                        return dwError;
            }

            wprintf(L"The named pipe (%s) is created.\\n", FULL_PIPE_NAME);

            // Wait for the client to connect.
            wprintf(L"Waiting for the client\'s connection...\\n");
            if (!ConnectNamedPipe(hNamedPipe, NULL))
            {
                if (ERROR_PIPE_CONNECTED != GetLastError())
                {
                    dwError = GetLastError();
                    wprintf(L"ConnectNamedPipe failed w/err 0x%08lx\\n", dwError);
                                cleanup(hNamedPipe);
                                return dwError;
                }
            }

                //
                // Let the system know that the client has been connected
                //
                Globals::textToUse = gcnew System::String("Client is connected.");
                Globals::form->Invoke(gcnew
System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form,
&CurrencyMeter::Form1::UpdateStatusStrip));

                // we initialize the number of messages to zero
                int numMessages = 0;

            //
            // Receive a request from client.
            //
                while(true) {
                        BOOL fFinishRead = FALSE;

                        double temp = 0;

                        string symbols = ""; // initialize to ""

                        char chRequest[BUFFER_SIZE];
                        DWORD cbRequest, cbRead;
                        cbRequest = sizeof(chRequest);
                        memset (chRequest, 0, BUFFER_SIZE);

                        // read from the input
                        fFinishRead = ReadFile(
                                hNamedPipe,     // Handle of the pipe
                                chRequest,      // Buffer to receive data
                                cbRequest,      // Size of buffer in bytes
                                &cbRead,        // Number of bytes read
                                NULL            // Not overlapped I/O
                        );

                        // check if there are any errors
                        if (!fFinishRead && (ERROR_MORE_DATA != GetLastError()))
                        {
                                dwError = GetLastError();
                                printf("Receive %ld bytes from client: \\"\\"\\n", cbRead);
                                wprintf(L"ReadFile from pipe failed w/err 0x%08lx %d\\n", dwError, cbRead);
                                cleanup(hNamedPipe);
                                return dwError;
                        }

                        printf("Receive %ld bytes from client: \\"%s\\"\\n", cbRead, chRequest);


                        // if the number of received bytes is greater than one
                        if(cbRequest > 0) {
                                stringstream s (stringstream::in | stringstream::out);
                                s << chRequest;

                                // WARNING: KLUDGE ignore first 20 characters
                                s.ignore(20, \' \');

                                // get these constants from the frame
```

```
s >> code;

if(code == 100) {
    s >> startIndex;
    s >> numBarsToDownload;
    s >> numSymbols;
    s >> symbols;

    // export data into the trade evaluation engine
    if(Globals::numSymbols == 0) {
        Globals::numSymbols = numSymbols;
        for(int i = 0; i < numSymbols; i++) {
            string str1 = symbols.substr(i*6, 3);
            string str2 = symbols.substr(i*6 + 3, 3);
            Globals::strarray[i] = gcnew System::String(symbols.substr(i * 6, 6).c_str());
            strarray2[i] = symbols.substr(i * 6, 6);
            currency_pair_index_map[symbols.substr(i * 6, 6)] = i;

            if(currency_pair_map.find(str1) == currency_pair_map.end()) {
                currency_pair_map[str1] = new vector<string>();
            }

            currency_pair_map[str1]->push_back(str2);

            if(currency_pair_map.find(str2) == currency_pair_map.end()) {
                currency_pair_map[str2] = new vector<string>();
            }

            currency_pair_map[str2]->push_back(str1);

```

```
                }

                // get the list of all the currencies

                for (std::map<std::string, std::vector<std::string>*>::iterator it = currency_pair_map.begin(); it != currency_pair_map.end(); it++) {
                    currencies.insert((*it).first);
                }
                Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::AddRowsToDataGridView));
                Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::AddSymbolsToListView));
            }

            // if the vectors are uninitialized
            if(high.size() == 0) {
                for(int i = 0; i < numSymbols + 1; i++) {
                    high.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                    low.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                    open.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                    close.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                    bid_bar.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                    ask_bar.push_back(new boost::circular_buffer<double>(NUM_BARS_TO_USE));
                }

                for(int i = 0; i < NUM_BARS_TO_USE; i++) {
                    high[numSymbols]->push_front(1.00);
                    low[numSymbols]->push_front(1.00);
                    open[numSymbols]->push_front(1.00);
                    close[numSymbols]->push_front(1.00);
                    high[numSymbols]->push_front(1.00);
```

