Fixed Fraction Position Sizing:
A Stock Trading Strategy.

Guy R. Fleury

Abstract:

This paper: *Fixed Fraction Position Sizing: A Stock Trading Strategy* is using a simple stock trading strategy. It is designed to make money for the long term resulting from a single equation. It reaches its objective by generating long term alpha as a by product of the methodology used. It is a worthwhile trading strategy based on a slightly different perspective of the stock trading problem.

The resulting trading program could be executed as an automated trading script or on a discretionary basis.

The paper is presented as an evolutionary process, the how it came about, and the reasons for why it will hold the test of time.

I hope it can be of some use and help you to also design better trading systems.

Acknowledgment: Special thanks go to Murielle Gagné for her support and numerous comments while writing this paper.

The Starting Point: A Question

In a forum debate on trading strategy position sizing, a simple question was asked:

> "*Why do you always go broke using fixed fractional position sizing on a game with a 50% chance of success*???"

I have not used fixed fractions in ages, nonetheless, I opted to look at the problem again to see for myself the "what if I did" implied? With better tools, better software, better machines and a better understanding of the game, it might help uncover something I might have missed at the time, or reassert the reasons why I did not use such trading procedures. Also, I use variable position sizing schemes in my own programs and it sounded like something I should investigate further since I was sure it would have some impact.

This started a research project that spanned a few weeks. A search for an answer to the "Why is that?" question above. The research led to some added insights: some interesting, others unexpected.

The problem can be viewed from two different perspective depending on how you define a fixed fraction of equity trading strategy. On one hand, you have the use of a fix fractional amount (a percent of the trading account) used as betting unit; and on the other, playing for a fix percent account return on equity using barrier-like limits such as profit targets and stop losses. Both these methods use fix fractions to define their respective trading methods. However, the premises on which these two trading strategy are based leads to quite different behaviors when applied to real market data.

To differentiate both trading methods, I will use "blue" to describe the system using a fix fraction of account as betting unit, while "red" (my view of the problem) will stand for the percent return on account play. Which is being referenced should be evident from the context.

The two trading strategies, their respective values and merits will be discussed making it a closed system with blue and red presenting their respective outlooks with math, equations, concepts and programs along with the interpretation of what the data is saying. Everything and anything needed being put on the table by both sides. I'll try to maintain both perspectives as separate views of the problem and will continue to reference them as questions that have been or could have been asked or remarks made by supporters of either trading method.

My early response to the question was:

> *Using a fixed fraction makes the game a percentage or compounding play. Two 50% increases followed by two 50% decreases does not return you to your initial state, but to 56.25% of your original amount; and that is not a way to get ahead even in a 50% chance of success game.*

This is exactly the problem encountered by the trading method analyzed: an equal fixed fractional position sizing (EFFPS) trading strategy. There is a delayed trade size imbalance due to the trading strategy used. To show that in fact it was the case, I presented a spreadsheet where a portfolio simulation on 10 stocks was set up where trades would be executed as if having a win/loss ratio of 1.00 (a 50/50 scenario).

Red's Equal Fix Fraction Position Sizing

The equation best representing the equal fix fraction of equity position sizing scheme has been around for quite some time. And since my interest is the long term, I set the number of trades to be analyzed at $k = 1{,}000$. A time horizon long enough to show ramifications that might not be noticeable under a smaller number of trades.

The equation best suited to represent red's trading strategy is:

$$A(k)(red1) = A(0)\cdot(1+f)^W\cdot(1-f)^L \qquad (1)$$

where A(k)(red1) is red's trading account value after k trades, A(0) the initial account size, f the fixed fraction of equity at stake, W the number of wins, while L being the number of losses. Note that: $(W + L) = k = 1{,}000$, the number of trades. Filling in the numbers for the variables in equation (1) gives the answer for any trading scenario based on this EFFPS trading method.

This trading strategy can be programed; its simplest pseudo-code being:

```
InstallProfitTarget( f );  // set profit target to f %
InstallStopLoss( f );      // set stop loss to f %
```

Most trading software have these functions built in. What the above instructions do is set barrier-like limits within which prices can fluctuate. When ever the stock price touches a limit, the corresponding trade is automatically executed, generating a win or a loss of f %. The account is automatically adjusted: the profit is added to the account or the loss subtracted resulting in the new account size. More precisely, the account will receive the proceeds of the stock sale, such that after a win: $A(k+1) = A(k)\cdot(1+f)$ and for a loss: $A(k+1) = A(k)\cdot(1-f)$.

From the spreadsheet, it was easy to see that as the number of trades increased, in this 50% winning ratio game, the account size was decreasing due to a trade size imbalance. And this deterioration was more important the more the trade number increased and the more the equal fixed fraction increased.

To generate a 50% winning ratio game, I used the equivalent of a coin flip to determine up and down price swings of ± f %: the equal fixed fraction. This way, each random test would have for expected outcome: 50% winning trades. Each test would be different, but overall, as the number of tests would increase, their mean would average out to this 50/50 scenario.

The expectation of this long term EFFPS trading strategy does not operate as the expectation of a coin tossing experiment but as a degenerative process resulting in the following equation:

$$E[A(k)(red1)] = A(0)*[(1+f)^W*(1-f)^L] \rightarrow 0 \qquad \text{as } k \rightarrow 1{,}000 \qquad (2)$$

where E[A(k)(red1)] = my expectation of what the most probably outcome might be as k increased. The variables have the same definition as in equation (1).

The equation states that using an EFFPS trading strategy will have the trading account value tend to zero, as the number of trades increases, making the trading method most certainly undesirable. It's a major statement since you are playing as if trading was a 50/50 game of chance. Therefore, the average on a number of trails, should have the winning ratio converge to 50% which it does. And if so, the game should resemble a coin tossing experiment. However, equation (2) says otherwise.

To gain an understanding of the process, you simply fill in some plausible numbers and you get your answer. Say after 100 trades with an equal fixed fraction f = 0.20: with the most expected outcome of a 50/50 game: W = L = k/2 = 50; one would get:

$$E[A(k=100)(red1)] = A(0)\cdot(1+0.20)^{50}\cdot(1-0.20)^{50} = 0.1298\cdot A(0)$$

Not exactly A(0) the original portfolio. It did not matter the size of the portfolio either, it is totally scalable. What is expected to remain in the account after 100 trades would be on average: 12.98% of the original account. This even if you won 50% of the trades.

You had a 50% chance of winning or losing on any single trade, yet you lost a game the equivalent of a coin tossing experiment with expectancy of zero which should have generated a no change scenario. On 100 trades, with an expected win ratio of 50%, you are expected to win 50 and lose 50 trades with equal probability. Yet, using the equal fixed fractional position sizing scheme you lost. And worst, you were almost guaranteed to lose your entire account the more you continued to trade (see equation (2) above).

There is no need to make complicated calculations. The strategy should and is expected to lose big time simply by increasing the number of trades. The EFFPS trading strategy when applied to trading stocks is simply an assured way to lose, and the longer you play, the more likely you will lose it all. At trade k = 200, under the same conditions: A(k=200)(red1) = 0.0169·A(0). Only 1.69% of the portfolio is expected to remain, the equivalent of losing 98.31% of the original trading account.

Blue's Equal Fix Position Sizing

Blue's trading strategy resembles more a betting system. A fix fraction of the account is bet on each trade in a 50/50 game. An equation to represent blue's trading strategy would be:

$$A(k+1)(blue) = A(k) + b\cdot A(k)\cdot pw - b\cdot A(k)\cdot pl \qquad (3)$$

where A(k+1)(blue) is the value of the trading account after trade k has occurred, A(k) the value of the account prior to the trade, pw the probability of a win, while pl the probability of a loss. Blue's most often cited b value (fraction of account bet) was: b = 0.10; meaning that 10% of the account A(k) was bet on each successive trade.

The expectation of such a trading system in a 50/50 scenario would be:

$$E[A(k+1)(blue)] = A(k) + [b \cdot A(k) \cdot pw - b \cdot A(k) \cdot pl] = A(0) \qquad (4)$$

The reasoning behind this trading strategy is that you place percent of account bets that have win/loss probability approaching a 50/50 type of game. And since it does resemble a coin tossing experiment, the same expectancy of no net benefit should prevail and therefore: E[A(k)(blue)] = A(0). Blue's view of the fix fraction position sizing problem has some limitations and drawbacks which will be covered in this paper.

I'll try to restrain my analysis to the asked question, blue's betting strategy as well as my own. The intent is not to show which trading strategy was best (blue's or red's, even if it turns out that red (mine) was better by at least an order of magnitude, and with some added parameter modifications, could do much better). This should be viewed as relative since there are out there better trading strategies.

The point of interest is to show how both blue's and red's trading strategies would behave under fire as in a real life market situation using real life market data. There might be things to learn in the process. Also, let's not forget that the primary objective was to find ways to improve the strategy, repair its inefficiencies and/or correct any problems that might be encountered.

Some might think that this kind of debate is trivial. I would say think again. It could in fact have a considerable impact on anyone's trading account. I don't feel concerned, I don't use this trading technique (but will make every effort to not use it ever). Any trading strategy developer should be aware of the ramifications of what the above question implies.

Any barrier-like exits could turn out to average out to the equivalent of an equal fixed fraction exit, and thereby this problem could become of major importance in any trading strategy you may design. The initial question can have repercussions over the entire life of your portfolio since the degradation is insidious and long term. In fact, a portfolio would suffer from an exponential decay the longer the technique would be used. But these deficiencies can be easily corrected.

The why of this paper.

Objectives

The objective was to first better understand the question raised, find other explanations (if any), then find solutions to the problem, and design improvements if possible. The last two items being a request and most probably why the question was asked.

Any program is just code, it simply executes. Any wrong code logic or misunderstood concepts leads to programs and procedures that might not survive beyond a "theoretical" or conceptual stage. A reality check is always required. At all times, the equal sign needs to turn out to be true. Your program code will not be satisfied with less.

Does blue's evaluation and concepts prevail in describing what would happen if his program was executed, or would red's program fail? And if it did not fail, then what would be the value of the blue's program and indirectly the value of his assumptions?

In the forum where this debate took place, I could not say that most adopted my point of view. It seemed as if all bets were on blue's side. I know having made the remark that there were only 2 on my side (the author of the original question and myself that agreed with his observations).

This was like going against all odds. In a forum dedicated to automated trading strategies (my interest is stocks), you dared contradict the majority, or as in this case, the one, as representative of the group since most agreed with blue's point of view.

You had these two views of the same problem, where even basic definitions would be questioned. Both blue and red had governing equations to represent their respective trading methodologies and therefore, from these trading rules could generate trading programs that could "execute" these rules. Notwithstanding, when ever you put some equations on the table, I do think it is a requirement that these equations hold. An equation has no opinions, there are no maybes, it bares the full force of an equal sign: it is true or someone has to prove it is not.

The Challenge

You had blue stating: E[A(0)(1+b)^W(1-b)^L] = A(0), used to describe the expectation of the trading procedure in a 50/50 game environment as having the same expectation as playing a coin tossing contest which almost everyone of age on this planet knows has an expectancy of zero. And therefore, his trading method would show the same: a no gain, no loss trading scenario having for end result the same value as the initial account, with for consequence that there was no need and absolutely no benefit of playing that game.

In blue's trading environment, you placed fractional bets of +b*A(k) of the portfolio based on a coin tossing game. Blue's most commonly cited value for b was 0.10; and therefore blue made 10% of account bets at a time. Only luck could make you a winner or a loser. Otherwise, your expected outcome was no gain, no loss. You are expected, if the coin is fair, to get out of the game with your original stake: A(0). However, there was this little problem in blue's reasoning. The expected outcome of his bet of +b*A(k) was to win or lose +b*A(k) on each bet. The same as playing head or tail with $50 bets. The expected outcome would then be: $50*W - $50*L → 0; if the convergence of the game was 50/50. Clearly in blue's trading scenario, you had: E[A(k+1)(blue)] = A(k) + b*A(k)*pw - b*A(k)*pl = A(0). A recursive equation which required to know the value of A(k) in order to determine the value of the outcome of each bet. Blue was simply using the wrong equation to express his view of the problem.

Blue was categorically stating that the original stake A(0) was the expected outcome for the equal fixed fractional position sizing scheme depicted in the original question. Because of the randomness of the game, and because this was a coin tossing problem after all, there was no other possible solution to this problem than: E[A(k)(blue)] = A(0), a no gain no loss scenario. No expected benefit from playing the game at all. And therefore, there was no need to play this game either.

But I don't think blue really understood that his view of the problem was not the only way to look at it. What red was using was equal fixed fractional "returns" on portfolio; while blue was looking at an equal fixed fraction betting system. Two very different problems, two different worlds.

The expected outcome of a coin flipping contest is indeed: zero. On a 1,000 coin flips, you are expected to converge to the mean value and be close to: 500 wins and 500 losses. So one could start analyzing an equation expressed in probabilities at its very center of gravity, and then explore "What if" scenarios that deviate from the mean.

On my side, I simply stated that: E[A(k)(red1)] → 0; meaning that, in time, the outcome of the trading procedure used would simply result in bankruptcy: the lost of the entire account. My interest was not the coin tossing part, but the use of the trading strategy itself.

Having not agreed to his views, blue's response was:

> *Guy, I would like to emphasize that a constant allocation fraction greater than 0 and less than 1 mathematically does not lead to zero account.*

If blue did not understand my point of view as a valid solution, no problem. I designed a simple spreadsheet simulation to show what really happens. With my familiarity of random processes it took just a few minutes of added work. Especially since all the specification of the problem had already been given in the initial question.

To make my point clear, I provided an Excel spreadsheet (available here), where an equal fixed fractional position sizing scheme is applied in a 50% winning ratio game on 10 stocks with for duration: 1,000 trades each. Pressing F9 multiple times, it was evident that an account would gradually decline to oblivion the longer you played. The procedure was simple, an equal fixed fraction was determined as the return measure (f = 20%). The return would be variable and would change depending on the output of the applied random function giving the series of winning or losing trades. The file may take some time to download (size > 660k).

The red1 equal fixed fraction of equity trading strategy requires only 4 lines of pseudo-code:

```
InstallProfitTarget( f );  // where f is the percent profit target
InstallStopLoss( f );     // where f is the percent stop loss
Do until you decide to quit or 12,500 trading days (about 50 years)
If NoActivePosition then Buy Q = Account / P(t) shares on next bar;
```

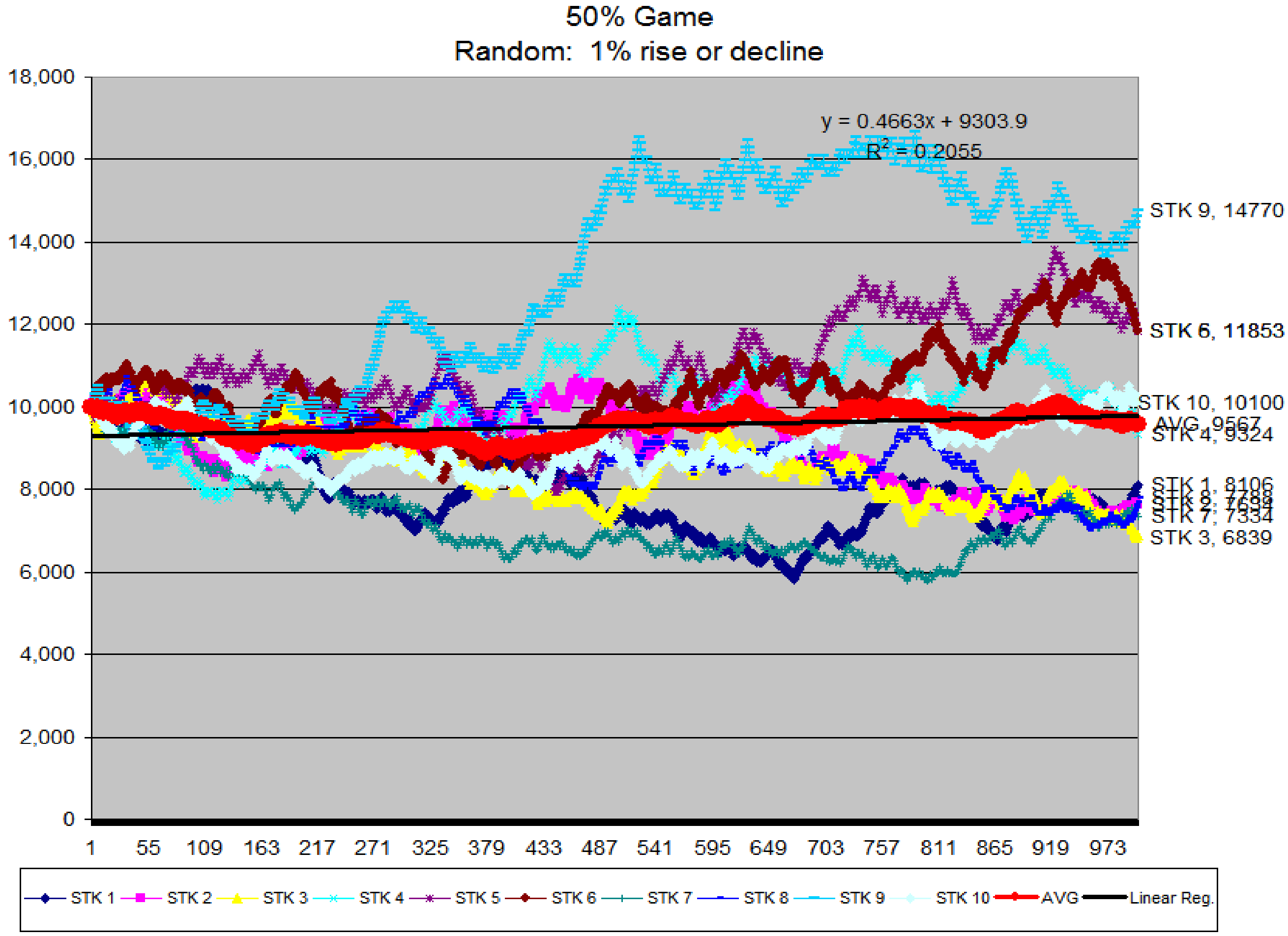
With the above trading procedure, whenever the stock price would hit either of the barrier-like limits: + f % or – f %, a trade would be executed to close the position with a profit or loss of magnitude ± f % that would automatically be added to or subtracted from the account.

In Excel, this would translate to: A(k+1) = A(k)*(1+0.20) for a win, or for a loss: A(k+1) = A(k)*(1-0.20). A random function (coin flip) would determine if you won or lost your bet. Really simple stuff. A single function copied and pasted over a 10 stock scenario with 1,000 trades each. It took more time to explain everything than to design the spreadsheet itself. The test did not need to be made; already very simple math had led to the same conclusion.

The EFFPS trading strategy is a lousy way to trade in a game environment with a 50% win ratio as it is almost a guarantee that your entire portfolio (the more you trade) will be lost. Not due to the coin flipping thing, but due to the trading method itself. The higher the barrier limits, the faster you portfolio would decline. I even provided charts where you could plainly see the portfolio's degradation as f (the fixed fraction percent) would increase.

It resulted in the following 5 case studies:

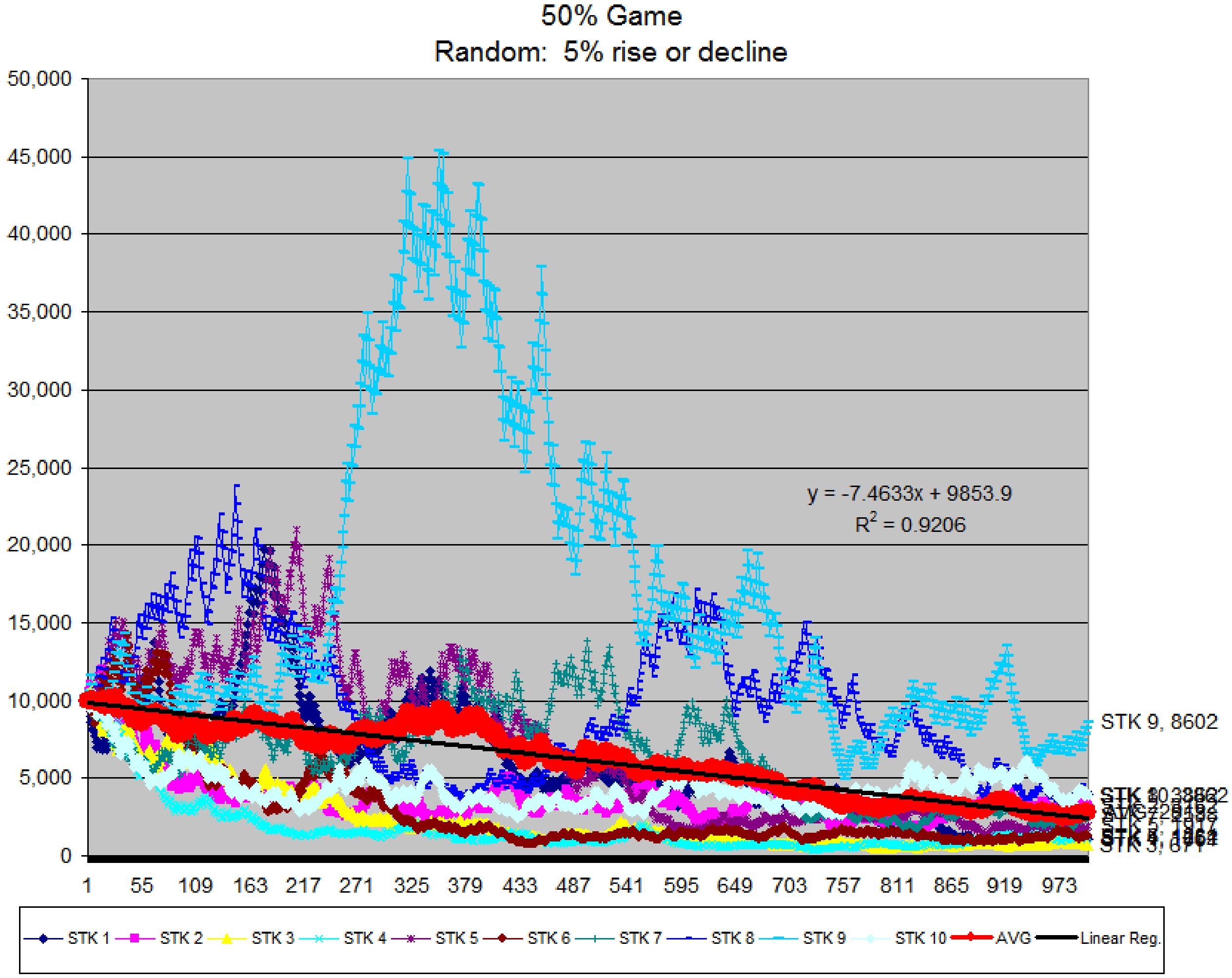
Fig. 1. Fixed Fraction 1% Profit Target - Stop Loss



The 1% degeneration is hardly noticeable when applied to randomly generated trades in a 50/50 environment. The expected average value would be about 0.95*A(0) which is close to the average shown on the chart above after 1,000 trades. There was no predictability in those price series. It would take 1,000 trades to lose 5% of the account as a result of the trading method. After k = 2 trades (with one win and one loss), and on a $10,000 trading account size it would represent only a $1.00 loss. But this would also mean that the stop loss as well as the profit target would be set at 1% from the entry price for each trade.

