Abstract

This dissertation demonstrates that there is high revenue potential in using limit order book imbalance as a state variable in an algorithmic trading strategy. Beginning with the hypothesis that imbalance of bid/ask order volumes is an indicator for future price changes, exploratory data analysis suggests that modelling the joint distribution of imbalance and observed price changes as a continuous-time Markov chain presents a monetizable opportunity. The arbitrage problem is then formalized mathematically as a stochastic optimal control problem using limit orders and market orders with the aim of maximizing terminal wealth. The problem is solved in both continuous and discrete time using the dynamic programming principle, which produces both conditions for market order execution, as well as limit order posting depths, as functions of time, inventory, and imbalance. The optimal controls are calibrated and backtested on historical NASDAQ ITCH data, which produces consistent and substantial revenue.

Statistical Arbitrage Using Limit-Order Book Imbalance

Anton D. Rubisov

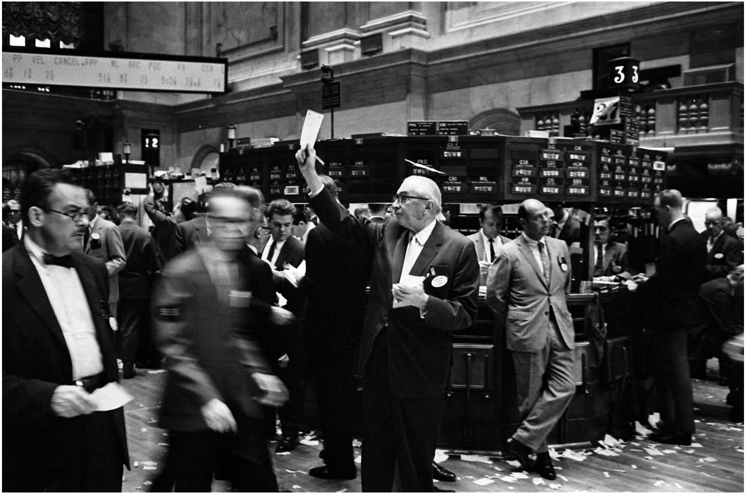
University of Toronto Institute for Aerospace Studies
Faculty of Applied Science and Engineering
University of Toronto

17 September 2015

Roadmap

Roadmap

NYSE Circa 1968



This image is available from the United States Library of Congress's Prints and Photographs division under the digital ID ppmsca.03199

NASDAQ Circa Now

Copyright 2000, The Nasdaq Stock Market, Inc. Reprinted with the permission of The Nasdaq Stock Market, Inc. Photo credit: Peter Aaron/Esto

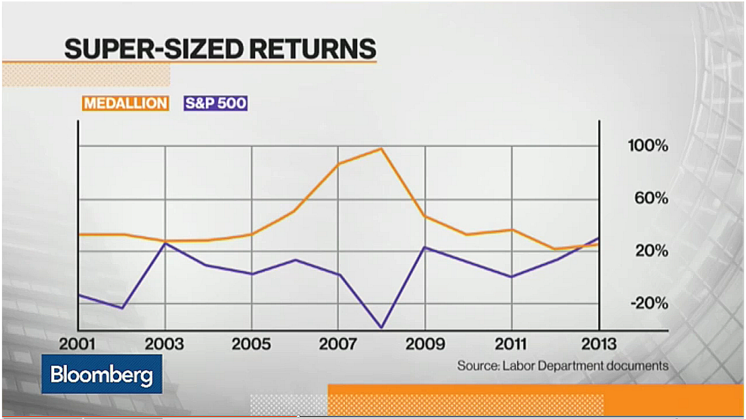
High-Frequency and Algorithmic Trading

Hallmarks of HF and Algo trading: algorithms, models, execution speed, timescale.

Benefits:

- reduction of human error
- closes arbitration holes, producing fair markets
- digest large sets of data

Why Do We Care?



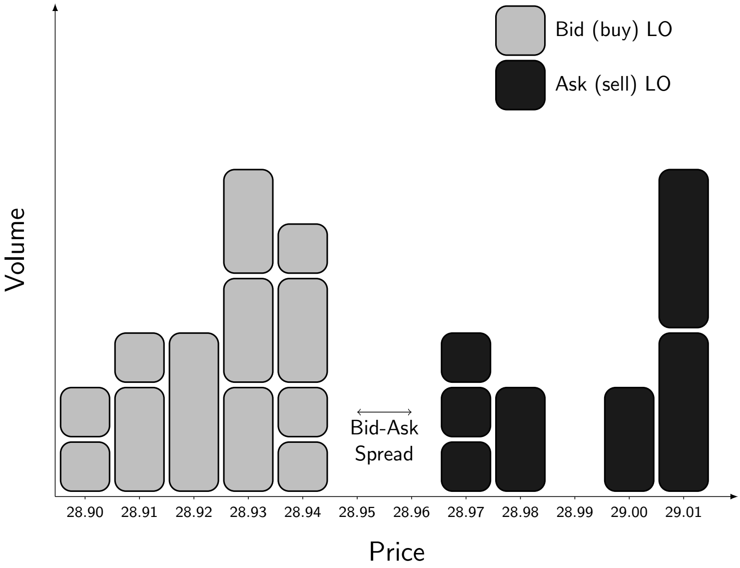
Rubin, R. and Collins, M. (2015). How an exclusive hedge fund turbocharged its retirement plan. *Bloomberg Business.*

