

Wiley Trading Series

Algorithmic
TRADING

WINNING STRATEGIES AND THEIR RATIONALE

+ website

ERNEST P. CHAN

WILEY

ALGORITHMIC TRADING

Founded in 1807, John Wiley & Sons is the oldest independent publishing company in the United States. With offices in North America, Europe, Australia, and Asia, Wiley is globally committed to developing and marketing print and electronic products and services for our customers' professional and personal knowledge and understanding.

The Wiley Trading series features books by traders who have survived the market's ever changing temperament and have prospered—some by reinventing systems, others by getting back to basics. Whether a novice trader, professional, or somewhere in-between, these books will provide the advice and strategies needed to prosper today and well into the future.

For a list of available titles, visit our website at www.WileyFinance.com.



ALGORITHMIC TRADING

Winning Strategies and
Their Rationale

Ernest P. Chan

WILEY

Cover image: © Oleksiy Lebedyev/istock

Cover design: Wiley

Copyright © 2013 by Ernest P. Chan. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the Web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Chan, Ernest P., 1966–

Algorithmic trading : winning strategies and their rationale / Ernest P. Chan.

pages cm. — (Wiley trading series)

Includes bibliographical references and index.

ISBN 978-1-118-46014-6 (cloth)

1. Investment analysis. 2. Stocks. 3. Exchange traded funds. 4. Algorithms.

5. Program trading (Securities) I. Title.

HG4529.C443 2013

332.63'2042—dc23

2013008380

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To my parents, HungYip and Ching, and my partner, Ben

CONTENTS

Preface	ix
CHAPTER 1 Backtesting and Automated Execution	1
CHAPTER 2 The Basics of Mean Reversion	39
CHAPTER 3 Implementing Mean Reversion Strategies	63
CHAPTER 4 Mean Reversion of Stocks and ETFs	87
CHAPTER 5 Mean Reversion of Currencies and Futures	107
CHAPTER 6 Interday Momentum Strategies	133
CHAPTER 7 Intraday Momentum Strategies	155
CHAPTER 8 Risk Management	169
Conclusion	187
Bibliography	191
About the Author	197
About the Website	199
Index	201

P R E F A C E

This book is a practical guide to algorithmic trading strategies that can be readily implemented by both retail and institutional traders. It is not an academic treatise on financial theory. Rather, I hope to make accessible to the reader some of the most useful financial research done in the past few decades, mixing them with insights I gained from actually exploiting some of those theories in live trading.

Because strategies take a central place in this book, we will cover a wide array of them, broadly divided into the mean-reverting and momentum camps, and we will lay out standard techniques for trading each category of strategies, and equally important, the fundamental reasons why a strategy should work. The emphasis throughout is on simple and linear strategies, as an antidote to the overfitting and data-snooping biases that often plague complex strategies.

In the mean-reverting camp, we will discuss the multiple statistical techniques (augmented Dickey-Fuller [ADF] test, Hurst exponent, Variance Ratio test, half-life) for detecting “time series” mean reversion or stationarity, and for detecting cointegration of a portfolio of instruments (cointegrated augmented Dickey Fuller [CADF] test, Johansen test). Beyond the mechanical application of these statistical tests to time series, we strive to convey an intuitive understanding of what they are really testing and the simple mathematical equations behind them.

We will explain the simplest techniques and strategies for trading mean-reverting portfolios (linear, Bollinger band, Kalman filter), and whether using raw prices, log prices, or ratios make the most sense as inputs to these tests and strategies. In particular, we show that the Kalman filter is useful

to traders in multiple ways and in multiple strategies. Distinction between time series versus cross-sectional mean reversion will be made. We will debate the pros and cons of “scaling-in” and highlight the danger of data errors in mean-reverting strategies, especially those that deal with spreads.

Examples of mean-reverting strategies will be drawn from interday and intraday stocks models, exchange-traded fund (ETF) pairs and triplets, ETFs versus their component stocks, currency pairs, and futures calendar and intermarket spreads. We will explain what makes trading some of these strategies quite challenging in recent years due to the rise of dark pools and high-frequency trading. We will also illustrate how certain fundamental considerations can explain the temporary unhinging of a hitherto very profitable ETF pair and how the same considerations can lead one to construct an improved version of the strategy. When discussing currency trading, we take care to explain why even the calculation of returns may seem foreign to an equity trader, and where such concepts as rollover interest may sometimes be important. Much emphasis will be devoted to the study of spot returns versus roll returns in futures, and several futures trading strategies can be derived or understood from a simple mathematical model of futures prices. The concepts of backwardation and contango will be illustrated graphically as well as mathematically. The chapter on mean reversion of currencies and futures cumulates in the study of a very special future: the volatility (VX) future, and how it can form the basis of some quite lucrative strategies.

