RESEARCH ARTICLE

AESTIMATIO, THE IEB INTERNATIONAL JOURNAL OF FINANCE, 2014. **8**: 114-131

A wave smoothing algorithm and applications to the financial markets

Ait Hellal, Omar
Meyer, Gerald H.

RECEIVED: 16 MAY 2013
ACCEPTED: 13 NOVEMBER 2013

Abstract

In this paper we present an algorithm that can be implemented recursively or iteratively, to smooth waves by filtering out "noise" until the base case is reached, a canonical form that we call the *wave's imprint*. Unlike other wave smoothing algorithms that consider extrema as outliers or noise, our wave smoothing algorithm considers extrema to be essential data as is the case with seismic activity, epileptic seizures, and daily highs and lows of the DJIA. It is applicable to any wave structure that has a fixed time period during which a high and low are recorded. We limit the scope of this paper to the analysis of financial markets, demonstrating commonality over a broad spectrum of financial markets including indexes, equities, commodities and currencies. As an application of the algorithm, we devise a simple trading system that is profitable over multiple markets.

Keywords:

Smoothing algorithm, Fibonacci retracement, Market structure, Waves, Trading system.

JEL classification:

G1.

Ait Hellal, O. ✉ *Department of Math, Engineering and Computer Science,LaGuardia Community College, CUNY 31-10 Thomson Avenue, Long Island City, New York, 11101. Email: oaithellal@lagcc.cuny.edu*

Meyer, G.H. Department of Math, Engineering and Computer Science,LaGuardia Community College, CUNY 31-10 Thomson Avenue, Long Island City, New York, 11101. Email: geraldm@lagcc.cuny.edu

AESTIMATIO, THE IEB INTERNATIONAL JOURNAL OF FINANCE, 2014. **8**: 114-131

Un algoritmo de suavizado de ondas **y sus aplicaciones a los mercados financieros**

Ait Hellal, Omar
Meyer, Gerald H.

Resumen

En este artículo se presenta un algoritmo, susceptible de implementación de forma recursiva o iterativa, para el suavizado de ondas mediante filtrado del "ruido" hasta alcanzar el caso base, una forma canónica que denominamos la *huella de la onda.* A diferencia de otros algoritmos de este tipo, que consideran los valores extremos como valores atípicos o ruido, el algoritmo que se presenta los considera datos esenciales, como en el caso de la actividad sísmica, ataques epilépticos y máximos y mínimos diarios del DJIA. Es aplicable a cualquier estructura de onda con periodo de tiempo fijo durante el cual tenga lugar un máximo y un mínimo. El alcance de este artículo se centra en el análisis de los mercados financieros, mostrándose la comunalidad existente sobre un amplio espectro de ellos que incluye índices, acciones, materias primas y divisas. Como aplicación del algoritmo, se plantea un sencillo sistema de trading rentable en múltiples mercados.

Palabras clave:

Algoritmo de suavizado, retroceso de Fibonacci, estructura de mercado, ondas, sistema de negociación.

1. Introduction

For years, attempts have been made to characterize markets according to a set of rules. Some not very complex theories such as Market Profile explained by Dalton *et al*. (1999), and Elliot wave theory, updated by Frost and Prechter (2005) espouse strong deterministic behavior using a set of rules; others use chaos theory in an attempt to prove or disprove determinism in the markets. Results of these latter studies have been mixed, even for the same market (Adrangi *et al*. 2001, Epaminondas and Vassilia 2000). Gao and Wang (1999) as well as Adrangi *et al*. (2001) found no evidence of deterministic chaos, whereas Epaminondas and Vassilia (2000) and Peters (1996) claim the contrary. Our approach differs from all the above in that we apply a heuristic to extract the "essential" flavor of the market.

This paper introduces a Wave Smoothing Algorithm (WSA) that has applications in many areas, in particular, the financial markets. Unlike Marchand and Marmet (1983) who use a least squares approach developed by Savitsky and Golay (1964) to eliminate noise such as outliers, our algorithm pivots on the extrema and recursively smooths the squiggly subwaves. As an example of how the WSA can be applied to financial markets, we develop a simple trend following trading system. The system requires input from the WSA and produces very promising results, comparing favorably to patterns such as the engulfing pattern (Ait Hellal and Meyer, 2012) as well as to a carefully designed random system. As has become very popular of late (Chetty *et al*., 2008), we used empirical testing to verify the efficacy of our trend trading system.

The paper is organized as follows: In section 2.1 we give a detailed description of the WSA using a set theoretic formulation of the recursive algorithm. Appendix B contains detailed pseudocode compatible with C++ or JAVA for those who are well-versed in programming. The methodology is explained in section 2.2. Section 3 is dedicated to the results that we derive from running the WSA algorithm over four different markets. In section 4, we develop a trend following trading system as an application of the WSA, followed by an enhancement that incorporates time constraints. Also in section 4, we compare our trading system to a randomized approach. In section 5, we conclude by summarizing our results and outlining possible areas for which the WSA can be applied.

2. The WSA

2.1. Description of the algorithm

Initially, our data is stored in a file *raw_data* that contains comma separated values. Each row may contain values such as: date, time, open, high, low and close[1], from which we extract only the high and low. Our data is presorted in ascending order according to date.

We define the **periodNumber** as the ordering of the rows of *raw_data* starting from 0 and ending at the length of the file *raw_data* –1.

A **DataPoint** object is an ordered 3-tuple: periodNumber, high/low value, flag (flag denotes a high or a low).

