MACHINE DESIGNED TRADING STRATEGIES

"How the TSL Machine Designed the
#1 rated Futures Truth Strategy"

Michael L. Barna
President and Founder
Trading System Lab

Trading System Lab®

REQUIRED DISCLAIMER

HYPOTHETICAL PERFORMANCE RESULTS HAVE MANY INHERENT LIMITATIONS, SOME OF WHICH ARE DESCRIBED BELOW. NO REPRESENTATION IS BEING MADE THAT ANY ACCOUNT WILL OR IS LIKELY TO ACHIEVE PROFITS OR LOSSES SIMILAR TO THOSE SHOWN.

IN FACT, THERE ARE FREQUENTLY SHARP DIFFERENCES BETWEEN HYPOTHETICAL PERFORMANCE RESULTS AND THE ACTUAL RESULTS ACHIEVED BY ANY PARTICULAR TRADING PROGRAM. ONE OF THE LIMITATIONS OF HYPOTHETICAL PERFORMANCE RESULTS IS THAT THEY ARE GENERALLY PREPARED WITH THE BENEFIT OF HINDSIGHT. IN ADDITION, HYPOTHETICAL TRADING DOES NOT INVOLVE FINANCIAL RISK, AND NO HYPOTHETICAL TRADING RECORD CAN COMPLETELY ACCOUNT FOR THE IMPACT OF FINANCIAL RISK IN ACTUAL TRADING. FOR EXAMPLE, THE ABILITY TO WITHSTAND LOSSES OR TO ADHERE TO A PARTICULAR TRADING PROGRAM IN SPITE OF TRADING LOSSES ARE MATERIAL POINTS WHICH CAN ALSO ADVERSELY AFFECT ACTUAL TRADING RESULTS.

THERE ARE NUMEROUS OTHER FACTORS RELATED TO THE MARKETS IN GENERAL OR TO THE IMPLEMENTATION OF ANY SPECIFIC TRADING PROGRAM WHICH CANNOT BE FULLY ACCOUNTED FOR IN THE PREPARATION OF HYPOTHETICAL PERFORMANCE RESULTS AND ALL OF WHICH CAN ADVERSELY AFFECT ACTUAL TRADING RESULTS.

Mike Barna, CTA

- Founder and President, Trading System Lab
- BS Mathematics, Arizona State University
- MS Astronautical and Aeronautical Engineering, Stanford University
- SVP Regency Stocks and Commodities Fund, LP,LLC (QEP Hedge Fund)
- R-MESA*, BIGBLUE*, MESA BONDS, MESA NOTES
- Former Defense Industry Rocket-Ramjet and Guidance Systems Engineer
- Star Wars Research and Development Management Engineer
- 2 CFTC Commodity Licenses, 12 FAA pilot certificates or ratings

*Multiple awards received

WHO IS TSL?

- Trading System Lab-US Based trading research and development company with a small team of international and domestic programmers, third party developers and testers. Founded by former Hedge Fund Manager.

  www.tradingsystemlab.com

- Register Machine Learning, Inc.-US Based company with a team of international and domestic machine learning scientists, IP attorneys, statisticians and programmers. Involved in government contracts.

  RML licenses the patented Discipulus to TSL exclusively

  http://www.rmltech.com

Trading System Lab®

AN EASY LANGUAGE EXAMPLE

- Moving Averages use 1 time series: Close
- Stochastics use 3 time series: H, L, C
- What Indicator uses all 5 standard time series?
- Open
- High
- Low
- Close
- Volume or Tick Volume
- We will take this indicator and adapt it with the Hilbert Dominant Cycle

Hilbert Adaptive Williams Variable Accumulation Distribution

$$WVAD = \left| \sum_{n}^{i=0} Volume[i] * \frac{Close[i] - Open[i]}{High[i] - Low[i]} \right|$$

n~HilbertDC

References Robert W. Colby:
"The Encyclopedia of Technical Market Indicators",
John Ehlers-MESA Software
n~HilbertDC

Hilbert Adaptive Williams Variable Accumulation Distribution

```
//The Encyclopedia of Technical Market Indicators, Second Edition, Colby ISBN 0-07-012057-9
//MESA SOFTWARE
//@Copyright 2013 Trading System Lab(R)

inputs:  Period(14), smooth(3), Intraday(false);
vars:vad(0),n(0),vadold(0), hldiff(0), hdc(0), rvad(0), normal(0);

hdc = HilbertDC(close);

vadold = 0;
n = 0;
vad=0;

while n <= hdc begin
hldiff = high[n] - low[n];
if intraday=false and hldiff<>0 then vadold = ( (close[n] - open[n]  )  /  hldiff )*volume[n];
if intraday=true  and hldiff<>0 then vadold = (upticks-downticks)*( (close[n] - open[n]  )  /  hldiff )*(ticks)[n];
vad = vad + vadold;
n = n + 1;
end;

rvad = highest(vad, period) - lowest(vad, period);
if rvad<>0 then normal = (vad-lowest(vad, period))/rvad;
plot1(average(normal,smooth), "VAD");
//vad=0;
plot2(.2, "Upper");
plot3(.8, "Lower");
```

C to Easy Language Code

- ANTLR (Another Tool for Language Recognition) is used for CEL (C to Easy Language) translation. ANTLR developed by Prof. Terrance Parr at U.S.F.
- Comparison testing accomplished:
  - Machine code in Discipulus
  - ANSI C-Code DLL in TradeStation
  - In-Line Assembler Code DLL in TradeStation
  - CEL code in TradeStation
- Multiple GP outputs CEL translation
- C# GUI created
- “Click and Paste” simplicity

TSL/RML: Translates: Machine Code->C code->EasyLanguage & others

MACHINE DESIGNED TRADING ALGORITHMS

What would you do with a Million Equity Streams?