In the momentum camp, we start by explaining a few statistical tests for times series momentum. The main theme, though, is to explore the four main drivers of momentum in stocks and futures and to propose strategies that can extract time series and cross-sectional momentum. Roll returns in futures is one of those drivers, but it turns out that forced asset sales and purchases is the main driver of stock and ETF momentum in many diverse circumstances. Some of the newer momentum strategies based on news events, news sentiment, leveraged ETFs, order flow, and high-frequency trading will be covered. Finally, we will look at the pros and cons of momentum versus mean-reverting strategies and discover their diametrically different risk-return characteristics under different market regimes in recent financial history.

I have always maintained that it is easy to find published, supposedly profitable, strategies in the many books, magazines, or blogs out there, but much harder to see why they may be flawed and perhaps ultimately doomed. So, despite the emphasis on suggesting prototype strategies, we

will also discuss many common pitfalls of algorithmic trading strategies, which may be almost as valuable to the reader as the description of the strategies themselves. These pitfalls can cause live trading results to diverge significantly from their backtests. As veterans of algorithmic trading will also agree, the same theoretical strategy can result in spectacular profits and abysmal losses, depending on the details of implementation. Hence, in this book I have lavished attention on the nitty-gritties of backtesting and sometimes live implementation of these strategies, with discussions of concepts such as data-snooping bias, survivorship bias, primary versus consolidated quotes, the venue dependence of currency quotes, the nuances of short-sale constraints, the construction of futures continuous contracts, and the use of futures closing versus settlement prices in backtests. We also highlight some instances of “regime shift” historically when even the most correct backtest will fail to predict the future returns of a strategy.

I have also paid attention to choosing the right software platform for backtesting and automated execution, given that MATLAB[®], my favorite language, is no longer the only contender in this department. I will survey the state of the art in technology, for every level of programming skills, and for many different budgets. In particular, we draw attention to the “integrated development environment” for traders, ranging from the industrial-strength platforms such as Deltix to the myriad open-source versions such as TradeLink. As we will explain, the ease of switching from backtesting to live trading mode is the most important virtue of such platforms. The fashionable concept of “complex event processing” will also be introduced in this context.

I covered risk and money management in my previous book, which was built on the Kelly formula—a formula that determines the optimal leverage and capital allocation while balancing returns versus risks. I once again cover risk and money management here, still based on the Kelly formula, but tempered with my practical experience in risk management involving black swans, constant proportion portfolio insurance, and stop losses. (U.S. Supreme Court Justice Robert H. Jackson could have been talking about the application of the Kelly formula when he said we should “*temper* its doctrinaire logic with a little practical wisdom.”) We especially focus on finding the optimal leverage in realistic situations when we can no longer assume Gaussian distribution of returns. Also, we consider whether “risk indicators” might be a useful component of a comprehensive risk management scheme.

One general technique that I have overlooked previously is the use of Monte Carlo simulations. Here, we demonstrate using simulated, as opposed

to historical, data to test the statistical significance of a backtest as well as to assess the tail risk of a strategy.

This book is meant as a follow-up to my previous book, *Quantitative Trading*. There, I focused on basic techniques for an algorithmic trader, such as how to find ideas for new strategies, how to backtest a strategy, basic considerations in automating your executions, and, finally, risk management via the Kelly formula. Yes, a few useful example strategies were sprinkled throughout, but those were not the emphasis. If you are completely new to trading algorithmically, that is a good book to read. *Algorithmic Trading*, however, is all about strategies.

All of the examples in this book are built around MATLAB codes, and they are all available for download from www.wiley.com/go/algotrading or my website at www.epchan.com/book2. Readers will find the password embedded in the first example. Readers unfamiliar with MATLAB may want to study the tutorial in *Quantitative Trading*, or watch the free webinars on mathworks.com. Furthermore, the MATLAB Statistics Toolbox was occasionally used. (All MATLAB products are available as free trials from MathWorks.)

