Model-Based Trading Strategies

Financial-Hacker.com

Johann Christian Lotter / jcl@opgroup.de

oP group Germany GmbH

Disclaimer

- All trading systems herein are for education only.
- No profits are guaranteed.
- Don‘t blame me for losses.

U.S. Government Required Disclaimer - Commodity Futures, Trading Commission Futures, Derivatives and Options trading has large potential rewards, but also large potential risk. You must be aware of the risks and be willing to accept them in order to invest in the futures and options markets. Don't trade with money you can't afford to lose. This website is neither a solicitation nor an offer to Buy/Sell futures or options. The past performance of any trading system or methodology is not necessarily indicative of future results.

CFTC rule 4.41 - Hypothetical or simulated performance results have certain limitations. Unlike an actual performance record, simulated results do not represent actual trading. Also, since the trades have not been executed, the results may have under-or-over compensated for the impact, if any, of certain market factors, such as lack of liquidity. Simulated trading programs in general are also subject to the fact that they are designed with the benefit of hindsight. No representation is being made that any account will or is likely to achieve profit or losses similar to those shown.

Agenda

Modelling the Market

- Model based vs. data mining strategies
- The order book model
- The random walk model
- What we can learn from the random walk
- The general price curve model

Modelling the Market

Model vs. Data

- Two general approaches to strategy building:
  model based - data mining
- Model based system development starts with a market theory and attempts to find it reflected in the data.
- Data mining system development starts with the price data and attempts to find predictive patterns or rules.

Modelling the Market

Model = Simplified image of the reality

- We describe the trader behavior with a market model.
- Problem: A model is NOT the reality. The reality is unknown.
- The same reality can be described with many different models.
- The best model must be selected by experiment.

Modelling the Market

The „order book“ model

Order book:

- Ask 10k @ 1.03
- Ask 20k @ 1.02
- Ask 10k @ 1.01
- Bid 20k @ 0.99
- Bid 10k @ 0.98
- Bid 10k @ 0.97

Broker: Price = 1.01, Spread = 0.02

Modelling the Market

You buy 10k at market

- Ask 10k @ 1.03
- Ask 20k @ 1.02
- (Ask 10k @ 1.01 <- order filled at 1.01)
- Bid 20k @ 0.99
- Bid 10k @ 0.98
- Bid 10k @ 0.97

Broker: New price = 1.02, Spread = 0.03

-> A buy order pushes the price up, a sell order pushes it down

Modelling the Market

You buy 10k at market
Someone sells 10k at market

- Ask 10k @ 1.03
- Ask 20k @ 1.02
- Ask 10k @ 1.01 (your order filled at 1.01)
- Bid 20k @ 0.99 (other order filled at 0.99)
- Bid 10k @ 0.98
- Bid 10k @ 0.97

Broker: New price = 1.01, Spread = 0.02

-> buy and sell orders cancel each other

Modelling the Market

The Random Walk model

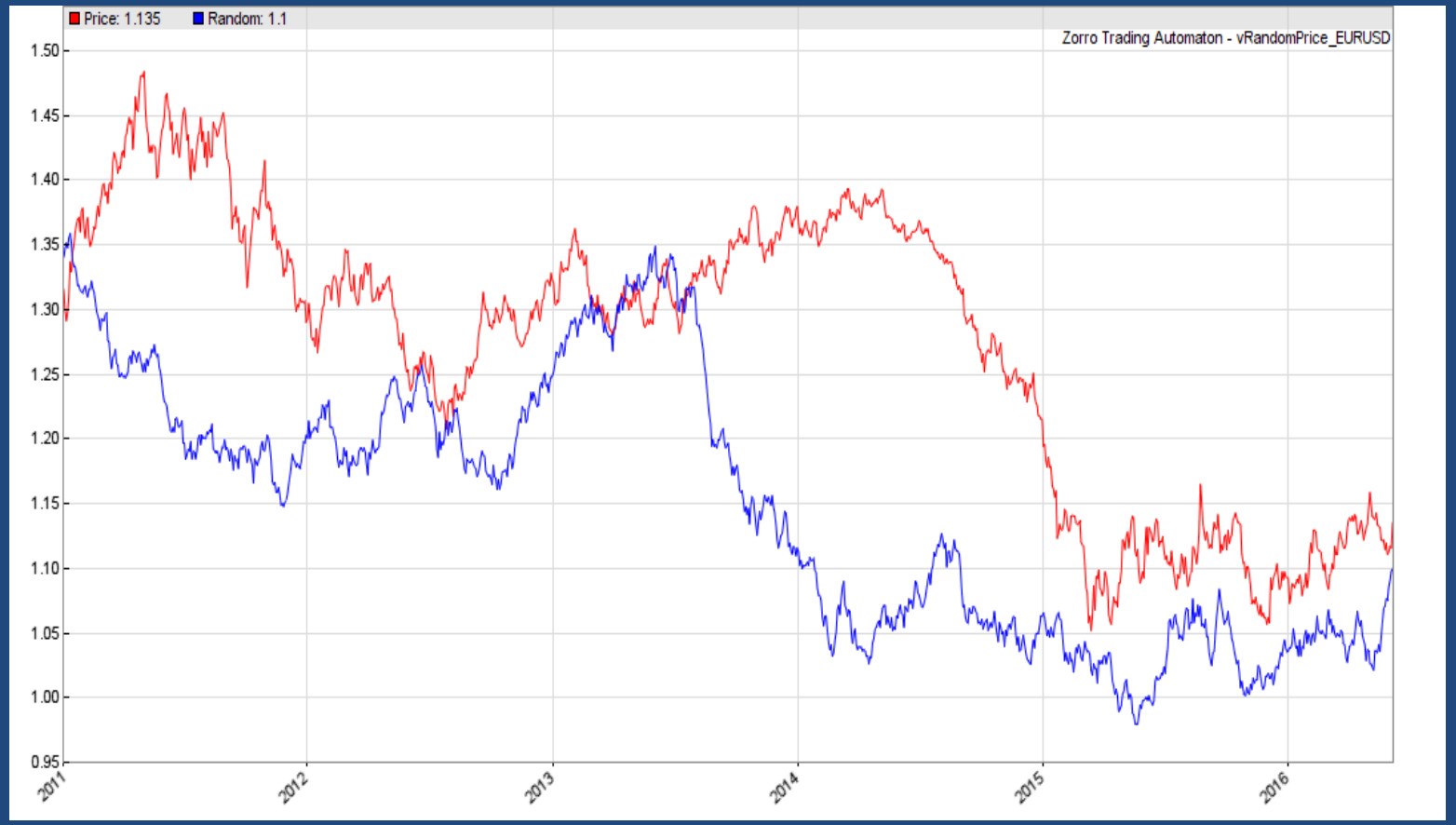
Buyers

Sellers

Price = PreviousPrice + Buyers - Sellers

Modelling the Market

Which price curve is real?