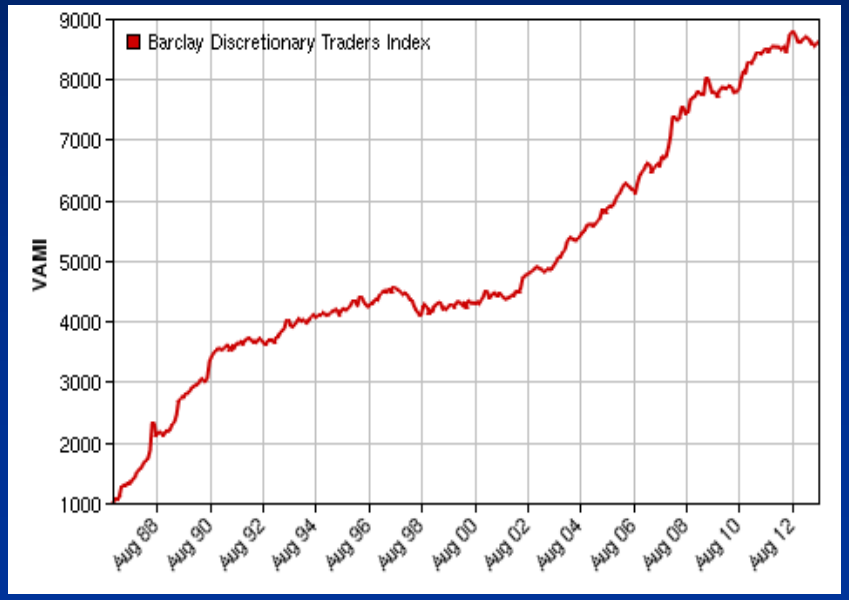
SYSTEMATIC VERSES DISCRETIONARY CTA VAMI

1987-2013

MECHANICAL SYSTEMS

466 programs, $260B

DISCRETIONARY



151 programs, $24B

Notice the 2010 to 2013 Drawdown

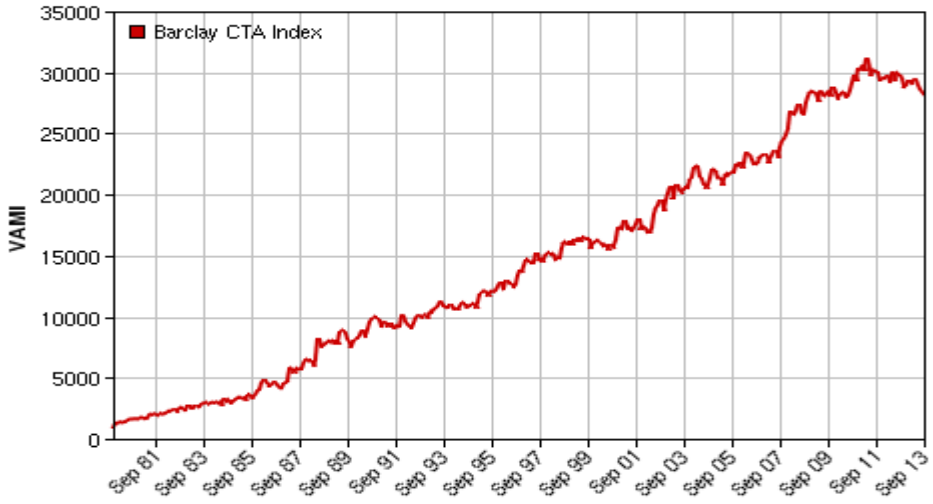
Source: BarclayHedge

Trading System Lab®

CTA PERFORMANCE IS DECAYING

1980	63.69%	**1992**	-0.91%	**2004**	3.30%
1981	23.90%	**1993**	10.37%	**2005**	1.71%
1982	16.68%	**1994**	-0.65%	**2006**	3.54%
1983	23.75%	**1995**	13.64%	**2007**	7.64%
1984	8.74%	**1996**	9.12%	**2008**	14.09%
1985	25.50%	**1997**	10.89%	**2009**	-0.10%
1986	3.82%	**1998**	7.01%	**2010**	7.05%
1987	57.27%	**1999**	-1.19%	**2011**	-3.09%
1988	21.76%	**2000**	7.86%	**2012**	-1.70%
1989	1.80%	**2001**	0.84%	**2013**	-3.04%†
1990	21.02%	**2002**	12.36%		
1991	3.73%	**2003**	8.69%		

†Estimated YTD performance for 2013 calculated with reported data as of October-7-2013 22:42 US CST



WHY?

Trading System Lab®

IS ALPHA GETTING MORE DIFFICULT TO GRAB?

- Getco Profits slide 81.8% in 2012
- HFT firm AienTech shuts down in September 2013
- CTA's report difficult returns
- Strategy tracking sites show performance decay

Alpha is much harder to capture than in past years. What is needed are deeper mining algorithms that design systems for us with minimal human workload.

Manual design techniques have been used for 40 years and now are just too inefficient. How much longer are we going to use manual design techniques?

ARE QUANT FUNDS IN TROUBLE?

The International Association of Financial Engineers is pleased to endorse

SQA Presents: The Near-Death Experience of Quant Asset Management

October 17, 2013
5:30-7:30pm
Jefferies
10th Floor Meeting Room
520 Madison Avenue
(between 53rd & 54th)
New York, NY 10022

AUM of "quant" equity managers remains substantially reduced from pre-crisis levels. The use of the term "quant" has been struck from the marketing materials of most asset managers and there has been a widespread loss of faith in quant investment methods. However, studies of actual performance of quant managers find the negative connotation of "quant" to be unwarranted.

AND BUY AND HOPE IS DEAD

$INDU 1993 to 2013

Trading System Lab®

SO WHY ARE WE STILL MANUALLY DESIGNING TRADING SYSTEMS?

- Software crisis-s/w costs are >10 times hardware costs
- 99% of CPU cycles are not used
- Lack of Industry Focus
- Past AI in Finance Failures
- Lack of confidence or understanding: fear?
- Note: Very few Futures Truth Trading Systems using AI

http://www.tradingsystemlab.com/files/CISC%20Architectures.pdf

WHAT IS A TRADING SYSTEM EQUITY CURVE?

Trading System Lab®

WHICH SYSTEM SHOULD YOU TRADE?

- Trader 1:

  35% accurate

  Average Win is 180% of Average Loss

- Trader 2:

  90% accurate

  Average Win is 10% of Average Loss

WHAT EQUATION PRODUCES ANY TRADING SYSTEM'S EXPECTATION?

$$EV = PW*AW - PL*AL$$

EV = expected value or average trade
PW = probability of a win
PL = probability of a loss
AW = amount won in winning trades
AL = amount lost in losing trades

FOR OUR 2 TRADERS:

Trader 1: EV = -.02

Trader 2: EV = -.01

So, neither Trader has a positive expectation

Note:

A system that is 90% accurate can have a negative EV

A system that is 30% accurate can have a positive EV

TOP COMPLAINTS OF TRADING STRATEGY USERS

- “I want systems but I can’t write code”
- “Easy Language is not that easy”
- “I need someone to write a strategy”
- “I wish strategies were more available”
- “I can’t HFT with these high commissions”

WHAT DOES TSL DO?

