Market Timing Portfolio Performance   
using Technical Indicators RSI, MACD and TD DeMark

Anuj Pradhan

Rutgers University, New Brunswick

CONTENTS

1. Abstract……………………………………………………………………………………………3
2. Introduction…………………………………………………………………………………...…...4
3. Literature Review………………………………………………………………………………….5
4. Dataset and Methodology…………………………………………………………………………7
5. Theory/Methods Involved………………………………………………………………………....8
6. Results and Discussion……………………………………………………………...……...……17
7. Appendix…………………………………………………………………………………………22
8. References…………………………………………………………….………………………….23

Abstract

We examine the performance of a portfolio of 27 stocks, selected from the top traded S&P 500 index, using technical indicators RSI, MACD and TD DeMark (Setup and Countdown) across 5 years of data. Market timing is of essence, as the point of entering a position is indicated by the signals generated by these technical, and subsequent fixed-period returns are backtested and compared with a random buy-and-hold strategy, to indicate excess returns. The portfolio is then optimized for beta neutrality and tested for 1 year of out-of-sample period. The results have shown to be statistically insignificant, as excess returns have often been outperformed by random strategies.

Keywords: Technical Analysis, Portfolio Optimization, Back-testing, Beta-Neutral, Financial Markets, S&P 500, Market Timing

Market Timing Portfolio Performance   
using Technical Indicators RSI, MACD and TD DeMark

INTRODUCTION

Technical analysis uses a variety of charts and calculations to spot trends in the market and individual stocks and to try to predict what will happen next. Technical analysts don't bother looking at any of the qualitative data about a company (for example, its management team or the industry that it is in); instead, they believe that they can accurately predict the future price of a stock by looking at its historical prices and other trading variables. Technical analysis assumes that market psychology influences trading in a way that lets them predict when a stock will rise or fall. For that reason, many technical analysts are also market timers, who believe that technical analysis can be applied just as easily to the market as a whole as to an individual stock. Critics of technical analysis, and there are many, say that the whole endeavor is a waste of time and effort. They point to academic studies like Burton Malkiel's "A Random Walk Down Wall Street" as evidence that there is no possible way to predict future prices using historical prices. Others contend that if any such systems were found to be successful, those who practiced them would be wealthy beyond their wildest dreams, and yet there aren't any billionaire technical analysts (yet).

This paper tries to test the predictability of some of these Indicators widely followed by traders and investors the world over.

# Literature Review

In an effort to have a firm foundation on the direction of this research, and under the guidance of Prof. Paisan Limratanamongkol, a number of previously published papers were studied and reviewed.

The paper closest to the goal of this project, was the one by Marco Lissandrin, Donnacha Daly and Didier Sornette published in November, 2015. In it, they statistically tested DeMark on the commodity markets over a period of 10 years, from 2004 to 2014.

The predictive power of 3 of the various elements of DeMark were studied in two steps. The first was to compare conditional returns (Lissandrin, Daly and Didier, 2015) on entry signals to exact unconditional return distributions (which represent the market). An over-performance of the conditional distribution compared with the market suggests that the tested indicator might have predictive power. The second step used approximated permutation tests to check if the initial suggestion was correct.

Theirs was a long and short strategy, and they found that all three indicators exhibit predictive power on some commodity futures. Most of the times, the entry signals provided by the indicators showed predictive power only for long or only for short positions. Having considered that long and short entry signals are generated by symmetrical algorithms, this confirmed the fact that uptrends and downtrends were asymmetrical in the markets.

But still, recent studies have suggested the existence of speculative periods in several markets (Sornette, 2009; Gilbert, 2010; Phillips and Yu, 2010). Investors and traders cannot always enter positions when a price correction is yet to come (Gromb and Vayanos, 2010). For example, market exposures in trading books are limited by capital constraints (Shleifer and Vishny, 1997) and by internal risk limits. In addition, even well-informed traders must formulate price expectations based on partial or uncertain data (Gorton et al., 2007; Khan, 2009) and this stimulates the use of rational herding behaviours, which have been described by Devenow and Welch (1996), Bikhchandani and Sharma (2001).

Research into MACD (Moving Average Convergence Divergence) and RSI (Relative Strength Index) has on the other hand been quite abundantly found over the past century (Faber 2010). Faber’s study used 10 Industry Portfolios (Consumer Non- Durables, Consumer Durables, Manufacturing, Energy, Technology, Telecommunications, Shops and Some Services, Health and Other) with monthly returns from July 1926 through December 2009, encompassing over eight decades of US equity sector returns. The study begins in 1928 since a year was needed for a ranking period, that they used to rank the portfolios and select the top x percentiles from each sector for varying portfolios.