From the file *raw_data,* one may create a .csv file (comma separated) *file_in* that contains rows consisting of the data in all the **DataPoints**. That is, rows alternate between a high value and a low value in ascending order by periodNumber (initially, every periodNumber will appear twice, once for the high and once for the low). Alternatively, one can implement the logic, as we did (see Appendix B) using an array of DataPoints.

A **wave** is defined by two consecutive DataPoints. Its **length** or **wavelength** is defined as the difference between their respective periodNumbers + 1. Therefore, the length of any wave is at least 1. The magnitude of the wave is defined as the absolute value of the difference of the high/low values of the two DataPoints.

A **wave structure** is defined by a set of DataPoints D where the highs and lows alternate and a high is always higher than the following low and conversely a low is smaller than the following high.

Relative High/Low: We define a relative high/low as any data item that is greater than or equal/less than or equal to its predecessor and greater than or equal/less than or equal to its successor. For the first and last point the definition is consistent looking only at the successor and predecessor respectively.

The algorithm that follows produces arrays or files (ordered sets) of DataPoints that represent successive wave structures where each successive wave structure is a refinement of the previous, eliminating "noise" until a base case is reached which we call the **wave imprint**. Below is a mathematical formulation of the algorithm.

Let H(k) be the set of high DataPoints in version k, where k=0 represents the initial set of data and L(k) the set of low DataPoints in version k. D(k), which is the version k wave structure, is the set resulting from merging H(k) and L(k) in ascending order according to periodNumber satisfying the set of constraints described below.

[1] *We used data readily available on fxhistoricaldata.com for FOREX and finance.yahoo.com for all other markets.*

Let $h_{k,i}$ be the ith element in set $H(k)$, similarly $l_{k,i}$ the ith element in set $L(k)$, and $d_{k,i}$ the ith element in wave structure $D(k)$. We generate successive sets of DataPoints recursively as follows:

Recall that H(0)={the ordered set of all DataPoints representing daily highs extracted from the initial data} with a similar definition for L(0). Thus, D(0) is the set resulting from merging L(0) and H(0) in ascending sequence by periodNumber with $h_{0,0}$ appearing before $l_{0,0}$ ($D(0) = \{h_{0,0}, l_{0,0}, h_{0,1}, l_{0,1}, \ldots, h_{0,N-1}, l_{0,N-1}\}$, where N is the length of the initial file). By construction, each periodNumber is repeated exactly twice in D(0) (e.g. $h_{0,0}$.periodNumber = $l_{0,0}$.periodNumber). Recursively, the set of highs in version k is $H(k) = \{h_{k-1,i} \mid h_{k-1,i} \geq h_{k-1,i+1}$ and $h_{k-1,i} \geq h_{k-1,i-1}\}$; the set of lows in version k is $L(k) = \{l_{k-1,i} \mid l_{k-1,i} \leq l_{k-1,i+1}$ and $l_{k-1,i} \leq l_{k-1,i-1}\}$.

The version k wave structure $D(k)$ is derived by first merging the two sets $H(k)$ and $L(k)$ according to the periodNumber, followed by weeding out anomalies from $D(k)$ according to the following set of rules:

1. If $d_{k,i}$.direction = $d_{k,i+1}$.direction and $d_{k,i}$.direction = high, then we have two consecutive high DataPoints; in this case, delete the smaller of the two. Similarly, delete the larger of any two consecutive low DataPoints occurring in $D(k)$.

2. If $d_{k,i}$.direction = high and $d_{k,i+1}$.direction = low and $d_{k,i} < d_{k,i+1}$; then we have a high followed by a low that is higher. In this case, delete $d_{k,i+1}$. Similarly, if $d_{k,i}$.direction = low and $d_{k,i+1}$.direction = high and $d_{k,i} > d_{k,i+1}$, then delete $d_{k,i+1}$.

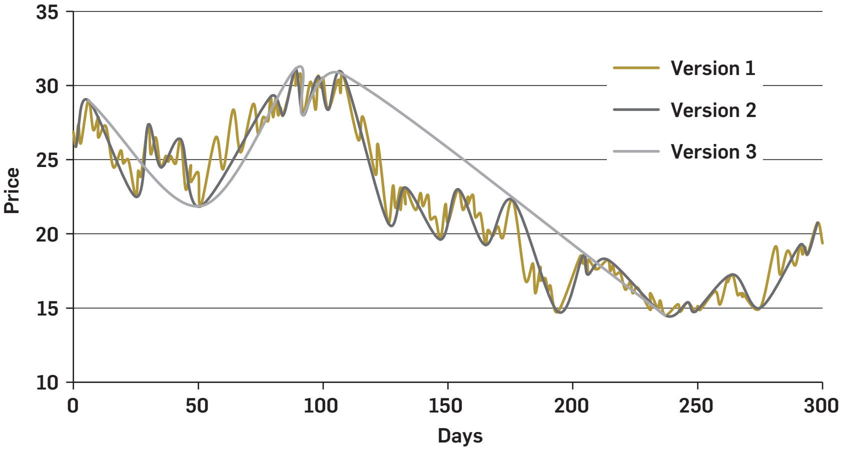
3. Repeat 1 and 2 until no changes are made.

4. Update $H(k)$ and $L(k)$ as $H(k) = \{d_{k,i} \mid d_{k,i}$.direction = high$\}$ and $L(k) = \{d_{k,i} \mid d_{k,i}$.direction = low$\}$ maintaining the order according to periodNumber.