CLICK TO DESIGN TRADING SYSTEM

TSL Automatically designs Trading Algorithms and Produces Easy Language Code

Trading System Lab®

HOW GOOD ARE THE TSL MACHINE DESIGNED STRATEGIES?

In 2008, and again in 2010, TSL submitted several “Machine Designed” Strategies to Futures Truth after running TSL. These strategies were initially held for over 18 months, then tested, compared and ranked against hundreds of submissions from worldwide “human” strategy designers. These systems have not been touched since and reporting on these “Machine Designed” Strategies continues.

Not just an academic exercise. These systems have to perform or traders don’t make money or managers don’t get paid!

The results?

THIRD PARTY FORWARD TESTS

3 Strategies released in 2007 and 3 in 2010, Designs Frozen, Post OOS, Independent Third Party Comparative Ratings published April 2013

Top 10 S&P Systems

Issue #1 2013 - published in April 2013

Ranking based on performance since their release to us.
Some systems have been out for a short period of time.
Results based on performance through January 31, 2013.
Return is based on three times the required margin.

Rank	System Name	Annual % Return
1.	TSL-SP_1.0Z	63.8%
2.	TSL-CEL_SP1	50.9%
3.	Impetus SP	48.6%
4.	Big Blue 2	36.2%
5.	%C Daybreaker	35.2%
6.		
7.		
8.		
9.		
10.		

Top 10 Systems For The Past 12 Months

Issue #1 2013 - published in April 2013

Results based on performance through January 31, 2013.
Return is based on three times the required margin.

Rank	System Name	Annual % Return
1	TSL_US1	170.5%
2	Ruggiero Bond	140.3%
3	ATS-3200	92.3%
4	MAR - SP Mini Sync	87.8%
5	Star ES	87.6%

Top 10 Systems Since Their Release Date

Issue #1 2013 - published in April 2013

Systems included in this table must have been released for at least 18 months. Results based on performance through January 31, 2013.
Return is based on three times the required margin.

Rank	System Name	Annual % Return
1.	TSL_CEL_NG_1.1	142.0%
2.	Delphi II EMD	76.9%
3.	MAR - NewBondLiveSync	73.5%
4.	TSL_US1	71.5%
5.	TSL-SP_1.0Z	63.8%
6.	MAR - SP Mini Sync	59.9%
7.	NatGator Silver	53.4%
8.	TSL-CEL_SP1	53.4%
9.	Propero ES	49.0%
10.	Impetus SP	48.6%

#1 Bonds
#9 Past 12 mo
Issue #3 2013

ROI based on 3*Margin and includes Slippage and Commission
Forward tested over the worst financial disaster of our lives.

Trading System Lab®

WHAT IS THE SIGNIFICANCE OF THE FT REPORTS?

- Top performing strategies produce business for the developers
- Cannot be the result of just chance. Too many systems and trades
- Shows TSL machine can design over numerous markets
- The results quiet those that think machines can't out-design humans
- Shows TSL #1 in blind testing over the **most volatile and difficult period in our history**
- Asserts that properly machine designed strategies not as brittle or over-fit as originally thought