Software and mathematics are the twin languages of algorithmic trading. Readers will find this book involves somewhat more mathematics than my previous one. This is because of my desire to inject more precision in discussing the concepts involved in financial markets, and also because I believe using simple mathematical models for trading can be more advantageous than using the usual “data-mining” approach. That is to say, instead of throwing as many technical trading indicators or rules at a price series to see which indicator or rule is profitable—a practice that invites data-snooping bias—we try to distill the fundamental property of that price series using a simple mathematical model. We can then exploit that model to our financial benefit. Nevertheless, the level of mathematics needed in the trading of stocks, futures, and currencies is far lower than that needed in derivatives trading, and anyone familiar with freshman calculus, linear algebra, and statistics should be able to follow my discussions without problems. If you find the equations too confusing, you can just go straight to the examples and see their concrete implementations as software codes.

When I wrote my first book, I was an independent trader, though one who had worked in the institutional investment management industry for many years. In the subsequent years, I have started and managed two hedge funds, either with a partner or by myself. I have survived the 2007 summer quant funds meltdown, the 2008 financial crisis, the 2010 flash crash, the

2011 U.S. federal debt downgrade, and the 2011–2012 European debt crisis. Therefore, I am more confident than before that my initial approach to algorithmic trading is sound, though I have certainly learned much more in the interim. For instance, I have found that it is seldom a good idea to manually override a model no matter how treacherous the market is looking; that it is always better to be underleveraged than overleveraged, especially when managing other people's money; that strategy performance often mean-reverts; and that overconfidence in a strategy is the greatest danger to us all. One learns much more from mistakes and near-catastrophes than from successes. I strove to record much of what I have learned in the past four years in this book.

My fund management experience has not changed my focus on the serious retail trader in this book. With sufficient determination, and with some adaptations and refinements, all the strategies here can be implemented by an independent trader, and they do not require a seven-figure brokerage account, nor do they require five-figure technology expenditure. My message to these traders is still the same: An individual with limited resources and computing power can still challenge powerful industry insiders at their own game.

■ The Motive

Books written by traders for other traders need to answer one basic question: Why are they doing it? More specifically, if the strategies described are any good, why would the trader publicize them, which would surely render them less profitable in the future?

To answer the second question first: Many of the strategies I will describe are quite well known to professional traders, so I am hardly throwing away any family jewels. Others have such high capacities that their profitability will not be seriously affected by a few additional traders running them. Yet others have the opposite properties: They are of such low capacity, or have other unappealing limitations that I no longer find them attractive for inclusion in my own fund's portfolio, but they may still be suitable for an individual trader's account. Finally, I will often be depicting strategies that at first sight are very promising, but may contain various pitfalls that I have not fully researched and eliminated. For example, I have not included transaction costs in my example backtest codes, which are crucial for a meaningful backtest. I often use

in-sample data to both optimize parameters and measure performance, which would surely inflate results. I am committing all these pitfalls in my examples because the simplified version is more illustrative and readable. These may be called “prototype strategies.” They are not meant to be traded “as-is,” but they are useful as illustrations of common algorithmic trading techniques, and as inspirations for the reader to further refine and improve them.

What about the basic motive question? It comes down to this: Crowdsourcing knowledge is often more efficient than any other method for gathering information. And so—as with my first book—I welcome your feedback on the strategies discussed in this book.

■ A Note about Sources and Acknowledgments

Naturally, I did not invent most of the materials presented here. Besides the traditional and commonly accessible sources of books, academic journals, magazines, blogs, and online trader forums (such as elitetrader.com and nuclearphynance.com), there are now new online expert networks such as Hightable.com and Quora.com where specific questions can be posted and often answered by true industry experts. I have personally benefited from all these sources and am grateful to the various online experts who have answered my questions with unexpected depth and details.

By virtue of my previous book and my blog (<http://epchan.blogspot.com>), I am also fortunate to have heard from a great many insightful readers, many of whom have contributed to my knowledge base.

I have also taught regular workshops in London and Singapore on various topics in algorithmic trading that were attended by many institutional analysts and traders. They have contributed valuable insights to me that may not be easily accessible in any public forums. Special workshops held for clients in Canada, China, Hong Kong, India, South Africa, and the United States have also exposed me to broad international perspectives and concerns.

I am also privileged to have collaborated with many knowledgeable finance professionals even as an independent trader and fund manager. Some of these collaborations are short-term and informal, while others lead to the formal formation of fund management companies. In particular, I thank Steve Halpern and Roger Hunter for their extensive discussions and countless joint projects and ventures.

I am indebted to Bryan Downing for introducing me to some of the trading technologies mentioned in Chapter 1, and to Rosario Ingargiola for showcasing his FXOne platform to me.