Fig. 1 shows all the profit targets and stop losses hit for each price series over the 1,000 trades scenario. It is not a time dependent chart but a trade dependent one.

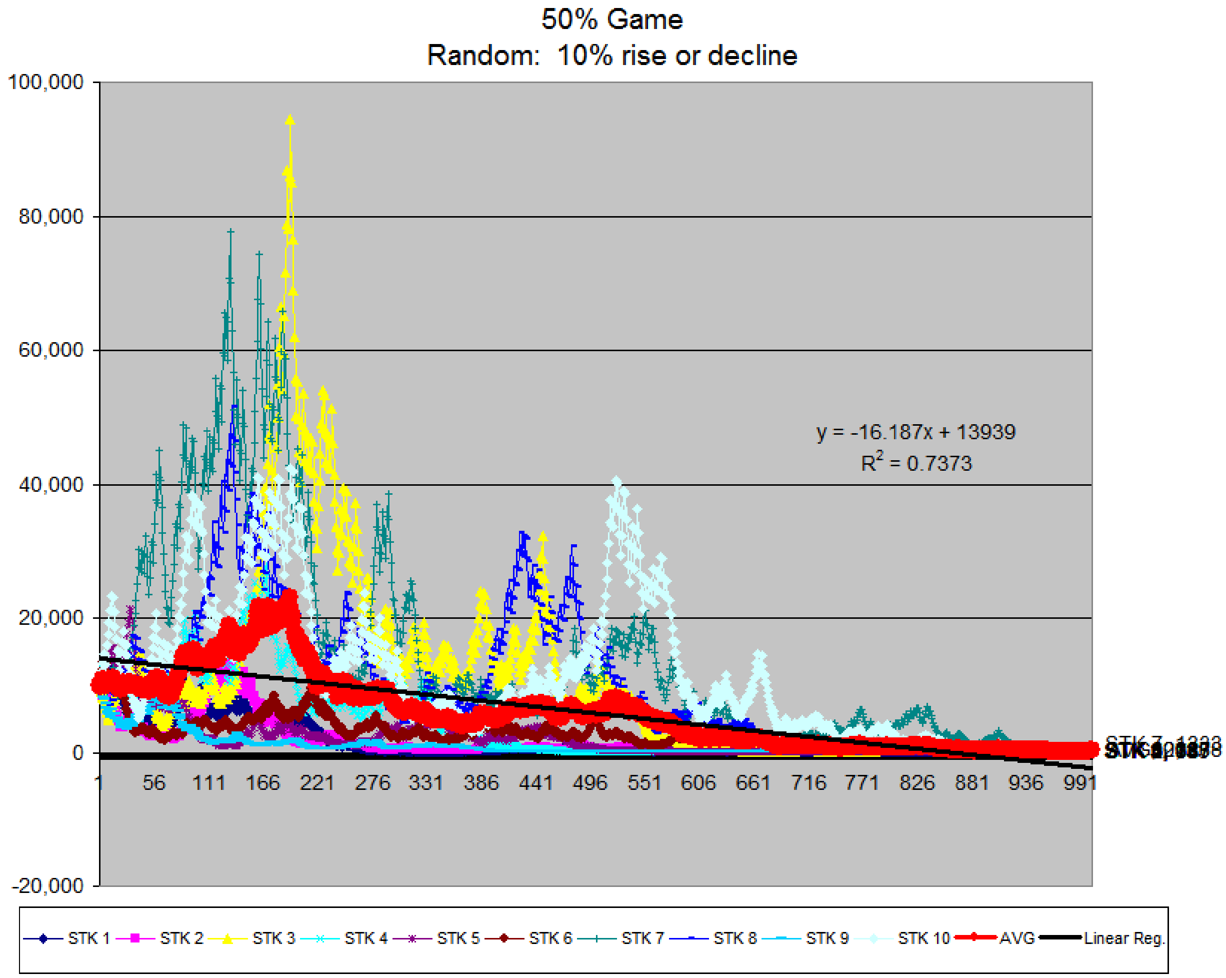
Fig. 2. Fixed Fraction 5% Profit Target - Stop Loss



By increasing the fixed fraction to 5%, we start to see more degradation of the portfolio as a whole, trades are still randomly generated, but a trend starts to be forming. At this level, the expected value would be: 0.2861*A(0). Expecting the portfolio to shrink down to 28.61% its original size. Again, this only due to the increase of the equal fixed fraction. And yet, a 5% price target could be considered by some as almost trivial in the stock market.

With the 50% win ratio, it does not stop a price series to be lucky (such as STK 9 early in the game), but the exercise here is to see the long term effect of the trading strategy.

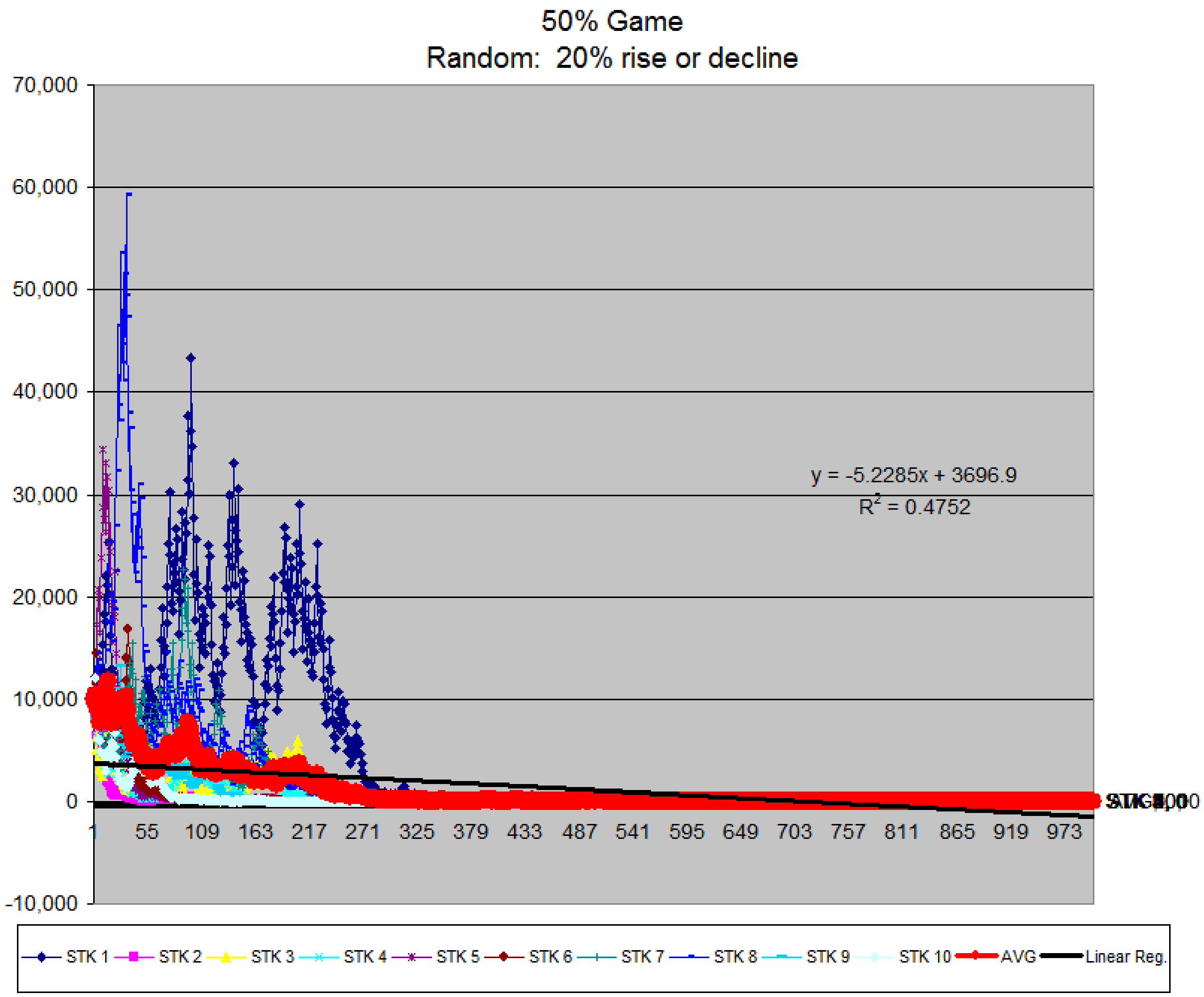
Fig. 3. Fixed Fraction 10% Profit Target - Stop Loss



Pushing further, by increasing the equal fixed fraction to 10% of account shows additional degradation. This time, the expected outcome shrinks to: 0.0657*A(0). Meaning that after those 1,000 trades, only 6.57% of the account is expected to remain. Not a good way to make money. Yet, all the trades were randomly distributed, all the ups and downs followed all the rules of coin flipping to a t.

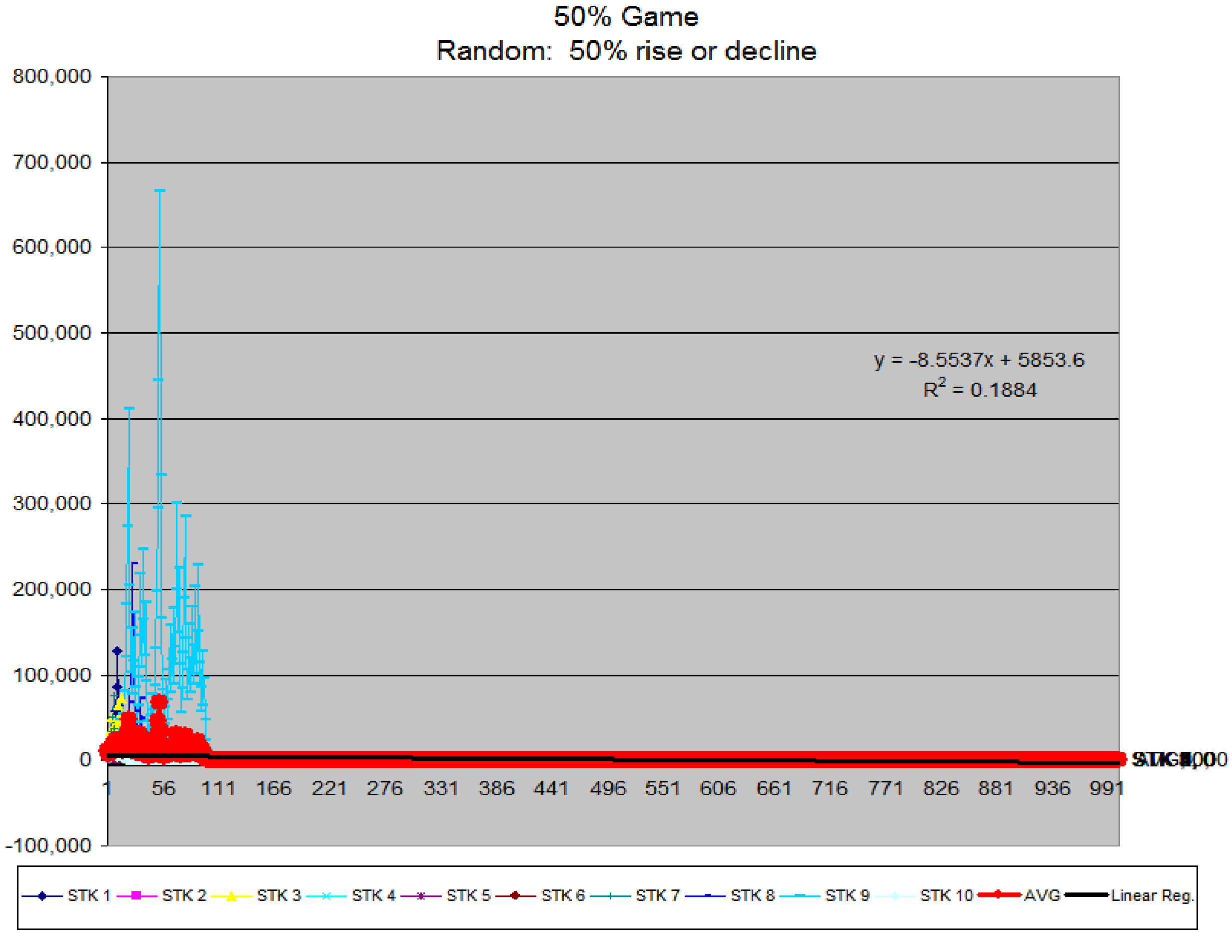
I've seen recently a trading software promoting in its training video a 10% profit target / stop loss trading strategy. Can I politely say it is not a great idea. Do these people really test their promotional stuff?

Fig. 4. Fixed Fraction 20% Profit Target - Stop Loss



At the 20% level for the equal fixed fraction the portfolio really degrades. The expected value is now down to one ten thousandth its initial size: 0.00001*A(0). It should be evident that increasing the fix fraction at play will reduce the value of any portfolio. At this level, 99.99% of the initial account value is lost. Yet it is the same equation that determines the outcome at play (f = 20%): A(k)(red1) = A(0)*[(1+f)^W*(1-f)^L]. All that changed was the increasing barrier-like trade limits. Yet, the random function was on a 50% winning ratio game!

Fig. 5. Fixed Fraction 50% Profit Target - Stop Loss



The equal fixed fraction set at 50% of portfolio is provided just as curiosity for this trading strategy. The value of the ending portfolio might now be down to: 3.39E-59 * A(0). Still a positive number but certainly not great for a trading account. And I'm sure your broker at this level has stop sending you monthly statements a long time before you could even reach it.

As the percent of the equal fixed fraction increased, it was evident that the portfolio value decreased as the number of trades increased. The charts above are representative of what would happen. Note that for any of those charts I would have a hard time to duplicated any of them since I don't use seeds in my random number generator. Each time I pressed F9, I got totally different answers. Therefore, the paths taken by each of the 5 scenarios above are just 50 price series out of E+301 possible cases. This won't change the probable outcome to be very close and on average to converge to what those charts show.

In the five charts above you have a progression of increasing equal fix fraction of equity bets from: 1%, 5%, 10%, 20% and 50%. The equation: A(k)(red1) = A(0)*(1+f)^W*(1-f)^L, had only the variable f gradually increasing. Note that red's game never really reaches 0, it can only tend to zero; but any account starting with A(0) that drops to 0.0001% of its original size should be considered as a losing proposition to say the least.

Note that the probability is not on the fixed fraction f since f does not change for the duration of a test. It is on the value of the fluctuating potential profit or loss (f * A(k)) and their respective frequency of occurrence (W and L) that will be won or lost as if on the flip of a coin with only constraint that the number of wins W plus the number of losses L sum to k the total number of trades: (W+L=k).

The Nature of This Trading Strategy

There is a separation of responsibility to made here. The coin flip determines the direction of the trade, the “who” won or lost, that's all. While the trading strategy has for responsibility to determine the size of the bet won or loss. It is not the coin that determines the size of the bet, it's the program's scaling fraction that determines that.

In the above trading strategy, it's the InstallProfitTarget(f) and InstallStopLoss(f) instructions that determine at what price a trade will be executed, and the coin will only tell which one was selected. Meaning that somehow, one of the two price barriers was hit. The coin flip will be just that, a “coin flip” with output: {head, tail}, {1, -1} or {win, lose}.

There are other ways to say the same thing in code as the installs above. For the profit target, one could use: SellAtLimit (entry price + profit target of f %) if current price > (entry price + profit target of f %) which would translate in pseudo-code to:

if CurrentPrice >= EntryPrice*(1+f) then SellAtLimit (EntryPrice*(1+f)) on next bar

It's just that the InstallProfitTarget is a pre-coded automated program response, requires one line of code and you know exactly what it is going to do for as long as you want to run the program. It will supersede any of the other trading rules, if encountered, that are outside these limits. Those two instructions are barriers, that if crossed, will automatically execute the trade either for a win or for a loss.

The stock market game is not a 50/50 game, but nonetheless, on 100 trades, in the 50% winning ratio game: a 50 wins and 50 losses is not only a probable outcome, it is the most expected outcome.

I can not provide even an estimate of what is the probable path of the next 100 consecutive trades, especially when looking into the future. It would be a futile exercise for the simple reason that the probability of any next move in the sequence is close to an unknown. There would be a googol paths (2^100) to analyze. In the end, only one of those possible paths will ever be realized, meaning that I can not choose the one in a googol that will most certainly happen what ever the estimate or guess I might have made.

Probabilities

After showing the above charts ensued another debate on probability theory, the meaning of expectations and that in a (50/50) game, the probabilities are that the expected value of such a game is zero. And therefore, again, the account should not be lost since the expectation as blue dictated was: E[A(k)(blue)] = A(0). The original account size "is" the expected value at the end of the game. Yes, if the game is a 50/50 game and one uses what amounts to a variable size betting system.

The expectation on a coin tossing game is indeed zero, it has been put in mathematical terms and on paper some 3 centuries ago and has been experienced by people since the dawn of time. That is not my contention here, it's the trading methodology used that is being analyzed and what it does to a trading account.

The further this developed, the further I became convinced that we both were looking at the problem from different perspective: blue from the theoretical point of view of randomly distributed coin flips while my only concern was the real output of the trading strategy itself. Blue was looking at the problem as if only his point of view was the only possible theoretical solution to this problem, while I was looking at the trading strategy on the practical side, and what its implemented trading program would do under real life conditions.

There was nothing I would put on the table, even saying over and over again, that there was a separation to be made between a trading strategy and a series of coin flips that could only provide direction (+ or -). As if blue could not understand that a trading strategy with its fixed trading rules is just a program that will do what it's told; and that a coin tossing experiment is just that, a coin experiment keeping all the properties of coin experiments.

I was categorically stating that the expected value of the trading strategy using an EFFPS scheme in a 50% win ratio environment was: E[A(k)(red1)] → 0, the longer you played. It was supported by my spreadsheet and the very nature of an EFFPS trading strategy. While blue was pounding on the table more vociferously then ever (even shouting), that the only outcome possible was his point of view: E[A(k)(blue)] = A(0).

The five charts above were still not sufficient evidence to show the deterioration was due to the equal fixed % profit target stop loss combination, and despite the randomness of up or down moves, the portfolio would still disintegrate. It was starting to be annoying trying to explain the difference to someone who evidently did not want to understand, could not understand, or maybe had another agenda.

Some might not see the importance of the difference, consider it trivial, but let me say: should you trade live with equal fixed percent profit target and stop loss, you can bet that your trading account will most certainly see the difference.

A trading strategy is just a set of trading rules. In this case, using an equal fixed percent profit target and stop loss meant that to get a profit of f%, it was required that the price of the stock to also rise by f% for a win, or drop by f% for a loss.

The initial question did put on the table a problem that affects anyone using barrier-like limits in their automated or discretionary trading programs. Using an equal profit target stop loss combination as illustrated above is just the poster child of such techniques. But, I think that illustrating the problem as in the above was the easiest way to show that there is indeed a degenerative process at work when using equal fixed fractional position sizing.

Anyone ignoring this phenomena, maybe, should not design automated trading strategies for outside consumption. At least, this way, it would only deteriorate their own account and not someone else's. It should be the responsibility of any trading strategy developer to protect himself and/or his clients against this type of portfolio degradation.

This is a severe judgment, especially on a process that almost everyone ignores or consider as trivial, but what would be the point of designing a trading strategy that is inherently flawed from the start and will only get worst as trading evolves?

My advice to all is this: please, investigate this phenomena on your own in order to protect yourself from this insidious way of losing your trading account. See what it does to your own trading strategies and then apply some of the simple corrective measures as described in this paper.

I am not responsible for the way anyone trades. The above is just a piece of advice. But if this phenomena is real (and it is), and you did nothing in trying to understand what it is all about, or did nothing about it to protect yourselves and with time see your account slowly degrade, know that, from my point of view, you had been properly warned.