Recent techniques have relied on eyeballing and visual intuition for their development (as readily found in all TA blogs and discussion forums), but this aspect is usually neglected during statistical tests. With the goal of keeping our approach visual, we examine one corner of this vast topic by studying the performance of DeMark chart indicators on commodity markets. In particular, this paper uses a Monte-Carlo based backtesting framework (Aronson, 2007; Masters, 2010) to determine whether these individual indicators have statistically significant predictive power. This family of indicators is, commercially speaking, one of the most popular and it is also possible to make use of it as an upgrade in leading financial market terminals such as Bloomberg Professional® and Thomson Reuters® which, combined, cover roughly 60% of the market share (Stafford, 2015). Despite this, no previous study has analysed its effectiveness, which is why the point of this study was to see how DeMark, RSI and MACD fared against a random buy-and-hold strategy, created using Monte Carlo methods.

## Dataset and Methodology

The data was sourced from Yahoo finance, for 27 of the top traded stocks on the S&P 500, from Jan 3rd 2011 to Dec 31st 2015. The Open-High-Low-Close data was then reappropriated to adjust for dividends and stock splits over the period in question, to give the columns Adj\_Open, Adj\_Close, Adj\_High and Adj\_Low.

Since the strategies being tested were fixed period buy-and-hold ones, forward returns were calculated for each stock. These returns were then saved in a separate file to form covariance matrices.

The strategies were then programmed, either through the available algorithms provided on Python’s TA-Lib, which contains RSI and MACD information, or in the case of DeMark according to instructions given in Jason Perl’s DeMark Indicators.

In an effort to backtest the strategies on a portfolio containing these 27 assets, the signals were then added to the original dataset containing price information and regular returns were compared with returns from the days that our RSI, MACD and DeMark signals fired, over 200 simulations of Monte Carlo for the random strategy, and Bootstrapping for the technical strategies.

The returns calculated from the various strategies were then taken as Alphas for Portfolio optimization. The portfolio weights were optimized with respect to a benchmark portfolio containing an equally weighted positions in all 27 stocks, throughout the time period under consideration.

Taking a leaf out of Lissandrin et al. 2015, Hit-Rates were calculated for each strategy, which is the percentage of correct predictions (excess returns positive when strategy implemented). Another feature of the analysis is the Lift statistic, that shows how much better the strategy is than the Random Buy-And-Hold one by showing the difference in Hit-Rates.

**Theory Involved**

**TD DeMark Sequential**

*TD Setup* is one component of *TD Sequential*; the other component, *TD Countdown*, cannot come into play until a TD Setup formation is complete.TD Setup, however, is not only a prerequisite for the broader trend-reversal TD Countdown signal; it is also an indicator, one that can help determine whether a market is likely to be confined to a trading range or starting a directional trend.TD Setup, of course, has both buy and sell indicators, and I will address them separately.

Bearish TD Price Flip

A Bearish TD Price Flip occurs when the market records a close greater than the close four bars earlier, immediately followed by a close less than the close four bars earlier.

**TD** **Buy** **Setup**

After a bearish TD Price Flip, there must be nine consecutive closes, each one less than the corresponding close four bars earlier.

TD Buy Setup “Perfection”

The low of bars eight or nine of the TD Buy Setup or a subsequent low must be less than, or equal to, the lows of bars six and seven of the TD Buy Setup.

TD Setup perfection is deferred until that happens, and, as long as that situation remains, the risk exists for a retest of the price low of TD Buy Setup bars six or seven, prior to the minimally expected response of a one- to four-bar consolidation/reversal. Before the trader enters a long position based on a completed TD Buy Setup, TD Buy Setup perfection is needed to increase the probability of his entering the market at or near an interim price low.

**Jason Perl’s Rules for Trading TD Buy Setups**

Objectively Many people believe, mistakenly, that one should initiate a long position following every completed TD Buy Setup. i advise against doing that except under the following conditions:

1. When the TD Buy Setup has been perfected, that is, the low of TD Buy Setup bar eight or nine is less than the lows of TD Buy Setup bars six and seven,

2. When none of the bars within the TD Buy Setup has closed below TDST support, and

3. When the close of TD Buy Setup bar nine is in close proximity to TDST support

**TD** **Countdown**

TD Countdown compares the current close with the low two bars earlier for a potential buy, and compares the current close with the high two bars earlier for a prospective sell. This price relationship is an important distinction from TD Setup, because the market must be trending for TD Countdown to objectively identify the likely exhaustion point for a trend reversal.