Finally, many thanks to my editor Bill Falloon at John Wiley & Sons for being always enthusiastic and supportive of my book ideas, to development editor Meg Freeborn for her unfailingly valuable suggestions, and to production editor Steven Kyritz for shepherding this book to its final form.

Backtesting and Automated Execution

While the focus of this book is on specific categories of strategies and not on general techniques of backtesting, there are a number of important considerations and common pitfalls to all strategies that need to be addressed first. If one blithely goes ahead and backtests a strategy without taking care to avoid these pitfalls, the backtesting will be useless. Or worse—it will be misleading and may cause significant financial losses.

Since backtesting typically involves the computation of an expected return and other statistical measures of the performance of a strategy, it is reasonable to question the statistical significance of these numbers. We will discuss various ways of estimating statistical significance using the methodologies of hypothesis testing and Monte Carlo simulations. In general, the more round trip trades there are in the backtest, the higher will be the statistical significance. But even if a backtest is done correctly without pitfalls and with high statistical significance, it doesn't necessarily mean that it is predictive of future returns. Regime shifts can spoil everything, and a few important historical examples will be highlighted.

The choice of a software platform for backtesting is also an important consideration and needs to be tackled early on. A good choice not only will vastly increase your productivity, it will also allow you to backtest the broadest possible spectrum of strategies in the broadest variety of asset classes.

And it will reduce or eliminate the chances of committing the aforementioned pitfalls. We will also explain why the choice of a good backtesting platform is often tied to the choice of a good automated execution platform: often, the best platform combines both functions.

■ The Importance of Backtesting

Backtesting is the process of feeding historical data to your trading strategy to see how it would have performed. The hope is that its historical performance tells us what to expect for its future performance. The importance of this process is obvious if you have developed a strategy from scratch, since you would certainly want to know how it has performed. But even if you read about a strategy from a publication, and you trust that the author did not lie about its stated performance, it is still imperative that you independently backtest the strategy. There are several reasons for this.

Often, the profitability of a strategy depends sensitively on the details of implementation. For example, are the stock orders supposed to be sent as market-on-open orders or as market orders just after the open? Are we supposed to send in an order for the E-mini Standard & Poor's (S&P) 500 future just before the 4:00 P.M. stock market closing time, or just before the 4:15 P.M. futures market closing time? Are we supposed to use the bid or ask price to trigger a trade, or are we supposed to use the last price? All these details tend to be glossed over in a published article, often justifiably so lest they distract from the main idea, but they can affect the profitability of a live-traded strategy significantly. The only way to pin down these details exactly, so as to implement them in our own automated execution system, is to backtest the strategy ourselves. In fact, ideally, our backtesting program can be transformed into an automated execution program by the push of a button to ensure the exact implementation of details.

Once we have implemented every detail of a strategy as a backtest program, we can then put them under the microscope and look for pitfalls in the backtesting process or in the strategy itself. For example, in backtesting a stock portfolio strategy with both long and short positions, have we taken into account the fact that some stocks were hard to borrow and cannot easily be shorted at any reasonable size? In backtesting an intermarket pair-trading strategy in futures, have we made sure that the closing prices of the two markets occur at the same time? The full list of pitfalls is long and tedious, but I will highlight a few common ones in the section "Common Pitfalls of

Backtesting.” Often, each market and each strategy presents its own very specific set of pitfalls. Usually, a pitfall tends to inflate the backtest performance of a strategy relative to its actual performance in the past, which is particularly dangerous.

Even if we have satisfied ourselves that we have understood and implemented every detail of a strategy in a backtesting program, and that there is no pitfall that we can discover, backtesting a published strategy can still yield important benefits.

Backtesting a published strategy allows you to conduct true out-of-sample testing in the period following publication. If that out-of-sample performance proves poor, then one has to be concerned that the strategy may have worked only on a limited data set. This is actually a more important point than people realize. Many authors will claim in their articles that the backtest results were “verified with out-of-sample data.” But, actually, if the out-of-sample testing results were poor, the authors could have just changed some parameters, or they could have tweaked the model substantially so that the results look good with the “out-of-sample” data. Hence, true out-of-sample testing cannot really begin until a strategy is published and cast in stone.

Finally, by backtesting a strategy ourselves, we often can find ways to refine and improve the strategy to make it more profitable or less risky. The backtesting process in trading should follow the “scientific method.” We should start with a hypothesis about an arbitrage opportunity, maybe based on our own intuition about the market or from some published research. We then confirm or refute this hypothesis by a backtest. If the results of the backtest aren’t good enough, we can modify our hypothesis and repeat the process.

