Statistical Arbitrage Using Limit-Order Book Imbalance

Anton Rubisov

University of Toronto

17 September 2015

Roadmap

Roadmap

Roadmap

Limit-Order Book Imbalance

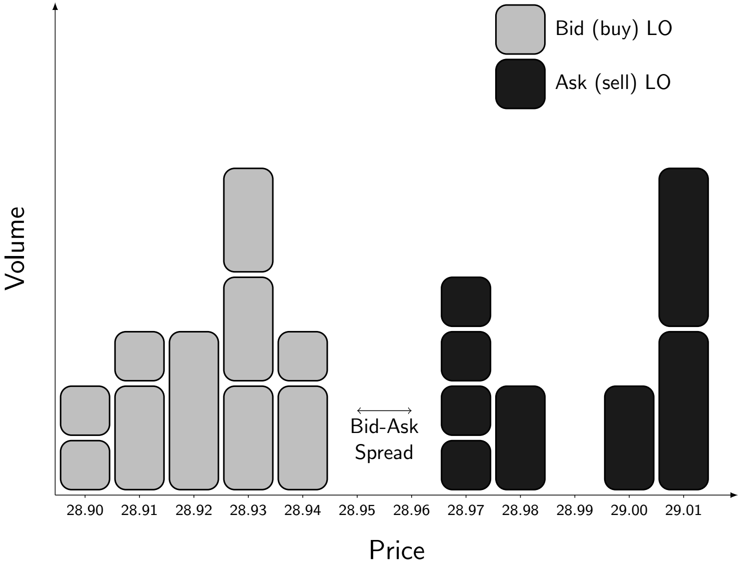
Imbalance is a ratio of quoted limit order volumes between the bid and ask side.

Limit-Order Book Imbalance

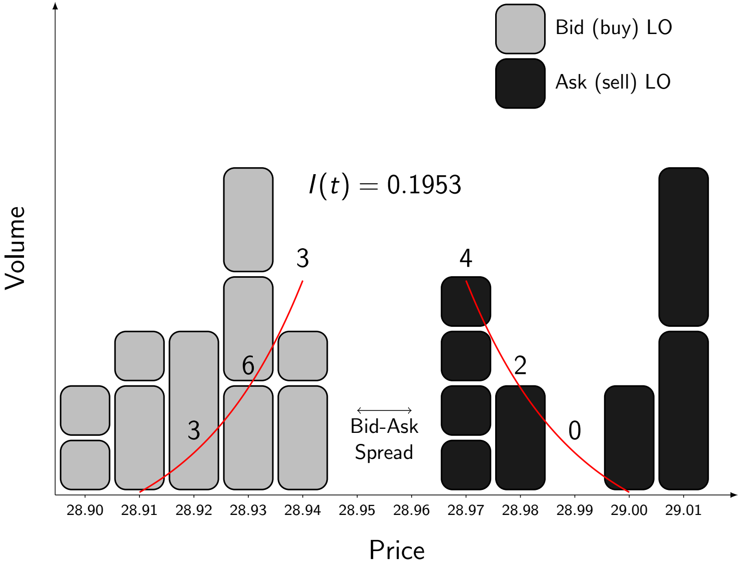
Imbalance is a ratio of quoted limit order volumes between the bid and ask side.

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

Limit-Order Book Imbalance

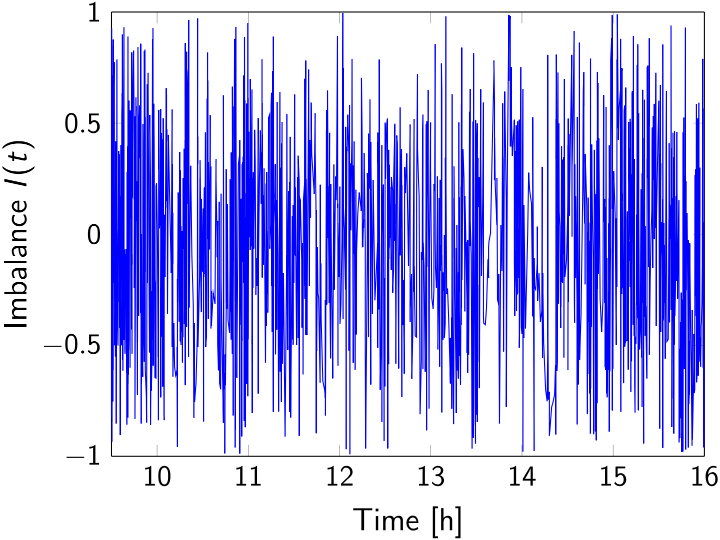


Limit-Order Book Imbalance



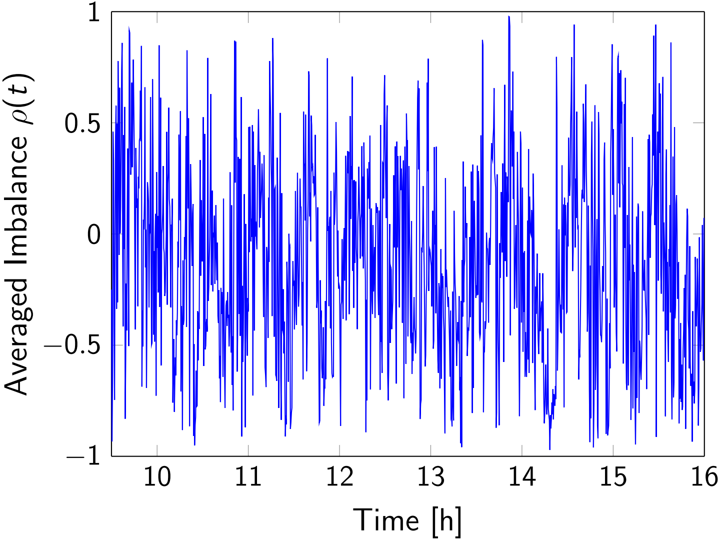
Limit-Order Book Imbalance

Resulting timeseries is noisy.