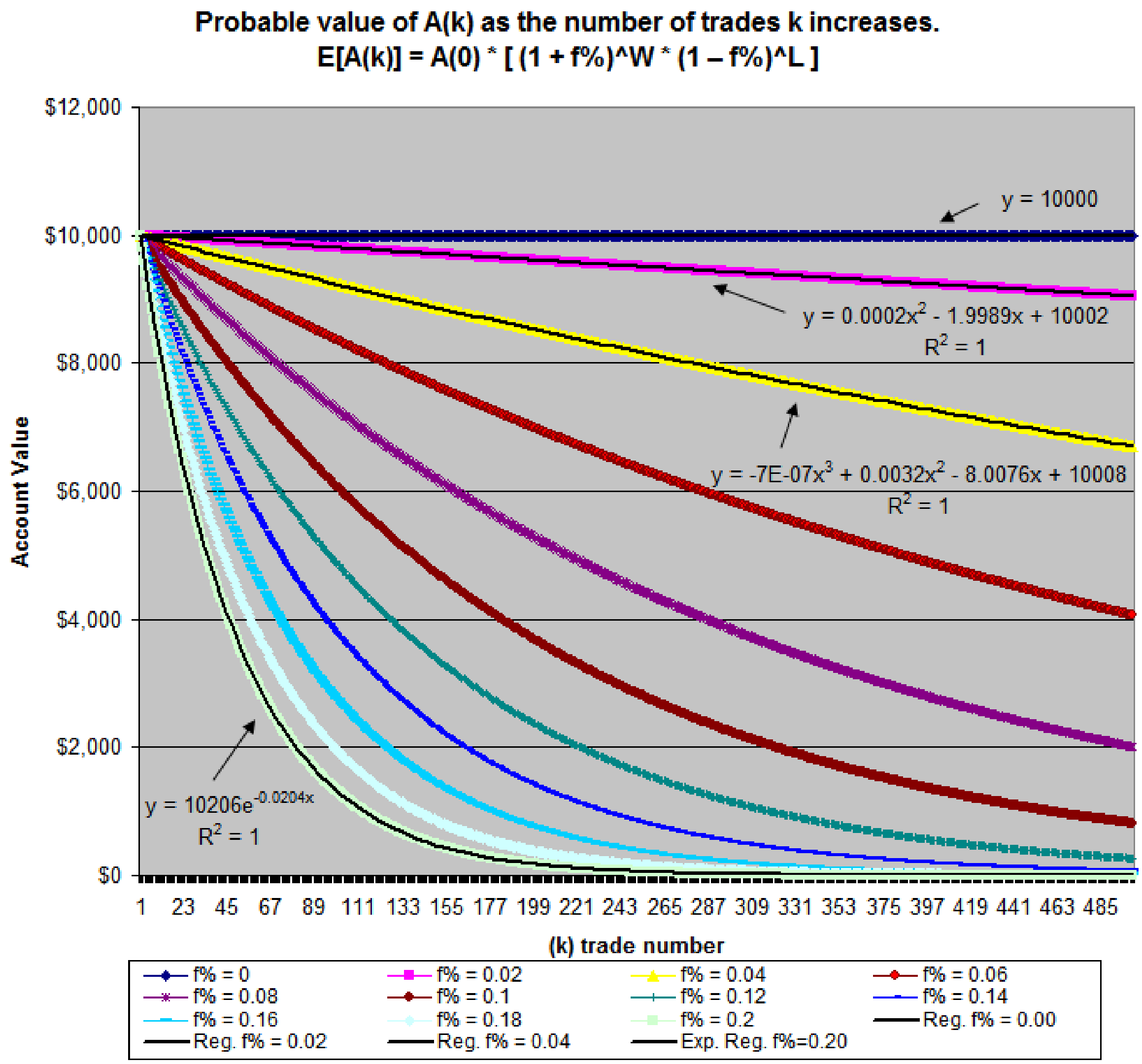
Portfolio Degradation

To show that there was indeed degradation, I analyzed the equal fixed fraction of equity put at risk at the most expected value in a 50% win ratio scenario with W = L = k/2 = 500 over a 10 stock tests of 1,000 trades each. It was a good point to start with. Later I could always study the implications of gradually moving away from the expected mean.

In the chart below, the degradation can easily be seen as the equal fixed fraction f% increases as k increase. The higher the equal fixed fraction, the faster the degradation. The higher the number of trades k, the higher the impairment to the account.

It should have been enough to show that there was indeed account degradation and that the only culprit was the equal fixed fractional position sizing trading strategy.

Fig. 6. Fixed Fraction Percent Deterioration



But it still was not enough. The impact of the equal fixed fraction f at the most probable outcome in a 50/50 game is all what the above charts are about. If you increased f in steps of 0.02 (2%), the degradation would simply accelerate. The study of moving away from the mean could be done later. First, it was important to see the trading strategy's behavior at the most expected values.

No matter what I was putting on the table, blue stuck to his guns: E[A(k)(blue)] = A(0), (the expected value of the game is still no loss he clamored). While I stuck to mine: E[A(k)(red1)] → 0 (the longer you played an EFFPS trading strategy with an equal fixed fraction of equity, the more you would lose, and the higher you put your % profit target, the faster you lost).

It was as if blue could not, or did not want to understand the difference between playing a series of independent random bets and a series of compounded returns on equity under uncertainty. Two different games, it should only be natural that they achieve different answers.

The EFFPS Trading Strategy

The EFFPS is a trading strategy in its own right. It's very easy to set up. In my software, it can be done using under 10 lines of code. The Excel spreadsheet which does the equivalent can simulate the strategy on the whims of the rand() function which does all the work. On my part, not too much: one equation, repeated (copy and paste) 10,000 times.

The equation most representative of the equal fixed fractional position sizing trading method is: A(k) = A(0)*(1+f)^W*(1-f)^L where f is the equal fixed fraction of the current account's profit target or stop loss taken as price moves about. With W being the number of times a profit target was hit, while L the number of times a stop loss was taken. In the spreadsheet, the probability of a W (win) or a L (loss) was set at 0.50, the same as a coin toss, and the same as in a 50% hit ratio scenario.

The variable f is not a percent of portfolio bet, it is a percent of portfolio return, as in:

A(k) = A(0)*(1+f)*(1-f)*(1+f)*(1-f)*(1+f)*(1-f)*(1+f)*(1-f)*(1+f)*(1-f)*(1+f)*(1-f)*(1+f)*(1-f)

where a series of returns are applied to the account, a multiplicative process. And not as a series of won or lost bets as blue is playing this game, an additive process:

A(k)(blue) = A(0) +bet -bet +bet -bet +bet -bet +bet -bet +bet -bet +bet -bet +bet -bet

Red's equation could be simplified to:

A(k)(red1) = A(0) * (1 + f)^7 * (1 - f)^7

while in blue's betting system you would have: A(k)(blue) = A(0) + 7*bet - 7*bet = A(0)

With the fixed fraction f = 0.10, and with initial account A(0) = $10,000, the above series of returns will give: A(k)(red1) = A(0) * (1 + 0.10)^7 * (1 - 0.10)^7 = $9,321. The mean of the 50% winning ratio taken at W = L = k/2 = 7.

For those wishing to see more while not wishing to exasperate anyone since it is the same as in the chart above. I'll present the first 100 trades (k) by steps of ten:

A(0)(red1) = A(0) * (1 + 0.10)^ 0 * (1 – 0.10)^0 = $10,000
A(10)(red1) = A(0) * (1 + 0.10)^ 5 * (1 – 0.10)^5 = $ 9,509
A(20)(red1) = A(0) * (1 + 0.10)^10 * (1 – 0.10)^10 = $ 9,043
A(30)(red1) = A(0) * (1 + 0.10)^15 * (1 – 0.10)^15 = $ 8,600
A(40)(red1) = A(0) * (1 + 0.10)^20 * (1 – 0.10)^20 = $ 8,179

A(50)(red1) = A(0) * (1 + 0.10)^25 * (1 – 0.10)^25 =$ 7,778
A(60)(red1) = A(0) * (1 + 0.10)^30 * (1 – 0.10)^30 =$ 7,397
A(70)(red1) = A(0) * (1 + 0.10)^35 * (1 – 0.10)^35 =$ 7,034
A(80)(red1) = A(0) * (1 + 0.10)^40 * (1 – 0.10)^40 =$ 6,690
A(90)(red1) = A(0) * (1 + 0.10)^45 * (1 – 0.10)^45 =$ 6,362
A(100)(red1) = A(0) * (1 + 0.10)^50 * (1 – 0.10)^50 =$ 6,050

There is a pattern in the above series; it is not increasing. And it does not seem like it wants to return to the mean either.

A single question is shaking the foundation of some accepted belief systems. Sure, betting fixed amounts in a head or tail game, the expected outcome is: no win, no loss. However, playing % returns using percent profit targets and stop losses does not behave the same. There will be a bet size imbalance. A 10% down move is not entirely recuperated by a 10% increase. It is how you deal with it that matters since anyone's trading account could be at stake here.

Anybody would agree that if you only played black at the roulette wheel at a casino with $100 on each bet; where you could lose your bet, or win $98, would not seem like a winning proposition. And yet, when looking at trading in the stock market, where about the same phenomena is at work, some people can't see that it is exactly what they are doing.

Do You Add or Do You multiply?

Since the EFFPS trading strategy had been so simple to program (see Appendix red1 code); could this trading strategy's deficiencies be corrected? The answer is: well yes, no problem.

The account degradation was due to the simple fact that it takes a higher "percentage" to rise from a decline in order to get back to even. To go up by 10% (1 + 0.10) and then go back down by 10% (1 - 0.10) is not an additive process, it is not +10% - 10% = 0; but a multiplicative process as in: (1 + 0.10) * (1 - 0.10). It depends on the player's implemented trading methodology if he wants to use a fix amount betting system or a fix % return system (playing percentage) which are respectively an additive and a multiplicative process.

This very notion is at the center of the debate. Do you use: 2 + 2 = 4, or do you use 2 * 2 = 4? To me, that's the real question here! And it has a simple answer. Do you play partial amounts of equity from account as in a betting system (additive), or do you go for percentage returns on account (multiplicative) resulting in compounding returns? The stock market is a long term compounding return "game".

Anyone can design either a trading strategy where the outcome of each bet is added or subtracted from the account (additive), or as a fixed % return on account, which is then a multiplicative process. Both blue and red may seem to play the same game, but in reality, they are not. It was as if blue could not see that both trading methods could coexist.

In a funny side note, both blue and red1 have the same answer for k = 1 when f = 0.10 and b = 0.10. But it is the only time, the only k value where both strategy give the same answer:

A(1)(red1) = A(0) * (1 + f)   or   A(1)(red1) = A(0) * (1 - f)

A(1)(blue) = A(0) + b*A(0) = A(0)*(1 + b) or A(1)(blue) = A(0) - b*A(0) = A(0)*(1 - b)

For k > 1, the outcome of both strategies diverge as k increases. Note that the price at which each trade took place are not the same. Red1 required a 10% price move while blue needed a 100% price increase to reach his target exit for trade #1.

Blue came back with, and I quote:

> ...but the properties of the random walk regulating the values of W and L mean that with k -> infinity the account will recover to $1,000, $1,000,000, $1,000,000,000 with the probability 1 (100%). This is a clear “drawback” of the formula.

I think everyone understands what is being said in blue's statement. But whatever, here is my interpretation: the properties of a head or tail game make it that it will reverse to the mean as k → infinity, and might even go from $10,000 (the account used as reference) right up to a billion with probability 1, should you want to wait long enough like reaching infinity. Infinite time is really a very very long time to wait...

My response was: could we settle for stuff that can happen while we are still alive, that do not exceed our respective life span? A million years from now has little interest except in documentaries and/or science fictions.

Also, there is no recovery, no return to the mean for: A(k)(red1) = A(0)*[(1+f)^W*(1-f)^L] → 0, for instance, make f > 0.20 and k → infinity. The equation is totally degenerative, even in a 50/50 game like scenario.

Tell the croupier at the casino the next time you continuously play black at the roulette wheel. I assure you the casino will invite you to be their guest anytime you desire. If you want to put up much higher stakes on the table; they will provide free, for you and your guest, the high roller suite ($5,000 a night), free airplane round trip tickets and all you can eat or drink during any of your stays, all this with a smile (kind of their way of saying: thank you).

Random walk properties do not have mean reversions, they don't have any obligation to do so. One can observe mean reversions all the time on past data, but can't predict that a mean reversion will start on the very next move. A random walk (coin flip) only have probabilities, and basically, it has only one: ½ for a fair coin. For any coin flip, for any duration you want, the probability will remain ½ on the next flip. Mean reversion implies that the probability in flipping a fair coin is not ½. Or that a martingale is not really following the definition of a martingale. If in your sequence of heads or tails, you are ahead by say 10, 50 or 100 tosses after an undermined number of flips to get there, your probability of head on your very next flip is still ½. And this means that your expected outcome from the point reached 10, 50 or 100 flips ahead will still be 10, 50 or 100 at the end of the game, and not 0, 0, 0 respectively as a mean reversion would suggest.

However, under a martingale, the expected outcome after 1,000 coin tosses is 0, it's the mean, the average. And if you chart a {+1, -1} sequence of 100,000 random walks for 100,000 coin flips each, you will see it all tends to zero on average. This does not mean that a martingale is reversing to its mean, only that the mean, the average, is a strong attractor as the series converges to the mean from both sides.

The house has an advantage when you play roulette, its origin is simply the zero and double zero. This won't stop you from winning. All it says is that the house is assured over the long term to win on average 1 / 37 or 2 / 38 of what is put on the table. And it is taken from all bets made on all tables, not on a single individual. The house does not need to gamble, discriminate or classify gamblers, even if high rollers do have some hidden privileges.

There is no house in the stock market, but the EFFPS trading strategy produces the same effect. The degradation of the account (for say f% = 0.10) is due to the 1% given away because the trader adopted a strategy designed to do just that, what ever the odds on the game itself may be.

The portfolio degradation is entirely due to the multiplicative nature of the betting system used: the equal fixed fractional position sizing scheme itself. An up trade followed by a down trade will have for value: (1 + 0.10) * (1 - 0.10) = 0.99. And therefore, even after just two bets (k=2) where 1 is won and 1 is lost, the account is reduced to 99% of its original value: A(k=2)(red1) = 0.99 * A(0). I should expect to lose 1% on each pair of bets (where I was presumed to get back to even) and the longer I play this game, the more of those pairs I will get. The account after k trades could look like this: A(k)(red1) = 0.99^(k/2) * A(0) keeping the W = L = k/2 at their most probable values. I'm assured with time that a large fraction of the trade pairs will cancel each other out.

After   10 trades: 0.99^(k/2 =   5) * A(0)(red1) = **0.95** * A(0)
After 100 trades: 0.99^(k/2 =   50) * A(0)(red1) = **0.61** * A(0)
After 250 trades: 0.99^(k/2 = 125) * A(0)(red1) = **0.28** * A(0)
After 500 trades: 0.99^(k/2 = 250) * A(0)(red1) = **0.08** * A(0)

No wonder that the longer you trade using this trading technique the more you are bound to lose.

Correcting the Problem

To correct the EFFPS trading strategy's deficiency problem is very simple. Make each bet equivalent as in having both bets equal the same amount that the price rise or fall.

This translates to solving the equation: (1 + a) * (1 - 0.10) = 1.00. The answer is not: a = 10%. It is: a = 0.10/0.90 = 11.11%. Therefore, correcting for the degradation, it is sufficient to add 1.11% on the winning side to produce: (1 + 0.1111) * (1 - 0.10) = 1.00. And the above scenario of possible trades becomes:

After 10 trades: 1.00^(5) * A(0)(red2) = 1.00 * A(0)
After 100 trades: 1.00^(50) * A(0)(red2) = 1.00 * A(0)
After 250 trades: 1.00^(125) * A(0)(red2) = 1.00 * A(0)
After 500 trades: 1.00^(250) * A(0)(red2) = 1.00 * A(0)

The correction factor needed to escape from a total portfolio disaster is simply: a / (1 - a), in the above case 0.10 / 0.90 = 0.1111. This compensation factor will totally erase the effect of degradation occurring in the EFFPS trading method used (see Appendix program red2). It will also make it a positive expectancy game as a side effect which will be covered later.

One could also compensate for the portfolio degradation by adjusting the percent stop loss fraction. This is finding a solution for the equation: (1 + 0.10) * (1 - a) = 1.00, and this equation will resolve to: - a = -0.090901 = - 9.0901%. The result would be the same as compensating for the decline in price: (1 + 0.10) * (1 - 0.090901) = 1.00. Reducing the stop loss by: a/(1+a) would have the same effect as increasing the profit target by: a/(1-a), see Appendix program red3.

To compensate account deterioration, there are two equivalent alternatives: increase the profit target by: a/(1-a), or reduce the stop loss by: a/(1+a). Each taken individually will totally compensate the EFFPS scheme. And on the scenario above, it means increasing the profit target from +10.00% to +11.11%; or reducing the stop loss from -10.00% to -9.0901%. These two compensation factors are just minor adjustments to be made to any automated or discretionary trading strategy.

Double Compensation

The above adjustments are minor modifications to a trading strategy. Using both at the same time can reverse the degradation process.

This would result in: (1 + 0.1111) * (1 - 0.09091) = (1.1111) * (0.9091) = 1.0101; which would be the equivalent of having a trading edge (see Appendix program red4).

After 10 trades: 1.0101^(5) * A(0)(red4) = 1.051 * A(0)
After 100 trades: 1.0101^(50) * A(0)(red4) = 1.653 * A(0)
After 250 trades: 1.0101^(125) * A(0)(red4) = 3.512 * A(0)
After 500 trades: 1.0101^(250) * A(0)(red4) = 12.336 * A(0)

Two minor modifications, and the trading strategy has made what was a really degenerative process (red1) into a profitable trading strategy with exponential long term growth for outlook. A complete reversal of the trading technique. A trading script only controls the decision taking process, it sets the barrier-like limits in the automated trading strategy. But the price still has to move to where these limits are.

On the two initial requests: can the EFFPS trading strategy (red1) be corrected? Yes, no problem. Use either of the compensated scenarios: CFFPS (red2 or red3). Can it be improved again? Yes, and easily, by using both compensators at the same time, as shown the DCFFPS (red4): the double compensated FFPS trading strategy.

This turns the tide. Using the 2 compensators at the same time, you have changed a trading strategy that was exponentially eroding your portfolio into nothingness due to its negative trade size imbalance into a strategy with exponential growth built in; and now having a positive trade size imbalance in your favor. A major task and yet with only minor modifications to the program code.

This trading strategy (DCFFPS: red4) is now worth playing, from start to finish. It wants to participate fully and wants to take all trades at all times. It will generate positive alpha over its nemesis: the Buy & Hold trading strategy. Note also that all this could be done using pen and paper if desired, no computer required.

The Programs Mentioned

The red programs to do the above use the same basic equations as in the spreadsheet. Adding the compensator(s) change the nature of the game itself, from an equal fixed fraction (red1) to the compensated fixed fraction (red2 or red3), to the double compensated version (red4). This can be summarized in pseudo-code programs (also described in the Appendix).

```
//red's 50% win/loss programs.
Account = 10000;              // Could be scaled to any amount, simply add zeros

//without compensation factor. red1
fw% = 10%;                    // profit target: percent of equity gain
fl% = 10%;                    // stop loss: percent of equity lost

// with profit target compensation factor.  red2
fw% = fw%/(1-fw%);            // adjustment factor: a/(1-a) to correct for degradation
// resulting in fw% = 11.11%,  // for version one, comment out line fw% above

// with stop loss compensation factor.  red3 (not shown in forum)
// fl% = fl%/(1+fl%);         // adjustment factor: a/(1+a) to correct for degradation
// to implement both compensators, remove "//"  on line above
// which will result in red4 (not shown in forum)
```

```
InstallProfitTarget(fw%);      // Stock rises, take profit, sell
InstallStopLoss(fl%);          // Stock falls then take loss, sell
Do while until you decide to quit or 12,500 trading days (50 years)
If NoActivePosition then       // take one
 BuyAtMarket Q shares =  Account/P(t);
// with total cash in account buy Q shares on next bar
```

Here is a simplified version of blue's trading strategy:

```
//blue's betting system program, // version two
Account = 10000        // Could be scaled to any amount
b = 0.10               // can be modified: 0 < b < 1, percent of account put at play
Bet = b*account        // amount bet on each trade
InstallProfitTarget(b*account);    // Stock gains amount = b*account, take profit
InstallStopLoss(b*account);        // Stock falls by amount = b*account, sell, take loss.
Do while until you decide to quit or 12,500 trading days (50 years)
If NoActivePosition then           // take one, buy for b*Account of shares
.BuyAtMarket Q shares =  b*Account/P(t);      // use 10% of account for next bet
```

To recap, these programs have set trade barriers that if crossed trigger execution of either their profit target or stop loss under their respective conditions. Red uses a fixed percentage of account for profit target or stop loss, while blue's entries are determined by a fixed percentage “b” of the account size as betting unit. Each having their respective accounts: A(k) (blue) and A(k)(red#) where A(k) stands for the account value after trade k and where k is numbered from 0 to 1,000. A(0) is the initial account size, in this case, A(0) = $10,000 for both players. Note that both accounts could be scaled to any amount.

Red made program modifications designed to compensate for bet size imbalances, and armed with these modification, can chose simple compensation: (CFFPS: red2 or red3), or double compensation (DCFFPS: red4).