```
                        low[numSymbols]->push_front(1.00);
                    }
                }

                for(int i = 0; i < numSymbols; i++) {
                    for(int j = startIndex; j < startIndex + numBarsToDownload; j++) {
                        s >> temp;
                        high[i]->push_front(temp);
                    }
                }

                for(int i = 0; i < numSymbols; i++) {
                    for(int j = startIndex; j < startIndex + numBarsToDownload; j++) {
                        s >> temp;
                        low[i]->push_front(temp);
                    }
                }

                for(int i = 0; i < numSymbols; i++) {
                    for(int j = startIndex; j < startIndex + numBarsToDownload; j++) {
                        s >> temp;
                        open[i]->push_front(temp);
                    }
                }

                for(int i = 0; i < numSymbols; i++) {
                    for(int j = startIndex; j < startIndex + numBarsToDownload; j++) {
                        s >> temp;
                        close[i]->push_front(temp);
                    }
                }

            } else if (code == 1000) {
                s >> numSymbols;

                for(int i = 0; i < numSymbols; i++) {
```

```
        s >> temp;
        high[i]->push_front(temp);
    }

    for(int i = 0; i < numSymbols; i++) {
        s >> temp;
        low[i]->push_front(temp);
    }

    for(int i = 0; i < numSymbols; i++) {
        s >> temp;
        open[i]->push_front(temp);
    }

    for(int i = 0; i < numSymbols; i++) {
        s >> temp;
        close[i]->push_front(temp);
    }

    for(int i = 0; i < numSymbols; i++) {
        s >> temp;
        bid_bar[i]->push_front(temp);
    }

    for(int i = 0; i < numSymbols; i++) {
        s >> temp;
        ask_bar[i]->push_front(temp);
    }

    calculate_strength(120);

    Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::UpdateDataInTable));
    Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::UpdateListView));
} else if(code == 10000) {
    s >> numSymbols;

```

```
                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_high[i] = temp;
                    }

                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_low[i] = temp;
                    }

                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_open[i] = temp;
                    }

                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_close[i] = temp;
                    }

                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_bid[i] = temp;
                    }

                    for(int i = 0; i < numSymbols; i++) {
                        s >> temp;
                        latest_ask[i] = temp;
                    }

                    calculate_strength(120);

                    Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::UpdateDataForTickInTable));
                    Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::UpdateListView));
                }

                numMessages++;
                Globals::textToUse = gcnew System::String("Receieved data frame # " + numMessages);
                Globals::form->Invoke(gcnew System::Windows::Forms::MethodInvoker((CurrencyMeter::Form1 ^)Globals::form, &CurrencyMeter::Form1::UpdateStatusStrip));
            }
        }

        DisconnectNamedPipe(hNamedPipe);

        getchar();

        return 0;

    }

    template <class T>double mean(int n, T *data)
    {
        double m=0;
        int i=n/8;
        while (i>0)
        {
            m += data[0];
            m += data[1];
            m += data[2];
            m += data[3];
            m += data[4];
            m += data[5];
            m += data[6];
            m += data[7];
            data += 8;
            i--;
        }

        switch (n%8)
        {
        case 7: m+=data[6];
        case 6: m+=data[5];
        case 5: m+=data[4];
```

```
			case 4: m+=data[3];
			case 3: m+=data[2];
			case 2: m+=data[1];
			case 1: m+=data[0];
			}

			return m/n;
	}

	template <class T>double pearson(int N, T *data, T *data1)
	{
			double med = mean<T>(N, data),
				med1 = mean<T>(N, data1),
				rez = 0, rez1 = 0, rez2 = 0;

			for(int i=0; i<N; i++)
			{
				rez+=(data[i]-med)*(data1[i]-med1);
				rez1+=(data[i]-med)*(data[i]-med);
				rez2+=(data1[i]-med1)*(data1[i]-med1);
			}

			return rez/sqrt(rez1*rez2);
	}

	double calculate_covariance_of_price_changes(string currency1, string currency2) {
			bool invert_currency1 = true;
			bool invert_currency2 = true;

			double mean1 = 0;
			double mean2 = 0;
			double covariance = 0;

			int ind1;
			int ind2;

			std::map<std::string, int>::iterator it1;
			std::map<std::string, int>::iterator it2;