Modelling the Market

Two rules from the random walk model

Rule 1: A pure random walk curve can not be traded

(Rule of **No Roulette System**)

Rule 2: The volatility of a random walk curve is proportional to the square root of its duration

(Rule of **Square Root Volatility**).

Modelling the Market

Forces, pulling at the price curve

- Previous price (~ 99%)
- Random buyers and sellers
- Fundamental buyers and sellers
- Technical buyers and sellers

Inefficiency = systematic deviation from the Random Walk

Modelling the Market

General price curve model

$$y_t = y_{t-1} + f(y_{t-1}, \cdots y_{t-n}) + \varepsilon_f + \varepsilon_r$$

Agenda

Exploiting Market Inefficiencies

- Momentum
- Mean Reversion
- Cycles
- Stat Arb
- Constraints
- Clusters
- Patterns
- Gaps
- Seasonality
- Heteroskedasticity

Exploiting Market Inefficiencies

Momentum

$$y_t = y_{t-1} + a_1(y_{t-1} - y_{t-2}) + a_2(y_{t-2} - y_{t-3}) + \ldots + \varepsilon_f + \varepsilon_r$$

Exploiting Market Inefficiencies

A simple momentum strategy

- Detect the market regime: trend or mean reversion?
- Get a trend line with a lowpass filter
- When market regime is trending:
- Enter long on a trend line valley
- Enter short on a trend line peak

Exploiting Market Inefficiencies

Mean Reversion

$$y_t = y_{t-1} - \frac{1}{\lambda}(y_{t-1} - \hat{y}) + \varepsilon_f + \varepsilon_r$$

Exploiting Market Inefficiencies

A simple mean reversion strategy

- Detect the market regime: trend or mean reversion?
- Remove trend with a highpass filter
- When market regime is mean reverting:
- Enter short when the price exceeds a high threshold
- Enter long when the price falls below a low threshold

Exploiting Market Inefficiencies

Cycles

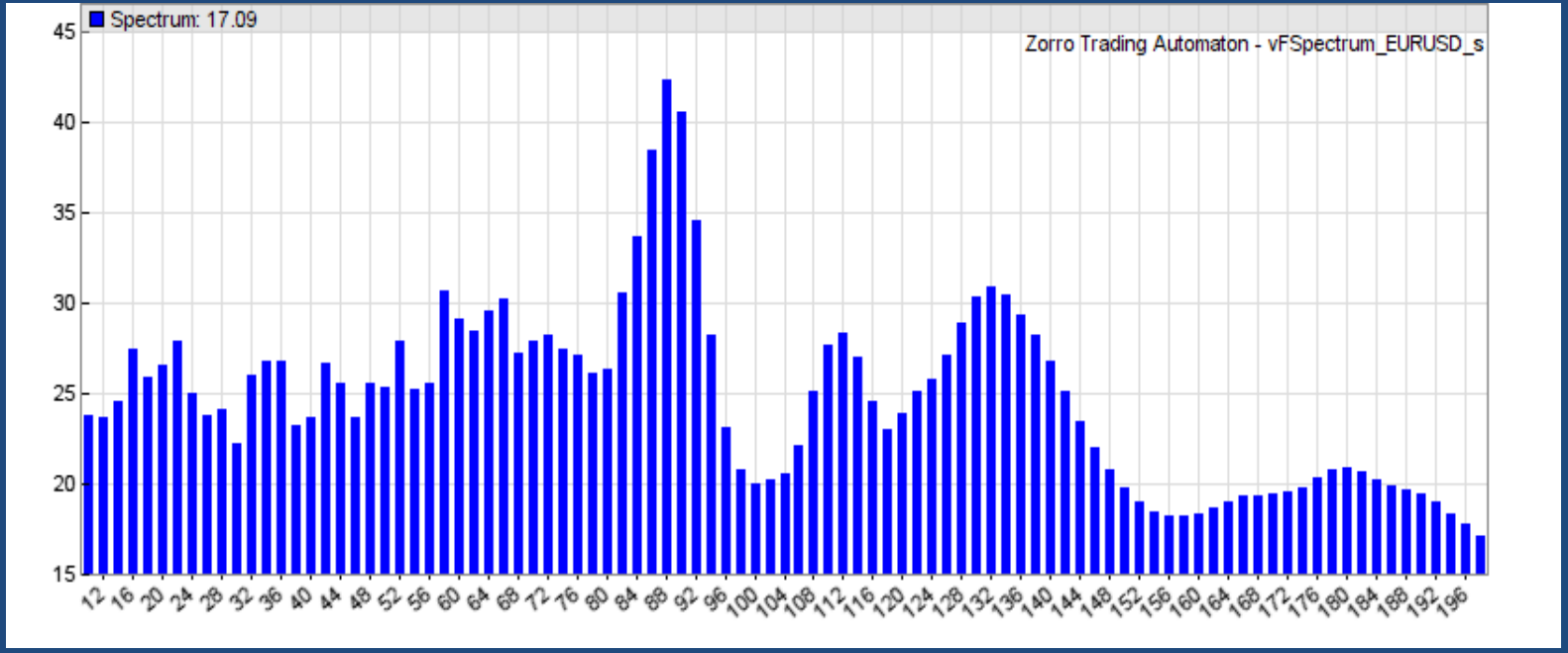
$$y_t = y_{t-1} + a_1 sin(\frac{2\pi}{c_1} t + d_1) + a_2 sin(\frac{2\pi}{c_2} t + d_2) + \ldots + \varepsilon_f + \varepsilon_r$$

a_1 = Amplitude of the first cycle

c_1 = Bar period of the first cycle

Exploiting Market Inefficiencies

Frequency spectrum of a price curve



Exploiting Market Inefficiencies

A simple cycle strategy

- Detect the dominant cycle c_1 and phase d_1.
- Get the current amplitude of the dominant cycle.
- When amplitude is above a threshold:
- Enter short when the phase d_1 is approaching a sine peak.
- Enter long when the phase d_1 is approaching a sine valley.

