High-Frequency Algorithmic Trading with Momentum and Order Imbalance

Anton Rubisov

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

A thesis submitted in partial fulfilment for the degree of
Master of Applied Science

October 2015

Abstract

Robots in da skies.

Acknowledgements

And I would like to acknowledge ...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Hi, my name is Stereo Mike.

Yeah, we got three tickets to the Bran Van concert this Monday night at the Pacific Pallisades. You can all dial in if you want to answer a couple of questions; namely, what is Todd's favorite cheese? Jackie just called up and said it was a form of Roquefort. We'll see about that.

Give us a ring ding ding, it's a beautiful day.

Yeah Todd, this is Liquid, ring-a-ding-a-dinging, I want those three Bran Van tickets, man. Whaddya think? Todd?

Limit order book imbalance is a ratio of limit order volumes between the bid and ask side, and can be calculated for example as $I(t) = \dfrac{V_b(t) - V_a(t)}{V_b(t) + V_a(t)} \in [-1, 1]$.

- We bin the bid/ask volume imbalances in the Limit Order Book into K bins, each being dubbed a "regime" of the limit order book.
- Z_t is a continuous-time Markov chain that tracks which regime we're in. Z_t takes values in $\{1, \ldots, K\}$, and has an infinitesimal generator matrix G.
- Conditional on being in some regime k, the arrival of buy and sell market orders follow independent Poisson processes with intensities $\lambda_k^{\pm}$.

We have observations of arrivals of buy/sell market orders and of regime switches occurring, all of which are timestamped. Pictorially, a timeline might look like:

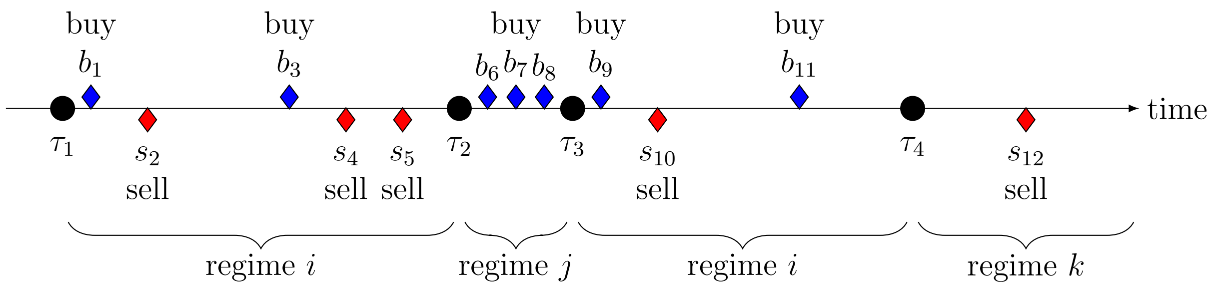


Figure 1.1: Hypothetical timeline of market orders arriving during changing order imbalance regimes.

Chapter 2

Exploratory Data Analysis

2.1 Maximum Likelihood Estimate of a Doubly Stochastic Poisson Process

2.1.1 Maximum Likelihood Estimation of G

Let G be the generator matrix for Z_t, so $G = \{q_{ij}\} \in \mathbb{R}^{K \times K}$ where q_{ij} are the transition rates from regime i to regime j for $i \neq j$, and $q_{ii} = -\sum_{j \neq i} q_{ij}$ so that the rows of G sum to 0.

When Z_t enters regime i, the amount of time it spends in regime i is exponentially distributed with rate $v_i = \sum_{j \neq i} q_{ij}$, and when it leaves regime i it will to go regime j with probability $p_{ij} = \dfrac{q_{ij}}{v_i}$.

From our observations we want to estimate the components of G. The holding time in a given regime i is exponentially distributed with pdf $f(t; v_i) = v_i e^{-v_i t}$. For the fictional events in the timeline given in Figure 1.1, the likelihood function (allowing for repetition of terms) would therefore be:

$$
\begin{aligned}
\mathcal{L}(G) &= (v_i e^{-v_i(\tau_2 - \tau_1)} p_{ij})(v_j e^{-v_j(\tau_3 - \tau_2)} p_{ji})(v_i e^{-v_i(\tau_4 - \tau_3)} p_{ik}) \dots && (2.1) \\
&= \prod_{i=1}^{K} \prod_{i \neq j} (v_i p_{ij})^{N_{ij}(T)} e^{-v_i H_i(T)} && (2.2)
\end{aligned}
$$

$$= \prod_{i=1}^{K} \prod_{i \neq j} (q_{ij})^{N_{ij}(T)} e^{-v_i H_i(T)} \tag{2.3}$$

where:

$$N_{ij}(T) \equiv \text{number of transitions from regime } i \text{ to } j \text{ up to time } T$$
$$H_i(T) \equiv \text{holding time in regime } i \text{ up to time } T$$

So that the log-likelihood becomes:

$$\ln \mathcal{L}(G) = \sum_{i=1}^{K} \sum_{i \neq j} \left[N_{ij}(T) \ln(q_{ij}) - v_i H_i(T) \right] \tag{2.4}$$

$$= \sum_{i=1}^{K} \sum_{i \neq j} \left[N_{ij}(T) \ln(q_{ij}) - \left(\sum_{i \neq k} q_{ik} H_i(T) \right) \right] \tag{2.5}$$

To get a maximum likelihood estimate $\hat{q}_{ij}$ for transition rates and therefore the matrix G, we take the partial derivative of $\ln \mathcal{L}(G)$ and set it equal to zero:

$$\frac{\partial \ln \mathcal{L}(G)}{\partial q_{ij}} = \frac{N_{ij}(T)}{q_{ij}} - H_i(T) = 0 \tag{2.6}$$

$$\Rightarrow \boxed{\hat{q}_{ij} = \frac{N_{ij}(T)}{H_i(T)}} \tag{2.7}$$

2.1.2 Maximum Likelihood Estimation of $\lambda_k^{\pm}$

Now we want to derive an estimate for the intensity of the Poisson process of market order arrivals conditional on being in some regime k. We'll look first at just the market buys for some regime k. In the above timeline, the market order buy arrival times are indexed by b_i. Since we're assuming that the arrival process is Poisson with the same intensity throughout trials, we can consider the inter-arrival time of events conditional on being in regime k. Then the MLE derivation follows just as for the CTMC:

$$\mathcal{L}(\lambda_k^+; b_1, \ldots, b_N) = \prod_{i=2}^{N} \lambda_k^+ e^{-\lambda_k^+ (b_i - b_{i-1})} \tag{2.8}$$

$$= (\lambda_k^+)^{N_k^+(T)} e^{-\lambda_k^+ H_k(T)} \tag{2.9}$$

where:

$$N_k^+(T) \equiv \text{number of market order arrivals in regime } k \text{ up to time } T$$
$$H_k(T) \equiv \text{holding time in regime } k \text{ up to time } T$$