			// check whether or not to invert
```

```
                if(currency1 != "USD") {
                        if((it1 = currency_pair_index_map.find(currency1 + "USD")) != currency_pair_index_map.end()) {
                                invert_currency1 = false;
                                ind1 = (*it1).second;
                        } else {
                                invert_currency1 = true;
                                it1 = currency_pair_index_map.find("USD" + currency1);
                                ind1 = (*it1).second;
                        }
                } else {
                        ind1 = numSymbols;
                }

                if(currency2 != "USD") {
                        if((it2 = currency_pair_index_map.find(currency2 + "USD")) != currency_pair_index_map.end()) {
                                invert_currency2 = false;
                                ind2 = (*it2).second;
                        } else {
                                invert_currency2 = true;
                                it2 = currency_pair_index_map.find("USD" + currency2);
                                ind2 = (*it2).second;
                        }
                } else {
                        ind2 = numSymbols;
                }

                for(int i = 0; i < NUM_BARS_TO_USE; i++) {
                        data1[i] = invert_currency1 ? (close[ind1]->at(i) - open[ind1]->at(i)) : (1 / (close[ind1]->at(i))) - (1 / (open[ind1]->at(i)));
                        data1[i] = (data1[i] == data1[i]) ? data1[i] : 0;
                        data2[i] = invert_currency2 ? (close[ind2]->at(i) - open[ind2]->at(i) ) : (1 / (close[ind2]->at(i))) - (1 / (open[ind2]->at(i)));
                        data2[i] = (data2[i] == data2[i]) ? data2[i] : 0;
                        data3[i] = invert_currency1 ? (high[ind1]->at(i) - low[ind1]->at(i)) : (1 / (high[ind1]->at(i))) - (1 / (low[ind1]->at(i)));
                        data3[i] = (data3[i] == data3[i]) ? data3[i] : 0;
                        data4[i] = invert_currency2 ? (high[ind2]->at(i) - low[ind2]->at(i)) : (1 / (high[ind2]->at(i))) - (1 / (low[ind2]->at(i)));
                        data4[i] = (data4[i] == data4[i]) ? data4[i] : 0;
                }

                double output = (pearson(NUM_BARS_TO_USE, data1, data2) + pearson(NUM_BARS_TO_USE, data3, data4)) / 2;

                // if value is nan, return 0
                if(output != output) {
                        return 0;
                } else {
                        return output;
                }
        }

        void calculate_strength(int n) {
                int numCurrencies = currencies.size();
                int i = 0; int j = 0;
                double covariance_matrix[MAX_NUM_SYMBOLS][MAX_NUM_SYMBOLS];
                double raw_strength_vector[MAX_NUM_SYMBOLS];
                double alpha = pow(0.5, 1.0/double(n));

                double alpha_sum = 0;

                for(int i = 0; i < MAX_NUM_SYMBOLS; i++) {
                        alpha_sum = alpha_sum + pow(alpha, i);
                }

                for(set<string>::iterator it1 = currencies.begin(); it1 != currencies.end(); it1++) {
                        i = 0;
                        for(set<string>::iterator it2 = currencies.begin(); it2 != currencies.end(); it2++) {
                                if(*it1 == "USD" && *it2 == "USD") {
                                        covariance_matrix[i][j] = 1;
                                } else if(*it1 == "USD" || *it2 == "USD") {
                                        covariance_matrix[i][j] = 0;
                                } else {
                                        covariance_matrix[i][j] = calculate_covariance_of_price_changes(*it1, *it2);
                                }
                                // i must always be incremented
                                i++;
                        }

                        // raw_strength vector starts at 0 and is incremented
                        raw_strength_vector[j] = 0;

                        // run code to calculate strength vector
                        // use alpha as exponential weighting factor
                        for(int k = 0; k < NUM_BARS_TO_USE; k++) {
                                raw_strength_vector[j] = raw_strength_vector[j] + data1[k] *
pow(alpha, k);
                        }