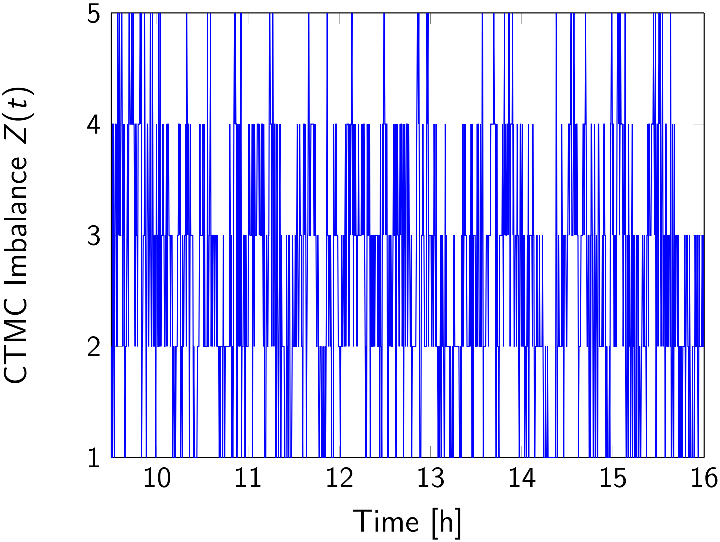
Limit-Order Book Imbalance

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

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

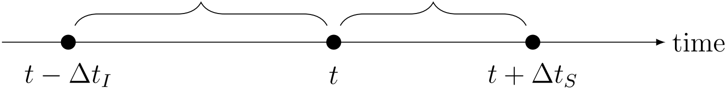
is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and

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

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

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

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**

Trading Strategies Informed by the Q Matrix

Naive Use market orders to buy (sell) if price is predicted to move up (down).

Naive+ Post at-the-touch limit orders when zero price change is predicted.

Naive++ Post a limit order to buy (sell) is price is predicted to move up (down).

Calibration of Naive Trading Strategies

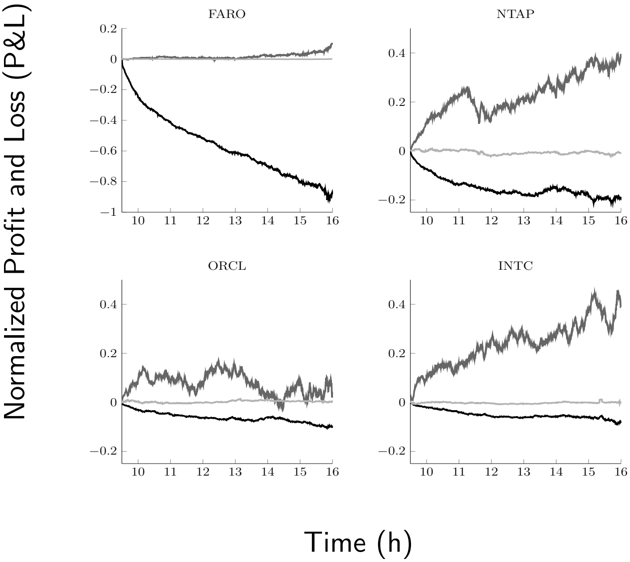
Need to select:

- price change observation period Δt_S
- imbalance averaging period Δt_I
- number of imbalance bins $\#_{bins}$

Calibration done on the first day of the trading year, same parameters used for all days.

Brute-force search of parameter space, using max Sharpe ratio criterion, found that $\Delta t_S = \Delta t_I = 1sec$, and $\#_{bins} = 4$

Results of Naive Trading Strategies



Conclusions from Naive Trading Strategies

Why is the Naive strategy producing, on average, normalized losses?

- Backtest is out-of-sample; evidence to reject time-homogeneity
- Calibration is done on first trading day; likely nonrepresentative of trading activity
- Price change probability matrix $\boldsymbol{Q}$ obtained using midprices, ignoring bid-ask spread; $\mathrm{sgn}(\Delta S)$ may be insufficient for create profit, especially on `FARO`

Conclusions from Naive Trading Strategies

Why do the Naive+ and Naive++ strategies outperform the Naive strategy?

- LOs vs MOs means different transaction price is being used (only MO loses value)
- Naive only executes when predicting non-zero price change
  - Only sign, not magnitude
  - Only *if one was already seen*

Roadmap

System Description

- Imbalance Averaging Time Δt_I
  A constant, specifying the time window over which the imbalance ratio $I(t)$ will be averaged.
- Price Change Time Δt_S
  A constant, specifying the time window over which price changes will be computed.
- Number of Imbalance Bins $\#_{bins}$
  A constant, specifying the number of bins (spaced by percentiles, symmetric around zero) into which $I(t)$ will be sorted.
- Imbalance ρ_t
  The finite, discrete stochastic process that results from sorting $I(t)$ into the imbalance bins $\{1, \ldots, \#_{bins}\}$, and which evolves in accordance with the CTMC $\boldsymbol{Z}$.

System Description

- Midprice S_t
  Stochastic process, evolves according to CTMC $\boldsymbol{Z}$.
- Midprice Change $\Delta S_t = \operatorname{sgn}(S_t - S_{t-\Delta t_S})$
- Imbalance & Midprice Change $\boldsymbol{Z}_t = (\rho_t, \Delta S_t)$
  Continuous-time Markov chain with generator $\boldsymbol{G}$.
- Bid-Ask Half-Spread ξ
  Assumed constant. 2ξ is equal to the bid-ask spread.
- Midprice Change $\{\eta_{0,\boldsymbol{z}}, \eta_{1,\boldsymbol{z}}, \dots\} \sim F_{\boldsymbol{z}}$
  i.i.d. RVs, with distribution dependent on the Markov chain state.

System Description

- Other Agent Market Orders $K_t^{\pm}$
  Poisson processes with rate $\mu^{\pm}(\boldsymbol{Z}_t)$. K^+ represents the arrival of another agent's buy market order.
- Our Limit Order Posting Depth $\delta_t^{\pm}$
  One of our controlled $\mathcal{F}$-predictable processes. δ^+ dictates how deep on the buy side we will post our buy limit order; $\delta^+ = 0$ implies at-the-touch.
- Our Limit Order Fill Count $L_t^{\pm}$
  Counting processes (not Poisson), satisfying

$$\mathbb{P}[\,\mathrm{d}L_t^{\pm} = 1 \mid \mathrm{d}K_t^{\mp} = 1] = e^{-\kappa \delta_t^{\pm}}$$

System Description

- Fill Probability Constant κ
  Fitted to satisfy the above relation, by considering the avg vol available at the first few depths relative to distribution of volumes of incoming market orders
- Our Market Orders $M_t^{\pm}$
  M^+ represents our buy market order. Assume we achieve the best bid/ask price.
- Our Market Order Execution Times
  $\boldsymbol{\tau}^{\pm} = \{\tau_k^{\pm} : k = 1, \dots\}$
  An increasing sequence of $\mathcal{F}$-stopping times.