One can start looking for the first bar of a TD Buy Countdown when a TD Buy Setup is in place.

**To Initiate TD Buy Countdown**

*After*

TD Buy Setup is in place, look for the initiation of a TD buy Countdown.

*If*

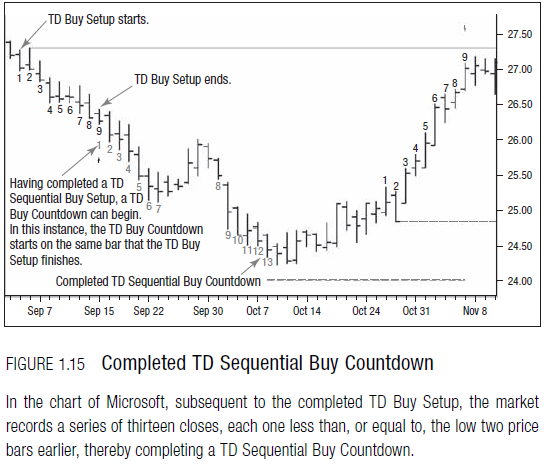
Bar nine of a TD Buy Setup also has a close less than, or equal to, the low two bars earlier,

*Then*,

Bar nine of a TD Buy Setup becomes bar one of a TD Buy Countdown.

*If*

That condition is not met, Then TD Buy Countdown bar one is postponed until it does, and the TD Buy Countdown continues until there are a total of thirteen closes, each one less than, or equal to, the low two bars earlier.



©Bloomberg Market Essentials

**TD DeMark Buy Basic Algorithm (Long only):**

1: **for** each (new) t-th price bar:

2: **proc** DeMark(Pc,Ph,Pl,n,m,q,p,k,options)

3: update Setup’s counter (s = s + 1)

4: **if** s = m then

5: compute Setup’s range & R

6: update Support/Resistance levels

7: **end if**

8: **if** indicator = ST then

9: **if** no open positions then

10: **if** Pc(t) > res(t) then

11: open new long position at t + 1

12: **else** **if** Pc(t) < sup(t) then

13: open new short position at t + 1

14: **end** **if**

15: **end** **if**

16: **else** **if** indicator = Sequential then

17: **if** s = m then

18: **if** no active Countdown phase then

19: activate new Countdown phase

20: reset Countdown’s counter (c = 0)

21: **end** **if**

22: **end** **if**

23: **if** active Countdown phase then

24: check recycle and ending conditions

25: **if** Combo then

26: update c

27: **else** (normal Countdown)

28: update c

29: **end** **if**

30: **if** c = p then

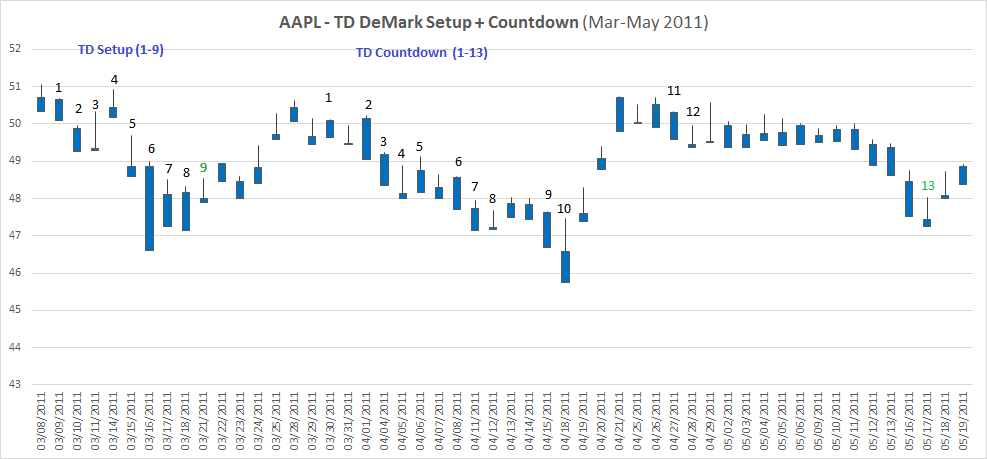
31: open new position

32: **end** **if**

33: **end** **if**

34: **end** **if**

35: **end** **proc**

****

**Example of DeMark Setup followed by Countdown Signal Generated from code**

**Momentum and Relative Strength Index (MACD and RSI)**

Momentum based strategies, in which we group both trend following and relative strength techniques, have been applied as investment strategies for over a century. Momentum has been one of the most widely discussed and researched investment strategies (some academics would prefer the term “anomaly”).