Exploiting Market Inefficiencies

Statistical Arbitrage

$$y_t = h_1 y_1 - h_2 y_2$$

y_t = price difference (mean reverting)

h_1, h_2 = hedge factors

Typically $h_1 = 1$, h_2 by linear regression of y_1, y_2

Exploiting Market Inefficiencies

Price constraints

- Price is restriced by an upper or lower hard boundary
- Or price is strongly mean reverting outside a soft boundary
- Classical example: The Swiss Franc cap 2011-2015
- But distant price constraints exist for most assets

Exploiting Market Inefficiencies

A simple price constraint strategy („Grid Trader“)

- Place lines at equal or increasing distances from a mid price.
- Whenever the price crosses a line:
- Close all open trades that are in profit.
- Open a new long and short trade if there isn‘t already one open at that line.
- Use a hedging method for avoiding open long and short positions at the same time.

Possible problems:

- Low short-term volatility
- Trading costs – especially rollover
- Exceeding boundaries -> margin call

Exploiting Market Inefficiencies

Price clusters

- Where do prices concentrate?
- „Support and Resistance“ -> two clusters
- „Fair price“ -> one cluster

Exploiting Market Inefficiencies

Curve patterns

- Not to be confused with „Candle Patterns“
- Some famous patterns – such as „Head and Shoulders“ – have no significance in real price curves and are probably myths.
- Other patterns – such as „Cups“ and „Half-Cups“ – really exist and can be explained by a behavior model („breakout“).
- Several algorithms for detecting curve patterns, f.i. the Frechèt algorithm.

Exploiting Market Inefficiencies

Gaps

- Overnight and weekend gaps can „amplify“ and „synchronize“ trader behavior patterns
- Trend and mean reversion before the gap reappears „with a revenge“

Exploiting Market Inefficiencies

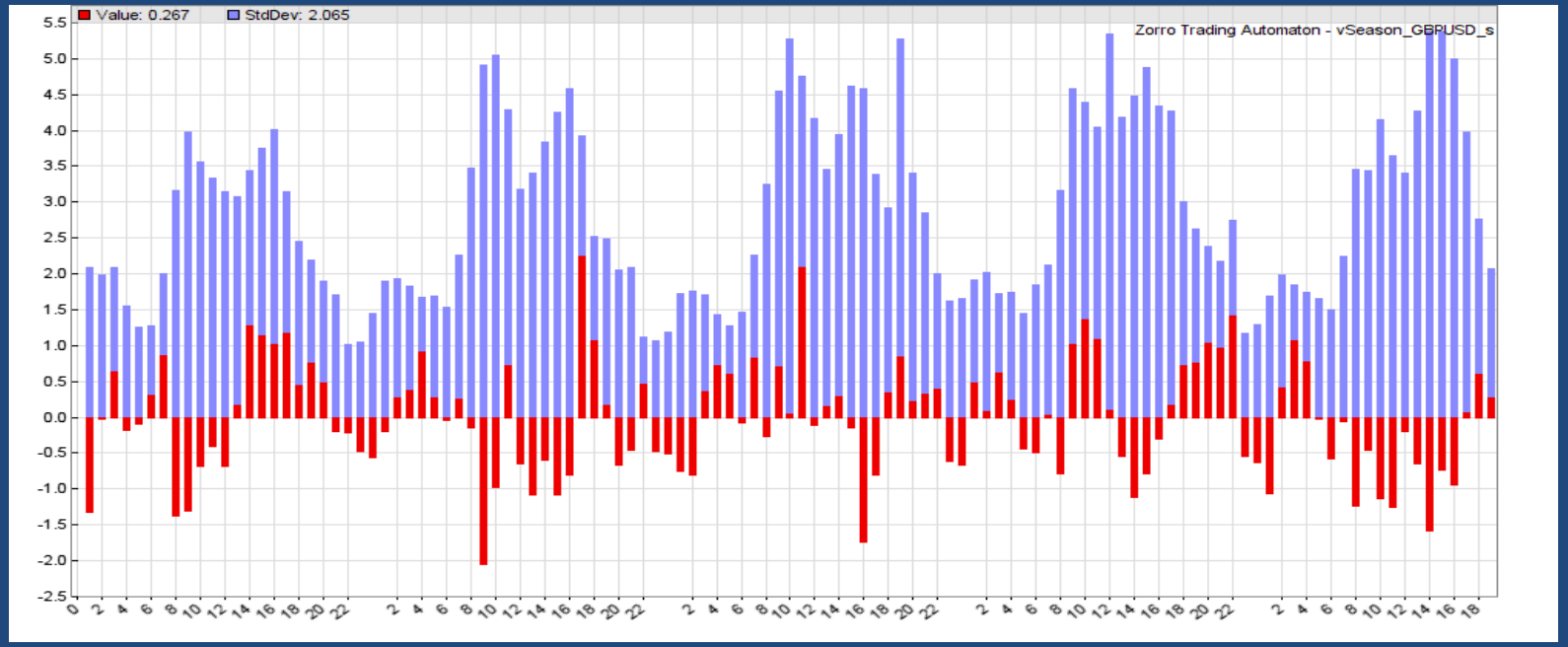
A simple gap trading strategy

- On an upwards trend, buy long on Friday when a 10-days high is reached.
- On a downwards trend, buy short on Friday when a 10-days low is reached.
- Close the position on Monday morning.
- Live trading can be followed on the Zorro forum.