System Description

- Cash $X_t^{\tau,\delta}$
  A stochastic variable representing our cash, initially zero, that evolves according to

$$
\begin{aligned}
\mathrm{d}X_t^{\tau,\delta} = &\underbrace{(S_t + \xi + \delta_t^-)\,\mathrm{d}L_t^-}_{\text{sell limit order}} - \underbrace{(S_t - \xi - \delta_t^+)\,\mathrm{d}L_t^+}_{\text{buy limit order}} \\
&+ \underbrace{(S_t - \xi)\,\mathrm{d}M_t^-}_{\text{sell market order}} - \underbrace{(S_t + \xi)\,\mathrm{d}M_t^+}_{\text{buy market order}}
\end{aligned}
$$

- Inventory $Q_t^{\tau,\delta}$
  A stochastic process representing our assets, initially zero, that satisfies

$$
Q_0^{\tau,\delta} = 0, \qquad Q_t^{\tau,\delta} = L_t^+ + M_t^+ - L_t^- - M_t^-
$$

Terminal Wealth

Call $W_t^{\tau,\delta}$ our net present value (NPV) at time t. Hence $W_T^{\tau,\delta}$ at terminal time T is our 'terminal wealth.'

At T, we:

- finish each trading day with zero inventory (avoid overnight positional risk)
- submit a market order (of a possibly large volume) to liquidate remaining stock
- price achieved will be $S - \xi \operatorname{sgn} Q - \alpha Q$
  - $\xi \operatorname{sgn} Q$ represents crossing the spread
  - α is a penalty constant
  - αQ represents receiving a worse price linearly in Q due to walking the book

Hence, $W_t^{\tau,\delta}$ satisfies:

$$W_t^{\tau,\delta} = \underbrace{X_t^{\tau,\delta}}_{\text{cash}} + \underbrace{Q_t^{\tau,\delta}\left(S_t - \xi \operatorname{sgn}(Q_t^{\tau,\delta})\right)}_{\text{book value of assets}} - \underbrace{\alpha\left(Q_t^{\tau,\delta}\right)^2}_{\text{liquidation penalty}}$$

Dynamic Programming Value Function

Our performance criterion will be to maximize terminal wealth:

$$H^{\boldsymbol{\tau},\delta}(t, x, s, \boldsymbol{z}, q) = \mathbb{E}_{t,x,s,z,q}\left[W_T^{\boldsymbol{\tau},\delta}\right]$$

The *value function* is given by

$$H(t, x, s, \boldsymbol{z}, q) = \sup_{\boldsymbol{\tau}\in\mathcal{T}_{[t,T]}} \sup_{\delta\in\mathcal{A}_{[t,T]}} H^{\boldsymbol{\tau},\delta}(t, x, s, \boldsymbol{z}, q)$$

Admissible trading strategies is the product of the set $\mathcal{T}$ of all $\mathcal{F}$-stopping times, with the set $\mathcal{A}$ of all $\mathcal{F}$-predictable, bounded-from-below depths $\delta \geq 0$.

- recognizing τ does not require knowledge of the future
- δ cannot 'see into the future'; measurable with respect to information at an earlier time

Dynamic Programming Equation

Theorem

Dynamic Programming Equation for Optimal Stopping and Control. *(Cartea et al., 2015) Assume that the value function $H(t, \boldsymbol{x})$ is once differentiable in t, all second-order derivatives in $\boldsymbol{x}$ exist, and that $G : \mathbb{R}^m \to \mathbb{R}$ is continuous. Then H solves the quasi-variational inequality*

$$\max \left\{ \partial_t H + \sup_{\boldsymbol{u} \in \mathcal{A}_t} \mathcal{L}_t^{\boldsymbol{u}} H \; ; \; G - H \right\} = 0$$

on $\mathcal{D}$, where $\mathcal{D} = [0, T] \times \mathbb{R}^m$.

Solving for the Infinitesimal Generator

$$
\begin{aligned}
\mathrm{d}H(t,x,s,\boldsymbol{z},q) &= \partial_t H \,\mathrm{d}t \\
&+ \underbrace{e^{-\kappa\delta^-}\left[H(t,x+(s+\xi+\delta^-),s,\boldsymbol{z},q-1)-H(\cdot)\right]\mathrm{d}K^+}_{\text{probability of our sell limit order being filled, times the change in value}} \\
&+ \underbrace{e^{-\kappa\delta^+}\left[H(t,x-(s-\xi-\delta^+),s,\boldsymbol{z},q+1)-H(\cdot)\right]\mathrm{d}K^-}_{\text{probability of our buy limit order being filled, times the change in value}} \\
&+ \sum_{\boldsymbol{j}} \underbrace{\mathbb{E}\left[H(t,x,s+\eta_{0,\boldsymbol{j}},\boldsymbol{j},q)-H(\cdot)\right]\,\mathrm{d}Z_{\boldsymbol{z},\boldsymbol{j}}}_{\text{change in value resulting from a CTMC state switch}}
\end{aligned}
$$

Compensated Markov Chain Process

To solve for $\mathcal{L}_t^{\boldsymbol{u}} Ht$ we need the compensated Markov chain process. For Poisson processes, this is

$$\mathrm{d}K^{\pm} = \mathrm{d}\,\widetilde{\mathcal{K}}^{\pm} + \mu^{\pm}(\boldsymbol{z})\,\mathrm{d}t$$

For the CTMC:

Compensated Markov Chain Process

To solve for $\mathcal{L}_t^{\boldsymbol{u}} Ht$ we need the compensated Markov chain process. For Poisson processes, this is

$$\mathrm{d}K^{\pm} = \mathrm{d}\,\widetilde{\mathcal{K}}^{\pm} + \mu^{\pm}(\boldsymbol{z})\,\mathrm{d}t$$