Market Data Feeds

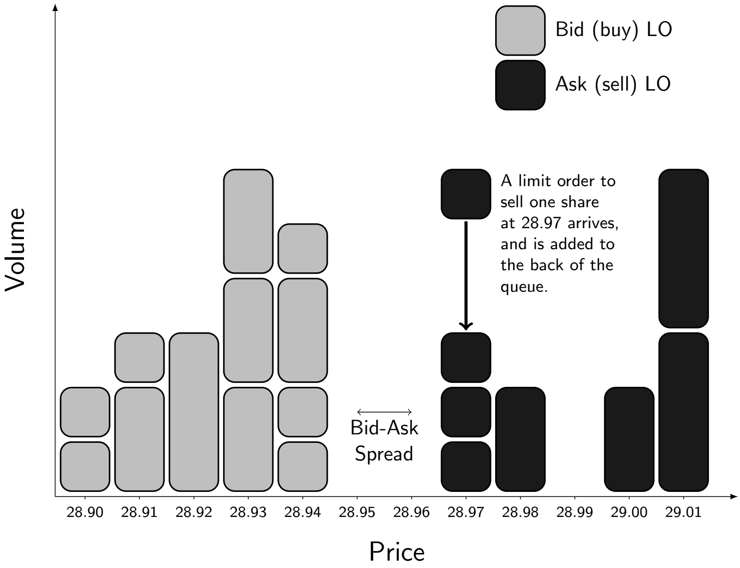
From the NASDAQ, we can subscribe to real-time data feeds showing order arrivals.

Time	Order ID	Event	Volume	Price
⋮	⋮	⋮	⋮	⋮
39960699	72408630	66	100	1107000
39960710	72408630	68	100	1107000
⋮	⋮	⋮	⋮	⋮