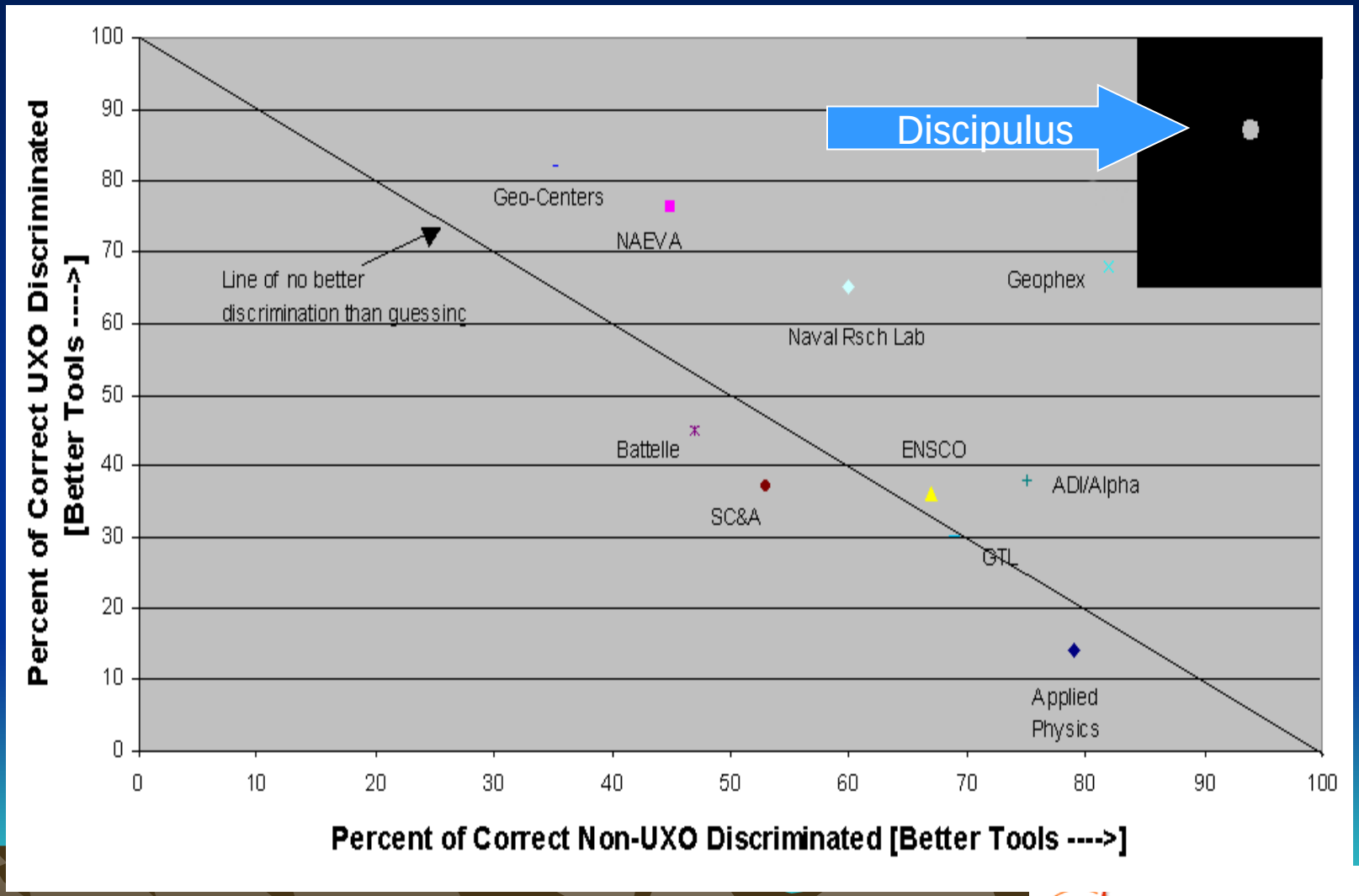
DISTRIBUTION ANALYSIS
WILCOXON TEST

Mann-Whitney

	t test	
1	Table Analyzed	Paired t test data
2	Column A	AVGRET/SDRET
3	vs	vs
4	Column B	AVGRET/SDRET
5		
6	Wilcoxon matched-pairs signed rank test	
7	P value	< 0.0001
8	Exact or approximate P value?	Gaussian Approximation
9	P value summary	****
10	Are medians signif. different? (P < 0.05)	Yes
11	One- or two-tailed P value?	Two-tailed
12	Sum of positive, negative ranks	9.596e+007 , -1.564e+008
13	Sum of signed ranks (W)	-6.043e+007
14		
15	How effective was the pairing?	
16	rs (Spearman, Approximation)	0.0370
17	P Value (one tailed)	< 0.0001
18	P value summary	****
19	Was the pairing significantly effective?	Yes

59,055 systems. Serial duplicates eliminated:
22,465 systems
SPY Long Only.
Distribution assumed Non-Gaussian.
WW Daniel, Applied Nonparametric Statistics

RML'S DISCIPULUS OUTPERFORMED EXISTING PUBLISHED RESULTS
UXO Discrimination Tests



Trading System Lab®

TSL Clients and Markets

- Major Wall Street Investment bank >$100M
- Small and mid size funds $10M-$100M
- Individual traders
- International Traders

- Note: TSL has not been widely distributed.

WHAT DID IT TAKE TO CREATE TSL?

- The work by two companies over many years
- Multiple U.S. patents
- Approximately one million lines of code
- World class scientists and programmers
- A lot of $
- Many obstacles along the way

WHAT IS TSL?

- TSL is a platform for the automatic design of Trading Strategies
- Linear Automatic Induction of Machine Code with Genetic Programming (LAIMGP) (algorithms are nonlinear)
- Trading Strategies are designed for you-NO PROGRAMMING!
- Code is written for you in different languages
- Strategies are tested OOS during design
- TSL is fast, very fast-can't do this work with normal software
- Patented and Trademarked
- Single Market Systems and Daytrading
- Pairs
- Portfolios
- Options

WHAT IS OUT OF SAMPLE (OOS) TESTING?

- Blind testing on data not used in design
- Hidden data
- Walk forward/backward data

ROBUSTNESS

(Over Fit Avoidance)

- Blind OOS Testing (walk forward)
- Run Path Logs (path intelligence)
- Unbiased Terminal Set (directionless inputs)
- Multi-Run, Randomized Criteria (global optimum)
- Zero Point Origin (no predefined initial point)
- Parsimony Pressure (Occam’s razor)
- Stat Tests-Distribution is exported
- TTPR

THIS IS OUR TRADING SYSTEM WHEN WE BEGIN A RUN IN TSL

(null starting point)