Note that version 1 is the result of running the algorithm on the raw data obtained from daily lows and highs, which in turn becomes input to the algorithm to generate Version 2, etc. Thus, version k+1 is a subset of version k where every DataPoint in version k+1 must be a relative high or relative low in version k. However, not every relative high and low of version k is a DataPoint of version k+1, since some are "weeded out" as described above. The recursion is continued until $D(k)$ is the empty set (the wave imprint is then $D(k$–1$)$).

The output of the algorithm over the first 300 days of Apple, Inc. (AAPL) is illustrated in Figure 1. The figure displays versions 1, 2 and 3 of the output of the algorithm. Note that for version 2, the red line appears truncated relative to the green line of version 1 and similarly, the blue line corresponding to version 3 appears truncated relative to the previous versions. This is due to the fact that we are showing a snippet from a much larger file.

Figure 1. **Different versions for APPL**



2.2. Methodology

We used financial data across four markets as input into the WSA. We considered stocks, commodities, indices, and Forex. For stocks, we selected 32 frequently traded stocks from the Nasdaq and NYSE. For commodities, we used 22 frequently traded ETFs that mirror the price action of the underlying commodity. For indices, we used 22 of the most heavily traded ETFs that mirror averages such as the DJIA, the Nasdaq 100, and the S&P 500. For Forex, we selected 14 commonly traded pairs including the eurusd, the gbpusd, and the gbpjpy. See Appendix A for a complete list of all stocks, commodity ETFs, indexes ETFs, and Forex currency pairs.

The heuristic uses time periods based on daily high/low values, some extending as long as 50 years, from which we extract the basic building blocks of our model, namely, the relative extrema. In each recursive version, the relative extrema are, by construction, relative extrema in earlier versions, but not necessarily in future versions. Some of these extrema are “weeded out” based on a set of rules that we apply as described in section 2.1. We then analyze the results of the successive versions generated by the WSA for each market. We generate descriptive statistics, involving length of the waves and their magnitudes and compare the magnitudes and lengths of consecutive waves. The magnitude/length ***retracements*** of the *k*th wave is defined as the ratio of the magnitude/length of waves *k* and *k*-1. These retracement statistics are generated for every version across all markets. Using this data we make inferences regarding the nature of the wave structures over different versions.

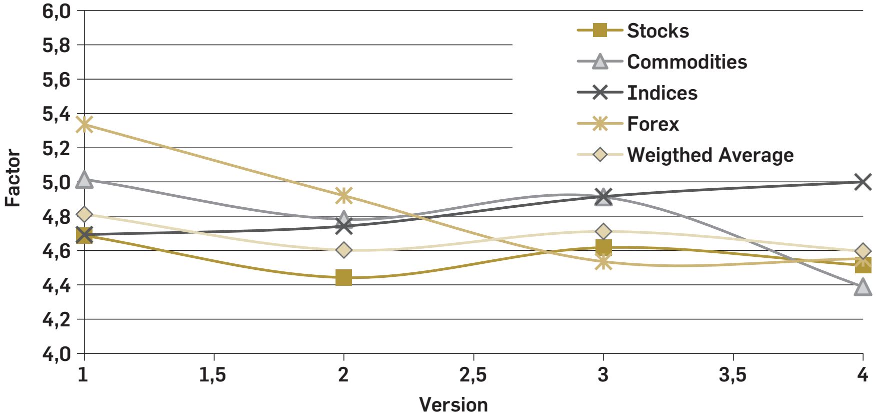
3. Results

The first chart represents the size reduction from one wave structure to the next. In this chart, we limit our analysis to the first five wave structures (versions 0 through 4, where version 0 is the initial set of data).

3.1. A factor of five

Surprisingly, there is commonality between all markets especially pronounced in the second, third and fourth versions. Indeed, the number of DataPoints, as well as waves, is reduced by a factor of approximately five between successive versions (see Figure 2). There has always been a certain mysticism surrounding the number five with respect to wave structure; in Elliot wave theory, for instance, impulse waves are comprised of five waves (Frost and Prechter, 2005).

Figure 2. **Ratio of number of waves from one version to the next**



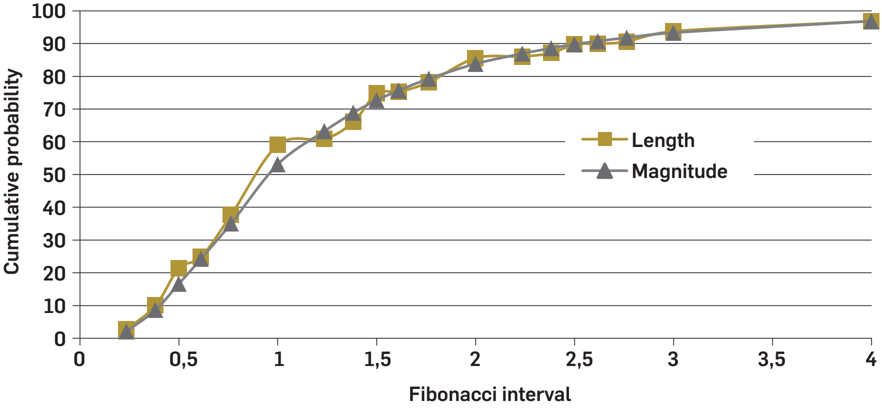
3.2. Order in the markets?