So that the log-likelihood becomes:

$$\ln \mathcal{L}(\lambda_k^+) = N_k^+(T) \ln(\lambda_k^+) - \lambda_k^+ H_k(T) \tag{2.10}$$

And the ML estimate for $\hat{\lambda}_k^+$ is:

$$\frac{\partial \ln \mathcal{L}}{\partial \lambda_k^+} = \frac{N_k^+(T)}{\lambda_k^+} - H_k(T) = 0 \tag{2.11}$$

$$\Rightarrow \boxed{\hat{\lambda}_k^+ = \frac{N_k^+(T)}{H_k(T)}} \tag{2.12}$$

2.2 Modeling $I(t)$: Continuous Time Markov Chain

Instead of modelling imbalance directly, an alternative approach is to discretize imbalance into subintervals (called bins), and model a stochastic process that tracks which bin $I(t)$ falls into. A naive model that can be employed is a continuous-time Markov chain (CTMC).

Let $Z(t)$ be a CTMC taking values in $\{1, \ldots, K\}$, and having infinitesimal generator matrix $\boldsymbol{G}$.[1] Conditional on being in some regime k, the arrival of buy and sell market orders follow independent Poisson processes with intensities $\lambda_k^\pm$ (and are hence Markov-modulated Poisson processes), where λ_k^+ (λ_k^-) is the rate of arrivals of market sells (buys).

Given a set of observations of buy/sell market orders and regime switches, we previously derived

[1]Define the terms $P_{ij}(t) = P\{Z(t) = j | Z(0) = i\}$. Then the matrices $\boldsymbol{P}(t) = \{P_{ij}(t)\}$ and $\boldsymbol{G}$ satisfy $\dot{\boldsymbol{P}}(t) = \boldsymbol{G} \cdot \boldsymbol{P}(t)$, and hence $\boldsymbol{P}(t) = e^{\boldsymbol{G}t}$

a maximum likelihood estimation (MLE) for both the entries of $\boldsymbol{G}$ and the values $\lambda_k^{\pm}$. Where $\boldsymbol{G} = \{q_{ij}\} \in \mathbb{R}^{K \times K}$, the q_{ij} represent the transition rates from bin i to j for $i \neq j$, and $q_{ii} = -\sum_{j \neq i} q_i j$ such that the rows sum to 0. We found that:

$$\hat{q}_{ij} = \frac{N_{ij}(T)}{H_i(T)}$$

where

$$\begin{aligned} N_i j(T) &\equiv \text{number of transitions from bin } i \text{ to } j \text{ up to time } T \\ H_i(T) &\equiv \text{holding time in bin } i \text{ up to time } T \end{aligned}$$

Similarly, for the Poisson process intensities $\lambda_k^{\pm}$, we found:

$$\hat{\lambda}_k^{\pm} = \frac{N_k^{\pm}(T)}{H_k(T)}$$

where

$$\begin{aligned} N_i j(T) &\equiv \text{number of market orders in bin } k \text{ up to time } T \\ H_i(T) &\equiv \text{holding time in bin } k \text{ up to time } T \end{aligned}$$

2.3 Calibrating a CTMC

We estimated parameters for a CTMC on a day's worth of LOB data. Using these parameters, we generated sample paths of the imbalance bins as well as arrival of market orders, and re-estimated parameters along the sample paths. By doing this for 10,000 paths we obtained histograms for the parameters (the individual entries of $\boldsymbol{G}$ as well as the intensities $\lambda_k^{\pm}$).

Using data for `ORCL` from 2013-05-15, averaging imbalances over a 100ms window, and taking the number of bins $K = 3$, we obtained the following mean values for the parameters:

$$\boldsymbol{G} = \begin{pmatrix} -0.112 & 0.098 & 0.0122 \\ 0.099 & -0.21 & 0.111 \\ 0.0115 & 0.112 & -0.1235 \end{pmatrix}$$

$$
\begin{array}{cc} & \begin{array}{ccc} k=1 & k=2 & k=3 \end{array} \\ \boldsymbol{\lambda} = \begin{array}{c} + \\ - \end{array} & \begin{pmatrix} 0.121 & 0.081 & 0.048 \\ 0.0263 & 0.062 & 0.153 \end{pmatrix} \end{array}
$$

2.4 Next Steps

1. Run cross-validation on the old CTMC imbalance model, also varying the averaging time.
2. Check for a unit root in the imbalance time series using the augmented Dickey-Fuller test, after transforming the data using the logit function.
3. Consider a CTMC where the state is actually the pair (I_{k-1}, I_k), with a $k^2 \times k^2$ transition matrix. Cross-validate and compare with regular CTMC.

2.5 Cross-validation of CTMC

To cross-validate the CTMC calibration, the following steps were taken:

1. An imbalance averaging time (in ms) and number of imbalance bins were fixed. The infinitesimal generator matrix $\boldsymbol{G}$ was calculated on the resulting timeseries.
2. An embedded discrete Markov chain transition matrix $\boldsymbol{A}$ was obtained from $\boldsymbol{G}$. This effectively says: conditional on a transition from bin i, what are the transition probabilities to bin j?
3. The stationary distribution, and number (n) of steps required to converge to the stationary distribution, was calculated. That is: for $\epsilon > 0$, calculate n such that $||\boldsymbol{A}^{n+1} - \boldsymbol{A}^n|| < \epsilon$.
4. Find the average number of steps in the timeseries that are required to observe n transitions. This is the size of the timewindow against which to cross-validate.
5. Remove the cross-validation timewindow (call this the "removed series") from the full timeseries (call this the "remaining series"). Calculate two infinitesimal generator matrices $\boldsymbol{G}_{removed}$ and $\boldsymbol{G}_{remaining}$.