Exploiting Market Inefficiencies

Seasonality

- Trader behavior depends on time of day, day of week, day of month, month of year
- Seasonal effects in a price curve can be detected by simple statistical methods



Exploiting Market Inefficiencies

Heterosketasticity

$$y_t = y_{t-1} + \varepsilon_t \sqrt{a + b(y_{t-1} - y_{t-2})^2}$$

GARCH model (Generalised Autoregressive Conditional Heteroskedasticity)

Agenda

The Development Process

1) Selecting the model. Confirming it with price data
2) Developing the trade algorithm
3) Developing the filter algorithm
4) Parameter adaption („optimizing“)
5) Test
6) Reality check
7) Implementing risk and money management

The Development Process

Step 1: Model selection

The three prerequisites for a financial model:

1) Has a **rational basis** in market structure / trader behavior
2) Can be expressed in a program flow or formula
3) Has **statistical significance** in real price curves

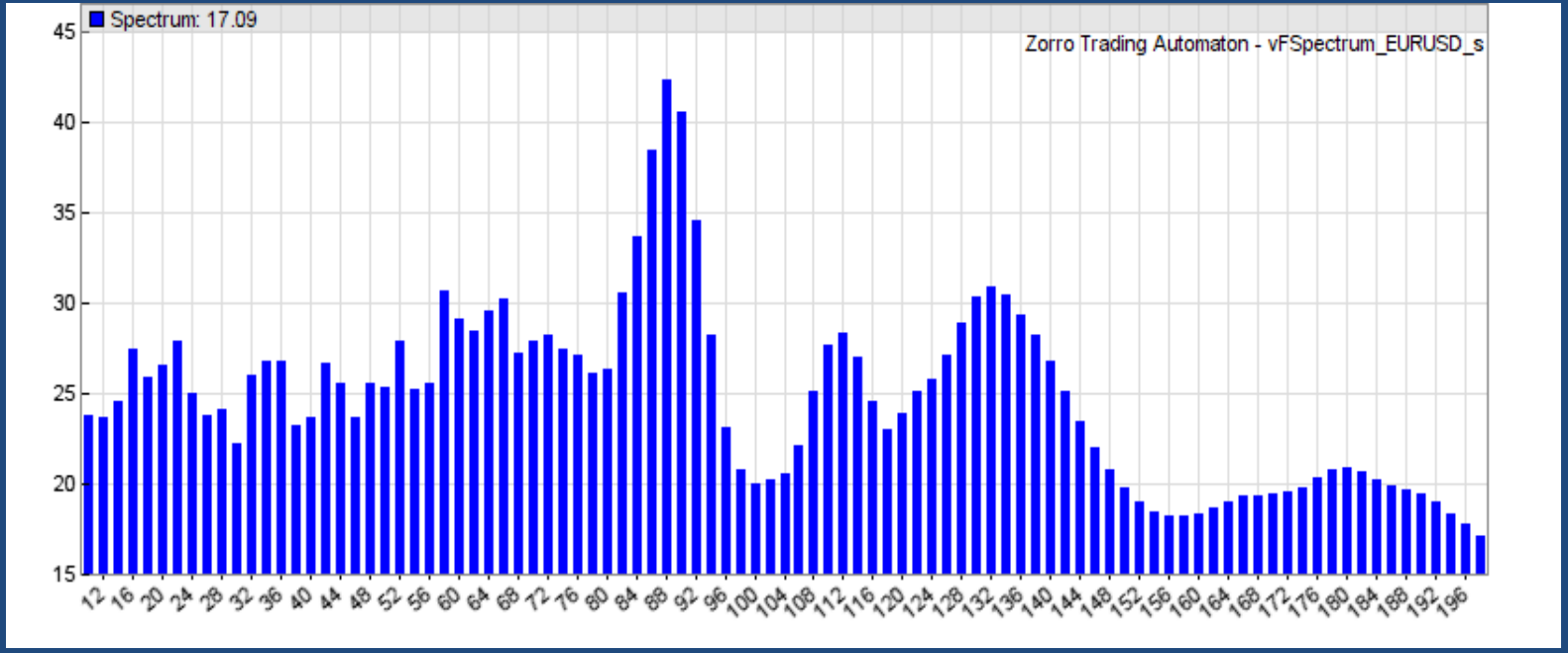
The Development Process

Confirming the model

- Find an algorithm that detects the inefficiency in price curves.
- Do a statistic. Plot a histogram.
- Compare with random walk curves or shuffled price curves. Difference should be significant.
- Do NOT rely on other people‘s research! Scam is ubiquitous (-> „Elliott Waves“, Rich Swannell)

The Development Process

Example: Frequency spectrum of a price curve



The Development Process

Step 2: Determining the algorithm

Example: Cycle strategy

- Detect the dominant cycle and phase.
- Generate a forerunning sine curve.
- Enter short at a sine peak.
- Enter long at a sine valley.
- Exit on reversal or after a half-period.

The Development Process

Step 3: The filter

A market inefficiency normally does not exist all the time. Therefore, we need a filter for determining if the inefficiency is present or not. In most cases the filter is more important than the algorithm.

Example: Cycle strategy

- Measure the amplitude of the dominant cycle.
- Trade only when the amplitude is above a threshold.

The Development Process

Step 4: Parameter adaption („Training“)

If the model has „free parameters“:

- Find out how the strategy reacts on parameter changes.
- Find the most robust parameter range („sweet spot“).
- Adapt the strategy to different assets.
- Adapt it to different market situations (even while live trading).