                        // divide by weighting factor
                        raw_strength_vector[j] = raw_strength_vector[j] / alpha_sum;

                        // j must always be incremented
                        j++;
                }

                double covariance_sum ;
                double current_strength;

                for(int i = 0; i < numCurrencies; i++) {
                        current_strength = 0;
                        covariance_sum = 0;
                        for (int j = 0; j < numCurrencies; j++) {
                                current_strength = current_strength + raw_strength_vector[j] *
covariance_matrix[i][j];
                                covariance_sum = covariance_sum + covariance_matrix[i][j];
                        }
                        processed_strength_vector[i] = current_strength * 10000 /
covariance_sum;
                }

                i = 0; // outer counter loop
                for(set<string>::iterator it2 = currencies.begin(); it2 != currencies.end();
it2++) {
```

```
                        processed_strength_vector_map[*it2] = processed_strength_vector[i];
                        i = i++;
                }
        }

        void cleanup(HANDLE hNamedPipe) {
            if (hNamedPipe != INVALID_HANDLE_VALUE)
            {
                CloseHandle(hNamedPipe);
                hNamedPipe = INVALID_HANDLE_VALUE;
            }
        }
```

```
#pragma once

#include "mql4ipc.h"
#include "globals.h"

namespace CurrencyMeter {

        using namespace System;
        using namespace System::ComponentModel;
        using namespace System::Collections;
        using namespace System::Windows::Forms;
        using namespace System::Data;
        using namespace System::Drawing;

        /// <summary>
        /// Summary for Form1
        /// </summary>
        public ref class Form1 : public System::Windows::Forms::Form
        {
```

```
	public:
		Form1(void)
		{
			InitializeComponent();
		}

	protected:
		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		~Form1()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::TabControl^  TabularView;
	protected:

	private: cli::array<System::Windows::Forms::ListViewItem^>^ listViewItemList;
	private: System::Windows::Forms::ImageList^ largeImageList;
	private: System::Windows::Forms::TabPage^  DataTab;
	private: System::Windows::Forms::TabPage^  GraphicalTab;
	protected:

	protected:

	protected:

	protected:


	private: System::Windows::Forms::StatusStrip^  StatusStrip;

	private: System::Windows::Forms::DataGridView^  dataGridView1;
	private: System::Windows::Forms::DataGridViewTextBoxColumn^  Symbol;
	private: System::Windows::Forms::DataGridViewTextBoxColumn^  Bid;
	private: System::Windows::Forms::DataGridViewTextBoxColumn^  Ask;
	private: System::Windows::Forms::DataGridViewTextBoxColumn^  Spread;
```

```

        private: System::Windows::Forms::ToolStripStatusLabel^  LoadingStatus;
        private: System::Windows::Forms::DataGridViewTextBoxColumn^  Strength;
        private: System::Windows::Forms::ListView^  listView1;


        private:
        // Implements the manual sorting of items by columns.
        ref class ListViewItemComparer: public IComparer
        {
        private:
           int col;

        public:
           ListViewItemComparer()
           {
                  col = 0;
           }

           ListViewItemComparer( int column )
           {
                  col = column;
           }

           virtual int Compare( Object^ x, Object^ y )
           {
                  return (Double::Parse((dynamic_cast<ListViewItem^>(y))->SubItems[ col ]-
>Text) - (Double::Parse((dynamic_cast<ListViewItem^>(x))->SubItems[ col ]->Text))) > 0.0;
           }
        };