6. Calculate two error terms for the resulting matrices:

$$err = \sqrt{\frac{1}{\#trials} \times \sum_{trials} \left(\frac{1}{\#bins^2} \sum_{ij} (\boldsymbol{G}_{remaining}(ij) - \boldsymbol{G}_{removed}(ij))^2 \right) x}$$

$$\boldsymbol{Err} = \sqrt{\frac{1}{\#trials} \times \sum_{trials} (\boldsymbol{1} - \boldsymbol{G}_{removed} \div \boldsymbol{G}_{remaining})^2}$$

where, for $\boldsymbol{Err}$, division and squaring are entry-wise and not matrix-wise.

This is following up on the cross-validation results from last time. In those results, in order to obtain the invariant distribution for the Markov chain, we calculated a transition probability matrix $\boldsymbol{A}$ for the embedded discrete-time Markov chain and took matrix powers $\boldsymbol{A}^n$ until it converged, and then observed the average number of timesteps that it took to see n transitions in the data.

In these results, we instead use the relationship $\dot{\boldsymbol{P}}(t) = \boldsymbol{P}(t)\boldsymbol{G} \;\Rightarrow\; \boldsymbol{P}(t) = e^{t\boldsymbol{G}}$. Thus we calculate the invariant distribution using the averaging time Δt and the number of such timesteps n and observe when $e^{\Delta t \boldsymbol{G} n}$ converges. This value n immediately tells us the timewindow size to remove for cross-validation.

Table 2.1 New results, convergence threshold 1e-05

# bins averaging time	stationary n	Timewindow size	err	$\boldsymbol{Err}$
3 bins, 100ms	478	47.8s (0.2% of series)	0.356402	644% - 11371%
3 bins, 500ms	144	72s (0.3% of series)	0.087631	236% - 985%
3 bins, 1000ms	89	89s (0.4% of series)	0.050605	150% - 480%
3 bins, 2000ms	57	114s (0.5% of series)	0.032076	122% - 725%
3 bins, 3000ms	45	135s (0.6% of series)	0.023662	98% - 552%
3 bins, 5000ms	35	175s (0.75% of series)	0.014182	70% - 514%
3 bins, 10000ms	29	290s (1.2% of series)	0.007361	52% - 496%
3 bins, 20000ms	22	440s (1.9% of series)	0.004447	43% - 1698%
5 bins, 100ms	546	54.6s (0.2% of series)	0.162690	452% - 6785%
5 bins, 500ms	162	81s (0.3% of series)	0.046204	187% - 2590%
5 bins, 1000ms	100	100s (0.4% of series)	0.029900	136% - 2962%
5 bins, 2000ms	65	130s (0.6% of series)	0.017340	86% - 2141%
5 bins, 3000ms	52	156s (0.7% of series)	0.012505	87% - Inf%
5 bins, 5000ms	42	210s (0.9% of series)	0.008035	66% - 978%
5 bins, 10000ms	31	310s (1.3% of series)	0.004563	45% - Inf%
5 bins, 20000ms	25	500s (2.1% of series)	0.002485	42% - Inf%

The large errors seen in the error matrix $\boldsymbol{Err}$ are attributable to the corner elements: in the case of 3 bins, this would be G_{13} and G_{31}. Or, for example, the error matrices for 5 bins at 100ms and at 20000ms looked like:

$$\boldsymbol{Err}_{100ms} = \begin{bmatrix} 6.86 & 8.48 & 5.92 & 9.68 & 11.02 \\ 7.57 & 6.82 & 8.80 & 67.58 & 8.31 \\ 6.33 & 5.08 & 4.52 & 8.55 & 16.79 \\ 14.64 & 54.50 & 8.12 & 6.41 & 7.77 \\ 6.82 & 36.76 & 5.47 & 5.86 & 5.04 \end{bmatrix}$$

$$\boldsymbol{Err}_{20000ms} = \begin{bmatrix} 0.79 & 0.99 & 3.63 & 20.23 & Inf \\ 1.10 & 0.44 & 0.82 & 1.36 & NaN \\ 2.07 & 0.64 & 0.42 & 0.88 & 3.83 \\ 3.64 & 1.66 & 0.85 & 0.57 & 2.81 \\ NaN & Inf & 1.42 & 1.08 & 0.87 \end{bmatrix}$$

2.6 2-dimensional CTMC

Next we considered a CTMC that tracks not only the imbalance bin, but jointly the imbalance bin and the price change over a subsequent interval. That is to say, the CTMC modelled the joint distribution $(I(t), \Delta S(t))$ where $I(t)$ is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and $\Delta S(t) = \text{sgn}(S(t+\Delta t_S) - S(t))$, considered individually for the best bid and best ask prices. For 3 bins, this was encoded into one dimension $Z(t)$ as follows:

$Z(t)$	Bin $I(t)$	$\Delta S(t)$
1	Bin 1	< 0
2	Bin 2	< 0
3	Bin 3	< 0
4	Bin 1	0
5	Bin 2	0
6	Bin 3	0
7	Bin 1	> 0
8	Bin 2	> 0
9	Bin 3	> 0

Here bid and ask prices were considered separately rather than considering the change in mid price. Calibrating a CTMC on the two resulting timeseries $Z_{bid}(t)$ and $Z_{ask}(t)$ yielded some interesting results:

imbalance Δt_I: 1000ms, price Δt_S: 500ms

$$\boldsymbol{G}_{Z_{bid}} = \begin{bmatrix} -0.9928 & 0.0217 & 0 & 0.2826 & 0.5870 & 0.0870 & 0 & 0.0145 & 0 \\ 0.0118 & -0.9647 & 0 & 0.1412 & 0.5882 & 0.2000 & 0 & 0.0118 & 0.0118 \\ 0 & 0.0909 & -1.0000 & 0 & 0.3636 & 0.5455 & 0 & 0 & 0 \\ 0.0146 & 0.0005 & 0 & -0.0792 & 0.0562 & 0.0034 & 0.0036 & 0.0006 & 0.0003 \\ 0.0016 & 0.0052 & 0.0003 & 0.0435 & -0.0897 & 0.0300 & 0 & 0.0080 & 0.0011 \\ 0.0003 & 0.0025 & 0.0022 & 0.0053 & 0.0919 & -0.1277 & 0 & 0.0017 & 0.0237 \\ 0 & 0.0345 & 0 & 0.4138 & 0.4138 & 0.1034 & -1.0000 & 0.0345 & 0 \\ 0.0179 & 0.0179 & 0 & 0.2232 & 0.5536 & 0.1250 & 0.0089 & -0.9732 & 0.0268 \\ 0.0094 & 0.0189 & 0 & 0.1132 & 0.5189 & 0.3113 & 0 & 0.0094 & -0.9811 \end{bmatrix}$$