Bad ideas:

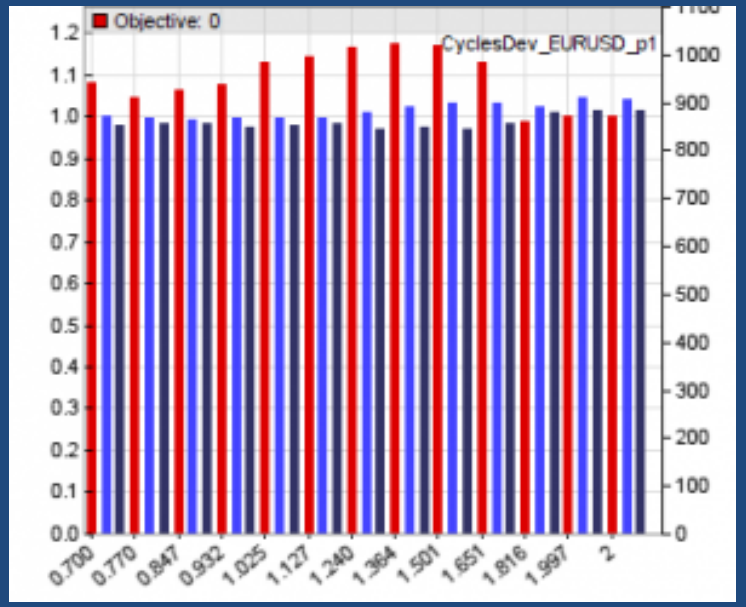
- Optimizing too many parameters.
- Optimizing for peaks (= brute force or genetic optimization).

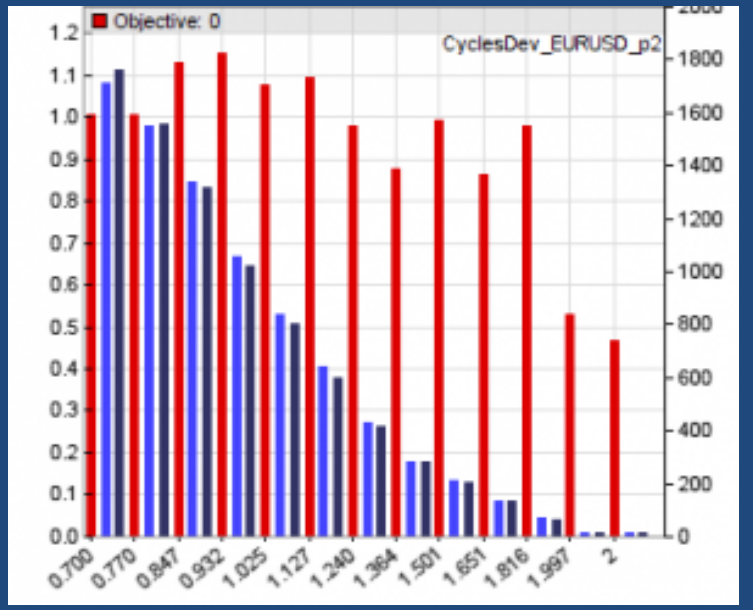
The Development Process

Example (Cycles system)

Adapted parameters: sine phase and threshold

Training results:





The Development Process

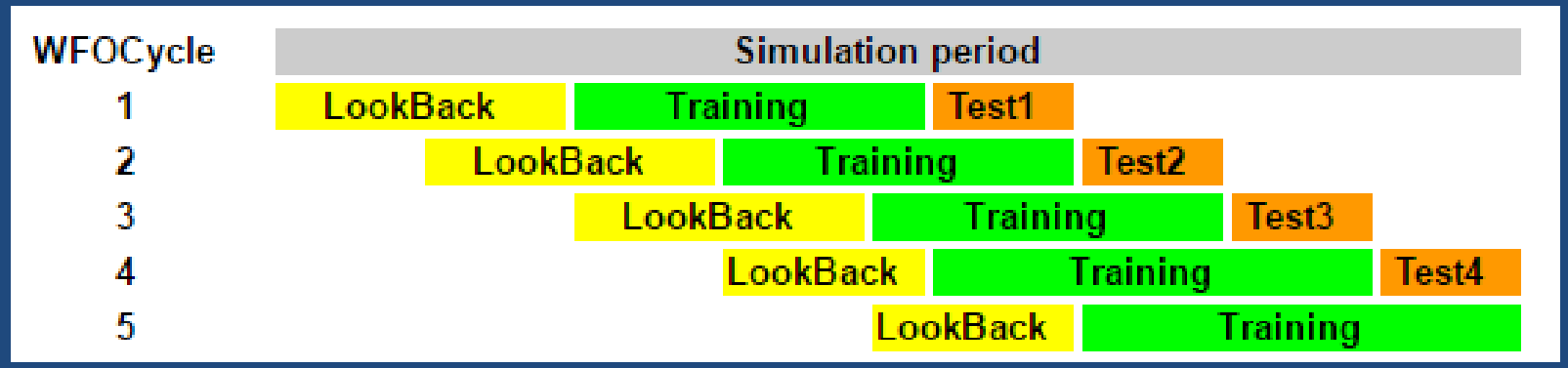
Step 5: Test

- Test should cover all significant market periods (5-10 years)
- Any parameter adaption introduce bias to the test result.
- The bias renders backtests completely useless.
- The solution: Testing the system with data not used for the adaption.

The Development Process

Walk-Forward Analysis

- Roll a window over the simulation period
- Separate the window in a training and test section.
- Good: The test is out-of-sample and still covers most of the data.
- Bad: The system depends on two more parameters.



The Development Process

Analyzing test results

Main performance parameters:

- Wins divided by losses (Profit Factor)
- Annual profit in relation to drawdown (Calmar ratio)
  (Drawdown must be normalized -> square root rule!)
- Annual return in relation to sigma (Sharpe ratio)
- Linearity of returns (R2 coefficient)

The Development Process

Monte Carlo method

For eliminating „randomness“ from the test results:

- Split the equity curve in small sections
- „Randomize“ the curve by shuffling without replacement
- Repeat 1000 times.
- Calculate test results from every shuffled curve.
- Sort test results by confidence intervals.