We have witnessed a kind of order in the chart above, showing that the WSA algorithm reduces larger wave structures into smaller ones by a factor of approximately five. We now show order in terms of the distributions of the retracements of magnitude and wavelength over all markets and across different wave structure versions.

We use the common Fibonacci retracement percentages as follows: we use the intervals [0, 0.236], (0.236, 0.382], (0.382, 0.5], (0.5, 0.628], (0.628, 0.764], etc. and we record the number of retracements that fall into each of the above intervals for both magnitude and length.

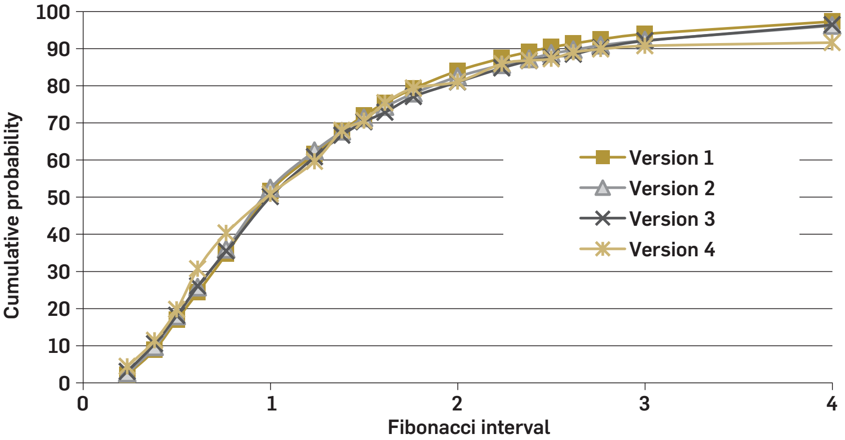
Below are charts showing the cumulative probability for a retracement to be less than a certain threshold for magnitude and length, i.e. $F(x) = P(\text{retracement} \leq x)$, where x is a retracement percent. In all charts the x-axis is the retracement threshold, corresponding to an endpoint of the above Fibonacci intervals and the y-axis is the cumulative probability (in percent) for a retracement not to exceed this threshold. We observe, as shown in Figure 3, that the cumulative probability retracements are strikingly similar for magnitude and length.

■ Figure 3. **Cumulative probability for magnitude and length retracements over all markets and all versions**



An interesting observation comes from Figure 4; it depicts the cumulative probability of magnitude retracements over all markets for different versions. We observe that although successive wave structure versions 2, 3, and 4 are each 80% smaller than the previous, they have the same intrinsic behavior as version 1 with respect to wave retracements (note: version 4 is approximately 92% smaller than version 1).

■ Figure 4. **Cumulative probability for magnitude retracements for different versions**



The next two charts, Figure 5 and Figure 6, show the magnitude as well as the length retracements for different markets across all versions. We can make the same observation here as well; the cumulative probability for different markets is very close, which suggests that different markets behave the same. This can be illustrated by noting the strong correlations that exist in markets such as the dollar versus gold or interest rates and spx.

Figure 5. **Overall cumulative probability for magnitude retracements for different markets across all versions**

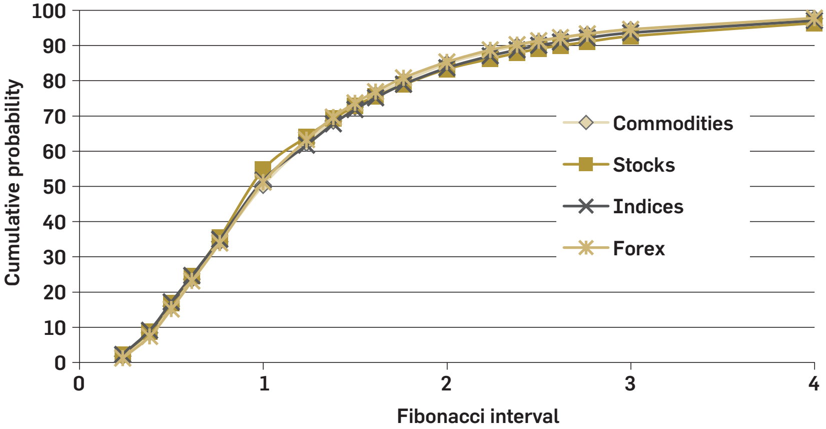
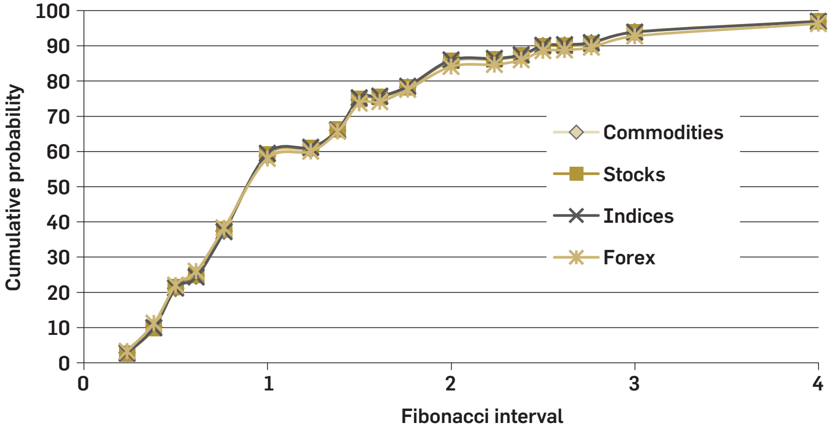


Figure 6. **Overall cumulative probability for length retracements for different markets across all versions**



In summary, our results suggest that markets behave the same in different timeframes! The behavior within small timeframes corresponding to lower versions of the wave structures is a microcosm of the larger timeframes. This lends credence to Elliot wave theory which states that the intrinsic behavior of the markets is the same for waves of what is called lesser and higher degree (Frost and Prechter, 2005). We examined data regarding retracements within the same market across multiple versions and found that they have the same probabilistic characteristics. Figure 4 shows the cumulative probabilities for retracements across all markets for different versions (note: individual markets behave similarly, see Appendix C for charts).