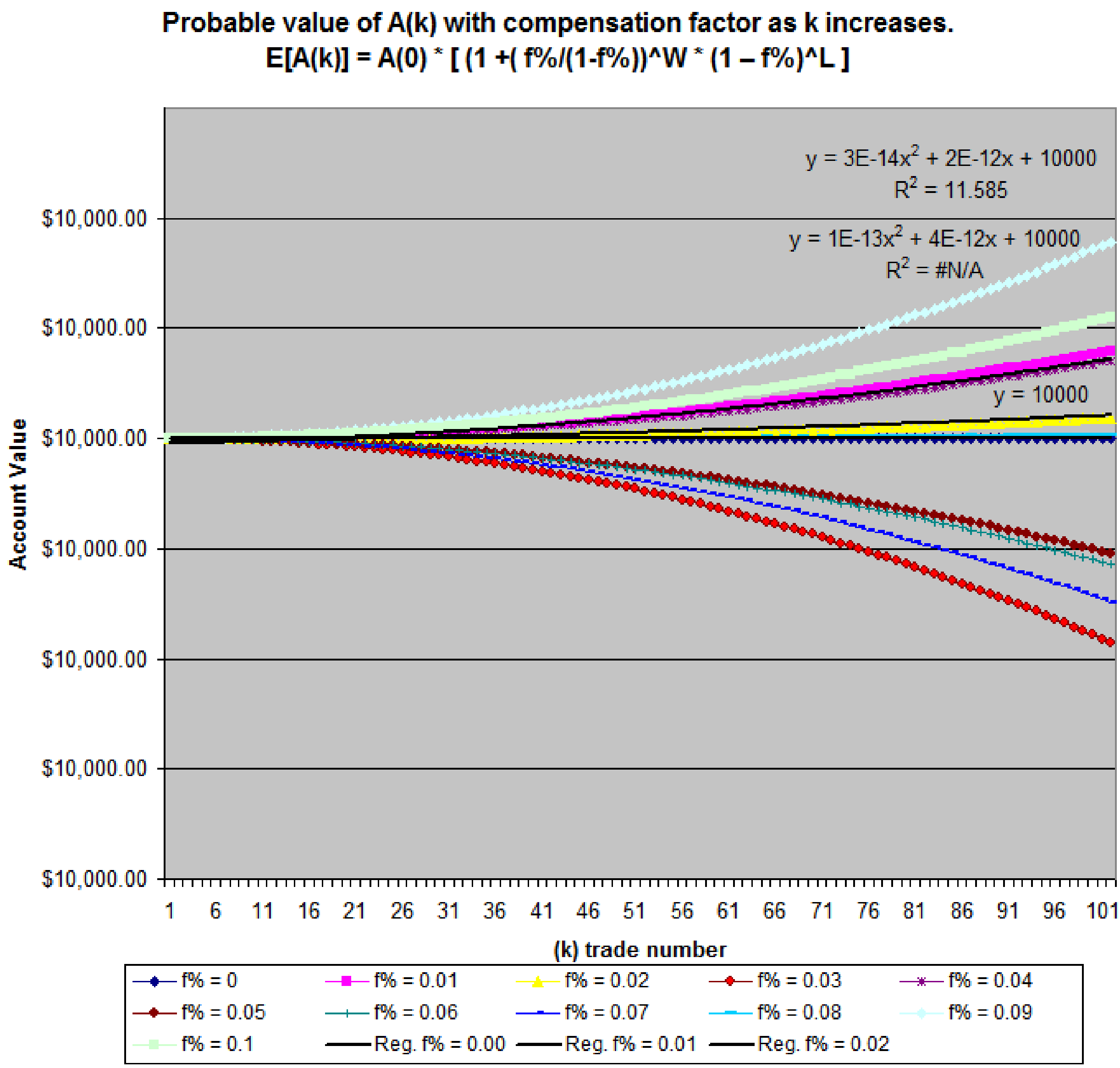
Meanwhile, blue was still forcibly holding on to his guns and was still pushing more adamantly than ever that the expectation of the game was: [A(k)(blue)] = A(0). A no need to play the game at all since no expected benefit could be gained. Still not understanding that playing a compounding game is not the same as playing a fix amount betting system.

Using red's programs, it was shown that no one should play using red1; it was a portfolio destroyer. It was not a question of holding on to the original capital, it was about losing it all. However, red's programs, red2 and red3, could compensate for the deficiencies of the red1 program. Making them worthwhile trading, but in a funny way, it was found that only the first trade of all mattered; a kind of do trade, but don't. The bad scenario of red1 was being converted into a Buy & Hold strategy having the same properties.

For the sake of correctness, adjustments were made to the spreadsheet to compensate for the red1 degradation. The following chart (Fig. 7) shows the residual error term on the 1,000 trades based on: [A(k)(red2)] → A(0); where the profit target compensator CFFPS was applied.

Note that the error term is expanding as the number of trades increase. However, it is not likely to influence any of the calculations since they are well below a penny even after 1,000 trades. The chart shows that with time, the residual error term increases due to rounding errors, and that at no point is this error term significant (E-12).

Fig. 7. Fixed Fraction Deterioration Compensation



It's all about Expectations

There are now 5 trading strategies; one by blue and 4 by red. Blue is expecting to stay even after the 1,000 trades while red has 3 solutions to the problem. So to resume, here are the equations for all 4 programs.

Blue has: [A(k)(blue)] = A(0). The no change scenario, and no need to trade at all.

Red's EFFPS: [A(k)(red1)] → 0 is a disaster in the making. A please don't trade scenario.

Red's CFFPS: [A(k)(red2 or red3)] → A(0). A total recuperation from red1's deficiencies resulting in: do trade, but only place the first bet, and then hold since it is equivalent to a Buy & Hold strategy.

Red's DCFFPS: [A(k)(red4)] > A(0). Creates a positive trade size imbalance which has for side effect to generate alpha. A do trade and take all the trades as it is an improves with time proposition.

Each of these strategies have end of game expectation. However, blue's scenario in real life is not made to perform as intended. It's like blue has not analyzed what his trading strategy implies. Especially since at one point in the debate blue did emphasis that f = b. Where I stated that the variable f was not equal to b, even if f = 0.10 and b = 0.10. The usage of each was different, representing different things, different notions. I'll be back on that.

To restate each player's equations:

Blue's stated equation had for expected value: E[A(k)(blue)] = A(0)*[(1+b)^W*(1-b)^L] = A(0). I stated before that this equation did not represent his trading method. Blue needs another equation to express his betting method. While the result: E[A(k)(blue)] = A(0) hold for his described trading strategy. However, it will be shown that even that does not hold up in real life.

Red's equal fraction program red1: E[A(k)(red1)] = A(0)*[(1+f)^W*(1-f)^L] → 0; proved to be a do not play for any reason program as it had for outcome the lost of the portfolio. In this sense, red1 is a lot worst than blue. This led to red's compensated program red2: E[A(k)(red2)] = A(0)*[(1+f/(1-f))^W*(1-f)^L] = A(0). Red2 and red3, are trading programs worth playing, however, the only trade that really matters is the first one since both strategies are equivalent to a Buy & Hold.

Time

The spreadsheet experiment did not consider time. Its only concern was trade execution. This enabled the analysis of the decision making process without loss of generality. But the market does not operate on the immediate result of a random function. It takes time to reach a profit target and/or stop loss.

What real market data will do is put back time in the equation. There will be no need for a random number generator since the market data itself will provide the random-like price movements.

The expected outcome of the random function obeyed all the laws of coin flipping where the most important one was that a toss would have a probability of 1/2. The advantage of the rand() function is no edge can be extracted and no prediction made show any value. If you won (lost), meaning having more heads(tails) than tails(heads) over the 1,000 trade process, it would be by luck(bad luck) alone.

The randomness of a random number generator is not in question here, what is, is what the outcome of the trading methodologies following their respective trading rules will be? It's the trading strategies themselves with their respective holding functions that make a difference in a trading methodology.

Therefore, applying the trading programs to real market data is the only way to show the real value of these trading methods. Real market data is not 50/50, on average, the stock market has a long term upward bias. Even in a coin flipping environment, this up side bias will show.

The Payoff Matrix

Since any trading strategy can be expressed using Schachermayer's payoff matrix, I'll use the function showing the evolution of blue and red's respective portfolios as: A(t)(blue) = A(0) + Σ(H(blue).*ΔP(blue)), and A(t)(red#) = A(0) + Σ(H(red#).*ΔP(red#)) with all starting with an initial account A(0) of $10,000. Note that the initial account could be any amount, it's totally scalable. Comparing trading strategy will be as simple as comparing: Σ(H(blue).*ΔP(blue)) to Σ(H(red#).*ΔP(red#)) which is the expression to represent the total generated profits or losses by any trading strategy.

Here, ΔP(blue) and ΔP(red#) use the **same price series** P(t), but slice it differently due to their respective entry and exit points. All have the same data series to contend with and that is P(t). If blue or red buy some shares at the same price in the series, they both have to pay the prevailing price. Here ΔP should be viewed as the entry and exit price difference matrix of 1,000 trades by 1, 10 or 100 stocks, (making the ΔP matrix a 1,000, 10,000 or 100,000 elements data array) but that is not the point being tested or debated here. I will use the same price series P(t) for all the trading strategies, they can slice it according to their respective trading procedures. This should prove sufficient for the purpose at hand.

The Inherent Degradation

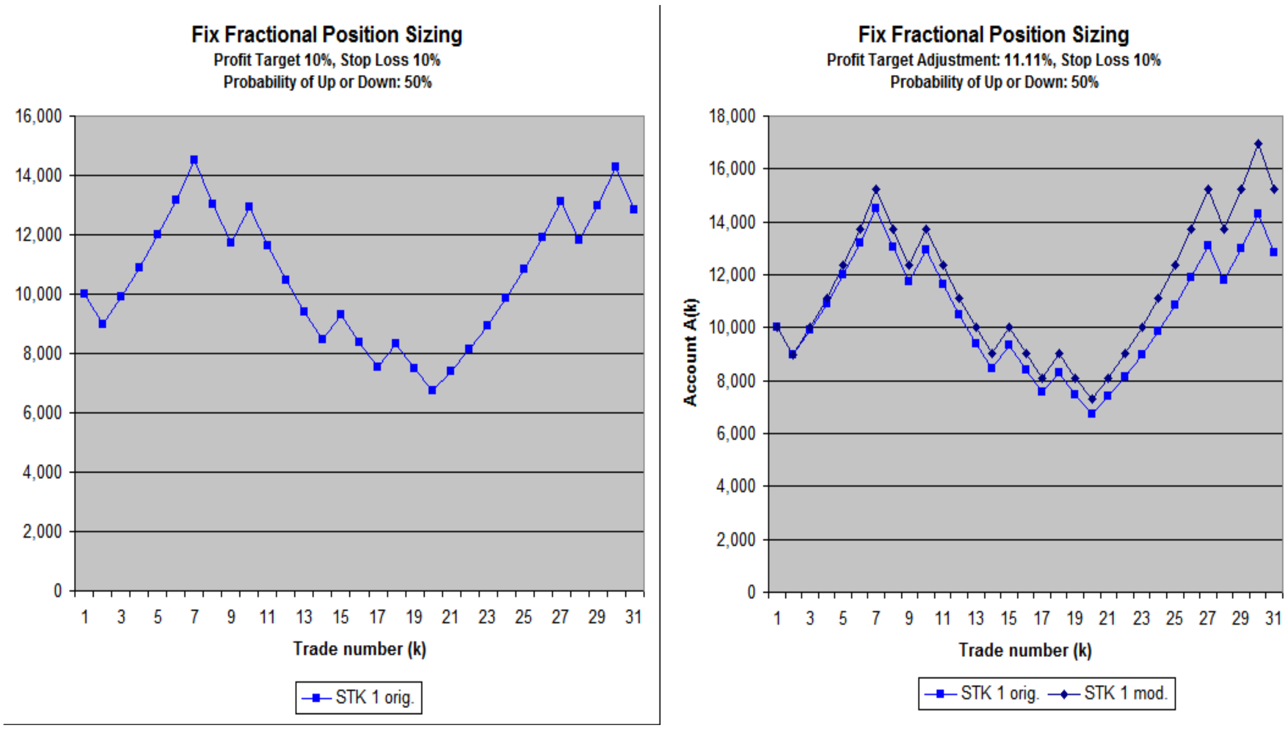
It was demonstrated that for red using the simple compensation factor as illustrated in program red2 that any down move was well balanced with an appropriate up move made to cancel each other out. What ever the price does going forward, it can only hit one of the barriers at a time, either the profit target or stop loss. I will limit initial comparisons to programs red1 and red2.

With the compensation factor in place, red2 will have the up move after a down move equal the same dollar amount value. This will vary depending on the price level reached, but still at each level, the up move after a down move will have the same dollar value.

The chart below shows on the left side the first 30 random trades taken according to the trading rules of red1. One would be hard pressed to say that there is degradation as it looks like any ordinary chart with price variations randomly generated. But what does not show on the account value chart on the left can be viewed using program red2 the modified version (CFFPS) made to compensate for the red1 degeneration.

The graph on the right is based on the same series of coin tosses with the same sequence of up and down moves but with the compensation factor in place (red2). The compensated curve shows divergence right from the start, it can be measured: it's simply the difference between the two curves. The CFFPS trading strategy is simply waiting for a higher profit target, meaning a slightly higher price for its trade execution, and this is what is showing on the chart on the right. Each time red2 hits a profit target, it was asking a slightly higher profit. And the sum of these additional profits would accumulate and show up in the final results.

Fig. 8. Fixed Fraction Plain and Compensated Chart (Profit Target: 11.11%)



You have both programs (red1 and red2) operating on the same price series. Both going down 10% on their stop loss with red2 compensating with an 11.11% profit target; while red1 only waits for a 10% recovery. This divergence will increase as explained previously at an average compounding rate of about 1% per corresponding trade pair.

That one trades using red1 or red2 won't alter the price series on which the trades are based only their entry and exit price points. What may differ is the time at which these trades may take place.

What ever the price may be, these trades will take place at an ever increasing divergence, with the modified red2 keeping its upper hand for the duration of the 1,000 trades. The divergence is a consequence of the position sizing asymmetry, a trade size imbalance which is being corrected by the 11.11% profit target while maintaining its 10% stop loss for down moves. The above tests were performed on randomly generated price data, but the same principles would apply to real life data.

Stock Prices

The stock price in Fig. 8 is a scaled image of the account value as profit targets and stop losses are executed over the portfolio's history. Already, in the compensated red2 program, the expected account value was set as:

E[A(k)(red2)] = A(0)*[(1+f/(1-f))^W*(1-f)^L] = A(0)

Being a scaled image, the stock price will also have the same series of up and down moves, in the same order, for the same % moves. Therefore, the expected price series at which trades will be executed for red2 will be:

P(k)(red2) = P(0)*(1+f/(1-f))^W*(1-f)^L = P(0)*(1+0.1111)^W*(1 - 0.10)^L = P(0)

The charts in Fig. 8, not only shows the account value as trades increases, but also the actual required price scaled by a constant: d * A(k). From Fig. 8, for instance, dividing the y-axis by 100 or 1,000 will give the prices ($100 or $10) at which the initial trade took place.

You know the account value at A(k), you know the price P(k) at k, you also know the quantity of shares on hand at k, A(k)/P(k) = Q(k). You only need to fill in the numbers (k, W, L and f) to get the answer. Knowing how stock prices move over the long term will somehow provide limits to the expansion of red2.

The Trading Environment

Red's trading strategies, red1 and red2, when suffering a 10% down move, suffer a 10% decline. Both their accounts are reduced by 10%: 0.90 * A(k)(red1), or 0.90 * A(k)(red2). On the winning side, red1 rises by 10.00% while red2 waits for a rise of 11.11% resulting in: 1.10 * A(k)(red1) and 1.1111 * A(k)(red2) respectively.

Both programs are based on a fixed fractional position sizing scheme, red2 compensating for red1 deficiencies. For red1, one loss followed by a win, and starting at n with n=k anywhere in the 1,000 trades sequence would show: A(n+1)=(1-0.10)*A(k)(red1), A(n+2)=(1+0.10)*A(n+1)(red1). After a loss, a winning trade would result in a 10% rise for red1: a partial recovery.

The down trade had for effect: (1-0.10)*A(n+1)(red1) = 0.90*A(n+1)(red1) and rising from there would give: A(n+2)(red1) = (1.10)*0.90*A(n+1)(red1) = (0.99)*A(n+1)(red1) = (0.90+0.09)*A(n+1)(red1), a real increase of only 9% in its attempt to recuperate. So a stock goes down by 10% in both versions, but red1 only recuperates 9% of every loss made. You could view the problem as when looking at a series of 1,000 trades, for red1, and for W = L = k/2, as a set of 500 trades showing 10% declines and 500 trades showing a 9% recuperation.

You lose 10% of your account and win 9% on your account in an attempt to recuperate in a trading strategy where the hit rate is 50% and you are doomed. It is just a matter of how many trades are play. There is nothing in the random function to help you out of this hole. All this does not stop you from being lucky, but this “luck” will have to compensate for the continuous degradation due to the trading method itself, no matter what.

You won at the roulette wheel after multiple bets, you still paid the house's advantage.

Each win after a loss results in a 9% recuperation for red1 while each win gives red2 a total recuperation for its decline (factor = a/(1-a) = 11.11%) bringing it back to even. The red1 program has symmetrical equal fixed fraction giving it a trade size imbalance and it is this imbalance that will slowly destroy its portfolio.

The equation for this degenerative process would be: e^(-(1-0.99)*(k/2)) = e^(-0.01*(k/2)). The divergence in the beginning is small, as shown in the above chart, but the function will slowly trim down the account as k increases. And this is entirely due to the equal fixed fraction position sizing scheme. It says that an automated, or a discretionary trading strategy for that matter, using equal fixed fraction position sizing is doomed from the start. And raising the equal fixed fraction to a higher value will only accelerate the portfolio's demise as shown in red1.

This degradation of the portfolio value is only due to the number of executed trades and the fixed fraction used, it has nothing to do with the randomness of trades. And this is something that entirely escapes blue's vision of the problem. The trading strategy program red1 has for expected value E[A(k)(red1)] → 0 while red2 has: E[A(k)(red2)] → A(0): a complete compensation of red1 deficiencies. Even here, red2 should more appropriately be given a higher expectation: E[A(k)(red2)] > A(0). To be discussed later.

The Compensator

If a compensation factor is required to force, correct or convert what was red1 into red2, doesn't that say that a correction factor was needed. And if after applying a correcting factor, the deficiency disappear, doesn't that say that there was in fact a deficiency in red1 in the first place, especially if 100% of the deficiency disappear simply by applying a small compensation factor.

One is not analyzing the randomness of the output of a random function, but analyzing a trading methodology using fixed fraction playing a seemingly random game. Using an equal fixed fraction of equity trading strategy won't change the final destination but just how randomly you will get there.

The expected value of the equal fixed fractional position sizing trading strategy (red1) has for expected value: E[A(k)(red1)] → 0. As k approaches large numbers (for say k>500), there may be no other solution. You may delay the outcome by using lower values of f, but by doing so, you increase the number k of trades. The degeneration is not due to the randomness of trades but to the very nature of the trading methodology itself.

In a winning scenario, this degradation might seem like not being there, like in the chart above, but it is there. It is a price you pay for every down trade. You have a stop loss, you just lose that 1% that you won't recuperate ever, except by chance. You play a sequence of 1,000 trade where stop losses are expected to number about 500, you will pay the 1% degradation fee on those 500 trades. And having had 500 losses, the 500 wins can not compensate for the degradation. Only an edge in your trading script (W > L) or pure luck could save you.

One could compensate for the deterioration by having more wins (W > L) which was not discussed much simply because I was not there yet. But here it goes, again after 100 trades with 55 and 56 wins:

A(100)(red1) = A(0)*(1 + 0.20)^55*(1 - 0.20)^45 = A(0)*0.98632 = $ 9, 863.20

A(100)(red1) = A(0)*(1 + 0.20)^56*(1 - 0.20)^44 = A(0)*1.47948 = $ 14, 794.80

You would need an “edge” in the game of at least 5 trades (W - L > 10) to compensate for the inherent degenerative process of the fixed fractional position sizing scheme. You would need a win ratio of 56% to compensate for red1 degenerative process.

Providing an “edge” is the sole purpose of designing automated trading strategies.

There are not that many variables in the equation. The other way to compensate is to change the percent profit target itself. I've provided the factor needed to do just that: a/(1-a). Doing so, will produce a total compensation of the degenerative process:

A(100)(red2) = A(0)*(1 + 0.25)^50*(1 - 0.20)^50 = A(0) * 1.0000 = $ 10,000.00

Since, you managed to design a trading strategy with an edge (see above), it could still be applied and produce:

A(100)(red2) = A(0)*(1 + 0.25)^56*(1 - 0.20)^44 = A(0) * 14.5519 = $ 145,519

And finally one sees a positive scenario. The equation is well suited for this kind of analysis. All the above is simple elementary strategy design.

One can be lucky and have a sufficient number of trades to coverup the red1 deficiency but it does not remove it. Just like winning at the roulette wheel won't remove the house's advantage. The higher the number of trades (k) using red1, the more it will cost you; to in the end finally eat up the entire portfolio. If k is large enough or if the equal fixed fraction is higher it will most likely accelerate the degenerative process.

It is not: I am not subject to this degenerative process. It is you use an equal fixed fractional position sizing scheme, or something that looks or works like it, you almost have a guarantee of seeing you portfolio slowly disintegrate due to the misconception that 10% down is equal to a 10% up. If any of your trading programs use such a technique, my advice is: modify them pronto. If you use this technique in your discretionary trading method, don't think you are exempted for this; it will do the same thing as in an automated trading script. The correcting factor for the profit target is just: a/(1-a). For a profit target of: a = 10%, this correcting factor will be to set the profit target to 11.11%, only a minor modification that will save your account from the long term degeneration of the EFFPS trading strategy.

There are many forms of FFPS schemes out there, they are all affected by this. Your trading strategy uses profit targets of any kind, stop losses based on some kind of indicator or what ever, then maybe the first thing to do is review your code and simulate what would be the outcome after a number of executed trades. It's not just looking at one trade ahead, it's over a hundred or more trades to come. Why play this kind of game if you are bound to lose, no matter what you do? What's the value of your trading strategy if over the long term its destiny is portfolio oblivion?