$$\boldsymbol{G}_{Z_{ask}} = \begin{bmatrix} -0.9915 & 0.0169 & 0 & 0.2881 & 0.5678 & 0.1017 & 0 & 0.0169 & 0 \\ 0.0106 & -0.9681 & 0 & 0.1277 & 0.5638 & 0.2340 & 0 & 0.0213 & 0.0106 \\ 0 & 0.0588 & -1.0000 & 0 & 0.2941 & 0.5882 & 0 & 0 & 0.0588 \\ 0.0121 & 0.0005 & 0 & -0.0775 & 0.0580 & 0.0034 & 0.0027 & 0.0005 & 0.0003 \\ 0.0016 & 0.0058 & 0.0002 & 0.0448 & -0.0898 & 0.0297 & 0 & 0.0065 & 0.0011 \\ 0.0003 & 0.0025 & 0.0039 & 0.0059 & 0.0907 & -0.1311 & 0 & 0.0008 & 0.0270 \\ 0 & 0.0476 & 0 & 0.1905 & 0.5714 & 0.1429 & -1.0000 & 0.0476 & 0 \\ 0 & 0.0440 & 0 & 0.1319 & 0.6374 & 0.1429 & 0 & -0.9890 & 0.0330 \\ 0.0085 & 0.0254 & 0.0085 & 0.0847 & 0.5169 & 0.3220 & 0 & 0.0169 & -0.9831 \end{bmatrix}$$

Using these matrices, we can compute conditional probabilities. For example, we can ask: conditional on being in bin 1 (more bid volume than ask) and on the bid price changing, what is the probability that the change will be greater than 0? less than 0?

Again, converting the generator matrix to the embedded discrete time Markov chain matrix proves enlightening for these calculations:

$$
\boldsymbol{A}_{Z_{bid}} = \begin{bmatrix}
0 & 0.0219 & 0 & 0.2847 & 0.5912 & 0.0876 & 0 & 0.0146 & 0 \\
0.0122 & 0 & 0 & 0.1463 & 0.6098 & 0.2073 & 0 & 0.0122 & 0.0122 \\
0 & 0.0909 & 0 & 0 & 0.3636 & 0.5455 & 0 & 0 & 0 \\
0.1839 & 0.0065 & 0 & 0 & 0.7097 & 0.0435 & 0.0452 & 0.0081 & 0.0032 \\
0.0174 & 0.0581 & 0.0029 & 0.4855 & 0 & 0.3343 & 0 & 0.0891 & 0.0126 \\
0.0022 & 0.0197 & 0.0175 & 0.0416 & 0.7199 & 0 & 0 & 0.0131 & 0.1860 \\
0 & 0.0345 & 0 & 0.4138 & 0.4138 & 0.1034 & 0 & 0.0345 & 0 \\
0.0183 & 0.0183 & 0 & 0.2294 & 0.5688 & 0.1284 & 0.0092 & 0 & 0.0275 \\
0.0096 & 0.0192 & 0 & 0.1154 & 0.5288 & 0.3173 & 0 & 0.0096 & 0
\end{bmatrix}
$$

Generator matrices $\boldsymbol{G}_{bid}$ and $\boldsymbol{G}_{ask}$ were estimated for the resulting timeseries. These were converted to one-step probability matrices $\boldsymbol{P}_{bid}$ and $\boldsymbol{P}_{ask}$ using the formula $\boldsymbol{P} = e\boldsymbol{G}\Delta t$, where Δt is the imbalance averaging period. What this matrix encodes are the conditional one-step transition probabilities - for each entry $\boldsymbol{P}_{ij}$ we have:

$$
\begin{aligned}
\boldsymbol{P}_{ij} &= \mathbb{P}\left[Z_n \in j \mid Z_{n-1} \in i\right] \\
&= \mathbb{P}\left[(\rho_n, \Delta S_n) \in j \mid (\rho_{n-1}, \Delta S_{n-1}) \in i\right]
\end{aligned}
$$

The aim is to use these $\boldsymbol{P}$ matrices to compute conditional probabilities of price changes. For example, we can ask: if we are currently in imbalance bin 1, and previous were also in bin 1 and saw a negative price change, what is the probability of again seeing a negative price change?

Since each state $(\rho_n, \Delta S_n) \in j$ is actually comprised of two states, say $\rho_n \in k, \Delta S_n \in m$, we can re-write these entries of $\boldsymbol{P}$ as being:

$$
\begin{aligned}
&\mathbb{P}\left[\rho_n \in i, \Delta S_n \in j \mid \rho_{n-1} \in k, \Delta S_{n-1} \in m\right] \\
&= \mathbb{P}\left[\rho_n \in i, \Delta S_n \in j \mid B\right]
\end{aligned}
$$

where we're using the shorthand $B = (\rho_{n-1} \in k, \Delta S_{n-1} \in m)$ to represent the states in the previous timestep. Using Bayes' Rule, we can write:

$$
\mathbb{P}\left[\Delta S_n \in j \mid B, \rho_n \in i\right] = \frac{\mathbb{P}\left[\rho_n \in i, \Delta S_n \in j \mid B\right]}{\mathbb{P}\left[\rho_n \in i \mid B\right]}
$$

The left-hand-side value is exactly the conditional probability in price change that we're interested in finding, the numerator is each individual entry of the one-step probability matrix $\boldsymbol{P}$, and the denominator can be computed as:

$$\mathbb{P}\left[\rho_n \in i \mid B\right] = \sum_j \mathbb{P}\left[\rho_n \in i, \Delta S_n \in j \mid B\right]$$