Granted, we cannot predict with high accuracy where the markets are going to be months, weeks, or even days from now. Nonetheless, markets will trend in an orderly way despite large shocks such as black swan events, which usually have an immediate but relatively short-lived impact. We show, based on our predictive trading system, that a small probabilistic edge can be achieved from taking advantage of this order.

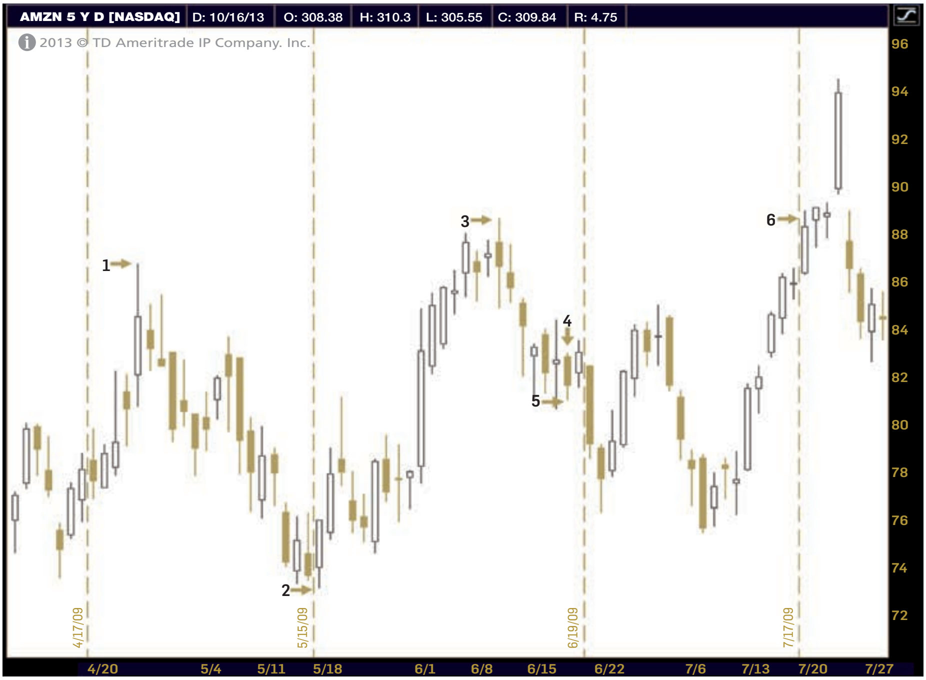
4. An application of the WSA

4.1. A pragmatic application of the algorithm

We looked for a predictive nature of the algorithm that we could use towards developing a trading system. Our aim was to show an example of how one can develop a trading system using the data ouput from the WSA. From our results (Figure 3), we observed that more than 52% of the waves retrace less than 100% which conforms to the adage "the trend is your friend". This led us to develop a trend following trading system. Version 1 waves are short in nature and very difficult to trade, so we decided to trade using version 2 waves. We define an intermediate uptrend as two consecutive version 2 high DataPoints $d_{2,i-2}$ and $d_{2,i}$, where the second is greater than the first; conversely, an intermediate downtrend is two consecutive version 2 low DataPoints $d_{2,i-2}$ and $d_{2,i}$, where the second is less than the first. The trading system is illustrated for the bullish case in Figure 7.

Figure 7 illustrates a bullish continuation pattern (the short pattern is analogous) for Amazon.com (AMZN). The arrows indicate key DataPoints for the setup. The first arrow shows a version 2 high and the second shows the last version 2 low. The third arrow shows a new version 2 high (the target), which must be higher than the previous version 2 high (first arrow) establishing the requisite intermediate uptrend. The fourth arrow, pointing down, shows the confirmation candle (this red candle confirms the completion of the relative high in version 1). The fifth arrow shows where to enter the long trade; the entry point is the low of the confirmation candle, and a required 50% retracement from $d_{2,i}$ to $d_{2,i-1}$ had already occurred during the candle before. Therefore, we wait to enter the trade when and if it trades at the low of the confirmation candle. In this example, it took two days before the trade was entered. The last arrow points to where the trade is closed: in this example, a winning trade, it took nearly a month to reach the target. We reject trades that meet our requirements with a greater gain/loss potential than 20%, thus limiting our risk. In all cases, the stop is chosen so that the distance from the entry point to the target $d_{2,i}$ is equal to the distance from the entry point to the stop in the opposite direction.

Figure 7. **An illustration of a bullish setup in AMZN**



We tested the above predictive algorithm in multiple markets over a twelve year time period beginning January 2000, except for ETFs that were created later. This includes Stocks, Commodities, Indices, and Forex. We considered 32 of the most heavily traded stocks in NYSE as well as NASDAQ, 22 ETFs tracking the most heavily traded commodities, 22 ETFs tracking the most followed indices, and 14 commonly traded Forex pairs. See Appendix A for a complete list of markets used in this study.