Trading System Lab®

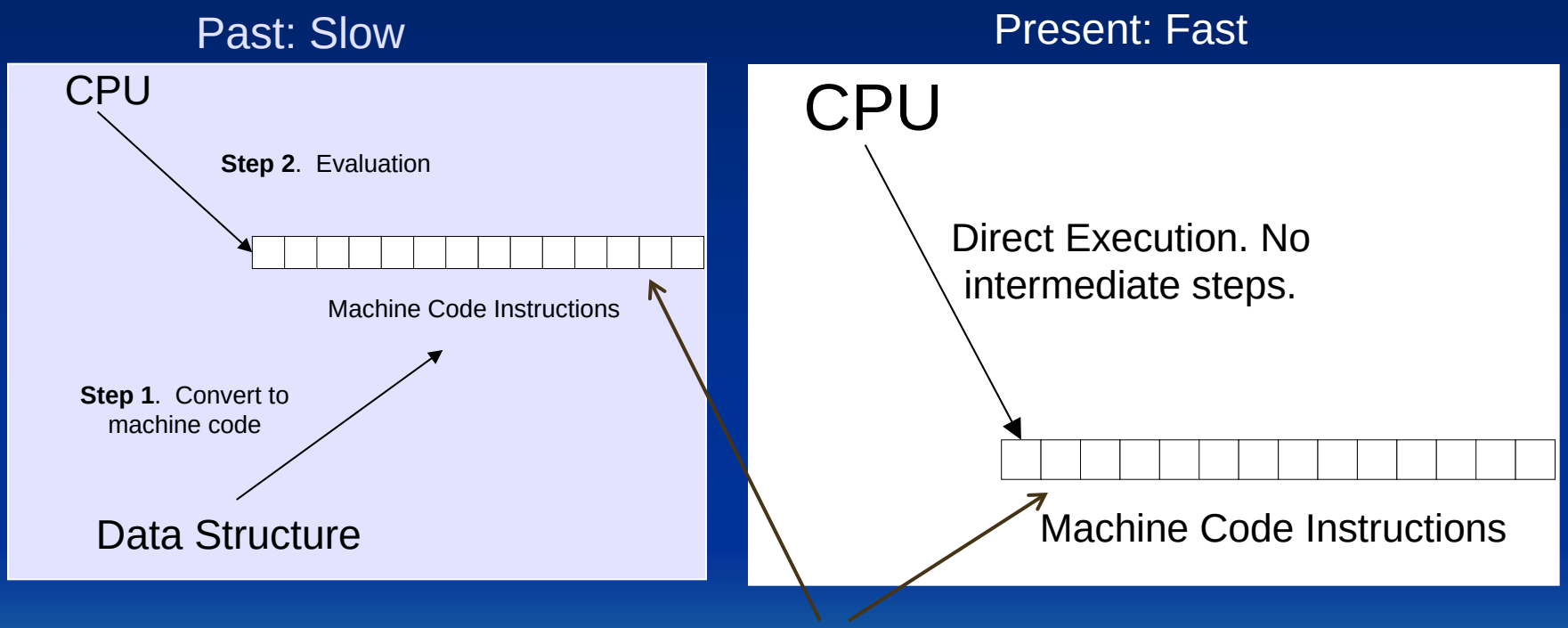
THIS IS OUR TRADING SYSTEM CORE LOGIC CODE AFTER A RUN HAS COMPLETED

MACHINE CODE TRANSLATED TO C, EL, JAVA, ETC.

```
long double f[8];
long double tmp = 0;
int cflag = 0;

f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;

L0:                     f[0]-=v[25];
L1:                     f[0]+=v[43];
L2:                     f[0]=fabs(f[0]);
L3:                     f[0]-=v[13];
L4:                     f[0]-=v[49];
L5:                     f[0]-=v[41];
L6:                     f[0]*=f[0];
L7:                     f[1]-=f[0];
L8:                     f[0]+=v[22];
L9:                     tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L10:                    cflag=(f[0] < f[2]);
L11:                    f[0]-=v[39];

if (!_finite(f[0])) f[0]=0;

return f[0];
```

Easy Language or many other platforms

INDUCTION OF MACHINE CODE



TRADING ALGORITHM CODE

Evolved Programs in AIMGP

To evaluate a evolved program:

Typcast the pointer to the program as a function pointer

To transform an evolved program:

Typcast the pointer as a pointer to an array of integers

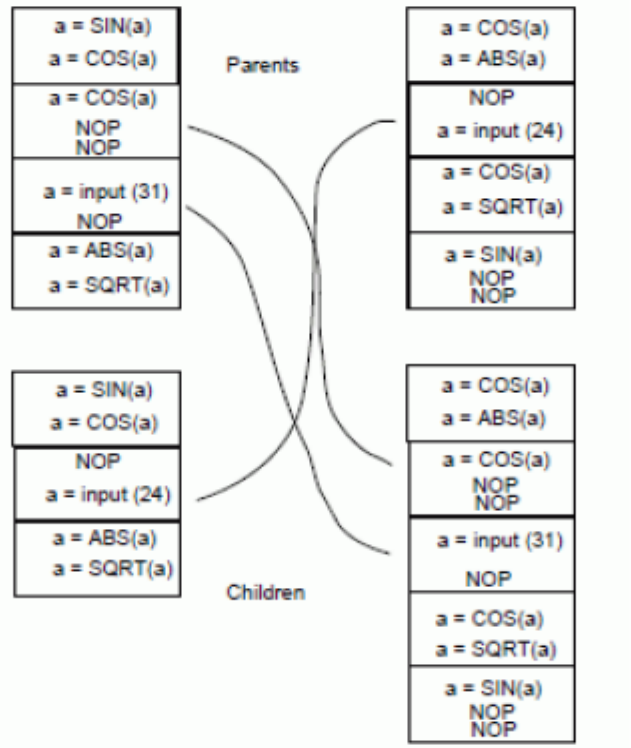
Machine code transforms the state of the CPU registers

Problem: How do you perform crossover and mutation on binary machine code?

LAIMGP REPRODUCTIVE CROSSOVER

Homologous and Non-Homologous

Trading Algorithm->



Reference: Frank D. Francone Licensiate Thesis (2009)

BIG PICTURE QUESTION:

How many CPU cycles does it take to compute the following?

$$X = Y + Z$$

BIG PICTURE QUESTION:

How many CPU cycles does it take to compute the following?

X= Y + Z

High level languages: 20 Clock cycles
CGPS(LAIMGP): 1 Clock cycle

So, TSL should be at least 20 times
faster than higher level languages
“on a single thread”

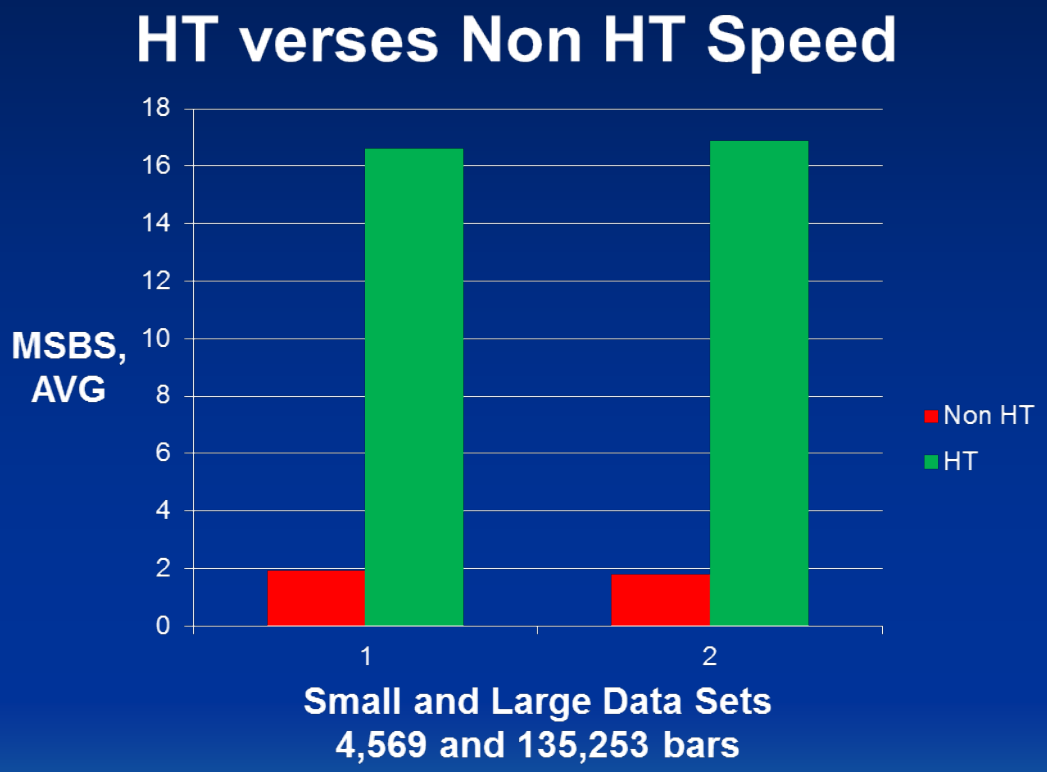
But What about Hyper-Threading?