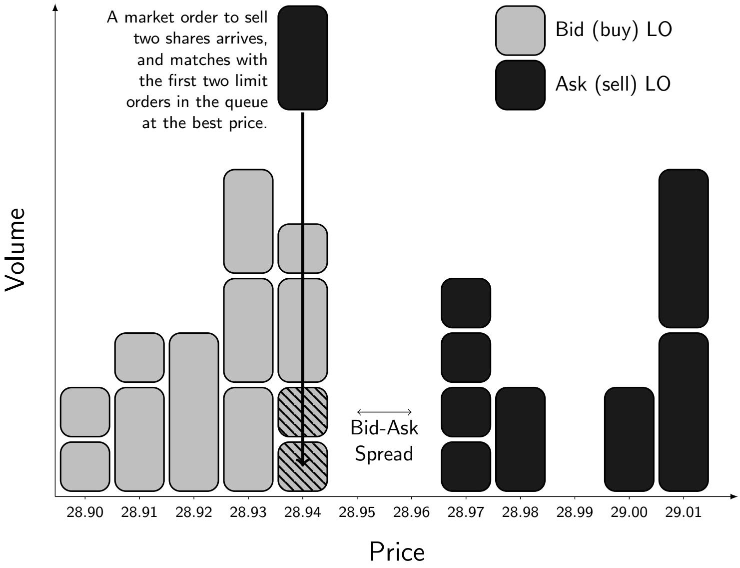
Limit-Order Book



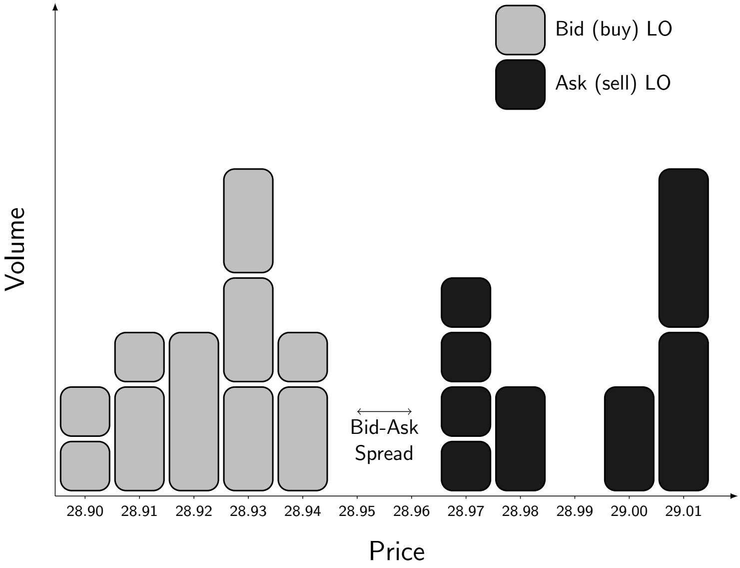
Limit-Order Book



Limit-Order Book



Limit-Order Book



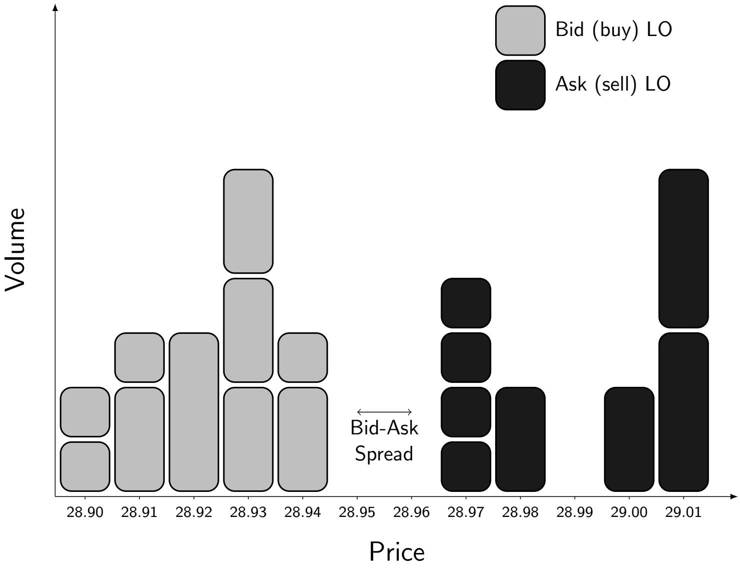
Roadmap

Limit-Order Book Imbalance

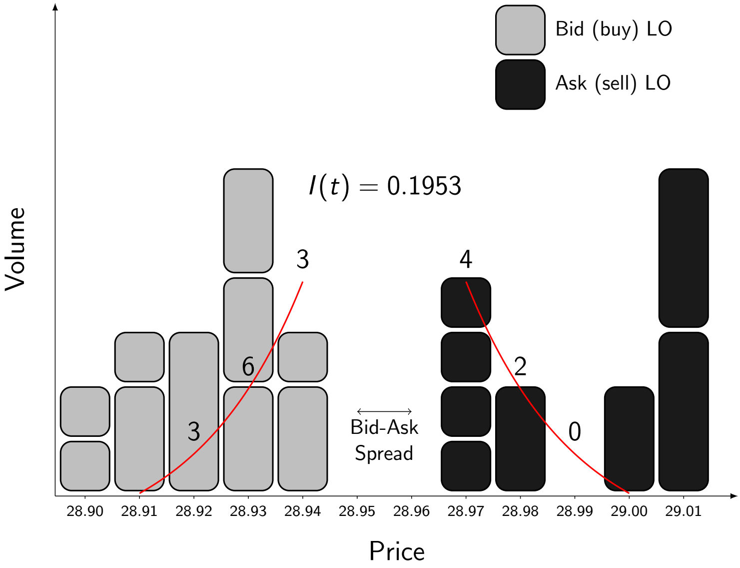
Imbalance is a ratio of quoted limit order volumes between the bid and ask side.

$$I(t) = \frac{V_{bid}(t) - V_{ask}(t)}{V_{bid}(t) + V_{ask}(t)} \in [-1, 1]$$

Limit-Order Book Imbalance

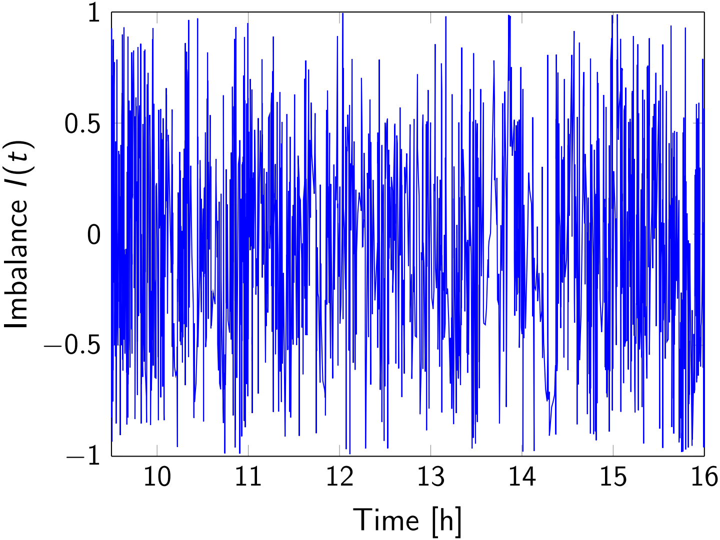


Limit-Order Book Imbalance



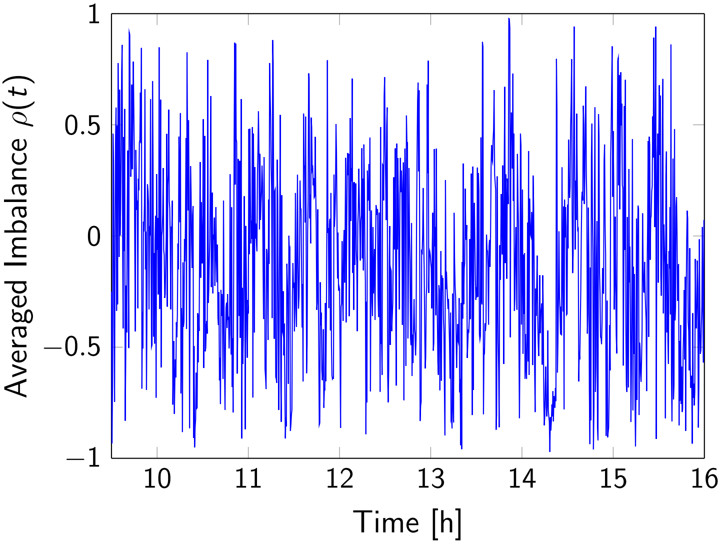
Limit-Order Book Imbalance

Resulting timeseries is noisy.