We tested the trading system and it performed very well over all markets, including longs and shorts. It produced a 53.2% winning percentage with an average of 0.42% profit per trade, which is outstanding considering that the total number of trades is 6450. It averaged 0.60% profit per trade for longs and 0.26% per trade for shorts.

4.2. Improvement

It is difficult in real time to discern the time at which a DataPoint is determined in each version because of the weeding out process described in section 2.1. A solution to this problem consists of estimating the likelihood of a relative extrema becoming a DataPoint in the next version. In order to do so, we implement a generalized DataPoint that includes a field called **timeRecorded** which can be defined recursively as follows: for version 0, the timeRecorded is the same as the periodNumber. Without loss of generality, we define the timeRecorded for a high DataPoint in version *k*, as the timeRecorded for the consecutive (by timePeriod) high DataPoint in version *k*-1 (note: the DataPoint in version *k* is necessarily also a version *k*-1 DataPoint). We use this enhanced implementation of the WSA to generate and analyze time related data. We define **confirmationTime** for a DataPoint as the difference between its timeRecorded and its periodNumber.

To have confidence that a version 2 real time DataPoint survives the weeding process, we incorporate timing. From our analysis, we found that the distribution of confirmationTime ratios from one DataPoint $d_{k,i}$ to its predecessor of the same direction $d_{k,i-2}$ has a mode within (0.764, 1]. For example, $d_{2,i}$.confirmationTime/ $_{2,i-2}$.confirmationTime is required to fall in a fixed interval as described below. Thus, we added a time constraint such that a real time DataPoint in version 2 is considered as having good potential to survive the weeding process if the ratio of its confirmationTime to its predecessor's confirmationTime is between 0.382 and 1.382 which is a symmetrical band around the mode. In Figure 7 above, the confirmationTime ratio constraint is satisfied; the confirmationTime for the new high $d_{2,i}$ is 7 and it is 10 for the previous version 2 high $d_{2,i-2}$ which yields a ratio of 0.7.

We tested the trading system with the time constraint; it achieved a 0.53% profit per trade on average, with long trades achieving 0.63% and short trades achieving 0.43% average profit per trade. The percentage of winning trades also improved slightly to 53.80%. The number of trades decreased to 4927 trades, approximately 400 trades a year, yielding on aggregate a profit of more than 200% (see caveat below).

Some Caveats:

We did not compute the effect of gap opens on any of the trades. This can skew the results even though, on average, the effects of the gaps should be a wash.

We did not compute dealer/broker commissions or spreads, which vary greatly by broker dealer and market. For example, the spread on the GBP/CAD can be as many as 12 pips or more. Discount brokers such as Scottrade and Ameritrade charge a flat rate between 7 and 10 dollars per trade for stocks and ETFs regardless of the size of the trade, thus the transaction costs are negligible for large trades (on the order of 0.02% round-trip for average trades of $100000). Some trades can take as long as a month or more to complete while others may complete within days. Therefore, in practice, one cannot expect the overall yield to be more than 200% as stated above. Finally, as in any trading system, past performance is no guarantee of future results.

4.3. Comparison with a randomized approach

We compared our trading system to a random trading system with characteristics that mimic ours. Our system produced the following statistics: 240074 days in our data, 4927 trades, average holding period of 14.28 days with a standard deviation of 26.86, and an average targeted profit in percent of 5.60% with a standard deviation of 4.05. The random system works as follows: the system initiates a trade with a uniform probability of 4927/240074 = 0.02*;* the direction of the trade long/short is defined by a random Boolean variable, as we had nearly the same number of short and long trades.

The trade is held a maximum number of days, if not closed before. These maximum durations are generated using a normal distribution with mean 14.28 and standard deviation of 26.86, as in our system. The targeted profit is generated according to a normal distribution with mean 5.60% and standard deviation of 4.05; the profit/loss ratio is kept to 1; the gain/loss on a single trade is limited to a maximum of 20%, as in our system.

We tested the system described above using the same data used in our system. We ran 1000 simulations, using the same data used for our system spanning the same 12 year period and 88 markets (see Appendix A), and averaged the results. On average, there were 4910 trades for each trial; the average Profit/Loss per trade over all simulations is a loss of 0.026%. Assuming the null hypothesis that the mean is 0 (the trading profit is 0), we calculated the standard deviation from 0 to be 0.1625. Thus the z-score for our result is –0.026/0.1625 = –0.1636. This lies at about the –6% quantile so that there is approximately an 88% chance that the null hypothesis is correct. Furthermore, the results of our trading system yield a z-score of 3.26 which lies well outside the 99% confidence level; therefore, we conclude that the results of our system are not the result of “friendly” data or random luck.

5. Conclusion

In this paper, we developed a recursive algorithm allowing us to smooth out noise in data, provided there is a recorded high and low in a prescribed period of time. We applied our algorithm to the financial markets and derived statistics that suggest order in the markets as evidenced by: the ratio of five (5) between consecutive versions in terms of number of waves and the consistency of the retracements of both magnitude and length for different markets and across versions. We also believe it may be useful to study the wave imprints as a way of classifying or categorizing waves.