        private:
                /// <summary>
                /// Required designer variable.
                /// </summary>
                System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
                /// <summary>
                /// Required method for Designer support - do not modify
                /// the contents of this method with the code editor.
                        /// </summary>
                        void InitializeComponent(void)
                        {
                                this->TabularView = (gcnew
    System::Windows::Forms::TabControl());
                                this->DataTab = (gcnew System::Windows::Forms::TabPage());
                                this->dataGridView1 = (gcnew
    System::Windows::Forms::DataGridView());
                                this->Symbol = (gcnew
    System::Windows::Forms::DataGridViewTextBoxColumn());
                                this->Bid = (gcnew
    System::Windows::Forms::DataGridViewTextBoxColumn());
                                this->Ask = (gcnew
    System::Windows::Forms::DataGridViewTextBoxColumn());
                                this->Spread = (gcnew
    System::Windows::Forms::DataGridViewTextBoxColumn());
                                this->Strength = (gcnew
    System::Windows::Forms::DataGridViewTextBoxColumn());
                                this->GraphicalTab = (gcnew System::Windows::Forms::TabPage());
                                this->StatusStrip = (gcnew
    System::Windows::Forms::StatusStrip());
                                this->LoadingStatus = (gcnew
    System::Windows::Forms::ToolStripStatusLabel());
                                this->listView1 = (gcnew System::Windows::Forms::ListView());
                                this->TabularView->SuspendLayout();
                                this->DataTab->SuspendLayout();
                                (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(
    this->dataGridView1))->BeginInit();
                                this->GraphicalTab->SuspendLayout();
                                this->StatusStrip->SuspendLayout();
                                this->SuspendLayout();
                                //
                                // TabularView
                                //
                                this->TabularView->Anchor =
    static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top
    | System::Windows::Forms::AnchorStyles::Bottom)
                                        | System::Windows::Forms::AnchorStyles::Left)
                                        | System::Windows::Forms::AnchorStyles::Right));
```

```
			this->TabularView->Controls->Add(this->DataTab);
			this->TabularView->Controls->Add(this->GraphicalTab);
			this->TabularView->Location = System::Drawing::Point(0, 0);
			this->TabularView->Name = L"TabularView";
			this->TabularView->SelectedIndex = 0;
			this->TabularView->Size = System::Drawing::Size(563, 316);
			this->TabularView->SizeMode = System::Windows::Forms::TabSizeMode::FillToRight;
			this->TabularView->TabIndex = 0;
			// 
			// DataTab
			// 
			this->DataTab->Controls->Add(this->dataGridView1);
			this->DataTab->Location = System::Drawing::Point(4, 22);
			this->DataTab->Name = L"DataTab";
			this->DataTab->Padding = System::Windows::Forms::Padding(3);
			this->DataTab->Size = System::Drawing::Size(555, 290);
			this->DataTab->TabIndex = 0;
			this->DataTab->Text = L"Tabular Data";
			this->DataTab->UseVisualStyleBackColor = true;
			// 
			// dataGridView1
			// 
			this->dataGridView1->AllowUserToAddRows = false;
			this->dataGridView1->AllowUserToDeleteRows = false;
			this->dataGridView1->Anchor = static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Bottom)
				| System::Windows::Forms::AnchorStyles::Left)
				| System::Windows::Forms::AnchorStyles::Right));
			this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;
			this->dataGridView1->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^  >(5) {this->Symbol,
				this->Bid, this->Ask, this->Spread, this->Strength});
			this->dataGridView1->Location = System::Drawing::Point(3, 3);
			this->dataGridView1->Name = L"dataGridView1";
			this->dataGridView1->ReadOnly = true;
			this->dataGridView1->RowHeadersVisible = false;
			this->dataGridView1->Size = System::Drawing::Size(549, 284);
```

```
                        this->dataGridView1->TabIndex = 0;
                        //
                        // Symbol
                        //
                        this->Symbol->HeaderText = L"Symbol";
                        this->Symbol->Name = L"Symbol";
                        this->Symbol->ReadOnly = true;
                        //
                        // Bid
                        //
                        this->Bid->HeaderText = L"Bid";
                        this->Bid->Name = L"Bid";
                        this->Bid->ReadOnly = true;
                        //
                        // Ask
                        //
                        this->Ask->HeaderText = L"Ask";
                        this->Ask->Name = L"Ask";
                        this->Ask->ReadOnly = true;
                        //
                        // Spread
                        //
                        this->Spread->HeaderText = L"Spread";
                        this->Spread->Name = L"Spread";
                        this->Spread->ReadOnly = true;
                        //
                        // Strength
                        //
                        this->Strength->HeaderText = L"Strength";
                        this->Strength->Name = L"Strength";
                        this->Strength->ReadOnly = true;
                        //
                        // GraphicalTab
                        //
                        this->GraphicalTab->Controls->Add(this->listView1);
                        this->GraphicalTab->Location = System::Drawing::Point(4, 22);
                        this->GraphicalTab->Name = L"GraphicalTab";
                        this->GraphicalTab->Padding =
System::Windows::Forms::Padding(3);
                        this->GraphicalTab->Size = System::Drawing::Size(555, 290);
```