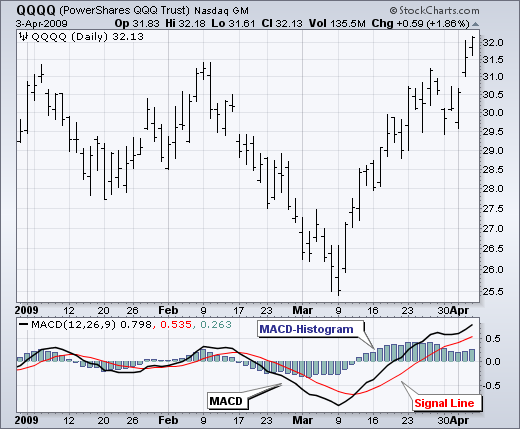
This paper tries attempts to test the effectiveness of the strategy in combination with RSI (Relative Strength Index).

**MACD Calculation**:

1. MACD Line: (12-day Exponential Moving Average (EMA) - 26-day EMA)
2. Signal Line: 9-day EMA of MACD Line
3. MACD Histogram: MACD Line - Signal Line

The MACD Line is the 12-day [Exponential Moving Average](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages) (EMA) less the 26-day EMA. Closing prices are used for these moving averages. A 9-day EMA of the MACD Line is plotted with the indicator to act as a signal line and identify turns. The MACD Histogram represents the difference between MACD and its 9-day EMA, the Signal line. The histogram is positive when the MACD Line is above its Signal line and negative when the MACD Line is below its Signal line.

The values of 12, 26 and 9 are the typical setting used with the MACD, however other values can be substituted depending on your trading style and goals.



(MACD algorithms were taken and used from the ones readily available on Python’s TA-Lib.)

**RSI Calculation:**

To simplify the calculation explanation, RSI has been broken down into its basic components: RS, Average Gain and Average Loss. This RSI calculation is based on 14 periods. Losses are expressed as positive values, not negative values.

The very first calculations for average gain and average loss are simple 14 period averages.

First Average Gain = Sum of Gains over the past 14 periods / 14.

First Average Loss = Sum of Losses over the past 14 periods / 14

The second, and subsequent, calculations are based on the prior averages and the current gain loss:

Average Gain = [(previous Average Gain) x 13 + current Gain] / 14.

Average Loss = [(previous Average Loss) x 13 + current Loss] / 14.



(RSI algorithms were taken and used from the ones readily available on Python’s TA-Lib.)

**Statistical Sampling and Calculation of Returns**

Monte Carlo and Bootstrap methods have been used to create simulations from the in comparison, smaller amounts of predictions recorded for each implementation of the trading signal.

Sampling Algorithm:

#bootstrap: test the confidence interval of the population (with replacement)

**def** bootstrap(data, num\_samples, statistic, alpha): # use: bootstrap(data[column].values, 1000, np.mean, 0.05)

"""Returns bootstrap estimate of 100.0\*(1-alpha) CI for statistic."""

n = len(data)

**if** (n > 0):

idx = np.random.randint(0, n, (num\_samples, n))

samples = data[idx]

stat = np.sort(statistic(samples, 1))

return (stat[int((alpha/2.0)\*num\_samples)],

stat[int((1-alpha/2.0)\*num\_samples)])

**else**:

**return**(np.NaN, np.NaN)

#monte\_carlo: test the bounds of a selected population (selection without replacement)

**def** monte\_carlo(data, n, num\_samples, statistic, alpha):

#Returns monte carlo simulation estimate of 100.0\*(1-alpha) CI for statistic.

**if** (len(data)):

stat = np.sort(np.array([np.mean(np.random.choice(data, size = n, replace=False)) **for** i in range (0, num\_samples)]))

return (stat[int((alpha/2.0)\*num\_samples)],

stat[int((1-alpha/2.0)\*num\_samples)])

**else**:

**return**(np.NaN, np.NaN)

Since both Monte Carlo and Bootstrap need to be applied to the same set of data, they have been rolled into one convenient function get\_stats() which returns the confidence interval for both statistics, as well as generates Hit-Rate, Lift and Excess Returns.

**Portfolio Optimization**

The portfolio optimization considered in this project is a convex quadratic optimization problem based on the following constraints:

* Strategies are Long only (All weights are non-negative)
* Portfolio is consigned to be Beta neutral
* Full investment should have taken place (All the money is invested)

The objective function for this optimization problem is given by

Non-negative weights

Beta neutrality

Full investment

Here, *lambda*  is a risk aversion free parameter. As per the problem construction *lambda* is chosen to limit the Tracking Error (TE) to under 3%.