For the CTMC:

- Define $K_l(t)$ to be the number of jumps with $Z_s - Z_{s^-} = l$ up to time t

Compensated Markov Chain Process

To solve for $\mathcal{L}_t^{\boldsymbol{u}} Ht$ we need the compensated Markov chain process. For Poisson processes, this is

$$\mathrm{d}K^{\pm} = \mathrm{d}\,\widetilde{\mathcal{K}}^{\pm} + \mu^{\pm}(\boldsymbol{z})\,\mathrm{d}t$$

For the CTMC:

- Define $K_l(t)$ to be the number of jumps with $Z_s - Z_{s^-} = l$ up to time t
- Define $\beta_l(x) = G_{x,x+l}$

Compensated Markov Chain Process

To solve for $\mathcal{L}_t^{\boldsymbol{u}} Ht$ we need the compensated Markov chain process. For Poisson processes, this is

$$\mathrm{d}K^{\pm} = \mathrm{d}\,\widetilde{\mathcal{K}}^{\pm} + \mu^{\pm}(\boldsymbol{z})\,\mathrm{d}t$$

For the CTMC:

- Define $K_l(t)$ to be the number of jumps with $Z_s - Z_{s^-} = l$ up to time t
- Define $\beta_l(x) = G_{x,x+l}$
- Then the compensated process (a martingale) is

$$\widetilde{\mathcal{K}}_l\,(t) = K_l(t) - \int_0^t \beta_l(Z_s)\,\mathrm{d}s$$

Solving for the Infinitesimal Generator

$$
\begin{aligned}
\mathcal{L}_t^{\delta} H = {} & \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} \left[H(t, x + (s + \xi + \delta^-), s, \boldsymbol{z}, q - 1) - H(\cdot) \right] \\
& + \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} \left[H(t, x - (s - \xi - \delta^+), s, \boldsymbol{z}, q + 1) - H(\cdot) \right] \\
& + \sum_{\boldsymbol{j}} G_{\boldsymbol{z},\boldsymbol{j}} \mathbb{E} \left[H(t, x, s + \eta_{0,\boldsymbol{j}}, \boldsymbol{j}, q) - H(\cdot) \right]
\end{aligned}
$$

Resulting Dynamic Programming Equation

Our dynamic programming equation simplifies to

$$0 = \max\Big\{\partial_t H + \sup_{\boldsymbol{u}\in\mathcal{A}_t} \mathcal{L}_t^{\boldsymbol{u}} H \;;\; H(t, x-(s+\xi), s, \boldsymbol{z}, q+1) - H(\cdot) \;; \\ H(t, x+(s-\xi), s, \boldsymbol{z}, q-1) - H(\cdot)\Big\}$$

with boundary conditions

$$\begin{aligned} H(T, x, s, \boldsymbol{z}, q) &= x + q(s - \xi\,\mathrm{sgn}(q)) - \alpha q^2 \\ H(t, x, s, \boldsymbol{z}, 0) &= x \end{aligned}$$

Ansatz for Value Function H

Introduce the ansatz

$$H(t, x, s, \boldsymbol{z}, q) = x + q(s - \xi \operatorname{sgn}(q)) + h(t, \boldsymbol{z}, q)$$

- separates out mark-to-market of the current position
- $h(t, \boldsymbol{z}, q)$ captures value due to optimal trading

Boundary conditions on h are

$$\begin{aligned} h(T, \boldsymbol{z}, q) &= -\alpha q^2 \\ h(t, \boldsymbol{z}, 0) &= 0 \end{aligned}$$

Simplified Infinitesimal Generator

$$
\begin{aligned}
\mathcal{L}_t^\delta H = {} & \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} \left[\delta^- + 2\xi \cdot \mathbb{1}_{q \geq 1} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q)\right] \\
& + \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} \left[\delta^+ + 2\xi \cdot \mathbb{1}_{q \leq -1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\right] \\
& + \sum_{\boldsymbol{j}} G_{\boldsymbol{z},\boldsymbol{j}} \left[q\mathbb{E}[\eta_{0,\boldsymbol{j}}] + h(t, \boldsymbol{j}, q) - h(t, \boldsymbol{z}, q)\right]
\end{aligned}
$$

Optimal Posting Depth

To find the supremum over δ^+, consider the first-order constraint:

$$0 = \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right]\right]$$

Optimal Posting Depth

To find the supremum over δ^+, consider the first-order constraint:

$$0 = \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right]\right]$$
$$= -\kappa e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right] + e^{-\kappa\delta^{+*}}$$

Optimal Posting Depth

To find the supremum over δ^+, consider the first-order constraint:

$$
\begin{aligned}
0 &= \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right]\right] \\
&= -\kappa e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right] + e^{-\kappa\delta^{+*}} \\
&= e^{-\kappa\delta^{+}}\left[-\kappa\left(\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right) + 1\right]
\end{aligned}
$$

Optimal Posting Depth

To find the supremum over δ^+, consider the first-order constraint:

$$\begin{aligned} 0 &= \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right]\right] \\ &= -\kappa e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right] + e^{-\kappa\delta^{+*}} \\ &= e^{-\kappa\delta^{+}}\left[-\kappa\left(\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right) + 1\right] \\ &= -\kappa\left(\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\le -1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right) + 1 \end{aligned}$$

Optimal Posting Depth

To find the supremum over δ^+, consider the first-order constraint:

$$\begin{aligned}
0 &= \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\leq-1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right]\right] \\
&= -\kappa e^{-\kappa\delta^{+*}}\left[\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\leq-1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\right] + e^{-\kappa\delta^{+*}} \\
&= e^{-\kappa\delta^{+}}\left[-\kappa\big(\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\leq-1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\big) + 1\right] \\
&= -\kappa\big(\delta^{+*} + 2\xi\cdot\mathbb{1}_{q\leq-1} + h(t,\boldsymbol{z},q+1) - h(t,\boldsymbol{z},q)\big) + 1
\end{aligned}$$

Rearranging, and recalling we want non-negative posting depths, the optimal buy depth δ^* is given by:

$$\delta^{+*} = \max\left\{0\,;\ \frac{1}{\kappa} - 2\xi\cdot\mathbb{1}_{q\leq-1} - h(t,\boldsymbol{z},q+1) + h(t,\boldsymbol{z},q)\right\}$$