```
			this->GraphicalTab->TabIndex = 1;
			this->GraphicalTab->Text = L"Graphical View";
			this->GraphicalTab->UseVisualStyleBackColor = true;
			// 
			// StatusStrip
			// 
			this->StatusStrip->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^  >(1) {this->LoadingStatus});
			this->StatusStrip->Location = System::Drawing::Point(0, 319);
			this->StatusStrip->Name = L"StatusStrip";
			this->StatusStrip->Size = System::Drawing::Size(566, 22);
			this->StatusStrip->TabIndex = 1;
			this->StatusStrip->Text = L"statusStrip1";
			// 
			// LoadingStatus
			// 
			this->LoadingStatus->Name = L"LoadingStatus";
			this->LoadingStatus->Size = System::Drawing::Size(199, 17);
			this->LoadingStatus->Text = L"Waiting for connection from client...";
			// 
			// listView1
			// 
			this->listView1->Anchor = static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Bottom) 
				| System::Windows::Forms::AnchorStyles::Left) 
				| System::Windows::Forms::AnchorStyles::Right));
			this->listView1->Location = System::Drawing::Point(0, 0);
			this->listView1->Name = L"listView1";
			this->listView1->Size = System::Drawing::Size(555, 290);
			this->listView1->TabIndex = 0;
			this->listView1->UseCompatibleStateImageBehavior = false;
			// 
			// Form1
			// 
			this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
			this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
			this->ClientSize = System::Drawing::Size(566, 341);
```

```
                this->Controls->Add(this->StatusStrip);
                this->Controls->Add(this->TabularView);
                this->Name = L"Form1";
                this->Text = L"Form1";
                this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
                this->TabularView->ResumeLayout(false);
                this->DataTab->ResumeLayout(false);
                (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this->dataGridView1))->EndInit();
                this->GraphicalTab->ResumeLayout(false);
                this->StatusStrip->ResumeLayout(false);
                this->StatusStrip->PerformLayout();
                this->ResumeLayout(false);
                this->PerformLayout();

            }
    #pragma endregion

    public: void UpdateStatusStrip() {
            LoadingStatus->Text = Globals::textToUse;
    }

    public: void AddRowsToDataGridView() {
                for(int i = 0; i < Globals::numSymbols; i++) {
                    dataGridView1->Rows->Add(gcnew System::String(Globals::strarray[i]), gcnew System::String(L""), gcnew System::String(L""), gcnew System::String(L""));
                }
    }

    public: void AddSymbolsToListView() {
                listView1->View = View::LargeIcon;

                // Create columns for the items and subitems.
                // Width of -2 indicates auto-size.
                listView1->Columns->Add("Currency", -2, HorizontalAlignment::Left);
                listView1->Columns->Add("Strength", -2, HorizontalAlignment::Left);

                largeImageList = gcnew ImageList;
                largeImageList->ImageSize = System::Drawing::Size(64, 64);

                // Initialize the ImageList objects with bitmaps.
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\very_high_strength.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\high_strength.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\strength.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\weak_strength.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\weak_weakness.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\weakness.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\high_weakness.png"));
                largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\very_high_weakness.png"));
                // largeImageList->Images->Add(Bitmap::FromFile("C:\\\\currencymeter\\\\no_indication.png"));

                listView1->LargeImageList = largeImageList;

                // Initialize the tile size.
                this->listView1->TileSize = System::Drawing::Size(128, 128);

                listViewItemList = gcnew array< System::Windows::Forms::ListViewItem^>(currencies.size());

                int i = 0;

                for(std::set<std::string>::iterator iter = currencies.begin(); iter != currencies.end(); iter++) {
                    listViewItemList[i] = gcnew ListViewItem(gcnew System::String((*iter).c_str()));
                    listViewItemList[i]->SubItems->Add(System::Convert::ToString(0));
                    // listViewItemList[i]->ImageIndex = 8;
                    i++;
                }