*Alpha*  is the forecast of return value for each strategy.

*Beta*  is the value of regression coefficient.

Sigma is the variance-covariance matrix for stock returns

*Weights*  are the active weights for each stock in the portfolio

### RESULTS AND DISCUSSION

The Strategies Tested:

1. RSI+MACD buy signal within a 10 day window. (Individually, not satisfactory)
2. DeMark Sequential buy signal (Setup + Countdown day 13)
3. DeMark Setup Day 6 buy signal
4. DeMark Setup day 9 buy signal
5. DeMark Countdown Day 5 buy signal
6. DeMark Countdown Day 8 buy signal
7. DeMark Countdown Day 11 buy signal

The ones that are shown are primarily the ones I wanted to highlight, since the other did not perform up to expectation.

As can be seen in the graph, DeMark slightly outperforms a random strategy, even as you increase the holding period. But, the difference is slight, and will erode as one starts including Transaction Costs (which have been assumed to be zero in this model).

The **best performing strategies** are given below, sorted by greatest out\_mean (strategy mean returns). When annualized, these returns can go up to as much as 30% (in the case of Setup day 6 buy, for 2 weeks). But gain, it must be kept in mind that transaction costs may hamper these numbers quite a bit.



The **Portfolio Optimization results** for the strategies taken in conjunction, were as follows:

**Tracking Error**: 0.7%

**Information Ratio**: -8.7%

However, it must be mentioned, on optimization, the Active weights came out to be nearly zero, save for 2 stocks - Home Depot and IBM - with close to 25% and 75% respectively. This shows that the strategy perhaps wasn’t successful after all, since the eventual Portfolio weights came out to be approximately equal to Equally Weighted Portfolio weights.

The Alphas were taken to be the (daily) excess returns, as calculated for each strategy through Bootstrap.

The Betas were calculated with a 4-year window on daily returns (2011-2014).

The Variance-Covariance matrix was taken for daily returns.

It must be noted that since daily returns were not sufficiently better for any strategy, the results also followed suit, with portfolio optimization not showing promise in the strategy. However, it may well happen that the better (longer held) strategies perform better on optimization.

Due to the limitations of technology (my laptop’s limited capability) the study could not be justifiably performed, but as a direction for future studies in this topic, it might be helpful to keep that in mind.

As you can see, in Setup Day 6 vs Random, the Red distribution is comfortably above the random strategy, making it the best performing strategy out of the ones tested.

#### Appendix

Instructions on how to use the code/files:

1. There are 2 excel files that are read into the code:
   1. Daily Prices (DataTwentyEight.xlsx) for signal generation
   2. Daily Returns Matrix (PTP Returns.csv) for Portfolio Optimization
2. The Signal Generation Code can be found on the Python file PTP\_Signals.py (or .ipynb if that is preferred). This code generates the following files:
   1. AllSignals.xlsx with the RSI, MACD and DeMark Buy Signals added (as 1 or 0)
   2. PTP\_results\_demark.csv, with the calculated returns distributions for all strategies. This Excel was later modified to generate charts that are used in the paper.
3. Since Python doesn’t have a Quadratic Programming element to it, the TE, IR calculation was done on R. The R file is named ‘Optimization and Tracking Error.r’. This file needs the following files to be present in the directory:
   1. A\_T.csv
   2. Mean.csv
   3. Beta.csv
   4. Cov\_Mat.csv

All these files are generated by the other Python File, PTP\_Computations.py. (or .ipynb if that is preferred)

**References**

Perl, J., 2008. *DeMark Indicators. Bloomberg Market Essentials, Wiley*.

Lissandrin, M., Daly, D., Sornette, D. (2015). *Statistical Testing of DeMark Technical Indicators on Commodity Futures. International Journal of Forecasting*.

Bikhchandani, S. (2001). *Herd behavior in financial markets. IMF Staff Papers Vol. 47*.

Welch, I., 1996. *Rational herding in financial economics. European Economic Review.*

Vayanos, D., 2010. *Limits of arbitrage. Annual Review of Financial Economics Vol. 2:1.*

Gilbert, C., 2010. *Speculative influences on commodity futures prices.*

Gorton, G., Hayashi, F., Rouwenhorst, K., 2007. *The fundamentals of commodity futures returns. Working paper no. 13249. Tech. rep., National Bureau of Economic Research (NBER).*

Khan, M., 2009. *The 2008 oil price "bubble". Policy brief 09/19. Tech. rep., Peterson Institute of International Economics, Washington DC.*

Shleifer, A., Vishny, R., 1997*. The limits of arbitrage. Journal of Finance Vol. 52:2, 737–783*.