Simplified Dynamic Programming Equation

$$
\begin{aligned}
0 = \max\Bigg\{ & \partial_t h + \mu^+(\boldsymbol{z}) e^{-\kappa \delta^{-*}} \left(\delta^{-*} + 2\xi \cdot \mathbb{1}_{q \geq 1} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q) \right) \\
& \quad + \mu^-(\boldsymbol{z}) e^{-\kappa \delta^{+*}} \left(\delta^{+*} + 2\xi \cdot \mathbb{1}_{q \leq -1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) \right) \\
& \quad + \sum_{\boldsymbol{j}} G_{\boldsymbol{z},\boldsymbol{j}} \left[q/\mathbb{E}\left[\eta_{0,\boldsymbol{j}}\right] + h(t, \boldsymbol{j}, q) - h(t, \boldsymbol{z}, q) \right] \ ; \\
& - 2\xi \cdot \mathbb{1}_{q \geq 0} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) \ ; \\
& - 2\xi \cdot \mathbb{1}_{q \leq 0} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q) \Bigg\}
\end{aligned}
$$

Interpreting the DPE - Market Orders

A buy market order will be executed at time τ_q^+ whenever

$$h(\tau_q^+, \boldsymbol{z}, q+1) - h(\tau_q^+, \boldsymbol{z}, q) = 2\xi \cdot \mathbb{1}_{q \geq 0}$$

and a sell market order whenever

$$h(\tau_q^+, \boldsymbol{z}, q-1) - h(\tau_q^+, \boldsymbol{z}, q) = 2\xi \cdot \mathbb{1}_{q \leq 0}$$

- 2ξ is the difference between purchase price and mark-to-market
- drives inventory back to zero when value unaffected

Interpreting the DPE - Limit Orders

Combining continuation region inequalities:

$$h(t, \boldsymbol{z}, q) \overset{\substack{\text{sell if } = \\ \downarrow}}{\leq} h(t, \boldsymbol{z}, q+1) \overset{\substack{\text{buy if } = \\ \downarrow}}{\leq} h(t, \boldsymbol{z}, q) + 2\xi, \qquad q \geq 0$$

$$h(t, \boldsymbol{z}, q) \underset{\substack{\uparrow \\ \text{buy if } =}}{\leq} h(t, \boldsymbol{z}, q-1) \underset{\substack{\uparrow \\ \text{sell if } =}}{\leq} h(t, \boldsymbol{z}, q) + 2\xi, \qquad q \leq 0$$

Interpreting the DPE - Posting Depths

Substituting the previous inequalities into our posting depth formula, we obtain bounds on our depths

$$\frac{1}{\kappa} - 2\xi \leq \delta^{\pm *} \leq \frac{1}{\kappa}$$

- possible interpretation: if buy depth is 'sufficiently' large, then actually optimal to sell instead

Roadmap

Background Information

Exploratory Data Analysis

Maximizing Wealth via Continuous-Time Stochastic Control

Maximizing Wealth via Discrete-Time Stochastic Control

Backtesting Results

Conclusion and Future Work

System Description

All names and variables are the same as in the continuous-time case.

We will consider the embedded discrete time Markov chain determined by

$$\boldsymbol{P} = e^{\boldsymbol{G}\Delta t}$$

for any time interval of size Δt. For our purposes, take $\Delta t = \Delta t_I = \Delta t_S = 1000$ ms

System Description

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

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

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

System Description - System Evolution

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

System Description - Impulse Control

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

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
- as in continuous time, assume order is lifted with probability $e^{-\kappa\delta^-}$.

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

Dynamic Programming Value Function

Again, our performance criterion is our *terminal wealth*:

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

So that our dynamic programming equations are

$$
\begin{aligned}
V_T(x, s, \boldsymbol{z}, q) &= x + q(s - \xi\,\mathrm{sgn}(q)) - \alpha q^2 \\
V_k(x, s, \boldsymbol{z}, q) &= \max\Big\{\sup_{\delta^\pm}\{\mathbb{E}_{\boldsymbol{w}}\left[V_{k+1}(f((x, s, \boldsymbol{z}, q), \boldsymbol{u}, \boldsymbol{w}_k))\right]\}\ ; \\
&\qquad V_k(x + s_k - \xi, s_k, \boldsymbol{z}_k, q_k - 1)\ ; \\
&\qquad V_k(x - s_k - \xi, s_k, \boldsymbol{z}_k, q_k + 1)\Big\}
\end{aligned}
$$

Ansatz for Value Function V

Introduce the same ansatz

$$V_k(x, s, \boldsymbol{z}, q) = x + q(s - \xi \operatorname{sgn}(q)) + h_k(\boldsymbol{z}, q)$$

with boundary condition $h_k(\boldsymbol{z}, 0) = 0$ and terminal condition $h_T(\boldsymbol{z}, q) = -\alpha q^2$.
The DPE simplifies to

$$\begin{aligned} 0 = \max\Bigg\{ & \sup_{\delta^\pm}\Big\{\mathbb{E}_{\boldsymbol{w}}\big[(s + \xi + \delta^-)L_k^- - (s - \xi - \delta^+)L_k^+ \\ & \qquad + (L_k^+ - L_k^-)\big(s + \eta_{0,T(\boldsymbol{z},\omega)} - \xi \operatorname{sgn}(q + L_k^+ - L_k^-)\big) \\ & \qquad + q\left(\eta_{0,T(\boldsymbol{z},\omega)} - \xi\left(\operatorname{sgn}(q + L_k^+ - L_k^-) - \operatorname{sgn}(q)\right)\right) \\ & \qquad + h_{k+1}(T(\boldsymbol{z},\omega), q + L_k^+ - L_k^-) - h_k(\boldsymbol{z}, q)\big]\Big\}\ ; \\ & - 2\xi \cdot \mathbb{1}_{q \geq 0} + h_k(\boldsymbol{z}, q+1)\ ; \\ & - 2\xi \cdot \mathbb{1}_{q \leq 0} + h_k(\boldsymbol{z}, q-1)\Bigg\} \end{aligned}$$

Optimal Posting Depth

To solve the supremum in the continuation region, again consider first-order condition and floor at zero. We obtain:

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

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

And similarly, the optimal buy posting depth is given by:

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

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

Optimal Posting Depth

Substituting one into the other, the optimal depth sell order posting depth is

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

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

$$\left.^{-(1-e^{\mu^+(\boldsymbol{z})\Delta t})e^{-\kappa\delta^{-*}}(2\xi\mathbb{1}_{q=0}-\aleph(q))\}}\ (2\xi\mathbb{1}_{q=0} - \aleph(q))\right\}$$

where

$$\aleph(q) = \sum_{\boldsymbol{j}} \boldsymbol{P}_{\boldsymbol{z},\boldsymbol{j}}\left[h_{k+1}(\boldsymbol{j},q-1) + h_{k+1}(\boldsymbol{j},q+1) - 2h_{k+1}(\boldsymbol{j},q)\right]$$

- solve numerically due to difficulty in isolating δ^{-*}

Simplified Dynamic Programming Equation

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

Interpreting the DPE

Again we have

$$
\begin{aligned}
h_k(\boldsymbol{z}, q) \overset{\substack{\text{sell if } = \\ \downarrow}}{\leq} h_k(\boldsymbol{z}, q+1) \overset{\substack{\text{buy if } = \\ \downarrow}}{\leq} h_k(\boldsymbol{z}, q) + 2\xi, \qquad & q \geq 0 \\
h_k(\boldsymbol{z}, q) \underset{\substack{\uparrow \\ \text{buy if } =}}{\leq} h_k(\boldsymbol{z}, q-1) \underset{\substack{\uparrow \\ \text{sell if } =}}{\leq} h_k(\boldsymbol{z}, q) + 2\xi, \qquad & q \leq 0
\end{aligned}
$$

- recalling $h_k(\boldsymbol{z}, 0) = 0$, again tells us h non-negative everywhere
- but no upper/lower bounds on $\delta^{\pm *}$

Roadmap

Background Information

Exploratory Data Analysis

Maximizing Wealth via Continuous-Time Stochastic Control

Maximizing Wealth via Discrete-Time Stochastic Control

Backtesting Results

Conclusion and Future Work

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

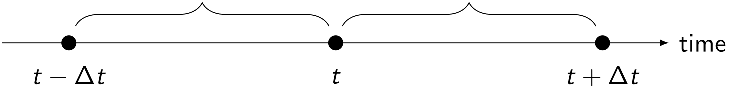
Parameter	Equation
$\boldsymbol{G}$	infinitesimal generator matrix
$\boldsymbol{P}$	transition probability matrix
$\mu^{\pm}$	market order arrival intensities
$H(t, x, s, \boldsymbol{z}, q)$	dynamic programming value function
$\delta^{\pm}$	limit order posting depths

Two Calibration Frameworks

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

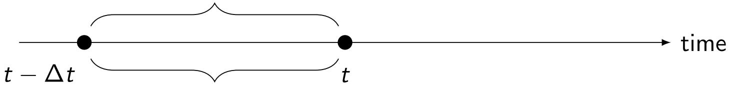
$\rho(t)$ is the imbalance bin of the time-weighted average of $I(t)$ over this past interval.

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



Regular calibration

$\rho(t)$ unchanged.

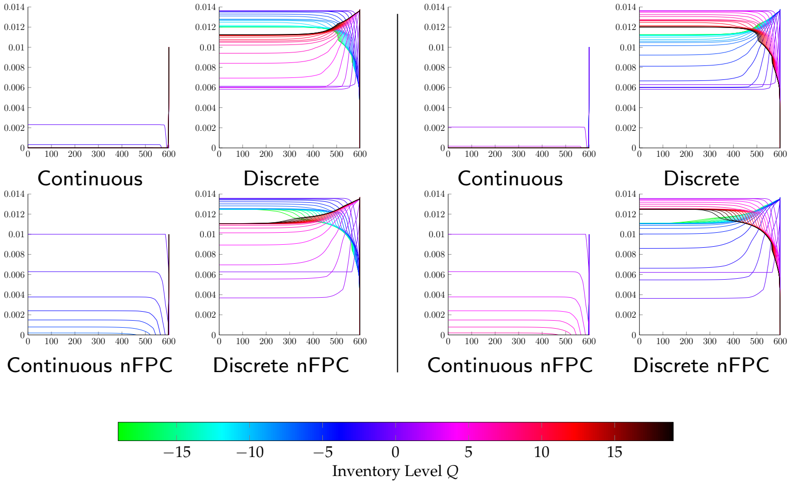


$\Delta S(t)$ calculated over the same past interval.

Dynamics of Posting Depths

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

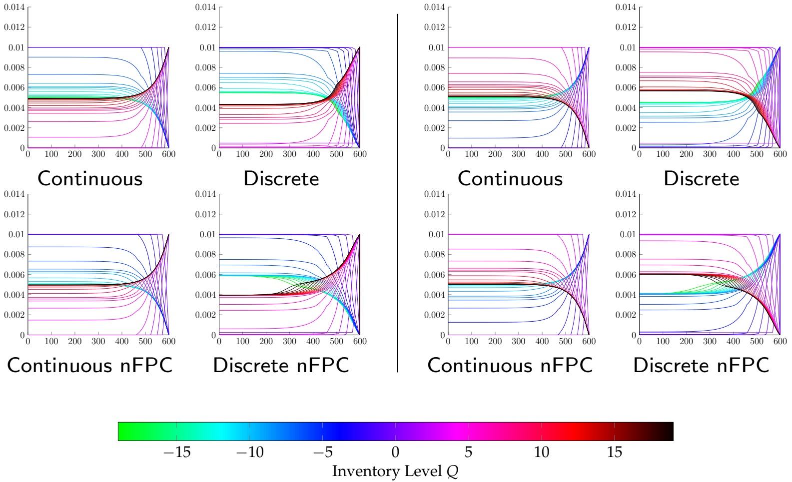
SELL Posting Depth δ^+ [$]
at $Z = (\rho = +1, \Delta S = +1)$



Dynamics of Posting Depths

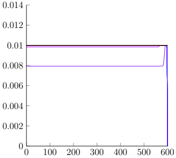
BUY Posting Depth δ^+ [$]
at $Z = (\rho = 0, \Delta S = 0)$