While we are most excited about the underlying WSA and its potential application across the disciplines, we also see the potential for its further use in predicting market movements. Indeed, we developed a very simple but tradable trading system in which our entry point is close to the 50% Fibonacci retracement and our target/stop is close to the previous version 2 relative high/lows. By varying these parameters one can look to improve the trading system. Many other setups, based on recursive versions, higher than version 2, can likely be utilized for a potential trading system, involving a longer holding period. There is no limit but the imagination as to what may work, should work, or deserves to be looked at. In the field of finance, there has been much research done to predict major turning points. Bianchi *et al*. (1999), explore several methods including an iterative least squares method. This is an example of a research topic where we believe the WSA has great potential. We expect that this paper will serve as a launching pad for others to come along with many variations of the theme. In addition, this paper may spur interest for researchers in other fields such as medicine, pharmacology, oceanography, astronomy, geology, etc. where data with wave-like structures need to be analyzed.

References

- Adrangi, B., Chatrath, A., Kanwalroop, K.D. and Rafiee, K. (2001). Chaos in Oil Prices? Evidence from Futures Markets, *Energy Economics,* **23**(4), pp. 405-425.
- Ait Hellal, O. and Meyer, G.H. (2012). The Engulfing Pattern: An Empirical Approach for Swing Traders, *Technical analysis of Stocks & Commodities Magazine,* **5**(4), pp. 44-49.
- Bianchi, M., Boyle, M. and Hollingsworth, D. (1999). A comparison of methods for trend estimation, *Applied Economics Letters,* **6**, pp. 103-109.

- Chetty, R., Looney, A. and Kroft, K. (2009). Salience and taxation: theory and evidence, *American Economic Review,* **99**(4), pp. 1145-1177.
- Dalton, J.F., Jones, E.T. and Dalton R.B. (1999). *Mind over Markets: Power Trading with Market Generated Information,* Traders Press, Cedar Falls, Iowa, USA.
- Epaminondas, P. and Vassilia, N. (2000). Are Oil Markets Chaotic? A Non-Linear Dynamic Analysis, *Energy Economics,* **22**(5), pp. 549-568.
- Frost, A.J. and Prechter, R.R. (2005). *Elliot Wave Principle: Key to Market Behavior,* 10th edition, New Classics Library, Gainesville, Georgia, USA.
- Gao, A.H. and Wang, G.H.K. (1999). Modeling Nonlinear Dynamics of Daily Futures Price Changes, *The Journal of Futures Markets,* **19**(3), pp. 325-351.
- Marchand, P. and Marmet, L. (1983). Binomial smoothing filter: A way to avoid some pitfalls of least square polynomial smoothing, *Rev. Sci. Instrum.,* **54**. pp. 1034-1041.
- Peters, E.E. (1996). *Chaos and Order in the Capital Markets: A New View of Cycles, Prices, and Market Volatility,* Wiley Finance, New York.
- Savitzky, A. and Golay, M.J.E. (1964). Smoothing and differentiation of data by simplified least squares procedures, *Analytical Chemistry,* **36**, pp. 1627-1639.

Appendix A

MARKETS:

a. Stocks

AAPL, AKS, AMZN, BAC, BBY, BCS, BIDU, BRCD, C, CAT, CSCO, DAL, DOW, EMC, EPI, F, GE, GLW, GOOG, IBM, INTC, JPM, LVLT, MSFT, NFLX, NOK, NVS, S, SFD, SIRI, SYMC, TSN.

b. ETFs tracking Commodities

DBA, DBC, GLD, GDX, IYM, SHY, SLV, SLX, TLT, UNG, USO, XHB, XLB, XLE, XLF, XLI, XLK, XLP, XLU, XLV, XLY, XME.

c. ETFs tracking Indices

DIA, EEM, EWA, EWC, EWG, EWH, EWJ, EWM, EWS, EWT, EWW, EWY, EWZ, FXI, IWD, IWF, IWM, QQQ, RSX, SPY, VWO, VXX.

d. FOREX

AUDJPY, AUDUSD, CADJPY, CHFJPY, EURAUD, EURCHF, EURGBP, EURJPY, EURUSD, GBPJPY, NZDUSD, USDCAD, USDCHF, USDJPY.