Using 3 bins, 1000ms imbalance averaging, and 500ms price change, we computed $\boldsymbol{P}_{bid}$:

$$\begin{array}{r} \Delta S_n<0\rightarrow \\ \Delta S_n=0\rightarrow \\ \Delta S_n>0\rightarrow \end{array} \left[\overbrace{\underbrace{\begin{matrix} .67 & .05 & .04 \\ .33 & .95 & .96 \\ .00 & .00 & .00 \end{matrix}}_{\Delta S_{n-1}<0} \underbrace{\begin{matrix} .01 & .03 & .04 \\ .99 & .97 & .96 \\ .00 & .00 & .00 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .05 & .05 \\ .41 & .93 & .95 \\ .58 & .02 & .00 \end{matrix}}_{\Delta S_{n-1}>0}}^{\rho_n=1} \overbrace{\underbrace{\begin{matrix} .02 & .50 & .12 \\ .96 & .49 & .87 \\ .02 & .01 & .00 \end{matrix}}_{} \underbrace{\begin{matrix} .01 & .00 & .02 \\ .98 & .99 & .97 \\ .01 & .01 & .01 \end{matrix}}_{\Delta S_{n-1}=0} \underbrace{\begin{matrix} .05 & .01 & .02 \\ .91 & .48 & .96 \\ .05 & .51 & .01 \end{matrix}}_{}}^{\rho_n=2} \overbrace{\underbrace{\begin{matrix} .00 & .00 & .52 \\ .98 & .95 & .47 \\ .02 & .04 & .01 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .01 & .00 \\ .95 & .96 & .93 \\ .05 & .03 & .02 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .00 & .00 \\ .98 & .88 & .34 \\ .02 & .12 & .66 \end{matrix}}_{}}^{\rho_n=3} \right]$$

2.7 In-Sample Backtesting of Naive Trading Strategies

As a refresher:

We are a considering a CTMC for the joint distribution $(I(t), \Delta S(t))$ where $I(t) \in \{1, 2, \ldots, \#_{bins}\}$ is the bin corresponding to imbalance averaged over the interval $[t - \Delta t_I, t]$, and $\Delta S(t) = \text{sign}(S(t + \Delta t_S) - S(t)) \in \{-1, 0, 1\}$, considered individually for the best bid and best ask prices. The pair $(I(t), \Delta S(t))$ was then reduced into one dimension with a simple encoding.

From the resulting timeseries we estimated a generator matrix $\boldsymbol{G}$ and used it to obtain a one-step transition probability matrix $\boldsymbol{P} = e^{\boldsymbol{G}\Delta t_I}$. The entries of $\boldsymbol{P}$ contain the conditional probabilities $\mathbb{P}\left[\rho_{curr}, \Delta S_{curr} \mid \rho_{prev}, \Delta S_{prev}\right]$, from which we can solve for the probability of now seeing a given price change (ΔS_{curr}) conditional on the current imbalance, the previous imbalance, and the previous price change.

For example, one such conditional probability matrix $\boldsymbol{P_C}$ (using 3 imbalance bins) was:

$$\begin{array}{r} \Delta S_n<0\rightarrow \\ \Delta S_n=0\rightarrow \\ \Delta S_n>0\rightarrow \end{array} \left[\overbrace{\underbrace{\begin{matrix} .67 & .05 & .04 \\ .33 & .95 & .96 \\ .00 & .00 & .00 \end{matrix}}_{\Delta S_{n-1}<0} \underbrace{\begin{matrix} .01 & .03 & .04 \\ .99 & .97 & .96 \\ .00 & .00 & .00 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .05 & .05 \\ .41 & .93 & .95 \\ .58 & .02 & .00 \end{matrix}}_{\Delta S_{n-1}>0}}^{\rho_n=1} \overbrace{\underbrace{\begin{matrix} .02 & .50 & .12 \\ .96 & .49 & .87 \\ .02 & .01 & .00 \end{matrix}}_{} \underbrace{\begin{matrix} .01 & .00 & .02 \\ .98 & .99 & .97 \\ .01 & .01 & .01 \end{matrix}}_{\Delta S_{n-1}=0} \underbrace{\begin{matrix} .05 & .01 & .02 \\ .91 & .48 & .96 \\ .05 & .51 & .01 \end{matrix}}_{}}^{\rho_n=2} \overbrace{\underbrace{\begin{matrix} .00 & .00 & .52 \\ .98 & .95 & .47 \\ .02 & .04 & .01 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .01 & .00 \\ .95 & .96 & .93 \\ .05 & .03 & .02 \end{matrix}}_{} \underbrace{\begin{matrix} .00 & .00 & .00 \\ .98 & .88 & .34 \\ .02 & .12 & .66 \end{matrix}}_{}}^{\rho_n=3} \right]$$

Immediately evident from $\boldsymbol{P_C}$ is that in most cases we are expecting no price change. In fact, the only cases in which the probability of a price change is > 0.5 show evidence of *momentum*; for example, the way to interpret the value in row 1, column 1 is: if $\rho_{prev} = \rho_{curr} = 1$ and previously we saw a downward price change, then we expect to again see a downward price change. In fact, the best way to summarize the matrix is:

$$\mathbb{P}\left[\Delta S_{curr} = \Delta S_{prev} \mid \rho_{prev} = \rho_{curr}\right] > 0.5$$

Algorithm 1 Naive Trading Strategy

$cash = 0$
$asset = 0$
for $t = 2$: `length`$(timeseries)$ **do**
    if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{+}= data.BuyPrice(t)$
        $asset \mathrel{-}= 1$
    else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{-}= data.SellPrice(t)$
        $asset \mathrel{+}= 1$
    end if
end for
if $asset > 0$ **then**
    $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
    $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

We backtested a number of naive trading strategies, outlined here, based on this key observation. In plain terms, the Naive trading strategies can be interpreted as follows:

Naive Trading Strategy Using the conditional probabilities obtained from $\boldsymbol{P_C}$, we will execute a buy (resp. sell) market order if the probability of an upward (resp. downward) price change is > 0.5.

Naive+ Trading Strategy Extending the naive trading strategy, if we anticipate no change then we'll additionally keep limited orders posted at the touch, front of the queue. We'll track MO arrival, assume we always get executed, and immediately repost the limit orders.

Naive++ Trading Strategy We won't execute market orders or keep limit orders at the touch. Using the conditional probabilities obtained from $\boldsymbol{P_C}$, if we expect a downward (resp. upward) price change then we'll add a limit order to the sell (resp. buy) side, and hopefully pick up an agent who is executing a market order going against the price change momentum.