As an added argument to make his point, blue, after many references and citations (Newton, Bernoulli, …): "...I had several years to think on what I said there." Well, may I say: me too.

The Possible Outcome

For red, using the compensated program (red2), to show a profit, it seemed required to at least have W – L ≥ 1. Meaning that over a 1,000 trades scenario, one win, a single win advantage, at any point in the series is sufficient to show this profit: A(t)(red2) = A(0)*(1+cfw) where cfw = f/(1-f), is the profit target compensation factor. It's the same as being ahead by a single positive coin flip over 1,000 tosses. The same as if your very first profit at (k=1), the very last one at (k=1,000), or anything in between (1 < k < 1,000) was a win for the game. A single trade, in a single position producing an advantage: W – L > 0 which could take years or decades to develop.

On this basis, if red2 ends up with say: W – L = 5, thereby being net 5 wins over the 1,000 trades, his net portfolio value would then be:

A(t)(red2) = A(0)*(1+cfw)*(1+cfw)*(1+cfw)*(1+cfw)*(1+cfw) = A(0)*(1+cfw)^5

no matter where in the series of the 1,000 trades only those 5 trades gave red2 its advantage. What ever the path taken to get to quitting time or end of game, where k = 1,000, only the net number of wins and losses would matter. The path taken to get there does not matter much, since you can't do anything about it, and you don't know which of the 5 trades in the 1,000 will stay as the net trades gained. There are after all over E+301 paths to reach the final destination point at k=1,000. There is no way an individual, or what ever organization, or all the computers in the world could select the one best path, the one in E+301, the ideal path that should be taken, or for that matter, which one will become reality.

The future only happens, and there is only one version of that. It is only after, that you can see which path was taken to get there, not before. Before, you can only make educated guesses.

You could make all the estimates you want, set probabilities on every move, put the tree diagram on paper (this one should be a lot of fun, paths > E+301.), calculate all the possible combinations or permutations of moves. It is all totally useless.

In a stock market trading game, all you can do is live up to the plate, make your “best” selection, then make your bet and manage it. The future is unknown and will happen only once no matter on how many possibilities you want to account for. Trying to assess the probability of the 10th or 100th move ahead in a coin flip game is just futility. The probability at the 9th or 99th spot is also 0.50. So don't count that it will help you in any way. The number of possible paths will be 2^10 or 2^100. At any point in a 50/50 random series, the next bet is still 0.50; if and only if, the underlying data series is also a 50/50 process as well. And this is a major consideration.

The Big Question

Using the red2 program, you could have 495 down moves distributed randomly in the 1,000 price series that could be canceled out by the 495 up moves of equal value and win in aggregate the 10 remaining bets distributed randomly in the series giving you a net 5 bets ahead. So you are still, even after 1,000 trades playing the basic win/loss ratio game with the same expectation, for up or down moves as if playing a single coin flip. The value of each bet, even if it depends on the sequence of trades is a matter of the trading methodology itself.

The way you bet is a trading strategy, a method and what you want to know is: if I apply my trading method, or my gambling strategy to what seems like a totally random phenomena, will it show profits? That is the question. It is the only question of interest. Will this make me money or will I lose it?

Then, in the debate, someone came up with a gem:

> *If the coin is fair then your expected equity will be unchanged in the long term. Whoever said it goes to zero by it's own nature is wrong. I built a simple model to prove the notion that "a 50% loss requires a 100% gain" is a fallacy of short sited thinking.*

Now imagine that he was not alone...

You read things like that and wonder where the old bean counters are? Has any of those people ever seen 2 + 2 = 4? Let's see now, you had 4 apples, you lost 2 apples (50%), therefore you have 2 left. Then how many apples do you need to get back to your 4 apples? Let's see now, give me a minute or two. Two, two..., yeah that is the answer. I'm almost sure... well, in my modest opinion, I would bet it is 2 apples are required to get back to my original 4, in all probability, a.s. Since you have 2 apples left, and that you need 2 apples to get back to your 4 apples, what percentage does that represent? Oh, let's see now, that's a toughy: I have 2 left, I need 2 to have 4, after all I can add you know, so that must be 2 over 4, then all I need is 50% more apples and I am there, the same 50% I lost. I'm not that short sighted you know. Sure... It's discouraging to see things like this. But then on the other hand, I can see a portfolio manager say to his client: you noticed we're down 50%, so we'll wait for a 50% rise to recuperate the loss. It must be less daunting for sure.

Blue, just as red, just as anyone else in the forum for that matter, knew very well the outcome of a coin flipping series. Randomness is not in question. The trading strategy is.

A randomly generated price series, like using Excel's rand() function, is an excellent addition to the methods that can be used to test trading strategies of all kinds. You know you can't win the coin tossing game no matter how long the series may be, except by luck. And should your trading strategy be confronted with a coin tossing series which will decide if you move up or down, it is then up to your trading strategy to live up to it and show its merits. If it wants to shine, it will have to survive, and this can only be done by managing the position sizing applied to the randomness seen in stock price variations.

To get back on track, you could also have at k = 1,000, W – L = - 5 where A(t)(red2) = A(0)*(1-fl%)*(1-fl%)*(1-fl%)*(1-fl%)*(1-fl%) = A(0)*(1-fl%)^5. But due to the length of time required to reach a large k like 1,000, this last scenario has lost much of its probability of occurring. Not because it can not occur, but simply because it took quite a while to get there. I'll explain later.

To end with W – L = 5, or W – L = -5, you have 1,000 trades to go through to reach the finish line. It therefore becomes a very long term endeavor with probably no stop time other than: do you want to quit the game or not?...

For k to increase in real life, you need the trading procedures to hit their respective barrier-like limits, time and time again: either with profit targets or stop losses. These barriers will be hit seemingly at random, you don't know the future, but you do know that an up move of 11.11% in a stock price is not a rare occurrence, nor is a 10% decline. You will be able over the life time of the trading strategy to hit these barriers repeatedly, and therefore see k increase trade by trade as time progresses.

The Output

What might not have been noticed by some is the output of red's compensated trading strategy (red2 or red3). It's in the market all the time. It has 100% market exposure. Compare this to blue's 10% market exposure.

As soon as red2 sells for what ever reason: profit target or stop loss, it immediately reestablishes a new position, sets up a new profit target and new stop loss. Being in the market all the time, it gains the same properties of any long term investments: full market exposure for the duration of time it takes to do 1,000 trades with red2 seeking 11.11% profit targets with 10% stop losses. The strategy is not expected to hit any of its limits every 10 minutes either, it will take time, and a lot of it.

The stock market over say a 50 year period (see the do while in program red2) has shown no negative returns for any 50 year rolling window since its creation: US market (1792). It's the same for any 25 or 30 year rolling window. Playing blue or red's red2 or red3 program versions for the long term will result in those strategies to exceed their respective initial accounts with a probability asymptotic approaching → 1 as the number of trades k increases to larger numbers. Meaning that blue: E[A(k)(blue)] > A(0), and red: E[A(k)(red2)] > A(0) as well; for the simple reason they stayed in the market long enough (a kind of participation prize).

The Surprise

What might not have been noticed is that red's programs (red2 and red3) camouflage a big Buy & Hold investment strategy in a robe of random-like trading. Except for the very first position taken, the trading itself is of no consequence. You would get the same results that you traded or not after the initial purchase. It would be like buying index funds for the long term like SPY or DIA for your 401(k) or IRA account.

The random trading activity does not matter that it be performed or not. It won't change the final outcome. It's kind of an illusion, something that can be done just for the fun of it. Since the strategy is made to last k = 1,000 trades, and that W – L will in probability be greater than zero, you finish not with an expected value of A(0), but will more probably be almost forced to accept: n * fw% compounded returns that are bound to be generated over the life of the portfolio.

To give an estimate of the expected value of the game, it might be sufficient just to gather a ball park figure of what might be close to the average of market indexes over past decades and see how many fw% could be squeezed in, from start to finish, on say a 50 year investment period.

This will give: A(k)(red2) ≈ A(0)*(1+fw%)^n and not A(0). Wishing to translate it over time instead of over trades could give: A(t)(red2) ≈ A(0)*(1+r(B&H))^t where the expectation of the portfolio value at end time t with average market return r(B&H) is more what you are seeking than the value at the end of the series of k trades where k = 1,000.

It is why the spreadsheet considered 10 stocks at a time; it was to average out volatility and randomness implied in all stock trading activity. You might have a dude, a super star in the lot, or just average performers, it would all average out. Playing a single stock is a more risky business, but this risk can be mitigated somewhat when trading 10, 50 stocks, indexes and/or certain ETFs.

Again, all the random trading for red's programs (red2 or red3) became irrelevant after the initial purchase. The distribution of the coin flips was irrelevant for these programs. The only trade of importance was the first one. It somehow became the only one of real interest.

Red's programs, red2 and red3, as well as blue's program end up as simple Buy & Hold equivalents with the same Buy & Hold attributes. You could just execute the first trade, the initial position, then sit down and wait, or perform unnecessary trading in and out of positions which would prove in the end to have not added anything after the initial purchase. Red2 and red3 would have 100% market exposure for blue's 10% portfolio exposure.

Both of red's trading programs, by their very nature, can not have for expectation A(0), mathematically or otherwise. Every additional net move of 11.11% up over a 50 year trading window would translate into A(0)(1+fw%)^(n+1) for red2 (with net up moves: W – L = n); all other moves would cancel out. And the higher the number of times a net 11.11% move up can be divided into the end points of a series like the Dow Jones or the S&P over say the past 50 years will give you an estimate of n. This is like setting profit levels, each new higher level being 11.11% apart, and n counts how many of these are in the index price series. The chart below (Fig. 9) illustrates the point.

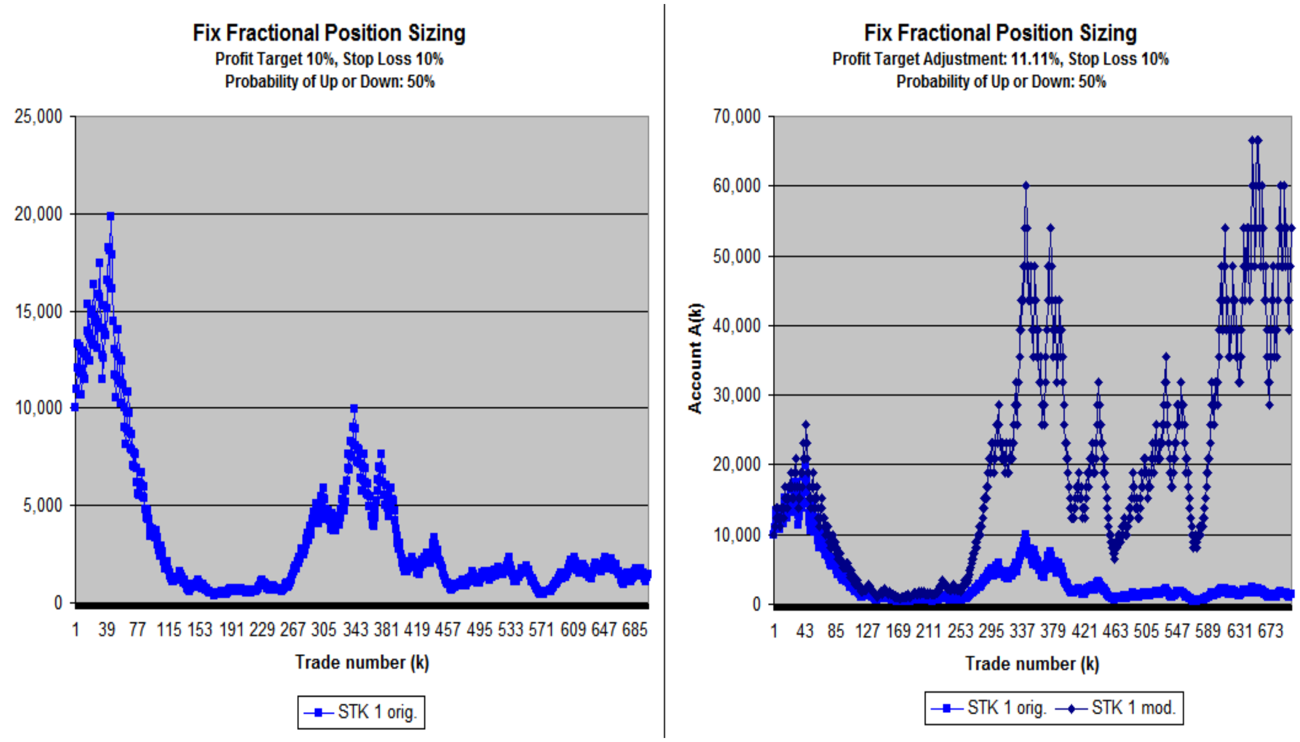
Fig. 9 covers k = 700 trades for red's red1 and red2 programs with only the compensated version (red2) having any value. The chart on the left shows red1. The chart on the right makes a comparative of the output of both programs. Only a slight modification (a/(1-a)) was required to achieve red2's performance level.

Notice that, being compensated, every trade up or down are of the same size at the same level n. They act the same as a coin toss, up 1, down -1, each with an equal amount. A consolidation range will have alternating coin flips (1,-1,1,-1,1,-1,1,-1,1,-1,1), a swing up will have more ups than downs (1,-1,1,-1,1,1,1,-1,1,1,1), while a swing down will show more downs than ups (1,-1,-1,-1,1,-1,1,-1,-1,-1,1). A spike up will have more ups (1,1,1,-1,1,1,1,-1,1,1,1), and a spike down will have more downs (1,-1,-1,-1,1,-1,1,-1,-1,-1,-1).

This is the same as you can find in any generated coin flip series. None of the coin flips are predictable and what ever pattern they may make is also unpredictable and has absolutely no relation to any of the previous flips. So, even if you see something in the data, there is nothing there, absolutely nothing that can help forecast what is coming. The next move, and every move in the chart had a probability of occurrence of 0.50. However, one should note that modeling the stock market as a 50/50 game of chance is just that, a model and not an accurate representation. Also note...

> *The nature of coin flipping does not change because you are the one flipping.*

Fig. 9. Compensated Chart (Profit Target: 11.11%)



The Trading Procedure Comparison

The important aspect of the chart on the right (Fig. 9) is in the trading itself, the compensated fixed fractional position sizing scheme (red2) resulted in every trade executed at level n to be at the same price no matter in which position they were in the sequence of the 1,000 trades (simply connect the dots horizontally). This has major significance. It is a game changer.

The only point of interest in the chart becomes n, the net number of levels reached. It's the net count of wins: W – L = n. Any new fw% up move to historical high, will increase n by 1. But it is still at k = 1,000, or at t = end of game, what ever n is, that accounts will finally be settled.

The trading is only for the entertainment. You are getting in and out, at random times, with no net real value added except when n increases, but this could have been done by just holding onto one's initial position for the duration. It turns this trading strategy into the equivalent of a buy and hold investment strategy.

This means that red using his modified program (red2), and being in the market all the time, with have for payoff matrix: Σ(H(red2).*ΔP(red2)) = Σ(H(B&H).*ΔP(red2)). It won't matter how things swing around. Except for the initial purchase, all the buying and selling at random times will have no additional value to the end of game (k = 1,000) as shown on the right chart above. However, one could declare a stop time (like quit the game) at any time in the series, at any k (1 < k < 1,000). The value of the account will be A(k) at that time, either: > A(0) or < A(0). It will be subject to the path taken to reach end of game at k or at t.

Red's compensated trading strategy (red2) is just a glorified Buy & Hold trading strategy hidden in some code to give the impression that it is working hard at trading for a living kind of thing. When in fact, its value depends on W – L > n. On those 1,000 trades, only a minority will matter.

But what will matter most is that very first trade, that first buy showing that participation in the game was more important and more valuable than trying to figure out how prices moved about. In an indirect way, it validates Mr. Buffett's view of the stock market: buy the best stocks you can and then manage your investment.

Blue's Scenario

Blue's scenario is a little different, it is a nuance, a variation on a theme of red's Buy&Hold scenario.

The generated profits will be scaled by "b = 0.10", (blue's most cited value) as the fraction invested in each trade: Σ(H(blue).*ΔP(blue)) = b * Σ(H(B&H).*ΔP(blue)). And therefore, the comparison of both trading strategy is very simple:

Σ(H(B&H).*ΔP(red2)) **>** b * Σ(H(B&H).*ΔP(blue))

The blue scenario has another particularity, it can only have rare losses. The strategy design requires stock prices to double to take a profit target of +b*A(k) which in the stock market takes time. But the stop loss - b*A(k) requires for the price to fall to zero to be executed. And therefore, only on the stock's bankruptcy will a stop be ever executed.

Blue will have a series of winning trades until a stock goes under; and then, will lose only the very last bet of the series. But nothing harmful since after just 3 trades, k > 3, blue can not lose enough to make the portfolio negative. Any series with k > 3 will have [A(k)(blue)] > A(0), and as the number of trades continue to increase over the years, the more [A(k)(blue)] will continue to be greater and greater than A(0). This is like saying that blue can only lose 10% of his portfolio, and this will require that his first stock selection goes bankrupt. To lose another 10% will require that the next stock chosen also go bankrupt. Talk about risk aversion.

So blue's expectation that the ending value be A(0) does not hold either, it will almost surely be > A(0) with probability → 1, simply because time is required for a stock price to double and double again. Blue's trading strategy is a double or nothing on each bet betting system and not that productive to boot.

Another thing blue can not see is a large number of trades (k) since a price has to double each time for k to increase by 1. For instance, in a 50 years Berkshire Hathaway scenario, k could reach 14 wins in a row and would be waiting for its 15^{th}. While AAPL over the last 20 years would be at k = 7 also waiting for its next double. A, k > 50, borders utopia.

The weakness here is associated with “b” the fraction of capital put at risk. It reduces market exposure and has for consequence: $b * \Sigma(H(blue).*\Delta P(blue)) = 0.10 * \Sigma(H(B\&H).*\Delta P(blue))$. This translates to taking only “b” percent (10%) of the potential profits that will be generated by a simple Buy & Hold scenario. The following blue scenario snapshot illustrates this.

Fig. 10. Blue and Red's Trading Strategies

	Blue's trading strategy							
	where it is said that: E[A(k)] = A(0) * [(1 + b)^W * (1 – b)^L] = A(0)							
	explains the behavior of the trading strategy (but it does not)							
APPL	starting price	$ 20.00	early 1990's some 20 years ago, but after Jobs took over					
		Bet	**First Bet**					
	Account	**Fraction**	**Size**					
Initial values	$ 10,000	0.10	$ 1,000				**Blue's**	**Red's**
					A(k)		**portfolio**	**portfolio**
Trade number	**Trade**		**Bet**		**Portfolio**	**Shares**	**value if**	**≈ B&H**
k	**Price**	**Bet**	**Result**	**Profit**	**Value**	**Bought**	**bet lost**	**Strategy**
1	$ 20.00	$ 1,000	won	$ 1,000	$ 11,000	50.00	$ 9,000	$ 10,000
2	$ 40.00	$ 1,100	won	$ 1,100	$ 12,100	27.50	$ 9,900	$ 20,000
3	$ 80.00	$ 1,210	won	$ 1,210	$ 13,310	15.13	$ 10,890	$ 40,000
4	$ 160.00	$ 1,331	won	$ 1,331	$ 14,641	8.32	$ 11,979	$ 80,000
5	$ 320.00	$ 1,464	won	$ 1,464	$ 16,105	4.58	$ 13,177	$ 160,000
6	$ 640.00	$ 1,611	won	$ 1,611	$ 17,716	2.52	$ 14,495	$ 320,000
7	$ 1,280.00	$ 1,772	still open			1.38	$ 15,944	$ 640,000
8	$ 2,560.00	$ -				-	$ -	$ -
9	$ 5,120.00	$ -				-	$ -	$ -
10	$ 10,240.00	$ -				-	$ -	$ -
11	$ 20,480.00	$ -				-	$ -	$ -
12	$ 40,960.00	$ -				-	$ -	$ -

The fraction b is the fraction of the portfolio being invested and as such, the first fraction invested is b = 10% of the current account A(0) = $10,000. Each successive trades will put b * A(k) at risk which includes the registered profits along the way.

For Berkshire Hathaway (Fig. 11), the scenario is similar. The number of trades is higher, but the return is still dismal. One of the success stories spanning some 50 years, where someone having just bought and held for the period would have tremendous results; you have blue achieving less than a 3% CAGR.