Reference: Efficient Evolution of Machine Code for CISC Architectures using blocks and Homologous Crossover,
Peter Nordin, Wolfgang Banzhaf, Frank Francone

HYPERTHREADED TSL

Million System-Bars per Second (**900 M Input SBS)**

core i7 990x, 3.47ghz



Trading System Lab®

GENETIC PROGRAMMING

- Based loosely on biological models of evolution and eucaryotic* sexual reproduction
- Simulates the path a biological species goes through as it evolves:

  -Starts off simple

  -Adapts to hostile environment

  -Strong Parents give birth to strong children

  -Random mutations may help

*Based on complex cells with membranes

TSL GP LEARNING

- Supervised Learning. No supervisory Signal.
- Population is initialized
- A Trading Strategy is initialized with random signals and a error produced
- Tournament is run within population
- Mutation causes random changes in winners
- Crossover exchanges DNA between winners
- Reproduction is applied on remainder
- Demes enhance genetic diversity
- Parsimony Pressure favor simpler solutions
- If n GWI occur then run restarts
- New trading algorithms emerge and improve based on the error
- Algorithms learn to trade better as they trade in simulation
- After x runs or user termination, all runs stop
- Finally, code is exported, translated and ported to a Trading OMS/EMS

TSL ACCEPTS 56 INPUTS

THEN REDUCES DOWN TO JUST A FEW

Example Boolean Indicators:

```
if h>=highest(h, dc)[1] and hhit=0 then begin
      hhit=1;
      lhit=0;
end;
if l<=lowest(l, dc)[1] and lhit=0 then begin
      hhit=0;
      lhit=1;
end;
```

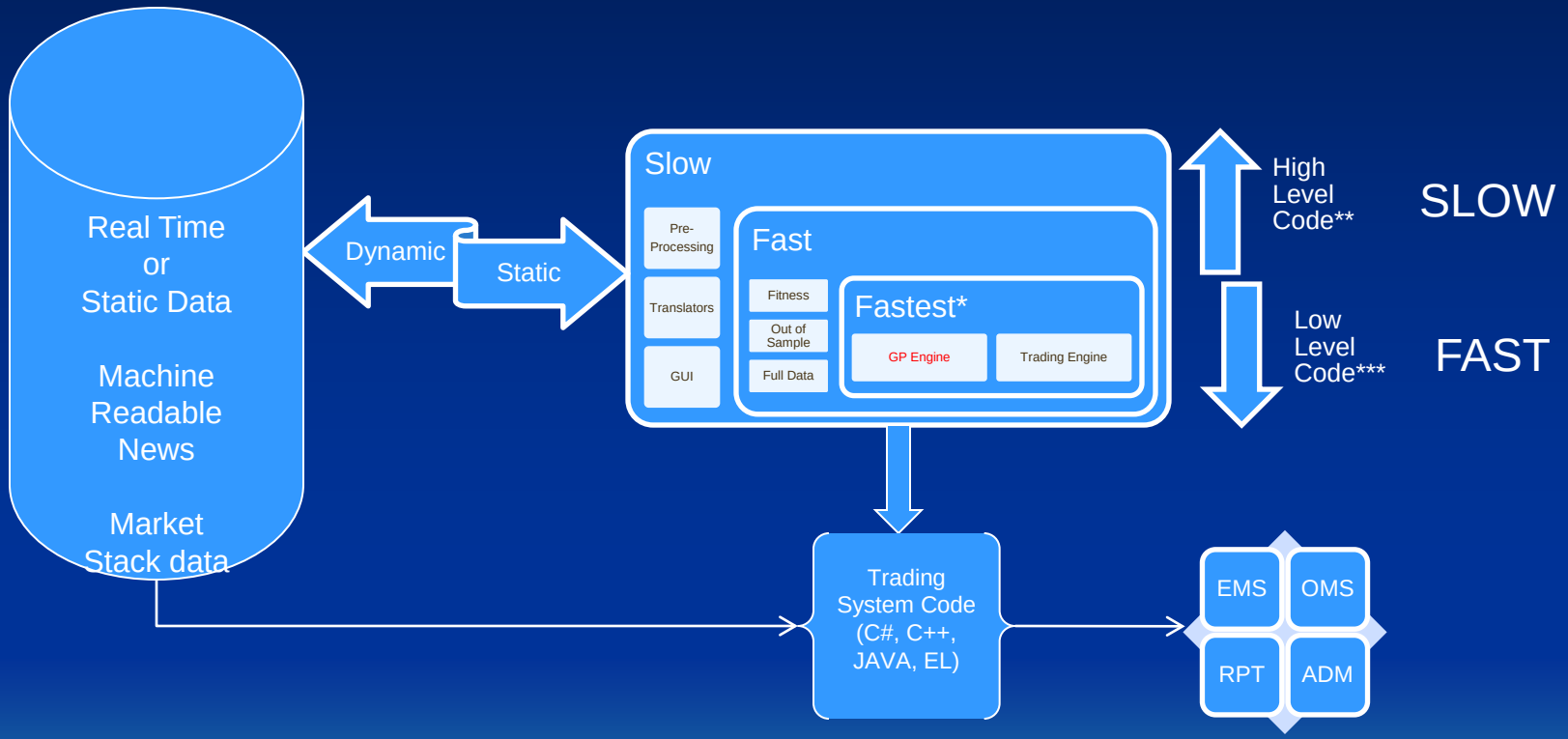
Example Boolean Patterns:

```
cond[45] = c>=c[1];
cond[46] = c>=c[2];
cond[47] = o>=c[1];
cond[48] = o>=c[2];
```

Consider: Intermarket data, Fundamental data, COT, Machine readable news, Twitter etc.