As I emphasized earlier, performance of a strategy is often very sensitive to details, and small changes in these details can bring about substantial improvements. These changes can be as simple as changing the look-back time period for determining the moving average, or entering orders at the open rather than at the close. Backtesting a strategy allows us to experiment with every detail.

■ Common Pitfalls of Backtesting

Although almost every strategy allows for unique opportunities in committing errors in backtesting, there are a number of common themes, some generally applicable to all markets, others pertain to specific ones.

Look-ahead Bias

As its name implies, look-ahead bias means that your backtest program is using tomorrow's prices to determine today's trading signals. Or, more generally, it is using future information to make a "prediction" at the current time. A common example of look-ahead bias is to use a day's high or low price to determine the entry signal during the same day during backtesting. (Before the close of a trading day, we can't know what the high and low price of the day are.) Look-ahead bias is essentially a programming error and can infect only a backtest program but not a live trading program because there is no way a live trading program can obtain future information. This difference between backtesting and a live trading program also points to an obvious way to avoid look-ahead bias. If your backtesting and live trading programs are one and the same, and the only difference between backtesting versus live trading is what kind of data you are feeding into the program (historical data in the former, and live market data in the latter), then there can be no look-ahead bias in the program. Later on in this chapter, we will see which platforms allow the same source code to be used for both backtest and live execution.

Data-Snooping Bias and the Beauty of Linearity

Data-snooping bias is caused by having too many free parameters that are fitted to random ethereal market patterns in the past to make historical performance look good. These random market patterns are unlikely to recur in the future, so a model fitted to these patterns is unlikely to have much predictive power.

The way to detect data-snooping bias is well known: We should test the model on out-of-sample data and reject a model that doesn't pass the out-of-sample test. But this is easier said than done. Are we really willing to give up on possibly weeks of work and toss out the model completely? Few of us are blessed with such decisiveness. Many of us will instead tweak the model this way or that so that it finally performs reasonably well on both the in-sample and the out-of-sample result. But voilà! By doing this we have just turned the out-of-sample data into in-sample data.

If you are unwilling to toss out a model because of its performance on a fixed out-of-sample data set (after all, poor performance on this out-of-sample data may just be due to bad luck), or if you have a small data set to start with and really need to tweak the model using most of this data, you should consider the idea of cross-validation. That is, you should select a

number of different subsets of the data for training and tweaking your model and, more important, making sure that the model performs well on these different subsets. One reason why we prefer models with a high Sharpe ratio and short maximum drawdown duration is that this almost automatically ensures that the model will pass the cross-validation test: the only subsets where the model will fail the test are those rare drawdown periods.

There is a general approach to trading strategy construction that can minimize data-snooping bias: make the model as simple as possible, with as few parameters as possible. Many traders appreciate the second edict, but fail to realize that a model with few parameters but lots of complicated trading rules are just as susceptible to data-snooping bias. Both edicts lead to the conclusion that nonlinear models are more susceptible to data-snooping bias than linear models because nonlinear models not only are more complicated but they usually have more free parameters than linear models.

Suppose we attempt to predict price by simple extrapolation of the historical price series. A nonlinear model would certainly fit the historical data better, but that's no guarantee that it can predict a future value better. But even if we fix the number of parameters to be the same for a nonlinear model versus its linear contender, one has to remember that we can usually approximate a nonlinear model by Taylor-series expansion familiar from calculus. That means that there is usually a simpler, linear approximation corresponding to every nonlinear model, and a good reason has to be given why this linear model cannot be used. (The exceptions are those singular cases where the lower-order terms vanish. But such cases seldom describe realistic financial time series.)

An equivalent reasoning can be made in the context of what probability distributions we should assume for returns. We have heard often that the Gaussian distribution fails to capture extreme events in the financial market. But the problem with going beyond the Gaussian distribution is that we will be confronted with many choices of alternative distributions. Should it be a Student's t -distribution that allows us to capture the skew and kurtosis of the returns, or should it be a Pareto distribution that dispenses with a finite second moment completely? Any choice will have some element of arbitrariness, and the decision will be based on a finite number of observations. Hence, Occam's razor dictates that unless there are strong theoretical and empirical reasons to support a non-Gaussian distribution, a Gaussian form should be assumed.

Linear models imply not only a linear price prediction formula, but also a linear capital allocation formula. Let's say we are considering a mean-reverting model for a price series such that the change in the price dy in