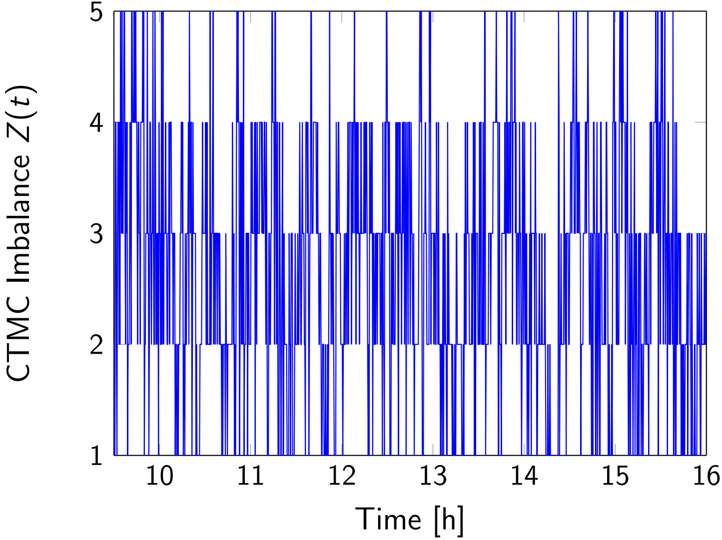
Limit-Order Book Imbalance

Smooth it by averaging on a sliding window (1 s).



Limit-Order Book Imbalance

Model as a continuous-time Markov chain $Z(t)$ with generator G.



Limit-Order Book Imbalance

Model as a continuous-time Markov chain $Z(t)$ with generator G.

$$Z = \begin{cases} 5, & \rho \in [+\frac{3}{5}, +1], \quad \text{buy-heavy} \\ 4, & \rho \in [+\frac{1}{5}, +\frac{3}{5}], \quad \text{buy-biased} \\ 3, & \rho \in [-\frac{1}{5}, +\frac{1}{5}), \quad \text{neutral} \\ 2, & \rho \in [-\frac{3}{5}, -\frac{1}{5}), \quad \text{sell-biased} \\ 1, & \rho \in [-1, -\frac{3}{5}), \quad \text{sell-heavy} \end{cases}$$

Incorporating Price Change

Next, consider a two-dimensional CTMC $Z(t)$ that jointly models imbalance bin $\rho(t)$ and price change $\Delta S(t)$, where

$$\rho(t) \in \{1, 2, \dots, \#_{bins}\}$$

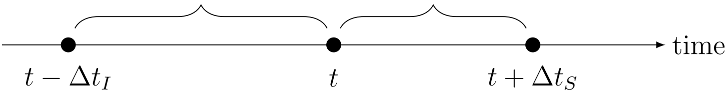
is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and

$$\Delta S(t) = \mathrm{sgn}(S(t + \Delta t_S) - S(t)) \in \{-1, 0, 1\}$$

is the *sign* of the change in midprice of the *future* time interval Δt_S.

$\rho(t)$ is the imbalance bin of the time-weighted average of $I(t)$ over this past interval.

$\Delta S(t)$ is the sign of the mid-price change over this future interval.



Transition Matrix

Using MLE, we obtain a generator matrix $\boldsymbol{G}$ for the CTMC. The transition matrix over a step of size Δt_I is given by

$$\boldsymbol{P}(\Delta t_I) = [p_{ij}(\Delta t_I)] = e^{\boldsymbol{G}\Delta t_I}$$

called our *one-step transition probability matrix*. Matrix entries give the probability of transition from one (imbalance, price change) pair to another over the time interval Δt_I. This can be written semantically as

$$p_{ij} = \mathbb{P}\left[\varphi(\rho_{\text{curr}}, \Delta S_{\text{future}}) = j \mid \varphi(\rho_{\text{prev}}, \Delta S_{\text{curr}}) = i\right]$$

Predicting Future Price Change

Using Bayes' Rule, we can transform the $\boldsymbol{P}$ matrix to

$$\mathbb{P}\left[\Delta S_{\text{future}} = j \mid \substack{\rho_{\text{curr}}=i \\ \rho_{\text{prev}}=k \\ \Delta S_{\text{curr}}=m}\right] = \frac{\mathbb{P}\left[\rho_{\text{curr}} = i, \Delta S_{\text{future}} = j \mid \substack{\rho_{\text{prev}}=k \\ \Delta S_{\text{curr}}=m}\right]}{\mathbb{P}\left[\rho_{\text{curr}} = i \mid \substack{\rho_{\text{prev}}=k \\ \Delta S_{\text{curr}}=m}\right]}$$

This allows us to predict future price moves. We'll call the collection of these probabilities the $\boldsymbol{Q}$ matrix.

Predicting Future Price Change

Sample $\boldsymbol{Q}$ matrix.