The Development Process

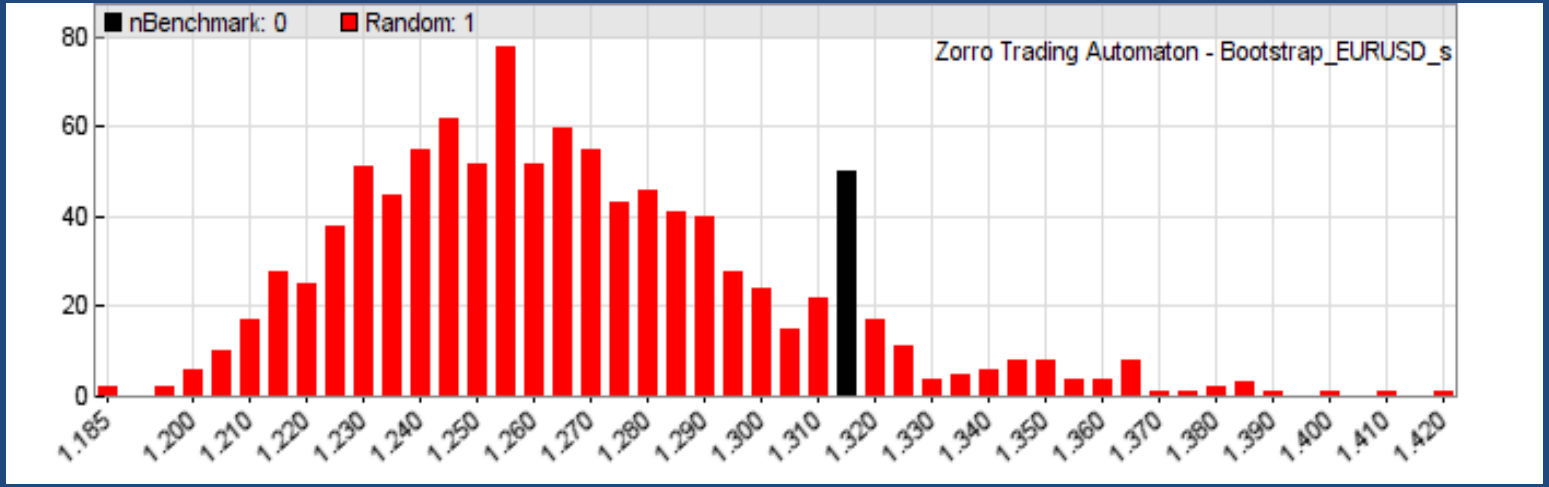
Step 6: Reality check

Even with walk forward and Monte Carlo analysis, test results still suffer from bias. Bias is introduced by the mere development process. Several methods to detect bias:

1) White's Reality Check: gives a quantitative measure of bias
2) Monte Carlo Reality Check: run the system with price curves randomized by shuffling with replacement. Plot a result distribution
3) Variants: run the system with inverted, detrended, or oversampled price curves
4) Real-out-of-sample test: Set aside a part of the data and only use it for this test.

The Development Process

White's Reality check



Details under:

http://www.financial-hacker.com/whites-reality-check/

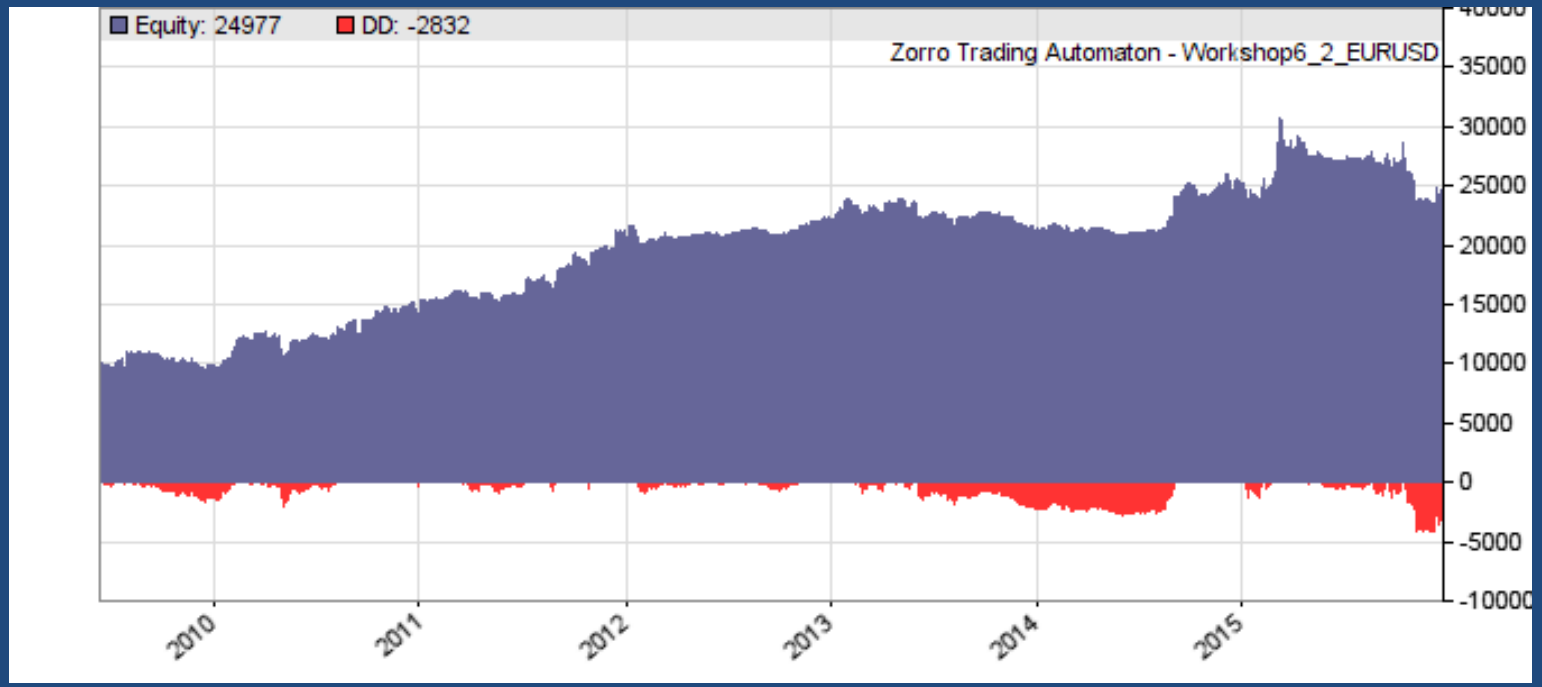
The Development Process

Step 7: Risk and Money Management

- Use a stop loss for eliminating negative outliers.
- Do not use profit targets. (If you really want to, use a profit-lock mechanism instead).
- Use an algorithm for calculating the optimal investment per portfolio component (Kelly, OptimalF, Markowitz).
- Re-invest only the square root of your profits.
- Supervise your system permanently and compare live results with backtest results (-> „Cold Blood Index“).