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

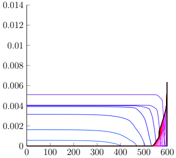


Dynamics of Posting Depths

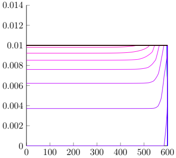
BUY Posting Depth δ^+ [$] at $Z = (\rho = +1, \Delta S = +1)$



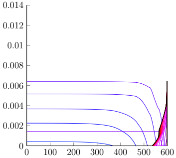
Continuous



Discrete

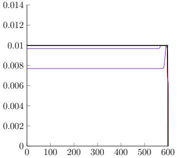


Continuous nFPC

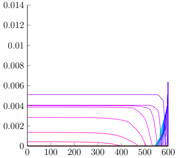


Discrete nFPC

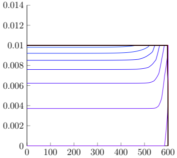
SELL Posting Depth δ^+ [$] at $Z = (\rho = -1, \Delta S = -1)$



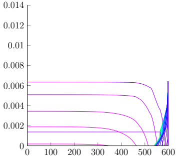
Continuous



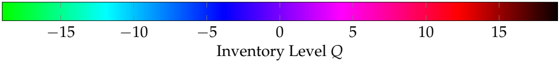
Discrete



Continuous nFPC

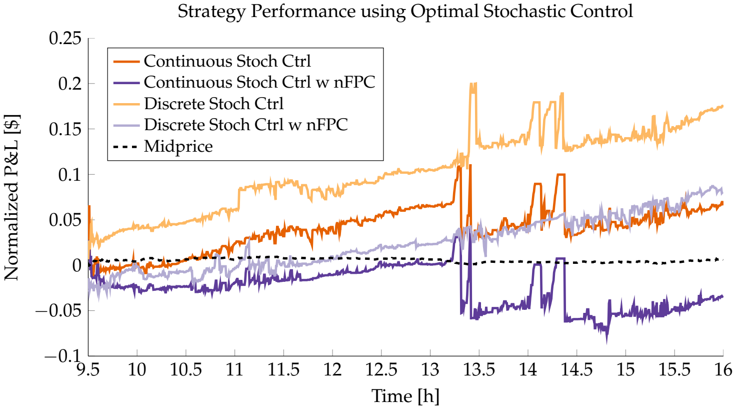


Discrete nFPC



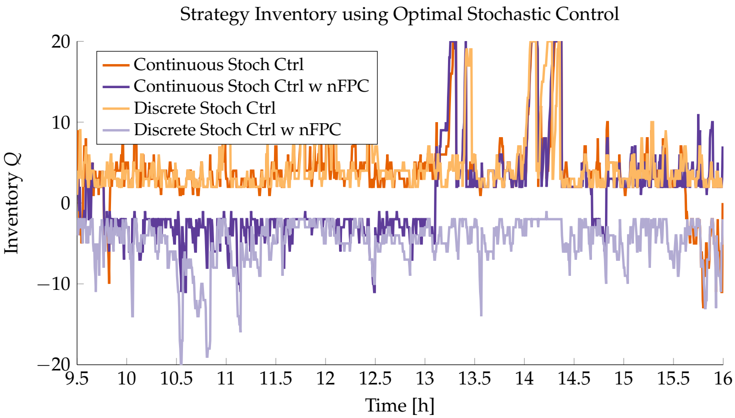
Sample Strategy Performance

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



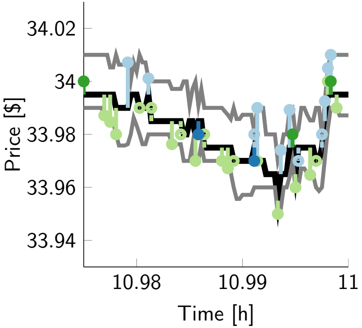
Sample Strategy Performance

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

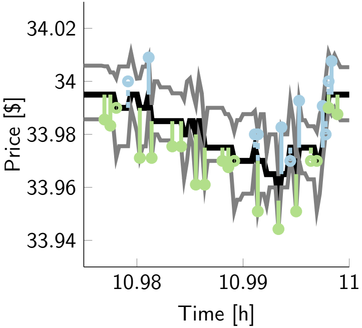


Sample Strategy Performance

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



Continuous

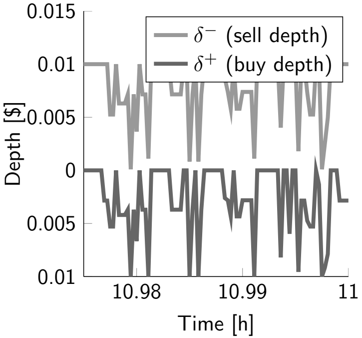


Discrete

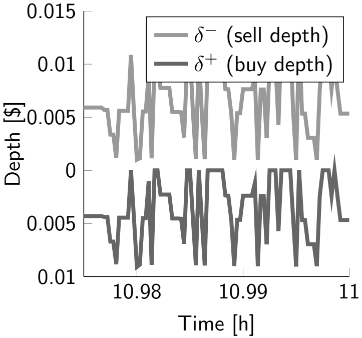
Midprice	Midprice $\pm\delta^{\pm}$
Our Sell MO	Our Buy MO
Ext Buy MO lifts our sell LO	Ext Sell MO lifts our buy LO
Ext Buy MO arrives	Ext Sell MO arrives