Algorithm 2 Naive+ Trading Strategy

$cash = 0$
$asset = 0$
$LO_{posted} =$ `False`
for $t = 2\ :\ \texttt{length}(timeseries)$ **do**
    if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{+}= data.BuyPrice(t)$
        $asset \mathrel{-}= 1$
        $LO_{posted} =$ `False`
    else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $cash \mathrel{-}= data.SellPrice(t)$
        $asset \mathrel{+}= 1$
        $LO_{posted} =$ `False`
    else if $\mathbb{P}\left[\Delta S_{curr} = 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
        $LO_{posted} =$ `True`
    end if
    if LO_{posted} **then**
        for $MO \in ArrivedMarketOrders(t, t+1)$ **do**
            if $MO == Sell$ **then**
                $cash \mathrel{-}= data.BuyPrice(t)$
                $asset \mathrel{+}= 1$
            else if $MO == Buy$ **then**
                $cash \mathrel{+}= data.SellPrice(t)$
                $asset \mathrel{-}= 1$
            end if
        end for
    end if
end for
if $asset > 0$ **then**
    $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
    $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

Algorithm 3 Naive++ Trading Strategy

$cash = 0$
$asset = 0$
$LOBuy_{posted} =$ `False`
$LOSell_{posted} =$ `False`
for $t = 2$: `length`$(timeseries)$ **do**
  if $\mathbb{P}\left[\Delta S_{curr} < 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
    $LOBuy_{posted} =$ `False`
    $LOSell_{posted} =$ `True`
  else if $\mathbb{P}\left[\Delta S_{curr} > 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
    $LOBuy_{posted} =$ `True`
    $LOSell_{posted} =$ `False`
  else if $\mathbb{P}\left[\Delta S_{curr} = 0 \mid \rho_{curr}, \rho_{prev}, \Delta S_{prev}\right] > 0.5$ **then**
    $LOBuy_{posted} =$ `False`
    $LOSell_{posted} =$ `False`
  end if
  for $MO \in ArrivedMarketOrders(t, t+1)$ **do**
    if $MO == Sell \ \wedge \ LOBuy_{posted}$ **then**
      $cash \mathrel{-}= data.BuyPrice(t)$
      $asset \mathrel{+}= 1$
    else if $MO == Buy \ \wedge \ LOSell_{posted}$ **then**
      $cash \mathrel{+}= data.SellPrice(t)$
      $asset \mathrel{-}= 1$
    end if
  end for
end for
if $asset > 0$ **then**
  $cash \mathrel{+}= asset \times data.BuyPrice(t)$
else if $asset < 0$ **then**
  $cash \mathrel{+}= asset \times data.SellPrice(t)$
end if

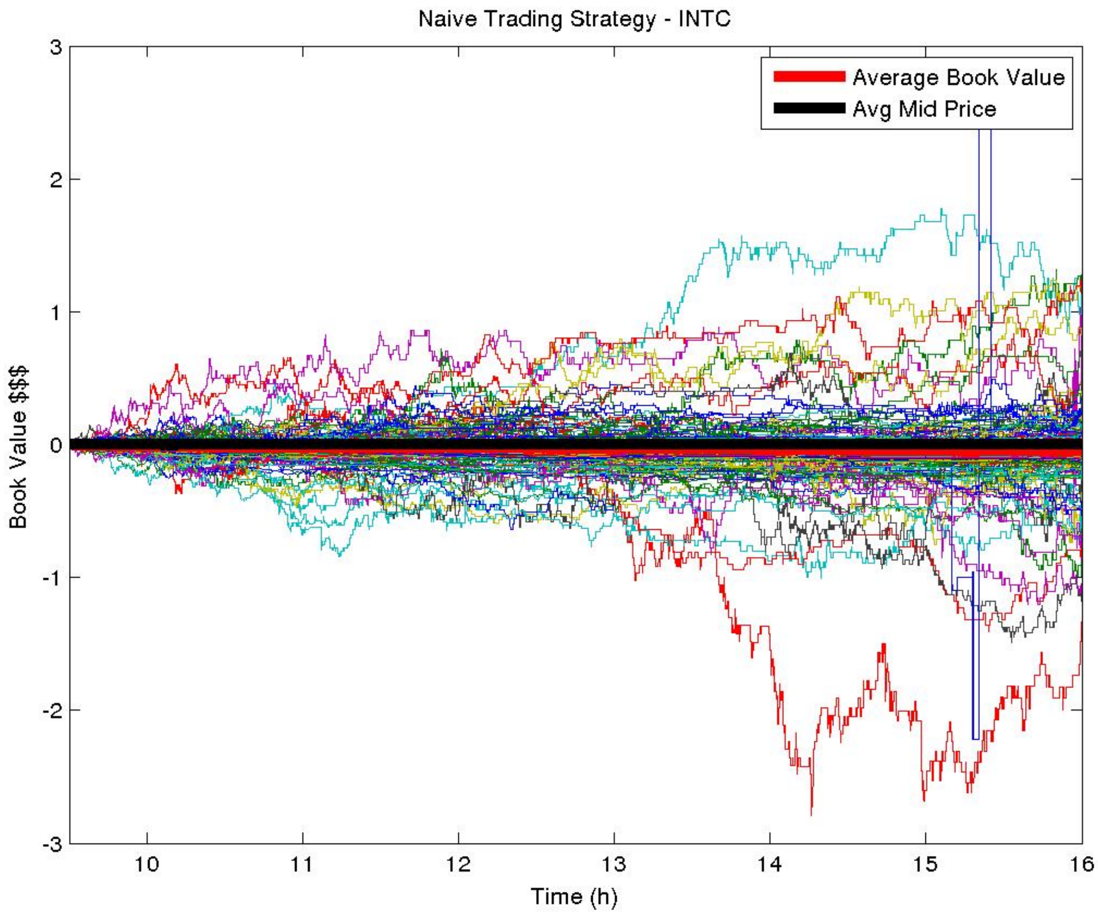


Figure 2.1: INTC: Book value against time of trading day.

Naive- Trading Strategy We additionally considered a trading strategy, for benchmark purposes, which used only current imbalance to predict future price change. But actually this predicted $\mathbb{P}[\Delta S_{curr} = 0] > 0.5$ at all times, so we could not run a strategy off it.

Backtesting these trading strategies required a choice of parameters for Δt_S, the price change observation period, Δt_I, the imbalance averaging period, and $\#_{bins}$, the number of imbalance bins. Through a brute force calibration technique we found that $\#_{bins} = 4$ provided the highest expected number of successful trades for most tickers, so this was chosen as a constant. Similarly, we empirically saw that calibration always yielded $\Delta t_S = \Delta t_I$, so this was taken as a given. Then each backtest consisted of first calibrating the value Δt_I from one day of data by maximizing the intra-day Sharpe ratio, then using the calibrated parameters to backtest the entire year.