	$\Delta S_{\text{curr}} < 0$			$\Delta S_{\text{curr}} = 0$			$\Delta S_{\text{curr}} > 0$		
	$\rho_{curr} = 1$	2	3	1	2	3	1	2	3
$\Delta S_{\text{future}} < 0$									
$\rho_{\text{prev}} = 1$	**0.53**	0.15	0.12	0.05	0.10	0.14	0.08	0.13	0.14
$\rho_{\text{prev}} = 2$	0.10	**0.58**	0.14	0.07	0.04	0.10	0.13	0.06	0.12
$\rho_{\text{prev}} = 3$	0.08	0.12	**0.52**	0.09	0.06	0.03	0.11	0.10	0.05
$\Delta S_{\text{future}} = 0$									
$\rho_{\text{prev}} = 1$	0.41	0.75	0.78	**0.91**	0.84	0.79	0.42	0.79	0.77
$\rho_{\text{prev}} = 2$	0.79	0.36	0.71	0.83	**0.92**	0.82	0.75	0.37	0.78
$\rho_{\text{prev}} = 3$	0.79	0.74	0.40	0.81	0.83	**0.91**	0.70	0.76	0.39
$\Delta S_{\text{future}} > 0$									
$\rho_{\text{prev}} = 1$	0.06	0.10	0.09	0.04	0.06	0.07	**0.50**	0.09	0.09
$\rho_{\text{prev}} = 2$	0.10	0.06	0.15	0.10	0.04	0.08	0.12	**0.57**	0.10
$\rho_{\text{prev}} = 3$	0.13	0.14	0.08	0.10	0.11	0.05	0.19	0.14	**0.56**

Roadmap

Background Information

Exploratory Data Analysis

Maximizing Wealth via Discrete-Time Stochastic Control

Backtesting Results

Conclusion and Future Work

System Description

$$\text{State } \vec{x}_k = \begin{pmatrix} x_k \\ s_k \\ \boldsymbol{z}_k \\ q_k \end{pmatrix} \quad \begin{matrix} \text{cash} \\ \text{stock price} \\ \text{Markov chain state, as above} \\ \text{inventory} \end{matrix}$$

$$\text{Control } \vec{u}_k = \begin{pmatrix} \delta_k^+ \\ \delta_k^- \\ M_k^+ \\ M_k^- \end{pmatrix} \quad \begin{matrix} \text{bid posting depth} \\ \text{ask posting depth} \\ \text{buy market order - binary control} \\ \text{sell market order - binary control} \end{matrix}$$

$$\text{Random } \vec{w}_k = \begin{pmatrix} K_k^+ \\ K_k^- \\ \omega_k \end{pmatrix} \quad \begin{matrix} \text{other agent buy market orders} \\ \text{other agent sell market orders} \\ \text{random variable uniformly distributed on [0,1]} \end{matrix}$$

System Description

Impulse Control

$$\begin{pmatrix} x_k \\ s_k \\ \boldsymbol{z}_k \\ q_k \end{pmatrix} = \begin{pmatrix} x_k \\ s_k \\ \boldsymbol{z}_k \\ q_k \end{pmatrix} + \begin{pmatrix} s_k - \xi \\ 0 \\ 0 \\ -1 \end{pmatrix} M_k^- + \begin{pmatrix} -(s_k + \xi) \\ 0 \\ 0 \\ 1 \end{pmatrix} M_k^+$$

System Evolution

$$\begin{pmatrix} x_{k+1} \\ s_{k+1} \\ \boldsymbol{z}_{k+1} \\ q_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ s_k + \eta_{k+1,T(\boldsymbol{z}_k,\omega_k)} \\ T(\boldsymbol{z}_k, \omega_k) \\ q_k \end{pmatrix} + \begin{pmatrix} s_k + \xi + \delta_k^- \\ 0 \\ 0 \\ -1 \end{pmatrix} L_k^- + \begin{pmatrix} -(s_k - \xi - \delta_k^+) \\ 0 \\ 0 \\ 1 \end{pmatrix} L_k^+$$

Fill Probability

Other agents' market orders are Poisson distributed, so

$$\mathbb{P}[K_k^+ = 0] = \frac{e^{-\mu^+(\boldsymbol{z})\Delta t}(\mu^+(\boldsymbol{z})\Delta t)^0}{0!} = e^{-\mu^+(\boldsymbol{z})\Delta t}$$

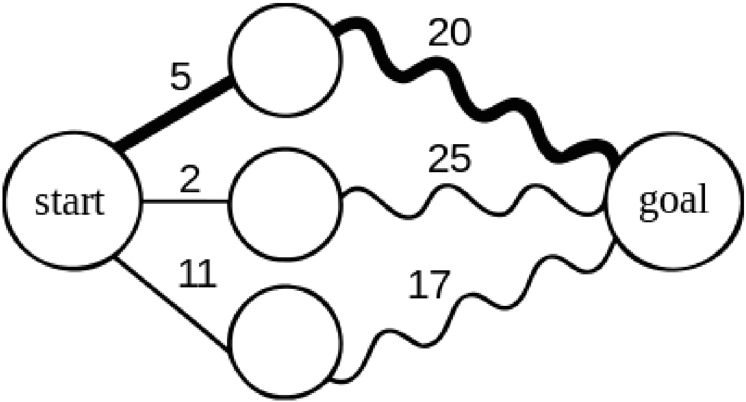
and

$$\mathbb{P}[K_k^+ > 0] = 1 - e^{-\mu^+(\boldsymbol{z})\Delta t}$$

- assume the *aggregate* of the orders walks the LOB to depth p_k
- if $p_k > \delta^-$, our sell limit order is lifted
- assume this occurs with probability $e^{-\kappa\delta^-}$.

$$\mathbb{E}[L_k^-] = (1 - e^{-\mu^+(\boldsymbol{z})\Delta t})e^{-\kappa\delta^-} = \underbrace{p(\delta^-)}_{\text{short-hand}}$$

Intro to Dynamic Programming



Made by Derrick Coetzee. Available under the Creative Commons CC0 1.0 Universal Public Domain Dedication.