THE ARCHITECTURE

Development followed no specific model since no specific model (~extreme programming) was readily available
~1 million lines of code, 9 languages, 2 companies



- *GPU/G80, EP or Many Core Implementations
- **VC++.NET, VB, C#, EL
- ***Assembler, C, C++

Trading System Lab®

FITNESS CAN BE MULTI GOAL

Machine Design Allows Us to Adjust Critical System Metrics as Targeted Fitness Function

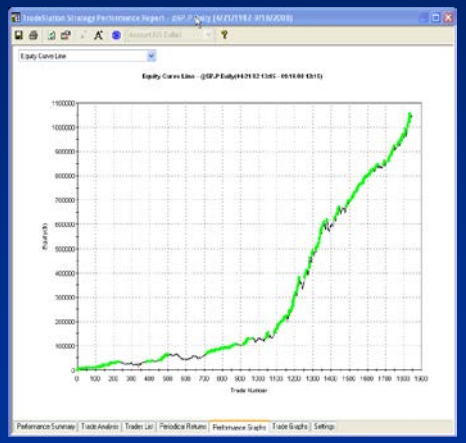
Net Profit

Drawdown

Percent Accuracy

Profit Factor

Average Trade

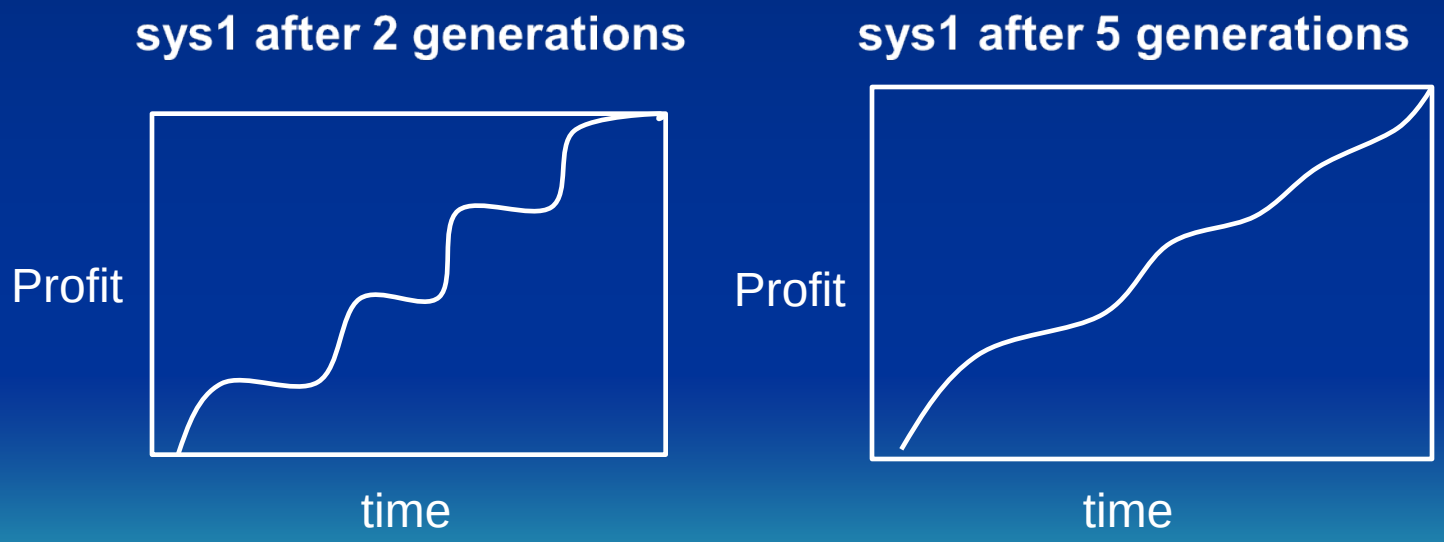


PRODUCES CODE

Trading System Lab®

FITNESS AS A SIMPLE EXPRESSION

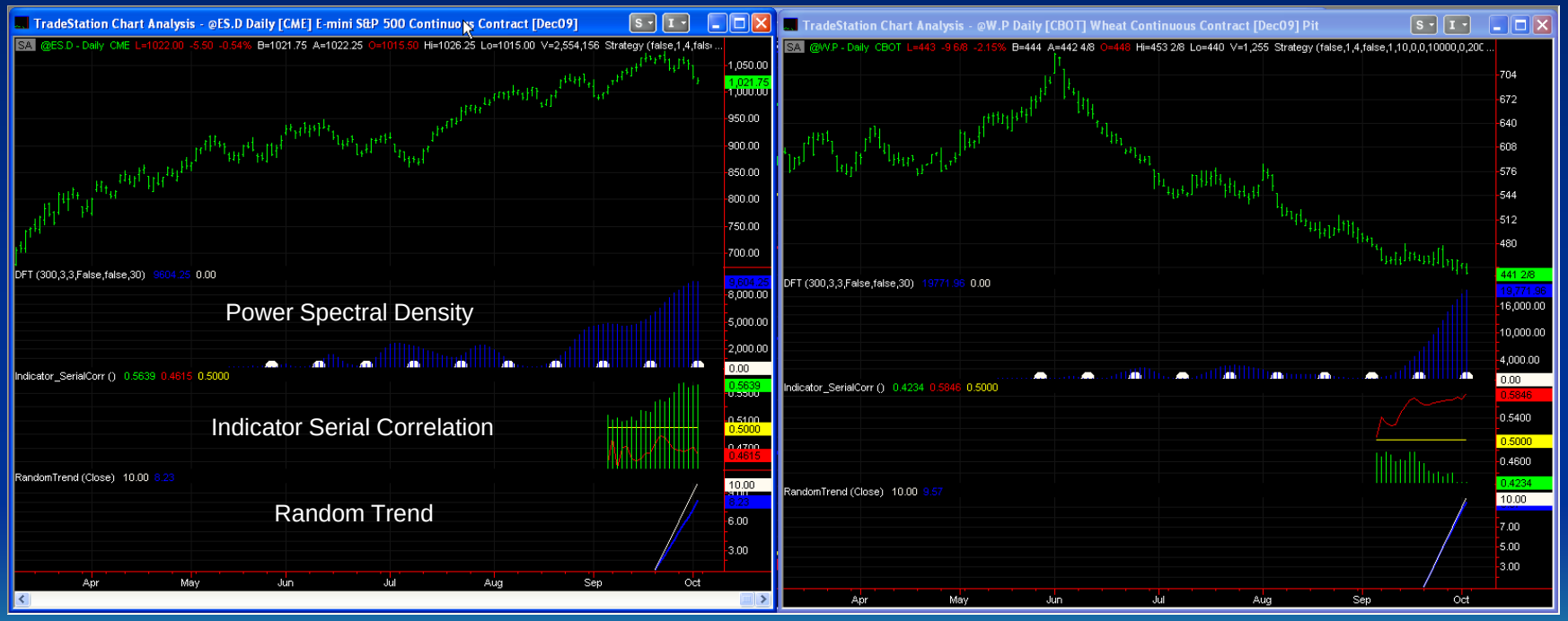
$$error = \frac{1}{NETPROFIT}$$



Notice drawdown improved even though DD was not part of fitness

MARKETS HAVE DIFFERENT DESCRIPTIVE STATISTICS

So Why Design Symmetrical Systems?