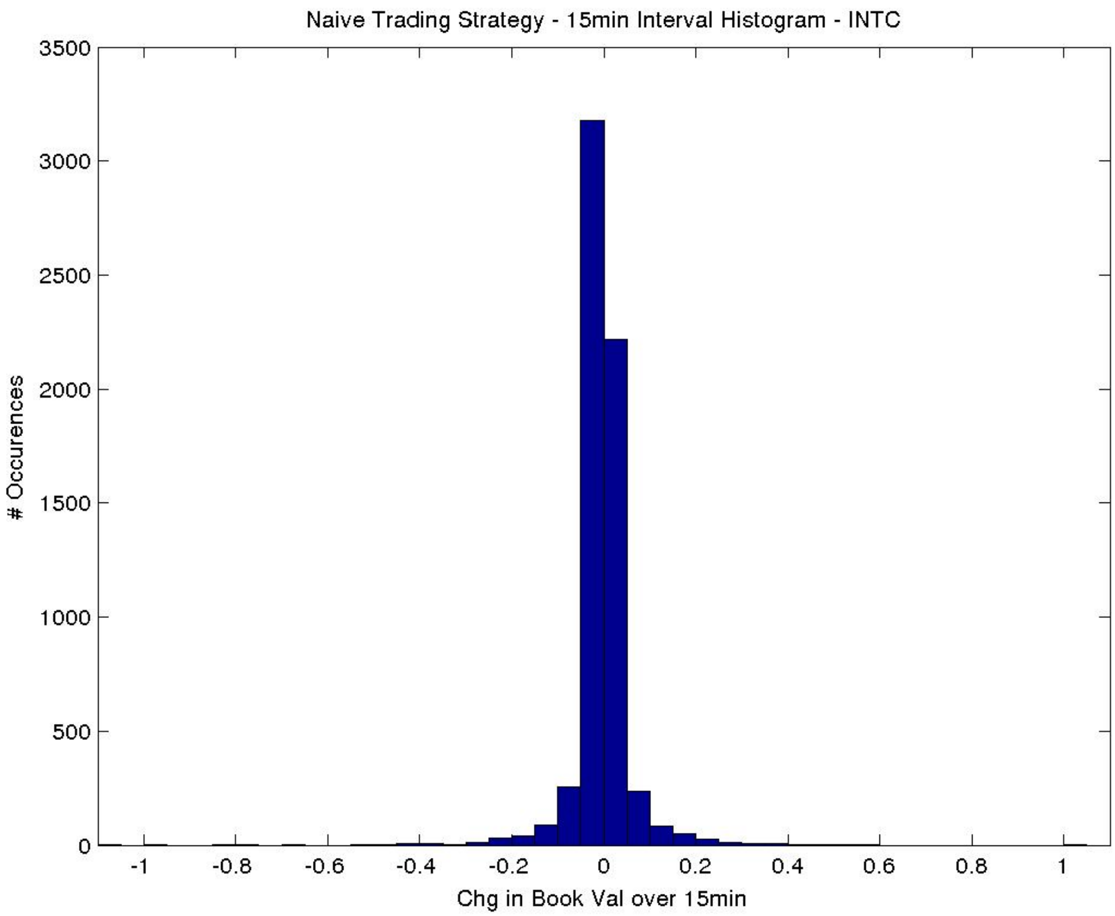


Figure 2.2: INTC: Histogram of 15min book value changes.

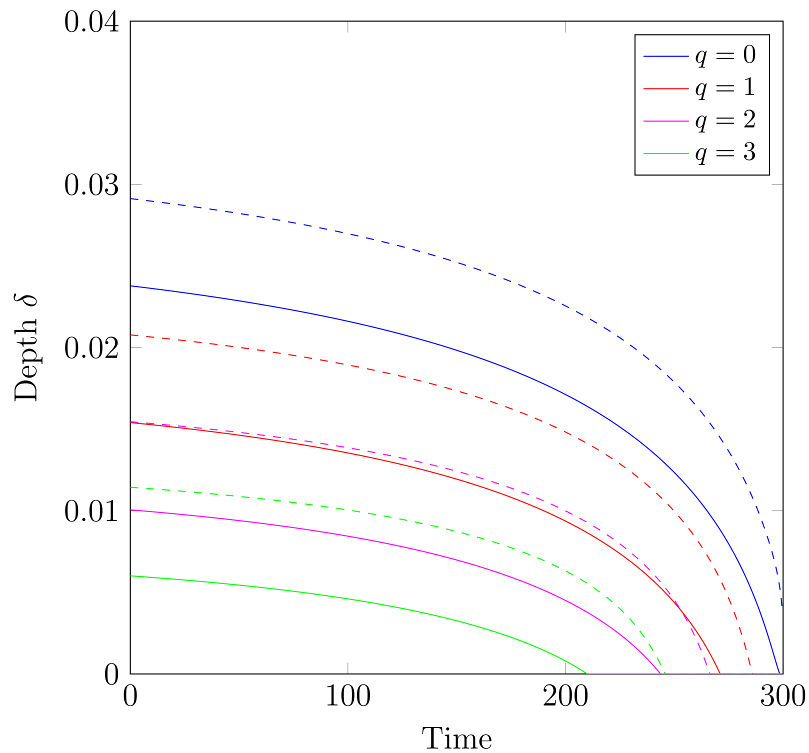
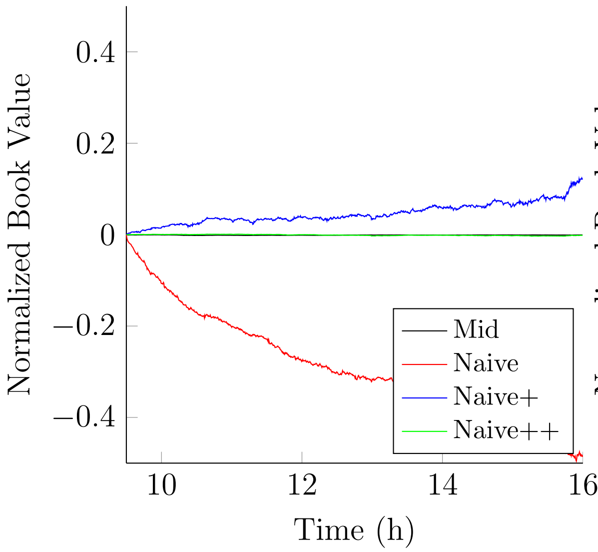