Sample Strategy Performance

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

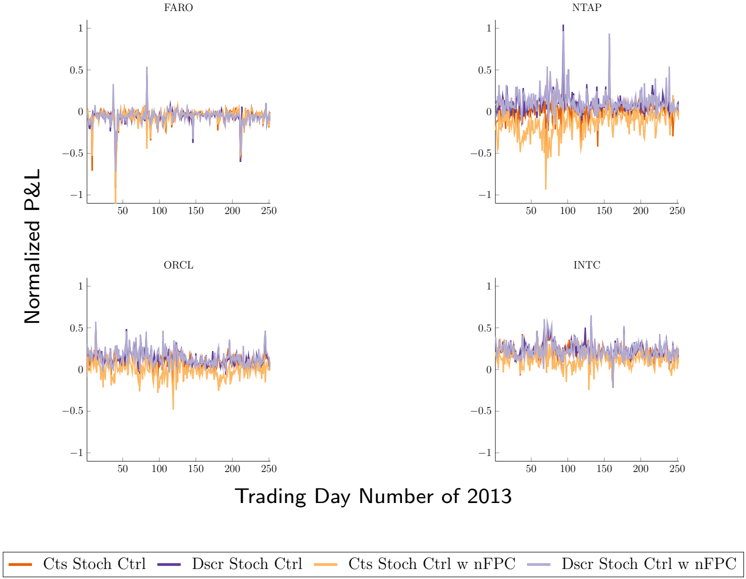


Continuous



Discrete

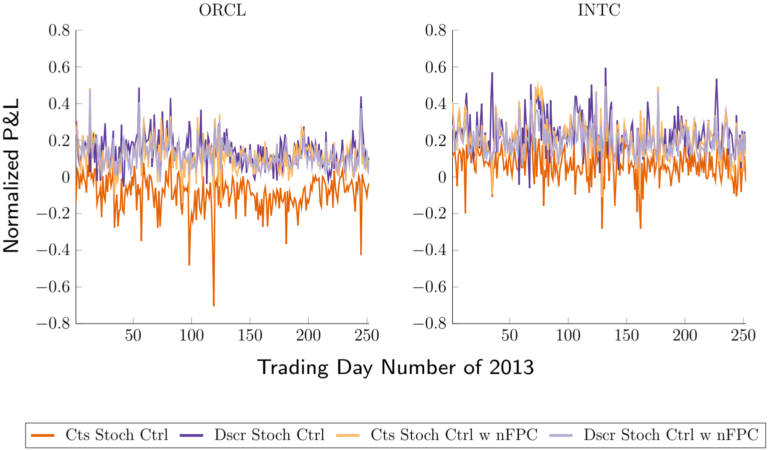
In-Sample Backtesting: Same Day Calibration



In-Sample Backtesting: Same Day Calibration

Strategy	Average Return	Risk Adj Return	# Trades	Average Inventory	% Win
ORCL					
Naive	-0.105	-0.253	484	1.40	28%
Naive+	-0.034	-0.011	4086	-55.18	61%
Naive++	0.002	0.006	132	0.61	52%
Continuous	0.115	1.348	1874	1.94	92%
Discrete	0.135	1.620	1898	3.93	98%
Continuous with nFPC	-0.010	-0.100	2455	1.32	48%
Discrete with nFPC	0.144	1.501	1759	2.85	97%
INTC					
Naive	-0.082	-0.228	258	-5.21	33%
Naive+	0.365	0.134	3962	-32.50	63%
Naive++	-0.001	-0.003	74	-0.84	48%
Continuous	0.214	2.159	1577	5.17	97%
Discrete	0.232	2.528	1642	4.48	98%
Continuous with nFPC	0.114	1.218	1894	2.01	90%
Discrete with nFPC	0.226	2.202	1569	4.28	98%

In-Sample Backtesting: Annual Calibration



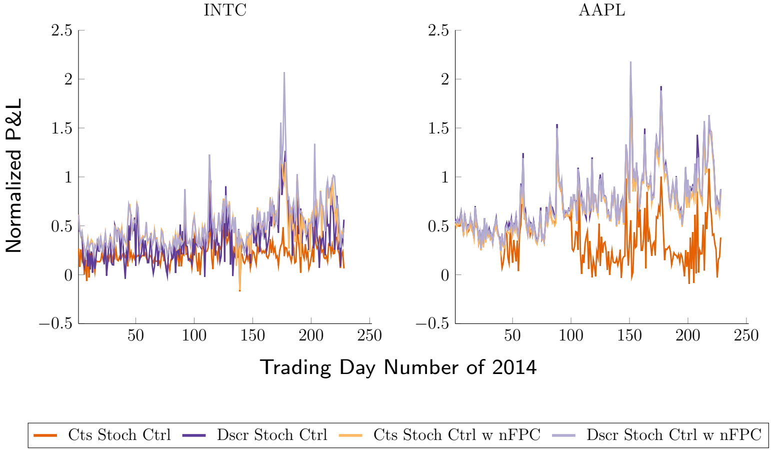
In-Sample Backtesting: Annual Calibration

Strategy	Average Return	Risk Adj Return	# MO	# LO	Average Invntry	% Win
`ORCL`						
Continuous	-0.089	-0.875	1540	1383	1.19	14%
Discrete	0.140	1.596	368	1344	0.46	96%
Continuous with nFPC	0.113	1.327	476	1338	2.67	94%
Discrete with nFPC	0.118	1.735	590	1337	3.43	99%
`INTC`						
Continuous	0.065	0.743	888	1207	1.22	84%
Discrete	0.235	2.189	380	1170	1.19	99%
Continuous with nFPC	0.209	2.030	396	1160	5.58	98%
Discrete with nFPC	0.197	2.588	576	1164	3.78	100%

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



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?*

Future Work

- Market order costs
  - LOs are free but MOs have a cost of c
  - presumed effect: upper bound on $\delta^{\pm}$ from $1/\kappa$ to $1/\kappa + c$.
  - presumed effect: fewer market orders
- Discrete posting depths in increments of 1 tick
  - Current depths are continuous variables
  - For most stocks, depth is in increments of $0.01
  - Can be solved by rounding...
- Short-term price impact
  - what effect does our trading have on the market?
  - 100 shares per execution = 600k shares per day = 1% daily vol

Future Work

- Accounting for non-homogeneity
  - data is nonhomogeneous, both intra-day and inter-day
  - calibrate first/last hour separately?
- Backtesting engine: information latency
  - minimizing latency is critical in high-frequency algorithmic trading
  - currently we have immediate execution
  - introduce simple 2-4 ms execution lag?
- Backtesting engine: algorithm latency
  - currently we feed a reconstructed LOB
  - requires redesign to process live data feed

Future Work

- Backtesting engine: tracking LOB queue position
  - $e^{-\kappa\delta}$ fill probability is highly flawed
    - random numbers determine whether our orders get filled
    - depth $\delta = 0$ is guaranteed execution
  - implicit cancellation and reposting of orders
  - queue position tracking adds a control of modifying/keeping existing order

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.