CME:E-MINI S&P

CBOT:WHEAT

Trading System Lab®

TACTICS, FITNESS AND PREPROCESSING

- (TT) Trade Types/Trading Tactics are entry techniques. Example: Enter on Limit
- (FF) Fitness Functions are “Targets” that TSL attempts to design to.

  Example: NetProfit/Max Drawdown
- (PP) Preprocessors are Patterns, Indicators or other facts used as DNA in TSL. Example: Close>Close[1]
- There are 25 TT’s, 40 FF’s and 10 PP’s allowing 10,000 possible setups to be tested.
- Testing over various bar sizes requires yet more runs
- Clearly there is a need for simplicity and runs reduction

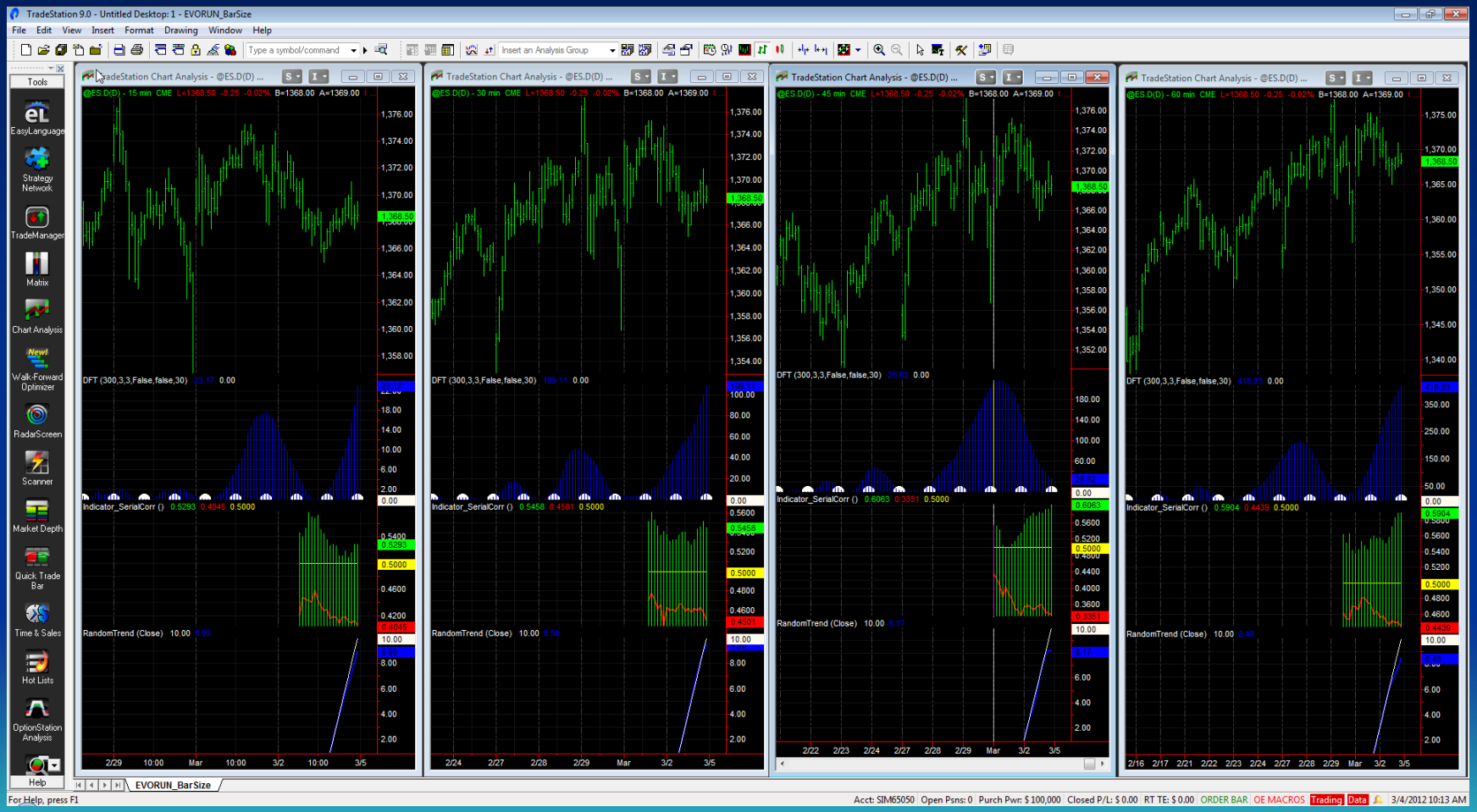
WHAT IS EVORUN™?

EVORUN is a multi run iterator:

1. Trade Type
2. Fitness Function
3. Preprocessor
4. Bar Size

EVORUN™

WHAT IS YOUR OPTIMUM BAR SIZE?



11/10/2013

Trading System Lab®

TSL DAYTRADING SYSTEMS

Low Frequency Daytrading

- Use TSL and design on lots of data
- Blend in additional filters if necessary
- Use TSL System Stats to filter out poor:

1. Months
2. Day of week
3. Time of day
4. Day of Week in Month
5. Day of Month

TRADING SYSTEMS DESIGNED IN 3 EASY STEPS

1. Preprocess (TradeStation EL)
2. Evolve (TSL)
3. Translate (TradeStation OMS/EMS)

TEXT BOOKS

- Francone-Genetic Programming
- Brameier/Banzhaf-Linear Genetic Programming
- Poli-A Field Guide to Genetic Programming
- Chen-Genetic Algorithms and Genetic Programming in Computational Finance
- Aldridge-High Frequency Trading

CONCLUSION

MACHINE DESIGNED TRADING SYSTEMS ARE HERE!

www.tradingsystemlab.com
408-356-1800

Trading System Lab®