Dynamic Programming Value Function

Our performance criterion is our *terminal wealth*:

$$\begin{aligned} V_k^{\delta^\pm}(x, s, \boldsymbol{z}, q) &= \mathbb{E}_{k,x,s,\boldsymbol{z},q}\left[W_T^{\delta^\pm}\right] \\ &= \mathbb{E}_{k,x,s,\boldsymbol{z},q}\Big[\underbrace{X_T^{\delta^\pm}}_{\text{cash}} + \underbrace{Q_T^{\delta^\pm}\left(S_T - \xi\,\mathrm{sgn}(Q_T^{\delta^\pm})\right)}_{\text{book value of assets}} - \underbrace{\alpha(Q_T^{\delta^\pm})^2}_{\text{penalty}}\Big] \end{aligned}$$

So that our dynamic programming equations are

$$V_T(x, s, \boldsymbol{z}, q) = x + q(s - \xi\,\mathrm{sgn}(q)) - \alpha q^2$$

$$V_k(x, s, \boldsymbol{z}, q) = \max\Big\{\sup_{\delta^\pm}\big\{\mathbb{E}_{\boldsymbol{w}}\left[V_{k+1}(f((x, s, \boldsymbol{z}, q), \boldsymbol{u}, \boldsymbol{w}_k))\right]\big\}\ ;\ V_k(x + s_k - \xi, s_k, \boldsymbol{z}_k, q_k - 1)\ ;\ V_k(x - s_k - \xi, s_k, \boldsymbol{z}_k, q_k + 1)\Big\}$$

Optimal Posting Depth

Solve one depth numerically (here the optimal sell depth):

$$\delta^{-*} = \max\Big\{0\ ;\ \frac{1}{\kappa} + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\geq 1} + \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}}\Big[h_{k+1}(\boldsymbol{j},q) - h_{k+1}(\boldsymbol{j},q-1)\Big]$$

$$- (1 - e^{\mu^-(\boldsymbol{z})\Delta t})e^{-\kappa \max\left\{0\ ;\ \frac{1}{\kappa} - \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\leq -1} + \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}}\left[h_{k+1}(\boldsymbol{j},q) - h_{k+1}(\boldsymbol{j},q+1)\right] - (1-e^{\mu^+(z)\Delta t})e^{-\kappa\delta^{-*}}(2\xi\mathbb{1}_{q=0} - \aleph(q))\right\}} (2\xi\mathbb{1}_{q=0} - \aleph(q))\Big\}$$

And substitute to solve for other depth:

$$\delta^{+*} = \max\Big\{0\ ;\ \frac{1}{\kappa} - \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\leq -1} + \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}}\left[h_{k+1}(\boldsymbol{j},q) - h_{k+1}(\boldsymbol{j},q+1)\right]$$

$$- p(\delta^-)\left(2\xi\mathbb{1}_{q=0} - \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}}\left[h_{k+1}(\boldsymbol{j},q-1) + h_{k+1}(\boldsymbol{j},q+1) - 2h_{k+1}(\boldsymbol{j},q)\right]\right)\Big\}$$

Simplified Dynamic Programming Equation

$$
\begin{aligned}
h_k(\boldsymbol{z}, q) = \max\Bigg\{ & q\mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] + \frac{1}{\kappa}(p(\delta^{+*}) + p(\delta^{-*})) \\
& + \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}} h_{k+1}(\boldsymbol{j}, q) \\
& + p(\delta^{+*})p(\delta^{-*}) \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}} \big[h_{k+1}(\boldsymbol{j}, q-1) \\
& \qquad + h_{k+1}(\boldsymbol{j}, q+1) \\
& \qquad - 2h_{k+1}(\boldsymbol{j}, q) \big] \; ; \\
& - 2\xi \cdot \mathbb{1}_{q \geq 0} + h_k(\boldsymbol{z}, q+1) \; ; \\
& - 2\xi \cdot \mathbb{1}_{q \leq 0} + h_k(\boldsymbol{z}, q-1) \Bigg\}
\end{aligned}
$$

- solve this numerically.

Roadmap

Data Set

Calibrate and backtest on the NASDAQ Historical TotalView-ITCH, timestamped to the millisecond

Ticker	Company	Average Daily Volume
FARO	FARO Technologies Inc.	200,000
NTAP	NetApp, Inc.	4,000,000
ORCL	Oracle Corporation	15,000,000
INTC	Intel Corporation	30,000,000
AAPL	Apple Inc.	50,000,000

Calibration

Global parameters for backtesting

Parameter	Value	Description
Δt_S	1000ms	time window for computing price change
Δt_I	1000ms	time window for averaging order imbalance
$\#_{bins}$	5	number of imbalance bins
κ	100	fill probability constant

$\kappa = 100$ implies:

- Orders posted at $\delta = 0$ filled with probability 1
- Orders posted at $\delta = \$0.01$ filled with probability 0.37
- Orders posted at $\delta = \$0.02$ filled with probability 0.13
- ...