Fig. 11. Blue and Red's Trading Strategies

	Blue's trading strategy							
	where blue says that: E[A(k)] = A(0) * [(1 + b)^W * (1 – b)^L] = A(0)							
	which most certainly does not explain the behavior of the trading strategy							
BRK-A	starting price:	$ 20.00	1960's some 50 years ago when Buffett was an unknown					
		Bet	**First Bet**					
	Account	**Fraction**	**Size**					
Initial values	$ 10,000	0.10	$ 1,000				**Blue's**	**Red2's**
					A(k)		**portfolio**	**portfolio**
Trade number	**Trade**		**Bet**		**Portfolio**	**Shares**	**value if**	**≈ B&H**
k	**Price**	**Bet**	**Result**	**Profit**	**Value**	**Bought**	**bet lost**	**Strategy**
1	$ 20.00	$ 1,000	won	$ 1,000	$ 11,000	50.00	$ 9,000	$ 10,000
2	$ 40.00	$ 1,100	won	$ 1,100	$ 12,100	27.50	$ 9,900	$ 20,000
3	$ 80.00	$ 1,210	won	$ 1,210	$ 13,310	15.13	$ 10,890	$ 40,000
4	$ 160.00	$ 1,331	won	$ 1,331	$ 14,641	8.32	$ 11,979	$ 80,000
5	$ 320.00	$ 1,464	won	$ 1,464	$ 16,105	4.58	$ 13,177	$ 160,000
6	$ 640.00	$ 1,611	won	$ 1,611	$ 17,716	2.52	$ 14,495	$ 320,000
7	$ 1,280.00	$ 1,772	won	$ 1,772	$ 19,487	1.38	$ 15,944	$ 640,000
8	$ 2,560.00	$ 1,949	won	$ 1,949	$ 21,436	0.76	$ 17,538	$ 1,280,000
9	$ 5,120.00	$ 2,144	won	$ 2,144	$ 23,579	0.42	$ 19,292	$ 2,560,000
10	$ 10,240.00	$ 2,358	won	$ 2,358	$ 25,937	0.23	$ 21,222	$ 5,120,000
11	$ 20,480.00	$ 2,594	won	$ 2,594	$ 28,531	0.13	$ 23,344	$ 10,240,000
12	$ 40,960.00	$ 2,853	won	$ 2,853	$ 31,384	0.07	$ 25,678	$ 20,480,000
13	$ 81,920.00	$ 3,138	won	$ 3,138	$ 34,523	0.04	$ 28,246	$ 40,960,000
14	$ 163,840.00	$ 3,452	won	$ 3,452	$ 37,975	0.02	$ 31,070	$ 81,920,000
15	$ 327,680.00	$ -	still open					
16	$ 655,360.00							
17	$ 1,310,720.00							
18	$ 2,621,440.00							

14 wins in a row! Was the probability of a win really 0.50. All stocks studied had the same pattern, can we still say that the doubling was on a coin flip? There is a reality check to be made here. It took BRK-A 50 years to double 14 times. A Buy & Holder would have made $81,920,000 on the deal while Mr. Blue would have his initial portfolio up to $ 37,975. A mere 3% CAGR.

It's like if blue was playing a double headed coin as long as he didn't lose the coin. This would result in a series of win as long as prices doubled followed by a lost resulting in the stock's bankruptcy. In the BRK-A case, with a mere 2.7% CAGR, doubling time would be over 25 years, meaning that to double the account size, it would require some 25 years. And another 25 years to double again. One would have been better off leaving their money in the bank.

To resume: the two trading programs (blue and red2) do not act in real life as in theory. They are both kind of deterministic scenarios which do not depend on the random outcome of their seemingly randomly generated trades. They are both Buy & Hold scenarios in disguise and have nothing to do with random-like price movements.

It's like shaking one's fingers and asking someone to count them. There was a lot of fuss for absolutely nothing, in the sense of generating some alpha superior to the Buy & Hold scenario. If you were looking to achieve over-performance, forget it. The solution was not red2 (as it performed the same as the Buy & Hold), neither was it blue with a performance level at least ten times less than red2.

One would have achieved the same end results using red2 as with a Buy & Hold over the long term trading intervals, no need to trade except for the initial purchase with no alpha generation. Maybe just a fun way, if not a futile way, to entertain oneself (and pay the commissions to do so).

Red's compensation factor might have transform what was a bad scenario (red1) into one where after the initial purchase, the trading does not matter at all. From a version where you should not play because you will lose everything (red1) to a scenario where you really don't need to trade except for the very first position: a strange phenomena of sort. You have a trading strategy (red2) where trades were randomly generated and where getting in and out of position did not matter, and performed the same as a Buy & Hold strategy, while its sibling red1 under the same randomly generated sequence of ups and downs moves would obliterate one's portfolio.

For red2, it's that first trade, the first purchase putting the entire portfolio on the line that becomes the only trade of interest or value. All others are just ways of passing the time, watching paint dry or something. There was no need to sell, no need to repurchase, even if the profit target or the stop loss was hit since you re-entered the trade at the very same price with the same quantity of shares. Also, as k increased, time advanced as well, resulting in years, decades where all the trades after the first where irrelevant. The Buy & Hold strategy did exactly the same thing having taken an initial position and then waited and waited (held).

Someone trading using program red1 will see his portfolio degrade with time as k increases where the equal fixed fraction used will determine the speed of the decline. However, one can compensate for this deficiency by a small compensator that is really easy to implement in code (see Appendix red2 or red3). But, by doing so, converts an absolutely losing scenario into a Buy & Hold one achieving with time the same level of performance. No alpha generation. It's much better than red1, for sure, but not a super star. Nonetheless, at least 10 times better than what blue's program could ever achieve (see Fig. 10 and Fig. 11).

The Partial Conclusion to this

The answer to the initial question is: yes, the equal fix fraction position sizing trading strategy ends in having your portfolio going broke as demonstrated in the red1 program. And one most certainly should not play this kind of game at all.

This paper showed that, as a bare minimum, one should compensate for red1's deficiencies. I would say the same for any trading strategy that uses barrier-like trade execution. Both trading programs (red2 and red3) as they are, generate no alpha when compared to the Buy

& Hold. But if executed will have both programs being fully invested for the long term as in an index fund, outperform blue's program by at least a factor of 10 to 1, as can easily be seen in the above 2 scenarios (AAPL, BRK-A).

Blue's no need to trade because the expected value of his trading system was A(0), has turned out to have the same expectation as a long term Buy & Hold scenario for the percent of account “b” put into play. And therefore, because of the long term view, blue's system had an higher expectation than A(0) in all probability, and almost surely; even if the overall result was rather poor.

Red's programs (red2 and red3) also behave as Buy & Hold scenarios. However, because of the methodology become a do play but do not trade scenario, in the sense that the randomness of all the trading after the initial purchase turns out to be unimportant and bring no added long term value to the game.

Nonetheless, the red2 program is still able to outperform blue by a factor of at least 10 to 1 due to blue's 10% (b = 0.10) market exposure. But since the stock market game is a compounding return game, and where time is the real measure of achievement, one would see, as in BRK-A (Fig. 11), blue ending with $ 37,975 while red2 would have over the same period increased his portfolio to: $ 81,920,000. I would consider the difference in performance as a lot more than just a 10 to 1 advantage. This does clearly show the difference between a betting system (blue, additive) and a compounded return system (red2, multiplicative).

You have a zero expectation for a 50/50 game, but we can't force a game that is not a 50/50 game to behave as such. You can kick it as much as you want, it will only go its own way. However, you can design trading systems that behave as if in a 50/50 game, but that does not mean that the real game underneath is a 50/50 game. It is a major nuance when designing trading systems. Another concept that escaped blue's vision on how to design trading strategies under uncertainty: the real world trading environment matters, it is where the game is played.

Even if someone says that the stock market behaves as a 50/50 game, it is not necessarily so. It is even kind of nonsensical. A single chart of the DOW over a 50 years period should be sufficient to show that there is an upward trend, an upward bias. Will it persist for the next 50 years is another question?

In these pages, was presented a trading strategy where all trades were randomly generated, where any single trade could not be predicted or anticipated. And that no matter where in the sequence of 1,000 trades, a single trade could have only a slight influence on the game, since the real score could not be determined before the end of game. And since 1,000 trades could take decades to perform there is no predictive knowledge that can be gained from the series of trades themselves, except maybe for the long term on the very thing that is monitored: the stock market itself.

When looking at the market in general and for the long term, one could make guesstimates of red's trading strategy's Buy & Hold equivalent (red2). And in that sense, trading was not really required. Only the first position, that very first “buy” gave all the needed opportunity required to achieve its long term results. It demonstrated that long term it was preferable to be in a position than trying to figure out what was the next trade or its expectation.

The whole exercise was an analysis of the fixed fractional position sizing scheme. What was learned had unexpected outcomes. The first of which was that the modified program really acted like a coin flip series providing no alpha. However, even with a zero alpha, the game would win long term, with large k, almost by default, making some of the assumptions unsupported. But the win would still not exceed the Buy & Hold.

The trading rules of both programs (blue and red's) could only be transformed to the equivalent of a Buy & Hold scenario where trading could have happened or not without change in the output under the only condition that the first trade be taken and held for the duration.

It resembles the conclusions I obtain in the early 1990's on an equivalent piece of research, if I remember correctly. It's like doing the same research again simply because you forgot the outcome obtained then. But even redoing the research with new tools has been educational.

Red's trading programs (red2 and red3) are close to a coin flipping strategy representation. Each position actually acting as if +1 or -1, having the same dollar value bets and acting the same as a random function would. Yet, simply because the number of trades was set high, the expectation of the game became not equivalent to a coin flip distribution but to a long term expected market return.

Therefore, the assumption that the stock market game is a flipping contest with the same attributes does not hold. Executing all the trades of red's program (red2) which were all based on a coin flip had higher expectation than the coin toss expectation. Thereby asserting $E[A(k)] = A(0)$ in this type of game, for large k, does not hold. Furthermore, it does not hold for either blue's or red's programs (red2 or red3). The most expected outcome would be: $E[A(k)] > A(0)$. It would also stress the notion that long term the stock market is not a 50/50 kind of game.

> “It is not one trade in particular that is of interest, it is the finish line.”

The Background

Any “theoretical” trading strategy just starts with a concept or collection of concepts. It tries to answer a question like: what if I did this or that? You then run some simulations to see if these concepts have what is needed to go from hypothetical to reality. The objective being to transform all these trading ideas into real life workable and executable trading strategies. It's like a reality check: can it be done or not?

Maybe most importantly, can it be done profitably?

And if so? What are the expectations of performing well over real market data going forward and for how long? It is not just I have this system that did this or that on “hypothetical” data or a mathematical formula. It is that this “system”, the one I hold in my hand will use my money in a real life situation and if I did not do my job correctly or properly; I will be the one to pay the piper. I would have worked hard to develop some trading program and then pay again (by losing) due to my misinterpretation, unlucky situation or strictly from having a flawed program design where from the beginning the premises were unfounded, under specified, or not grounded in reality.

One of the most severe test you can throw to a “pseudo” trading strategy is to feed it a randomly generated data series; something like a series of coin flips or the output of a rand() function. This is the most severe test for the reason that there is nothing you can do to influence what is coming your way. You won't and can't know what will be the value of the next coin flip, but on the other hand, you do know that it will be either head or tail with no in between. Coin flips can provide you with millions of historical data sets for having been performed, and in a way resemble past market data.

There is a lot to learn from coin flipping but in no way will it make you a better coin flipper.

Coin flipping offers a straight unpredictable yes or no: with no memory, no kindness, no if this, no if that, no sorry I didn't know it was you, no I knew you would, no I knew you could, no wait I'll mean reverse, no no problem I'm not moving from here, no wait I'll drop like a rock, no wait I'm about to skyrocket, no indicator correlation, no just for you and no alpha.

The coin flip test applied to a trading strategy is very unforgiving. You can not force or influence the outcome of the coin toss, you can not redirect it to suit your needs, or contain it in a little box. You ask for a randomly generated data series or trade series, then here it is; live with it! For sure, it will show you what random really means, and it won't be polite about it. I like very much these randomly generated data series, they teach me humility (if you can believe that) where it counts.

From theory you need to go to reality. Reality, in the stock market game stops to be “hypothetical”, it is measured by the dollar value in your broker's account where it counts, and where you do keep score (see Fig. 10 or Fig.11).

Does this trading strategy increase my trading account or not? It's a simple question. If my account blows up, I'm out of the game. If it takes me 50 years to double my account then why did I bother in the first place? Note that 50 years is about half, on the optimistic side, of one's adult lifetime! There is no: oh, let me try this a couple of more times.

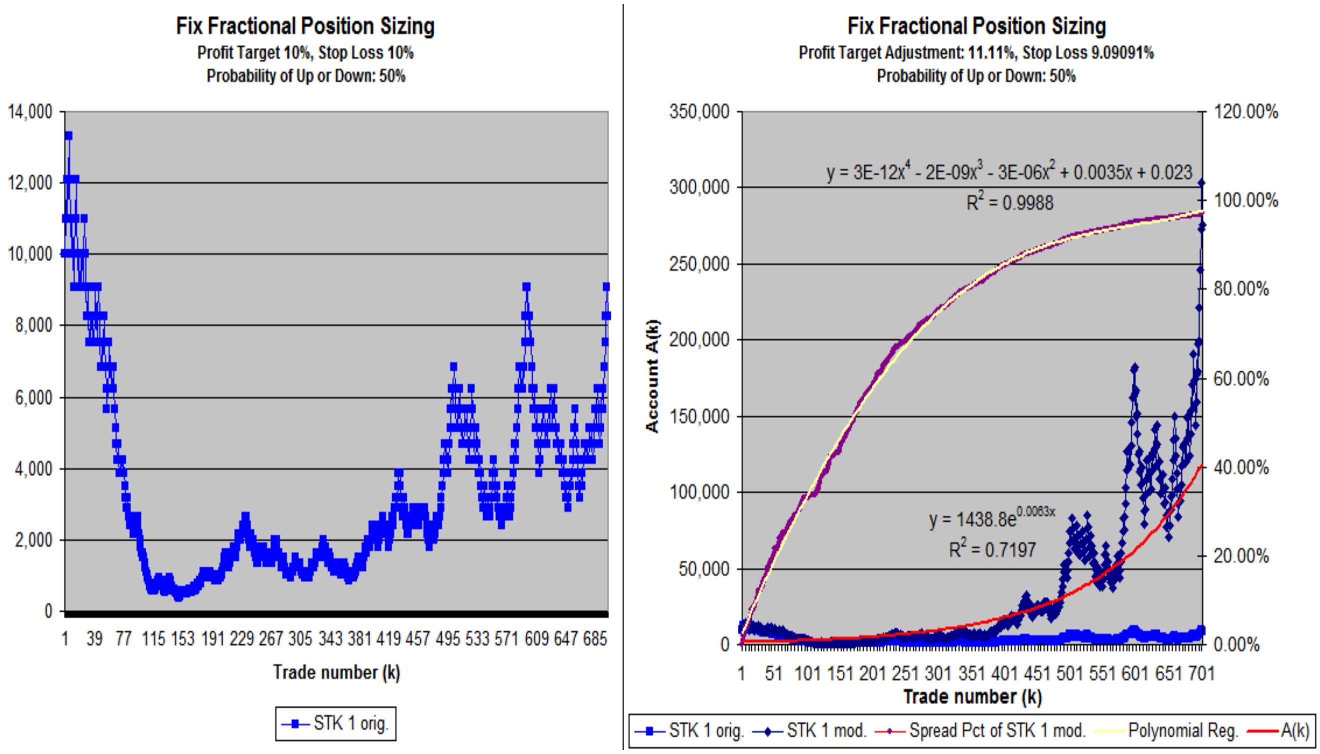
Could Improvements be Made?

Can the equal or compensated fixed fractional position sizing trading strategy made to work better? Well yes, naturally.

And it is also an easy solution. It won't change the randomness of the trading itself, by this, meaning the up or down of market moves, but it will change the value of the betting system itself and thereby the value of the game.

A better management of trades under the uncertainty of price movements. The trading strategy will be for the long term and this means large k. All that is required is changing red's trading strategy from red2 to red4; the double compensated program version which has a built in advantage in the game: a positive position sizing imbalance resulting in a long term exponential growth rate.

Fig. 12. Fix Fraction Double Compensation

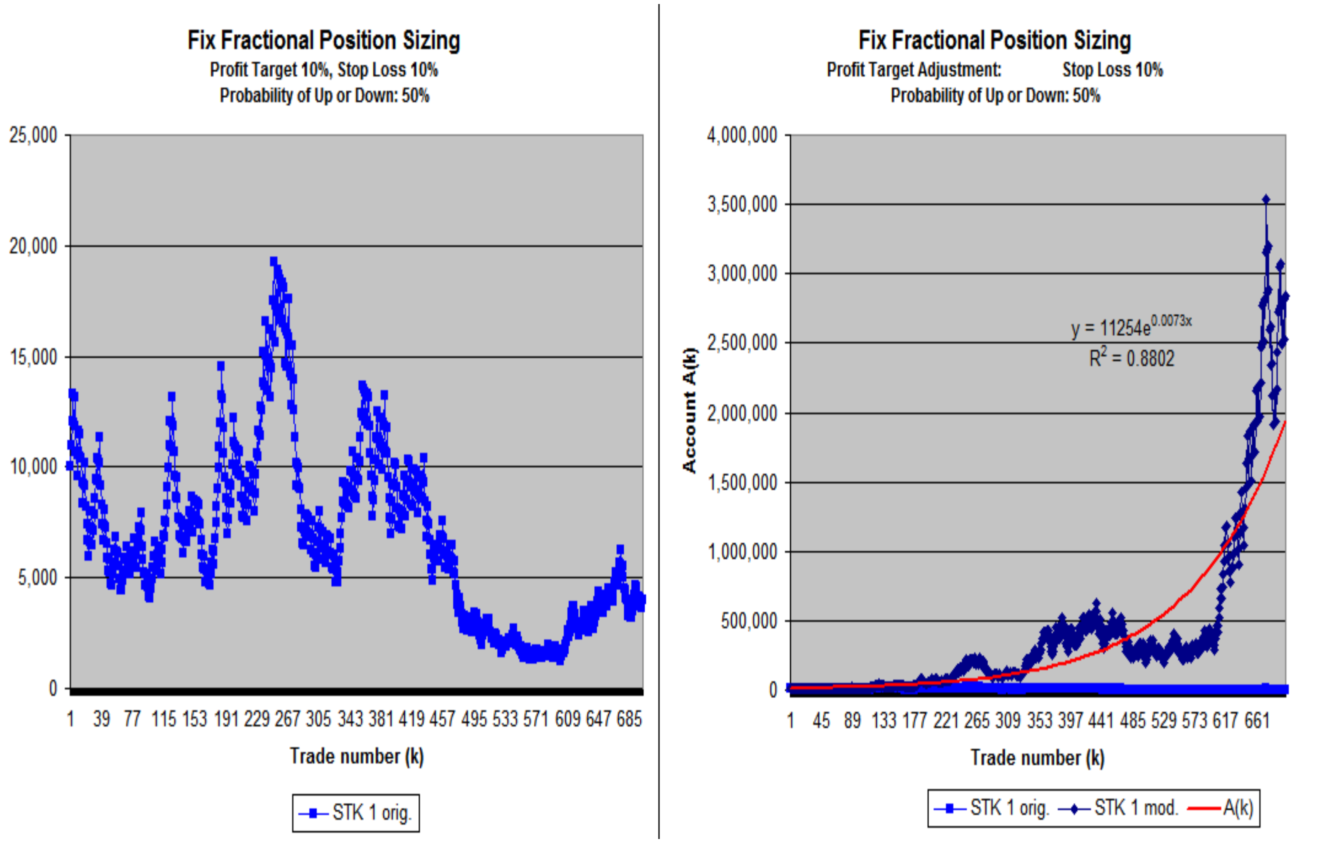


Program red4 uses double compensators, f/(1-f) for the profit target and f/(1+f) for the stop loss. This results for one win and one loss in: (1+0.1111)*(1-0.09091) = 1.010101. This will produce an exponential curve as shown in the chart above.

Red4 has a positive trade imbalance where one is not only invited to take the first trade, he is also invited to take all the trades as they come along. There is a positive expectancy to the game. It's long term value could be expressed as: A(k)(red4) = A(0) * (1.010101)^(k/2).

Another way to improve this trading strategy would be to add a controlling function to the mix. Adding a controlling function would force red's program (red4) to boost performance levels even higher than the Buy & Hold scenario. Since I'm still in the development process associated with the above, I'll only present a chart from a single stock for a single simulation. Notice that the regression line is of the exponential type, meaning that a portfolio would generate some alpha above and beyond the Buy & Hold strategy.

Fig. 13. Pushing for Higher Performance Levels



Notice that the chart on the left, generated by program red1 appears to almost flat line in the chart on the right. The chart on the right does show exponential growth.

I hope that the above does explain in detail what happens using an equal fixed fraction trading strategies and what kind of pitfalls it may encounter. I'll be working on a better controlling function for a short while just to see what could be the output, like how far can I push this thing. I still have red5, red6 and red7 to investigate, each promising higher and higher performance levels.

The Equation

The equation used for the equal fixed fractional position sizing strategy was:

A(k) = A(0)*(1+f)^W*(1- f)^L (1)

where f was the equal fixed fraction used for the profit target and the stop loss, and where W was the number of winning trades and L the number of losing trades.

Filling in your own historical numbers into the above equation should reveal some added insight as to where your trading strategy is going. It's use is to analyze trading strategies using the most basic of elements. The average rate of return for winning trades (f) with the average number of wins on say 100 trades. You fill in the numbers and you have an answer as to how your portfolio has or could behave over those 100 trades.

The equation is not the same as played by blue: A(k+1)(blue) = A(k) + (b* A(k) * 0.50) - (b * A(k) * 0.50), where to know the account value, it is required to know the outcome of every bet before k. Also, this last equation is just a betting system, and if played as if in a random game is expected to generate zero profits, or more clearly: A(k)(blue) = A(0) as blue hammer so often on the table. But because of the nature of the game played, should be expressed as A(k)(blue) > A(0) even if the output is inconsequential compared to other trading methods available since even a simple Buy & Hold would beat blue's performance level by a wide margin (by at least a factor of 10, see the BRK-A or AAPL scenarios above).

Equation (1) can be used to make rough estimates of some trading strategies. One could look at average figures over a number of trades based on their own past trading systems. By increasing the profit target (PT) and stop loss (SL) spread, one can more than compensate for the trade size imbalance and generate positive long term alpha.

For SL = 10%, and PT = 15%

A(100)(red) = A(0) * (1 + 0.15)^30 * (1 - 0.10)^70 =$ 415
A(100)(red) = A(0) * (1 + 0.15)^40 * (1 - 0.10)^60 =$ 4,814
A(100)(red) = A(0) * (1 + 0.15)^50 * (1 - 0.10)^50 =$ 55,849
A(100)(red) = A(0) * (1 + 0.15)^60 * (1 - 0.10)^40 =$ 647,994
A(100)(red) = A(0) * (1 + 0.15)^70 * (1 - 0.10)^30 =$7,518,377

As can be seen above, designing an edge with W > k/2 could be quite profitable. By increasing the spread further, even by a relatively small amount would more than over compensate for the degradation observed in Fig. 6.