                // Add the items to the ListView.
                listView1->Items->AddRange(listViewItemList);
    }

    public: void UpdateListView() {
                int numCurrencies = currencies.size();
                int i = 0;

                for(std::set<std::string>::iterator iter = currencies.begin(); iter != currencies.end(); iter++) {
                    ListViewItem^ currentItem = listView1->FindItemWithText(gcnew System::String((*iter).c_str()));
                    currentItem->SubItems[1]->Text = System::Convert::ToString(processed_strength_vector[i]);

                    if(currentItem != nullptr) {
                        double str = processed_strength_vector[i];
                        if(str > 5) {
                            currentItem->ImageIndex =  0;
                        } else if (str > 0.5) {
                            currentItem->ImageIndex =  1;
                        } else if (str > 0.2) {
                            currentItem->ImageIndex =  2;
                        } else if (str > 0) {
                            currentItem->ImageIndex =  3;
                        } else if (str > -0.2) {
                            currentItem->ImageIndex =  4;
                        } else if (str > -0.5) {
                            currentItem->ImageIndex =  5;
                        } else if (str > -5) {
                            currentItem->ImageIndex =  6;
                        } else {
                            currentItem->ImageIndex =  7;
                        }
                    }

                    i++;
                }

                this->listView1->ListViewItemSorter = gcnew ListViewItemComparer( 1 );

            }

    public: void UpdateDataInTable() {
                for(int i = 0; i < Globals::numSymbols; i++) {
                    std::string str1 = strarray2[i].substr(0, 3);
                    std::string str2 = strarray2[i].substr(3, 3);
                    dataGridView1->Rows[i]->Cells[0]->Value = Globals::strarray[i];
                    dataGridView1->Rows[i]->Cells[1]->Value = System::Convert::ToString(bid_bar.at(i)->front());
                    dataGridView1->Rows[i]->Cells[2]->Value = System::Convert::ToString(ask_bar.at(i)->front());
                    if(str2 != "JPY" && str1 != "JPY") {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 10000);
                    } else if (str2 == "JPY") {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 1000);
                    } else {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 100000);
                    }
                    dataGridView1->Rows[i]->Cells[4]->Value = System::Convert::ToString(processed_strength_vector_map[str1] - processed_strength_vector_map[str2]);
                }
                dataGridView1->Update();
    }

    public: void UpdateDataForTickInTable() {
                for(int i = 0; i < Globals::numSymbols; i++) {
                    std::string str1 = strarray2[i].substr(0, 3);
                    std::string str2 = strarray2[i].substr(3, 3);
                    dataGridView1->Rows[i]->Cells[0]->Value = Globals::strarray[i];
                    dataGridView1->Rows[i]->Cells[1]->Value = System::Convert::ToString(bid_bar.at(i)->front());
                    dataGridView1->Rows[i]->Cells[2]->Value = System::Convert::ToString(ask_bar.at(i)->front());
                    if(str2 != "JPY" && str1 != "JPY") {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 10000);
                    } else if (str2 == "JPY") {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 1000);
                    } else {
                        dataGridView1->Rows[i]->Cells[3]->Value = System::Convert::ToString((ask_bar.at(i)->front() - bid_bar.at(i)->front()) * 100000);
                    }
                    dataGridView1->Rows[i]->Cells[4]->Value = System::Convert::ToString(processed_strength_vector_map[str1] - processed_strength_vector_map[str2]);
                }
                dataGridView1->Update();
    }

    private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^  e) {

                Globals::form = this;

                pData =  (PCURRENCYDATA) HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, sizeof(PCURRENCYDATA));

                hThread = CreateThread(
                    NULL,                   // default security attributes
                    0,                      // use default stack size
                    getDataFromMql,             // thread function name
                    pData,                          // argument to thread function
                    0,                      // use default creation flags
                    &dwThread);                     // returns the thread identifier

                if( pData == NULL ) {
                    // If the array allocation fails, the system is out of memory
                    // so there is no point in trying to print an error message.
                    // Just terminate execution.
                    ExitProcess(2);
                 }

                if (hThread == NULL)
                {
                    ExitProcess(3);
                }
            }
    };
    }
```