Calibration

Calculated parameters for backtesting

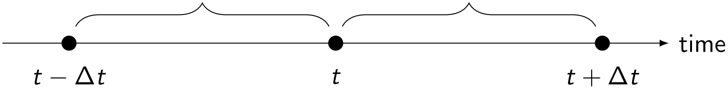
Parameter	Equation
$\boldsymbol{G}$	infinitesimal generator matrix
$\boldsymbol{P}$	transition probability matrix
$\mu^{\pm}$	market order arrival intensities
$H(t, x, s, \boldsymbol{z}, q)$	dynamic programming value function
$\delta^{\pm}$	limit order posting depths

Two Calibration Frameworks

Non-$\mathcal{F}$-predictable calibration

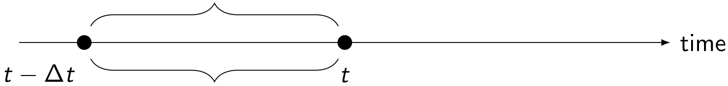
$\rho(t)$ is the imbalance bin of the time-weighted average of $I(t)$ over this past interval.

$\Delta S(t)$ is the sign of the midprice change over this future interval.



Regular calibration

$\rho(t)$ unchanged.

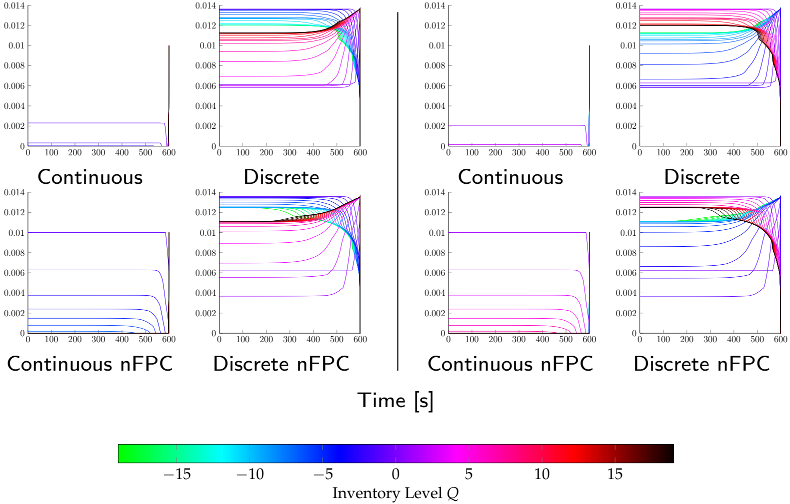


$\Delta S(t)$ calculated over the same past interval.

Dynamics of Posting Depths

BUY Posting Depth δ^+ [$]
at $Z = (\rho = -1, \Delta S = -1)$

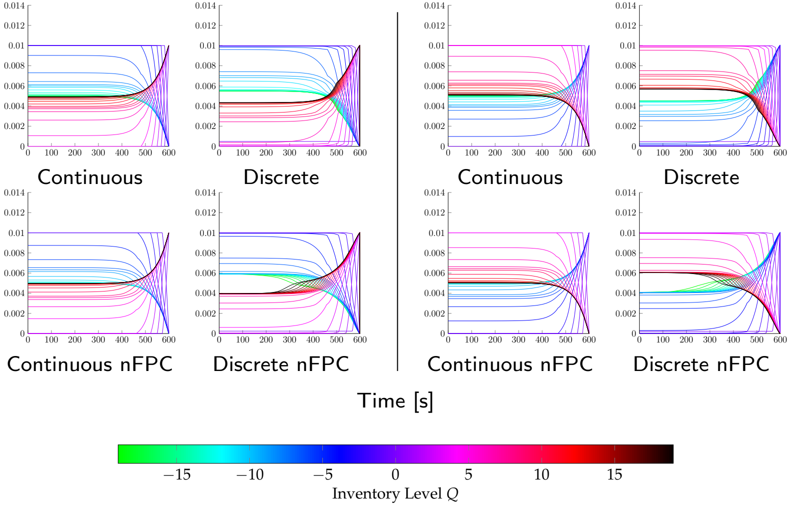
SELL Posting Depth δ^- [$]
at $Z = (\rho = +1, \Delta S = +1)$



Dynamics of Posting Depths

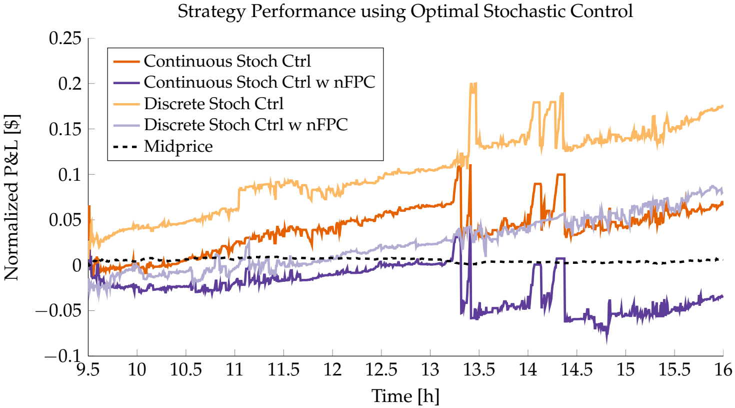
BUY Posting Depth δ^{+} [$]
at $Z = (\rho = 0, \Delta S = 0)$