For SL = 10%, and PT = 20%

A(100)(red) = A(0) * (1 + 0.20)^30 * (1 - 0.10)^70 =$ 1,487
A(100)(red) = A(0) * (1 + 0.20)^40 * (1 - 0.10)^60 =$ 26,412

A(100)(red) = A(0) * (1 + 0.20)^50 * (1 - 0.10)^50 =$ 469,016
A(100)(red) = A(0) * (1 + 0.20)^60 * (1 - 0.10)^40 =$ 8,328,660
A(100)(red) = A(0) * (1 + 0.20)^70 * (1 - 0.10)^30 =$147,898,070

Designing a trading edge (W-L>0) while also pushing for a higher profit target to stop loss ratio can push performance levels to new heights.

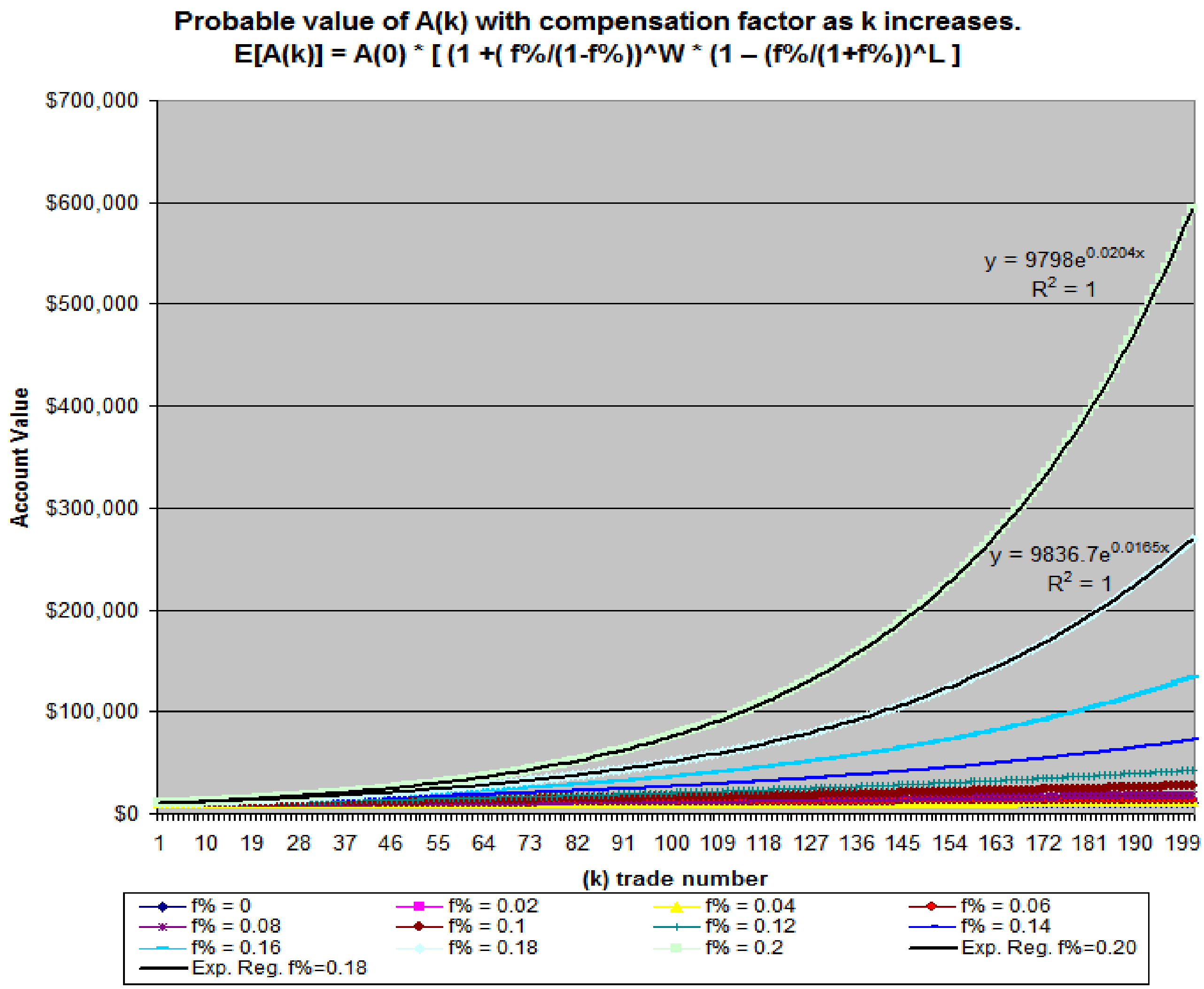
My Close to Final Analysis

Red4 uses a most interesting equation: A(k)(red4) = A(0)*(1+f/(1-f))^W*(1- f/(1+f))^L. It is rare to have so much crammed into so small a package. Not that many trading systems can be summarized in this way: into a single equation and still remain a complete trading system. Bearing an equal sign, red4 is not expressing an opinion or a maybe if this or that; it is making a statement.

By using the double compensation, one can transform what was red1 into red4 where the position size imbalance becomes an advantage to the long term portfolio perspective instead of having a negative imbalance as in red1 which has for only outcome losing one's trading account.

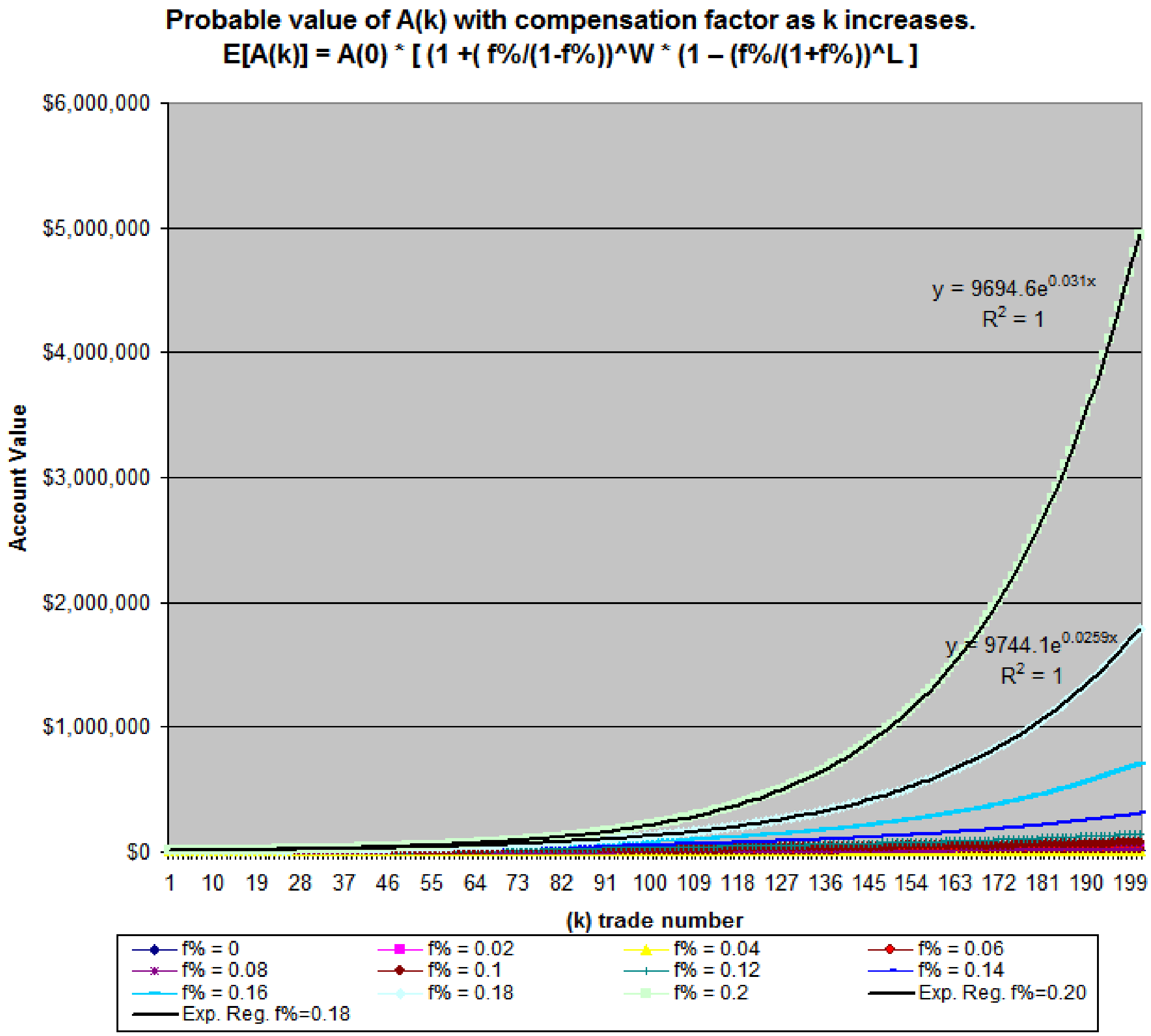
From the chart below, it is easy to see that the compensation factors do provide long term benefits even in a 50/50 game environment. Compare this chart to Fig. 6 which was a degenerative process. Fig. 14 shows that as k increases, the higher the fix compensated fraction used, the higher the expected return. Not only that, but these curves are exponential in nature, they are compounding over the number of trades made.

Fig. 14. The Double Compensated Fixed Fraction Position Sizing Scheme (red4)



It is also easy to simply over compensate for the degenerative process of red1. It would be sufficient to increase the spread between the profit target and stop loss. A small differential increment, due to compounding, could result in the following:

Fig. 15. Red4 (DCFFPS) Boosted making it red5



In both Fig, 14 and 15, it is difficult to differentiate the curves in the beginning. It is a long term process, it makes it more evident that the trading game is a long term endeavor. Note that Fig. 14 and Fig. 15 do provide alpha, and of the compounding type. The higher the trade number, the more valuable the importance of a single trade.

Elements not mentioned in red4's equation are time, stock selection and diversification. Each should be addressed as they can impact performance over the long term.

First, time has been dealt with in the sense that red2 or red3 showed that their trading strategies were Buy & Hold equivalents and therefore one should look at the trading game as a long term endeavor spanning not just years but decades. Red4 is an improvement over red2 and red3, and the boosted version (red5), one of the possible improvements over red4, led to the assertion that red4 and its boosted version were generating alpha over a long term horizon (as can be seen in Fig. 14 and Fig. 15 above).

Red4 is playing price differentials, not necessarily on the same stock. All it cares about is the next profit target and/or stop loss to be reached. Nonetheless, red4 is only looking at price differences, it does not matter on which stock this differential is taken. One could change on every single move on which stock this next move should be taken. And therefore one should look at the problem from a portfolio stand point and opt to diversify his/her portfolio in order to minimize single stock portfolio risk.

Boosting performance levels higher was not that hard either; it only required to increase the PT to SL spread. An easy way to do this is:

$$A(k)(red5) = A(0)*(1+f/(1-(f+c)))^W*(1- f/(1+(f+c)))^L \qquad (5)$$

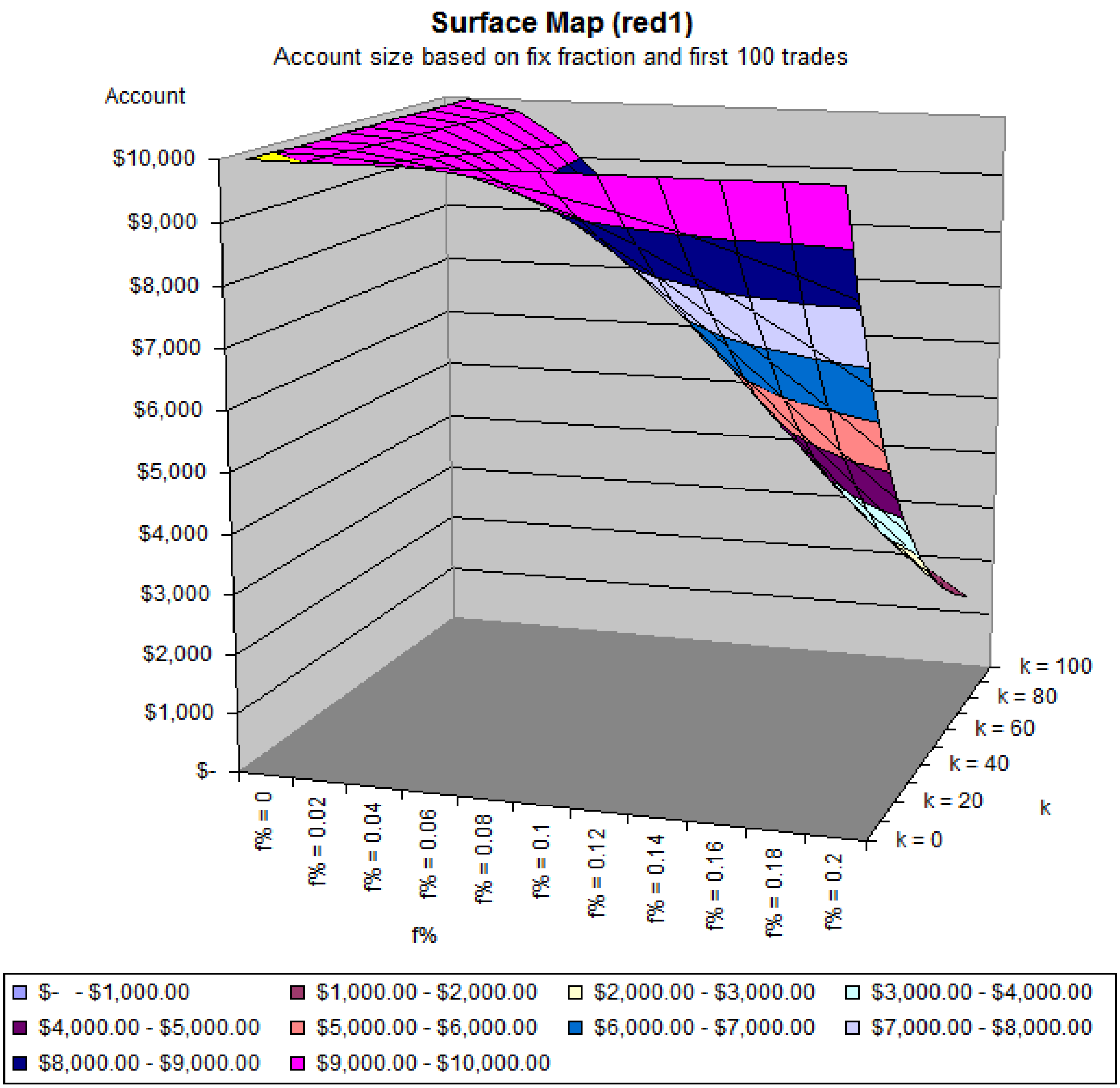
where c acts as an incremental compensation booster to improve performance. Even a marginal c value is sufficient to produce Fig. 15.

From one equation that had only one purpose which was to obliterate one's portfolio (red1), adding a small compensation factor led to red2 and red3 which not only repaired red1 deficiencies but generated a Buy & Hold trading strategy equivalent. By using both compensators at the same time, it not only reversed red1, it allowed to have a trading strategy with long term positive alpha generation as in red4. But, here, one could push even further by adding a booster to red4 to generate even better performance levels (red5).

Program	Formula	Long Term Output	Reason
Red1	A(0)*(1+f)^W*(1- f)^L	Lost of portfolio	(1+f)*(1- f) < 0
Red2	A(0)*(1+f/(1-(f)))^W*(1- f)^L	Buy & Hold	(1+f/(1-f))*(1- f) = 1.0
Red3	A(0)*(1+f)^W*(1- f/(1+(f)))^L	Buy & Hold	(1+f)*(1- f/(1+(f)) = 1.0
Red4	A(0)*(1+f/(1-(f)))^W*(1- f/(1+(f)))^L	Alpha generation	(1+f/(1-(f))*(1- f/(1+(f)) > 1.0
Red5	A(0)*(1+f/(1-(f+c)))^W*(1- f/(1+(f+c)))^L	Exponential Alpha	(1+f/(1-(f+c))*(1- f/(1+(f+c)) > 1.0

Red1 had a degenerative process as the number of trades increased and the fix fraction used increased, it is illustrated in the following surface map:

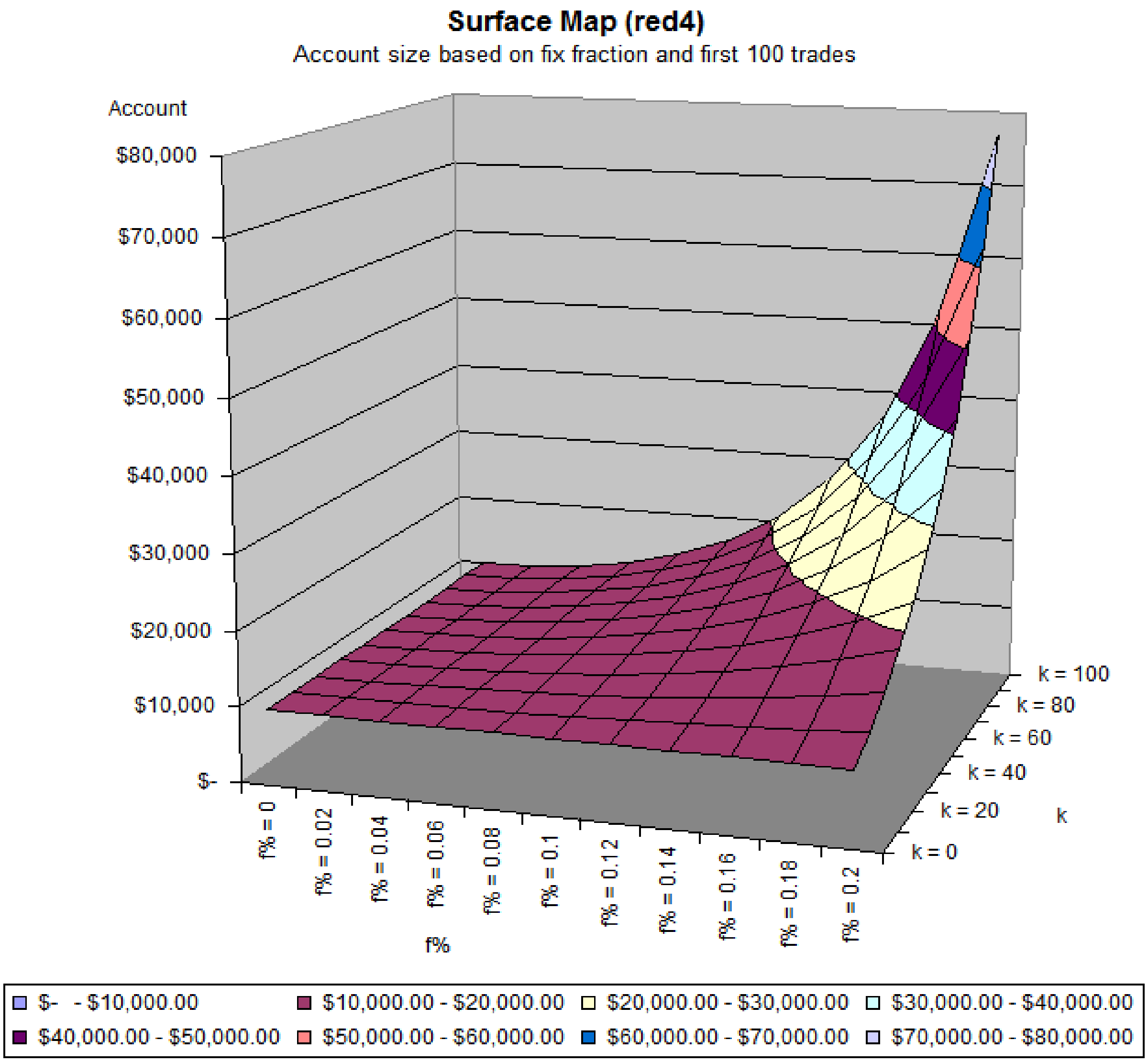
Fig. 16. Red1: Surface Map (first 100 trades)



The degenerative process is easy to see from Fig.16; the more the fraction (f%) increases and the more the number of trades increase, the more the expected outcome (account value) declined. It becomes only a matter of how slowly do you want to lose your entire portfolio.