The Development Process

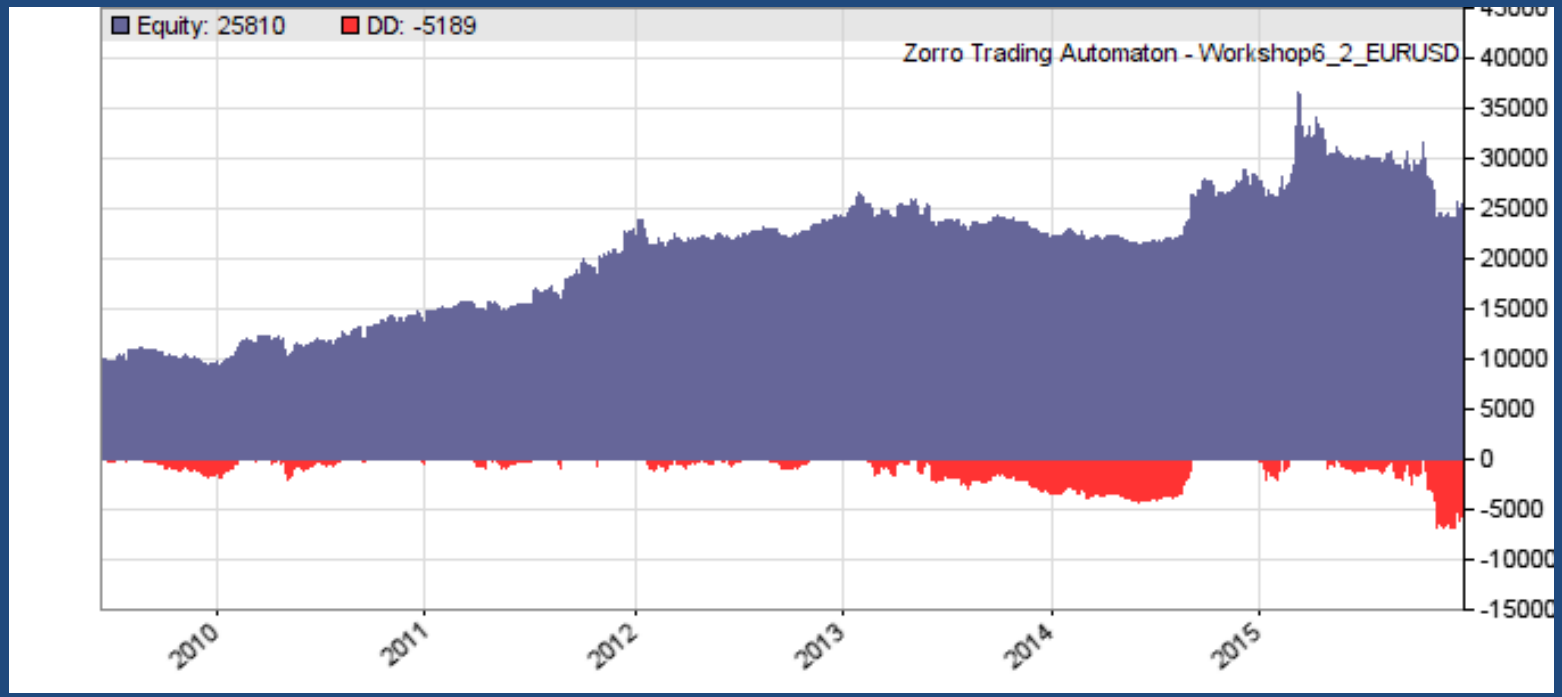
No Reinvestment



CAGR: 15%

The Development Process

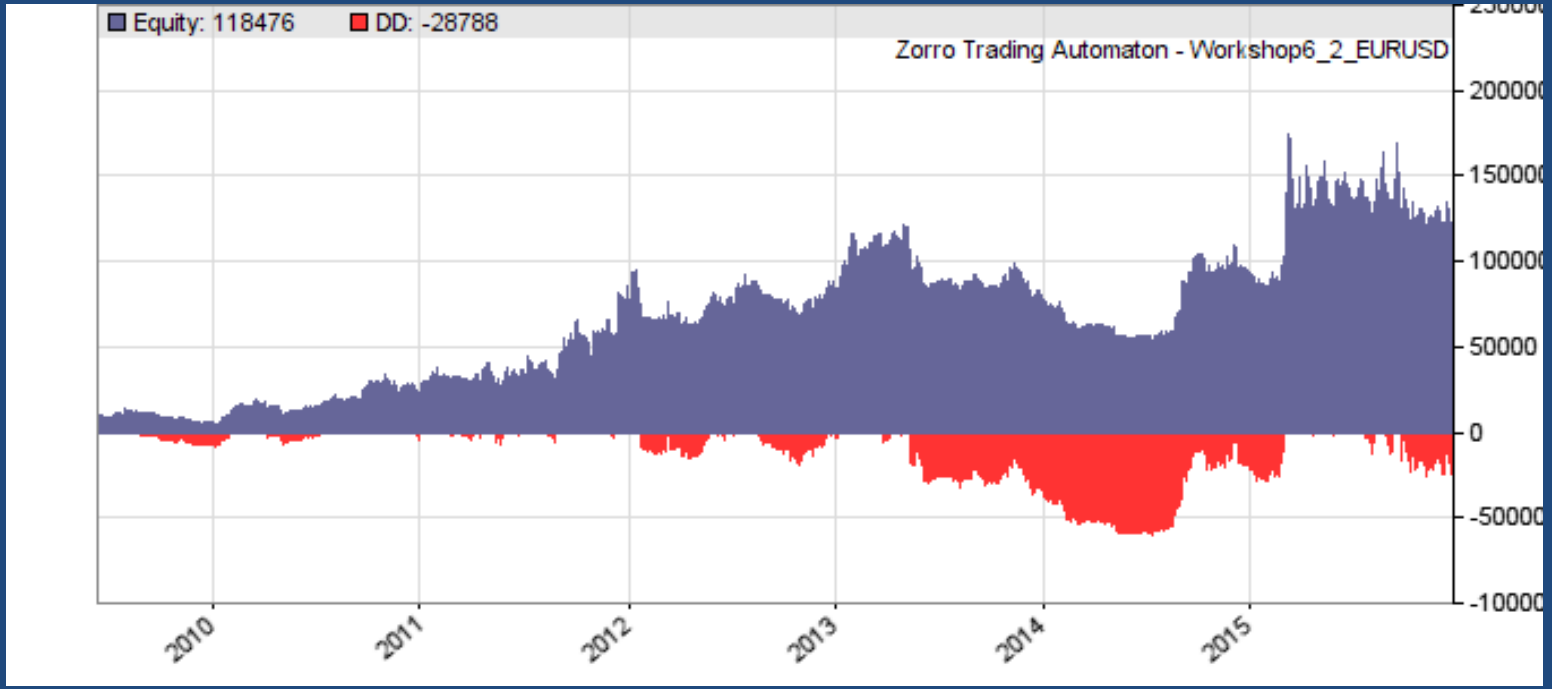
1% Reinvestment



CAGR: 16%

The Development Process

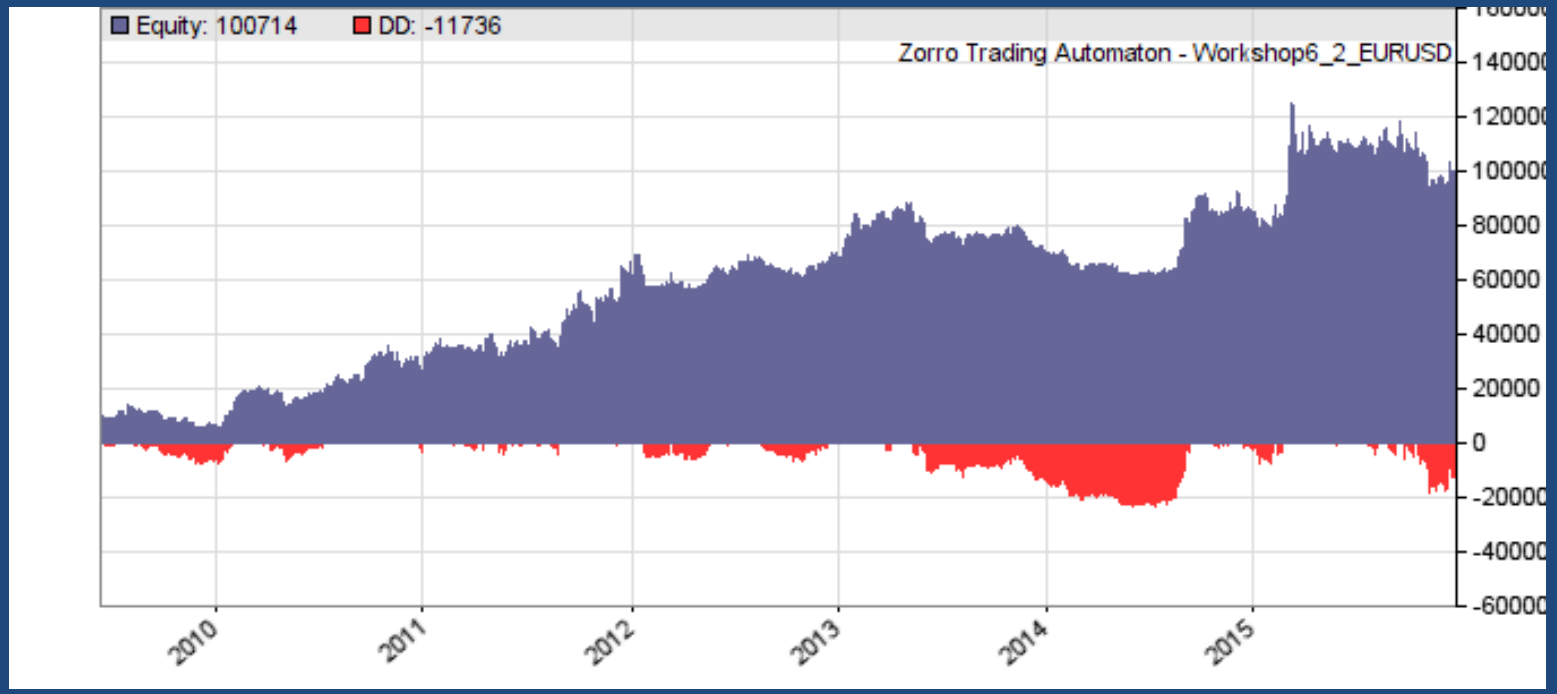
0.5 OptimalF Reinvestment



CAGR: 48%

The Development Process

Sqrt(P) OptimalF Reinvestment



CAGR: 43%

The Development Process

So far the theory…
Here's the real development process

Step 1. Visit trader forums. Look for the thread with new fabulous indicator.

Step 2. Implement the indicator with a long coding session. Ugh, the backtest does not look this good. Some coding mistake? Debug. Debug some more.

Step 3. Still no good result, but you have more tricks up your sleeve. Add a trailing stop. Run a week analysis. Tuesday is a bad day for this indicator? Add a filter for not trading on Tuesday. Add more filters for not trading between 12:00 and 14:00 and on any full moon except on Thursday. Wow, now we see some backtest profit!!

Step 4. Of course you're not fooled by in-sample results. After optimizing all 23 parameters, run a walk forward analysis. Ugh, the result does not look this good. Try different WFA cycles. Try different bar periods. Optimize more parameters. Finally, a sensational test result! And this completely out of sample!

Step 5. Trade the system live.

Step 6. Ugh, the result does not look this good.

Step 7. Hold many trading seminars for recovering your bank account.