SELL Posting Depth δ^{-} [$]
at $Z = (\rho = 0, \Delta S = 0)$



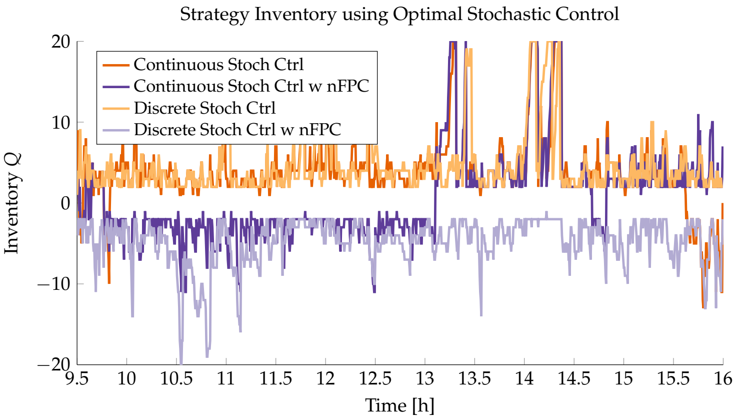
Sample Strategy Performance

Single day performance for `ORCL` on 2013-05-15



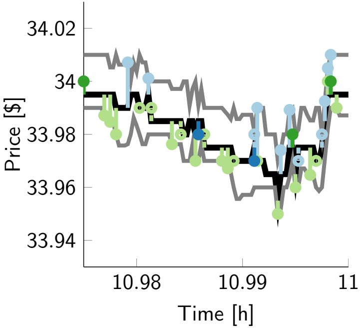
Sample Strategy Performance

Single day performance for `ORCL` on 2013-05-15

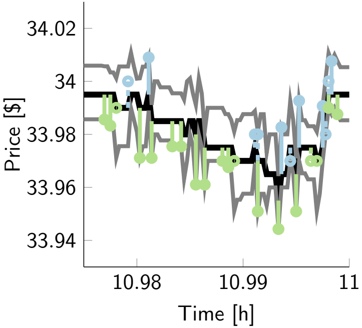


Sample Strategy Performance

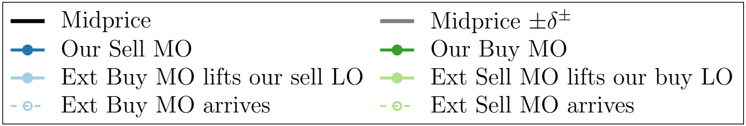
Single day performance for `ORCL` on 2013-05-15



Continuous



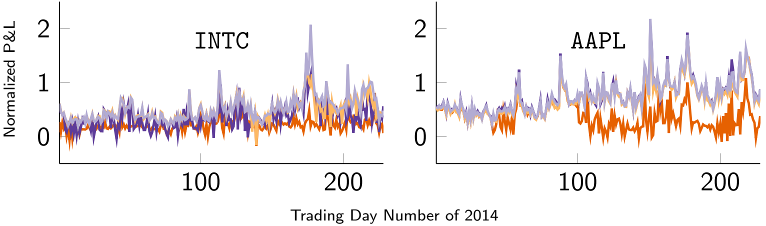
Discrete



In-Sample Backtesting: Conclusions

- average return increases as the underlying stock liquidity increases;
- average return increases as the underlying stock bid-ask spread decreases;
- average return is stable and risk-adjusted return is improved when calibrating over a larger period of time, and is therefore preferred;
- there is no clear victor between regular calibration and the nFPC method.

Out-Of-Sample Backtesting: Annual Calibration



Strategy	Average Return	Risk Adj Return	# MO	# LO	Average Invntry	% Win
INTC						
Continuous	0.209	2.112	2118	1758	0.44	98%
Discrete	0.372	1.591	949	1770	-5.89	98%
Continuous with nFPC	0.483	2.364	704	1693	1.46	100%
Discrete with nFPC	0.515	2.033	490	1629	2.81	100%
AAPL						
Continuous	0.378	1.571	3853	6297	-5.80	96%
Discrete	0.761	2.457	830	5566	4.05	100%
Continuous with nFPC	0.710	2.479	1276	5689	2.93	100%
Discrete with nFPC	0.764	2.442	796	5559	3.85	100%

Out-Of-Sample Backtesting: Annual Calibration

Back-of-the-envelope calculation:

Trade 100 shares at a time $\times$ average strategy return $\times$ average share price $\times$ 249 (trading days)

Trading `INTC` would have generated revenue of \$384,705.
Trading `AAPL` would have generated revenue of \$1,807,200.

Capital requirements: 100 shares $\times$ average share price $\times$ 20 (maximum inventory) $=$ \$250,000.

Return on investment (ROI) is 877%.

Roadmap

Conclusion

- 877% ROI on `INTC` and `AAPL`
- Factor in colocation fees, data subscription fees...
- ROI down to 359%
- Other high liquidity, low bid-ask spread stocks: `DELL`, `MSFT`
- *Can we take this strategy to market?*

Starting a Hedge Fund

Future Work

- Market order costs
- Discrete posting depths in increments of 1 tick
  - Can be solved by rounding...
- Our impact on the market (short-term price impact)
- Accounting for non-homogeneity
- Backtesting engine: information latency
- Backtesting engine: algorithm latency
- Backtesting engine: **tracking LOB queue position**
  - $e^{-\kappa\delta}$ fill probability is highly flawed

Thank you!

Questions?

Acknowledgement

Dr. Gabriele D'Eleuterio
For acting as my supervisor and supporting me through the graduate school tumult.

Dr. Sebastian Jaimungal
For taking me on as a surrogate student and guiding me through the research.

Dr. Dmitri Rubisov (my dad)
For pretty much being a supervisor too.