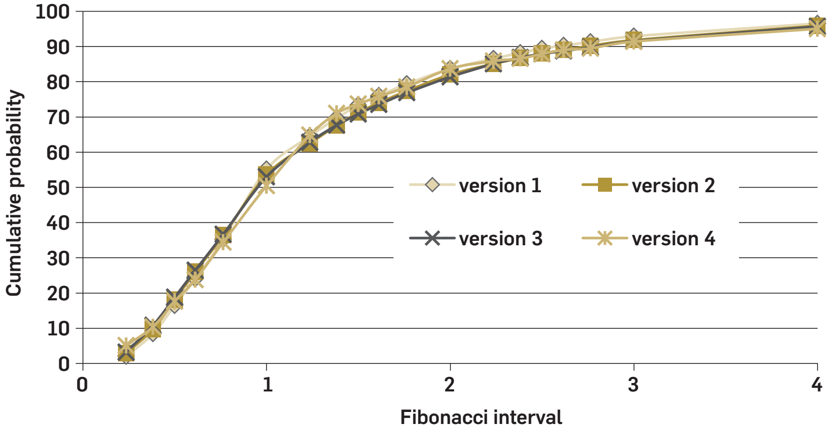
Appendix B

Pseudocode of WSA

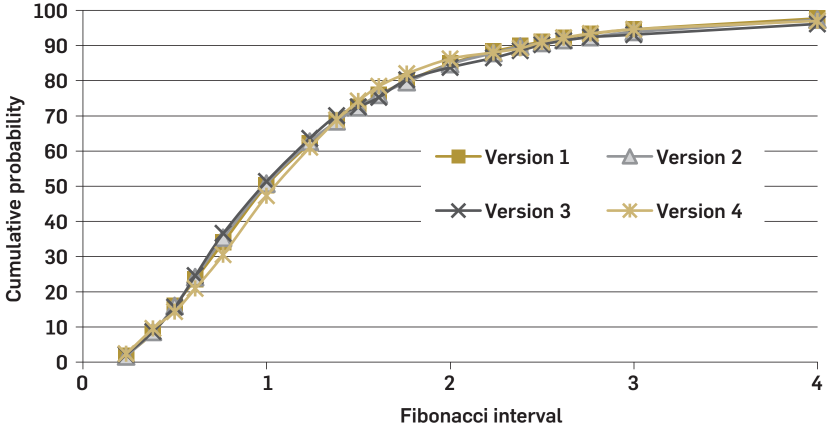
```
int version = 0; // Wave structure number (0 is the raw data)

void Waves (String file_in) // recursive function or method

{
        version++; //  next wave structure

        If (length of file_in==0)     //base case

            return;

        {

            Extract the relative highs and relative lows and their respective periodNumbers into two separate arrays low and high where low contains low DataPoint objects and high contains high DataPoint objects.

            Merge the relative highs and lows keeping the order according to the periodNumber into a single array highLow of DataPoint.

            Weed out anomalies in the highLow array in the following order:

            1. Eliminate consecutive relative highs (no relative low in between) and consecutive relative lows (no relative high in between) by choosing the greatest high and smallest low respectively. In case of a tie for the greatest or smallest, choose the DataPoint with the highest periodNumber.

            2. Check to make sure a relative high DataPoint is not followed by a relative low DataPoint that is greater or vice versa. In such case, without loss of generality, if a relative high is followed by a relative low that is greater, we remove the relative low. As a result, if the DataPoint that is removed is not the end of the file, we will be left with consecutive relative highs, that will be resolved according to 1.

            createOtherOutput(version,highLow); /*create any other files from the data in highLow; optional method or function */

            createNewVersion(version, highLow);   /*Create file_in+version, a .csv file (comma separated) that contains rows consisting of the data in all the DataPoints. This results in a reduction of the original data to a smaller version
            of itself, capturing the periodNumbers of the relative highs and lows which
            are used in turn as input into the recursion. */

            Waves (file_in + version);

        } //end recursion

    } //end waves
```

Appendix C

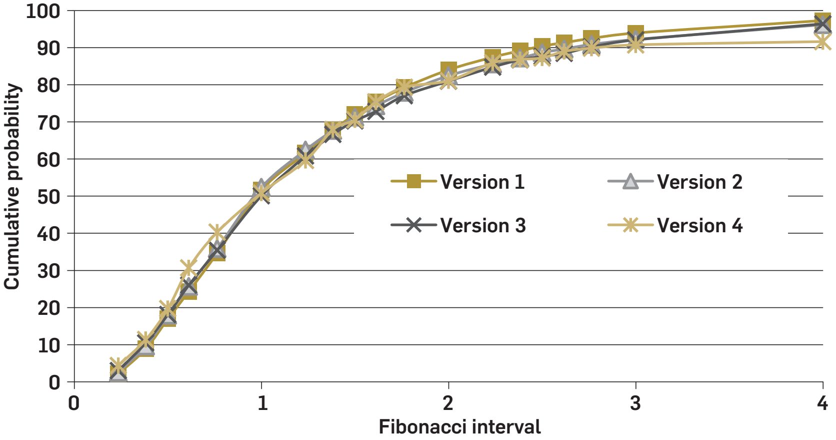
Cumulative probability for magnitude retracements for stocks, for different versions



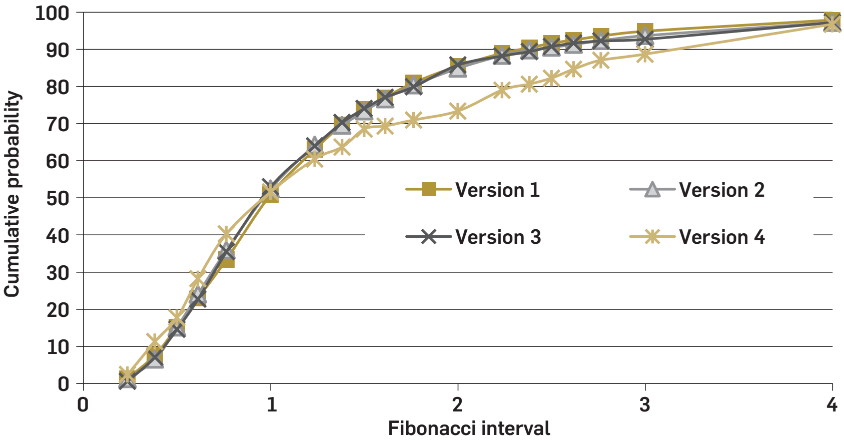
Cumulative probability for magnitude retracements for commodities, for different versions



Cumulative probability for magnitude retracements for indices, for different versions



Cumulative probability for magnitude retracements for Forex, for different versions


Cumulative probability
Fibonacci interval
Version 1
Version 2
Version 3
Version 4