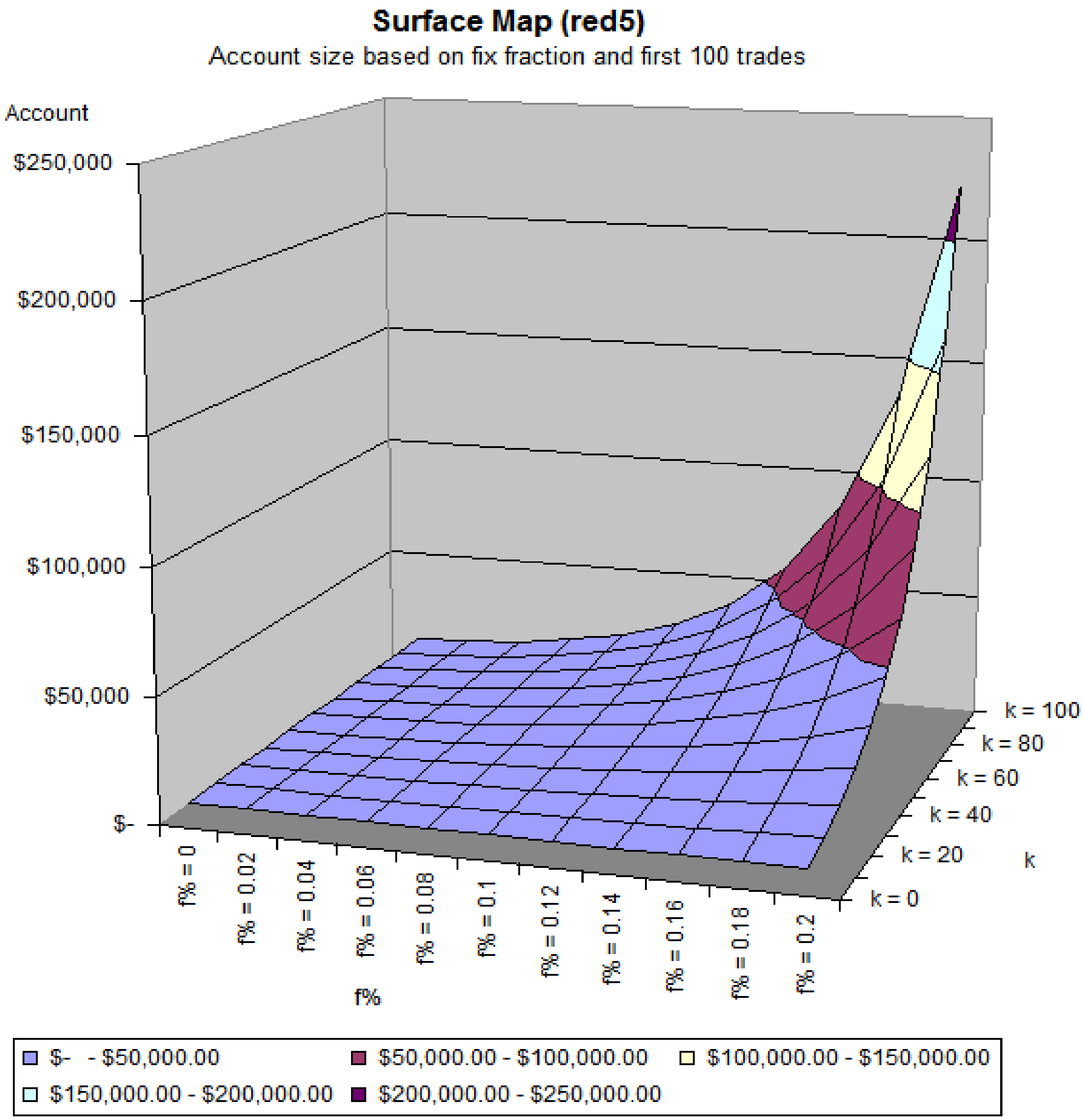
While using red4, the situation is reversed. Now, there is alpha generation due to the positive trade size imbalance; and it is on this regenerative process that all the emphasis is placed.

Fig. 17. Red4: Surface Map (first 100 trades)



By adding a compensation booster to the mix as in red5, one can push performance levels even higher. Yet, this “booster” adjustment is just a small incremental request when compared to the equation's output. Note that red4 and red5 are exponential growth functions, even a slight positive difference will compound with time (number of trades).

Fig. 18. Red5: Surface Map (first 100 trades)



It's the long term outlook that should be evaluated. That you win a trade here and there has little consequence over the long term, except if that single trade ends up to be your net gain over the period. But in trading, one is expected to do quite a number of trades over an extended period of time. And it is this long term view that should be put forward and analyzed. One only needs to compare Fig. 14 to Fig. 15 to see the effect of the compounding differential between red4 and red5 which is entirely due to the compensation booster.

One could be playing red1 simply because he has not yet recognized that he is doing so.

How could this be? Anyone using barrier-like exits can see, when averaging trade returns, what amounts to a fix fraction percent profit target. And there are numerous ways to accomplish this. In fact, I've seen a lot of trading strategies operating as if using red1 and where the author did not understand that his trading strategy was not only breaking down with time, but was really detrimental, to say the least, to his long term portfolio growth. Red1 is a flawed trading system from the start and can have for only outcome: the lost of one's portfolio. The only way to compensate for red1 is to have a sufficiently high win to loss ratio to cover up the inherent deficiency of this trading method.

All the talk has been on a fix fraction of portfolio at risk: you won f% or lost f% as if from the outcome of a coin tossing game. Another way of saying no edge present. However, equations red1 to red5 are predetermined sequence of events, in the sense that what ever the values you want to feed these equations, you get an exact answer. All the preceding analysis has been done on the premise of a random-like game where the win to loss ratio was 1.0, meaning that on 100 trades there were 50 wins to 50 losses which was the most expected outcome of a coin tossing game. Another analysis is now required, this one looking at the side effects of moving away from the mean: the what happens should you develop a long term edge to your trading game?

Some would like to say that mathematical models have little value when attempting to design trading systems, and yet, you have red1 which is a mathematical model transformed step by step into red5, another trading program where everything is predetermined, at least the sequence of what would happen should the stock price follow a particular path over say the next 100, or 1,000 trades. The point was to design red5 in such a way as it's most expected outcome was some long term exponential return. And I do think that this task has been accomplished with what could be considered minor modifications to the red1 program which in itself was a disaster in the making.

The Next Step

What's wrong with red5? Nothing. On the contrary, it leads to some interesting conclusions. One of the most basic ones being that the trading strategy is all predetermined. It does not care about how prices move as long as the barrier-like limits are hit time and time again. And when ever one of these limits is hit, a trade is executed to close the existing position.

Adding an edge, meaning designing a trading strategy such that W - L > 0, can only mean higher profit potential. Even red1 could be improved as this edge could compensate for its inherent deficiency. This was partially covered on page 47. There is an upward long term bias in stock prices, it should therefore be relatively easy to design a long term trading strategy using this edge.

Most importantly, red5 has exponential growth built in implying even better performance with a positive edge.

The Alpha Generation

Since red5 has some alpha generation built in, in the sense that it outperforms a Buy & Hold trading strategy, isn't it trying to redefine some old portfolio management concepts? The output of the efficient market hypothesis or the efficient frontier as taught in investment courses might not hold that much if with a simple trading strategy governed by a single equation it can outperform the Buy & Hold with practically no effort.

From the table below, the modifications brought to each of the programs (from red1 to red5) should be considered minor in nature. Yet, they totally changed the nature of the game.

Program	Basis: f = 0.1000	Basis: f = 0.1600	Long Term Output
Red1	fw%=0.1000, fl%=0.1000	fw%=0.1600, fl%=0.1600	Lost of entire portfolio
Red2	fw%=0.1111, fl%=0.1000	fw%=0.1905, fl%=0.1600	Buy & Hold equivalent
Red3	fw%=0.1000, fl%=0.0909	fw%=0.1600, fl%=0.1379	Buy & Hold equivalent
Red4	fw%=0.1111, fl%=0.0909	fw%=0.1905, fl%=0.1379	Low alpha generation
Red5	fw%=0.1176, fl%=0.0869	fw%=0.2025, fl%=0.1322	Exponential Alpha generation

The Red5 Trading Strategy

In this paper were presented 6 trading strategies. One from blue's side and 5 from red. Of all these trading strategies, only one really stands out and that is red5, the boosted double compensated program which in itself is an improvement over red4 (the double compensated version) which in turn was an improvement over red2 or red3.

How should red5 be played?

$$A(k)(red5) = A(0)\cdot(1+f/(1-(f+c)))^{W}\cdot(1- f/(1+(f+c)))^{L} \qquad (5)$$

Red5 is a trading strategy in its own right, and is also an alpha generator. In a single equation, the course of action is predetermined for the next 50 years. It answers the question: what should I do after each and everyone of the trades taken over a long term horizon? Red5 says what should be done when ever one of the barrier-like limits is hit which is to close the position (take the profit or loss) and establish a new one. Refer to Fig. 15 for the long term effect of the applied booster. Note that the regression lines are exponential.

Here is the red5 program:

```
// Red5:  adds control boosters to the compensators (see equation (5))
// author: Guy R. Fleury, January 2014
var Account, fw%, fl%, c, cUp, cDn, Q: integer;
```

```
// Initial account, could be scaled to any amount
Account = 10000;

// Red1 default equal fixed fraction without compensation factors.
// the profit target stop loss combination can be adjusted to other starting values: x%...
fw% = 10%;        // 10% profit target:  percent of equity gain
fl%  = 10%;       // 10% stop loss: percent of equity loss
// booster controllers
cUp = 0.05;       // value used to generate Fig. 18
cDn = 0.05;       // value used to generate Fig. 18
fw% = fw%/(1-(fw%+cUp));     // profit target booster adjustment factor: a/(1-(a+c))
fl%  = fl%/(1+(fl%+cDn);)    // stop loss booster adjustment factor: a/(1+(a+c))

InstallProfitTarget(fw%);        // Stock rises by fw%, take profit, sell
InstallTrailingStopLoss(fl%);   // Stock falls by fw% then take loss, sell

Do while until you decide to quit or 12,500 trading days (≈50 years)
 ApplyAutoStops( next bar );
 Begin
  If NoActivePosition then               // take one
   BuyAtMarket Q =  Account/P(t);     // P(t) = price on next bar
   // use total cash in account to buy Q shares at current price
 end;
End;
```

Red5 can be controlled by the compensation factors and its boosters. It does not say to which stock the formula is applied to. The reason is simple, it does not matter. It is not the concern of the trading strategy itself; all it looks for are price variations set as profit targets and stop losses. This also means that the strategy is not married to a particular stock in a particular direction. All the red5 program sets are the rules of engagement; if the stock price moves in such a way as to hit one of the barrier-like limits, the exit is executed.

No need to guess or predict the course of action, it is already preset by the trading strategy itself. All one has to do is follow the instructions: barrier limit is hit, exit position (take profit or loss) and set new position profit target and stop loss on the same stock or another, it does not matter; you are playing price differentials. Therefore, select the stocks you think have a higher probability of moving in the direction of your trade. It won't be perfect, but you don't need to be perfect to win this game.

The red5 program could be used just as a guideline. You've taken a position and estimate it won't go any higher; take your profit and establish a new position on another stock. The same as if you think the price will continue to decline, override the system, execute the stop before it is reached and take a new position on something else. When you get close to a profit target, there is nothing that forces you to take it either, you could move the target higher should you think prices could go higher. It's to say that you have some guidelines where you know you should look for boosted and compensated barrier-like limits as a means to generate exponential alpha. Nothing stops you from doing better than the red5 program.

From equation (5), it becomes evident that what ever you do to over-compensate for red1's deficiencies can lead to higher performance levels than expected. Any time you design your trading strategy to have (1+f/(1-(f+c))*(1- f/(1+(f+c)) > 1.0 will result in beating the Buy & Hold trading strategy.

All red's programs reduced their dollar amount bets on declining prices and increase them in rising prices as any fixed fraction of equity trading system would do. So, if the price went down, the dollar amount of the percent profit target would also go down. All 5 of red's programs play for percentages, they also have another interesting property: they all trade the same quantity of shares at a time, all the time (Q(0) = A(0)/P(0)). The trading volume is a constant in all program versions as can be seen from equation (5):

$$Q(k)(red5) = \frac{A(0)\cdot(1+f/(1-(f+c)))^W\cdot(1- f/(1+(f+c)))^L}{P(0)\cdot(1+f/(1-(f+c)))^W\cdot(1- f/(1+(f+c)))^L} = \frac{A(0)}{P(0)} \qquad (6)$$

This makes it very easy to remember what to do: you buy Q shares all the time and wait for an exit price to be hit which is simple to set: +x% or - y%.

The booster controller could be used something like a joystick (a slider function) to change in time the behavior of the red5 program. For example, making c = - f will degrade the program to the same level as red1 with the same long term consequence. Here are some outputs based on the booster controller value:

Controller	**Action**	**Result**	**Long Term Output**
c<-f	Accelerates degenerative process	Bad	Faster lost of entire portfolio
c=-f	Converts red5 into red1	Bad	Lost of entire portfolio
-f<c<0	Reduces compensation factors	Poor	Between lost of portfolio and Buy & Hold equivalent
cDn=-f	Converts red5 into red2	Ok	Buy & Hold
cUp=-f	Converts red5 into red3	Ok	Buy & Hold
c=0	No booster (red4)	Good but could do better	Low alpha generation
c>0	Boosts portfolio growth	Better	Exponential Alpha generation
c>0.05	Boosts portfolio growth	Much better	More exponential Alpha generation

By separating the booster controller one can add some refinement to the joystick process (see red5 program). The cUp variable controls the profit target compensation factor, and by raising it, it will have for effect to increase the profit target required for an exit, thereby, generating more profits when hit.

Red5 is a trading strategy where you preset your own rules of engagement for the duration and where you can, at your discretion, change its behavior in time using the booster controller. You are extracting from the ocean of variance what you want; you are following an equation on your own terms. Equation (5) is very flexible and can be changed at will; what's most important is to simply maintain an over-compensating posture. Anything above red4 will do the job, red5 has the advantage of having a booster controller built in for even higher performance levels.

The most important aspect of red5 is in the compounding. You will still lose on some of your trades, I don't think that is avoidable, however, the over-compensation factors in red5 will most certainly help you do a better job.

As was shown on page 47, over compensating will make you win the game.

A(100)(red5+) = A(0) * (1 + 0.20)^40 * (1 - 0.10)^60 =$ 26,412
A(100)(red5+) = A(0) * (1 + 0.20)^50 * (1 - 0.10)^50 =$ 469,016
A(100)(red5+) = A(0) * (1 + 0.20)^60 * (1 - 0.10)^40 =$ 8,328,660

And playing even longer (k=200) would produce:

A(200)(red5+) = A(0) * (1 + 0.20)^ 80 * (1 - 0.10)^120 =$ 69,759
A(200)(red5+) = A(0) * (1 + 0.20)^100 * (1 - 0.10)^100 =$ 21,997,613
A(200)(red5+) = A(0) * (1 + 0.20)^120 * (1 – 0.10)^ 80 =$ 6,936,657,968

Even a slight over compensation could mean a lot in the long term. Boosting the stop loss compensation from -10% to -8% over the 100 trades scenario would produce:

A(100)(red5+) = A(0) * (1 + 0.20)^40 * (1 - 0.08)^60 =$ 98,746
A(100)(red5+) = A(0) * (1 + 0.20)^50 * (1 - 0.08)^50 =$ 1,407,517
A(100)(red5+) = A(0) * (1 + 0.20)^60 * (1 - 0.08)^40 =$ 20,062,967

Or pushing a little more on the profit target by 2% would increase the above scenario to:

A(100)(red5+) = A(0) * (1 + 0.22)^40 * (1 - 0.08)^60 =$ 191,277
A(100)(red5+) = A(0) * (1 + 0.22)^50 * (1 - 0.08)^50 =$ 3,216,495
A(100)(red5+) = A(0) * (1 + 0.22)^60 * (1 - 0.08)^40 =$ 54,088,251

Opportunity Costs

There is also opportunity costs to be considered. It is not only about the PT and SL spread

and the win loss ratio. All four example above have A(0) = $10,000. And because those 4 scenarios are scalable, increasing the initial capital by an order of magnitude or two would add one or two zeros to the output of each of those 4 scenarios.
By not pushing to manage more initial capital, I think most traders short change themselves, and in a big way! The above scenarios could be only 1% or 10% of what they could be simply because they were unable to increase your initial stake. Being able to raise some 10M for a red5-like trading program would enable adding 3 zeros to the above scenarios making what you see represent only 0.1% of what could be possible. It's a high price to pay for not working to get as much initial capital as you can to do the best job you can.

Some Hidden Red1 Programs

If your trading program has the equivalent of an equal percent profit target stop loss combination, then you are operating a red1 program where only the win loss ratio can help you compensate for the portfolio degradation. And having a red1 program, you have a flawed program from the start that can only destroy your portfolio over the long term.

How could you design such a thing without the actual red1 program? Easy, you design an equivalent to red1. Say you have a system using Bollinger bands with a profit target stop loss set at ±2σ. Over a number of trades, these barrier-like limits will average out to a fix fraction position sizing scheme and you have your red1 program. What should you do? Simple, integrate in your program the over-compensation of red4 or red5: change the barrier-like limits to PT=2.2σ and SL=1.8σ to produce something close to red4. By pushing a little bit more, jump to red5 with PT= 2.4σ and SL=1.6σ. That's it, job done. What you are looking for is to have the following: $(1 + 2.4σ/P) * (1 – 1.6σ/P) > 1.00$. The equivalent of $(1 + fw\%) * (1 – fl\%) > 1.00$.

How to Use Red5?

You don't need to adopt the red5 program as is, what you need to do is make the minor adjustments proposed in red5 to your own trading programs, or at least make sure that your program is not of the red1 type. Red4 is acceptable, it does generate some alpha. But your best choice would be to incorporate what red5 is teaching into your own strategy designs simply by making sure you over-compensate. And this can be easily done; all it takes is to have a positive position sizing imbalance as in: $(1 + fw\%) * (1 – fl\%) > 1.00$.

If your trading strategies already over compensate for the equal fix fraction position sizing, all the above just says keep doing it, that's the way to go.

Red5 is a guideline for what to do in the face of uncertainty. It can't give you the probability of what is coming your way over the next 50 years nor can it give you what is the probability of the next price move. However, that might not that important.

What red5 does is set a few trading rules, that are not related to what the market does but to what you will do should the price move one way or the other. Red5 is a trading strategy in its own right and not a betting system. What ever the stock market may do, I don't know. However, I can certainly set my own rules of engagement what ever the market may do.

By setting your stock trading program to mimic the lessons from red5, all you are doing is improving your chances of achieving much higher returns than what most “theoretician” could propose. You are the one to select how you trade, you are the one to select on what you trade. All red5 does is show a slightly different way of looking at the problem which could have a major impact on your future portfolio.

It is not because you have a trading plan that can span years and decades, that common sense should be thrown out the window. There are many ways to improve on red5, especially in the win loss department. By looking at a chart like Fig. 15, it should become evident that one should have a long term perspective since the last few trades can have a major impact on a portfolio. Red5 plays a compounding game, the greater the number of trades, the better the hit rate (W/k>0.50), the more the program will make you money.

To resume all that has been said: I recommend over-compensating the equal fixed fraction position sizing trading scheme deficiencies, and transform a losing proposition into an exponential alpha generation machine. I know that for my own programs, I will make sure to add the booster controller function to my own strategy designs.

Hope this paper can be of some use and help you to also design better trading systems.

Good trading to all.

Guy

APPENDIX

Programs discussed in the paper.

The following restate the programs mentioned in this paper. They are in pseudo-code and therefore should be easily adaptable to most trading software out there. Variable declarations have been omitted on purpose. Are shown red's 5 programs (red1, red2, red3, red4, and red5); by commenting out the specified lines, one can move from one version to the next. I hope it is clear from the context what to comment out to get the desired version.

Of interest are: red2 or red3 (CFFPS), red4 (DCFFPS) and red5 (BDCFFPS) programs. I would have liked to say that blue's program has some value, but I can't simply based on the evidence and its rather low performance level. One thing is sure, please don't trade red1.

Programs

```
//Red's Fixed Fractional Position Sizing Programs

Account = 10000;         // Initial account, could be scaled to any amount

// Red1               //  plain equal fixed fraction without compensation factors.
Fw% = 10%;       // 10% profit target:  percent of equity gain
fl%  = 10%;        // 10% stop loss: percent of equity loss

// Red2: with a profit target compensation factor.
fw% = fw%/(1-fw%);     // adjustment factor: a/(1-a) to correct for degradation: fw% = 11.11%
// to keep red1, comment out line fw% above

// Red3: with stop loss compensation factor. (not shown in forum)
// fl% = fl%/(1+fl%);      // adjustment factor: a/(1+a) to correct for degradation
// to implement both compensators, remove comment symbol "//"  above
// Red4 will result
==
// Red5:  adds a booster to the compensators and requires both
// compensator factors to be modified (see equation (5))
c = 0.05;                              // value used to generate Fig. 18
// fw% = fw%/(1-(fw%+c));  // booster adjustment factor: a/(1-(a+c))
// fl%  = fl%/(1+(fl%+c));      // booster adjustment factor: a/(1+(a+c))
// to implement boosted compensators, remove comment "//"  for the two lines above

InstallProfitTarget(fw%);         // Stock rises by fw%, take profit, sell
InstallTrailingStopLoss(fl%);   // Stock falls by fw% then take loss, sell
Do while until you decide to quit or 12,500 trading days (≈50 years)
If NoActivePosition then         // take one
.BuyAtMarket Q shares =  Account/P(t);
// use total cash in account to buy Q shares at next price
```

Here is blue's simplified version of his trading strategy:

```
//Blue's Betting System Program

Account = 10000;     // initial account, could be scaled to any amount
b = 0.10;                  // percent of account put at play, can be modified: 0 < b < 1
Bet = b*account;      // amount bet on each trade
InstallProfitTarget(b*account);    // Stock gains amount = b*account, take profit
```



```
InstallStopLoss(b*account);       // Stock falls by amount = b*account, sell, take loss.
Do while until you decide to quit or 12,500 trading days (50 years)
If NoActivePosition then                              // take one, buy for b*Account of shares
 BuyAtMarket Q shares =  b*Account/P(t);      // use 10% of account A(k)(blue) for next bet
```

Here are some of my previous papers, all available free, for those that might wish to push the limits even further.

Alpha Power: Adding More Alpha to Portfolio Return (2007)

> *A trading strategy designed to profit even from randomly generated prices.*

A Jensen Modified Sharpe Ratio to Improve Portfolio Performance (2008)

> *A more elaborate view with the mathematics of a trading system designed to accumulate shares for the long term while trading over the process.*

Alpha Power: The Implementation

> *Simulations based on the Alpha Power methodology over real market data. Shows how the system would have performed on past data on 43 stocks data sets over a 6 years testing period.*

Visit my web site for more articles all related to the above papers:

My web site: AlphaPowerTrading

My e-mail: guyrfleury@gmail.com