Figure 2.3: Optimal posting depth δ

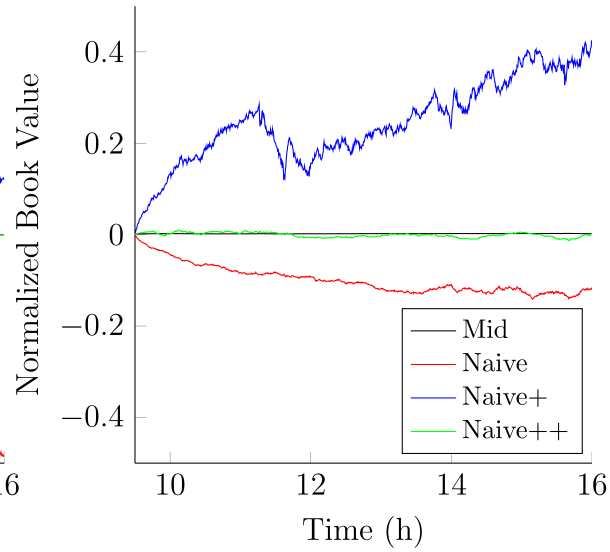
2.8 Conclusions from Naive Trading Strategies

To properly compare the Naive trading strategies, it must be understood that the Naive+ strategy has the Naive built into it - thus it's actually the difference between the two that needs to be assessed to ascertain the effect of posting Limit Orders when no price change is predicted. As seen in Figure 2.4, the Naive trading strategy on average underperformed the average mid price, while the Naive+ (adding at-the-touch limit orders when no change was predicted) and Naive++ (adding limit orders to adversely selecting agents that traded against the price change momentum) strategies both on average generated revenue.

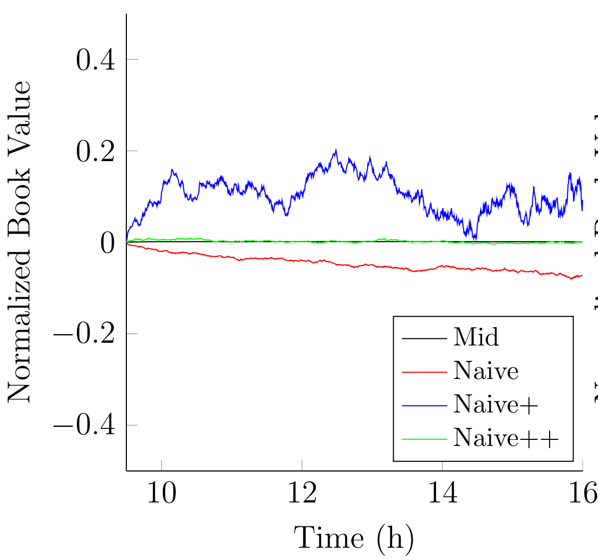
Question 1 Why is the Naive strategy producing, on average, normalized losses? Especially so when considering that we are in-sample backtesting. On calibration, we see that our intra-day Sharpe ratio is around 0.01 or 0.02 when we choose our optimal parameters, so at the very least on the calibration date the strategy produces positive returns. The remainder of the calendar days are out-of-sample, as the parameters are (likely) not optimal. This suggests non-stationary data, and in particular not every day can be modelled by the same Markov chain. The problem may be exaggerated by the fact that we're calibrating on the first trading day of the calendar



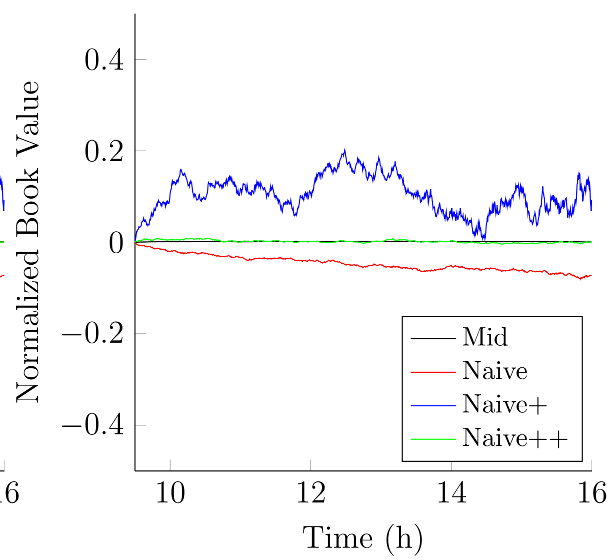
(a) FARO



(b) NTAP



(c) ORCL



(d) AAPL

Figure 2.4: Comparison of Naive (red), Naive+ (blue), and Naive++ (green) trading strategies, with benchmark Midprice (black). Plotted are book values against time of trading day, averaged across trading year.

year, when we might expect reduced, or at least non-representative, trading activity. Further, we're currently obtaining the $\boldsymbol{P_C}$ probability matrix using only bid-side data, not sell-side or mid, and we're ignoring the bid-ask spread. Thus predicting a "price change" may be insufficient when considering a monetizable opportunity, as we won't be able to profit off a predicted increase followed by a predicted decrease unless the interim mid-price move is greater than the bid-ask spread (assuming constant spread). This suggests a potential straightforward modification to the strategy.

Question 2 Why do the Naive+ and ++ strategies outperform the Naive strategy? This is particularly interesting since the probabilities are being obtained from the same matrix. The obvious difference between the successful and unsuccessful strategies is that the former (a) uses limit orders, and (b) executes when we predict a zero change, whereas the latter uses (a) market orders, and (b) executes when we do predict non-zero change.

(a) obviously leads to a different transaction price being used: if I buy with a LO I'm paying the bid price, whereas buying with a MO I pay the ask price. If I value the stock using the mid price, and the mid price doesn't move as a result of my transaction, then with LO I'm buying the asset for less than I'm valuing it at, and with MO I'm paying more than its value.

(b) seems to be the largest flaw in the Naive strategy, to which there are two factors. One, we are not predicting the magnitude of the price change, only whether it is zero or non-zero. Two, from the probabilities presented above, *we will only predict a price change if we've already seen a price change*. Thus we're effectively reacting too late.

Here's how this works adversely. Suppose a stock has bid/ask quotes of \$9.99/\$10.01, for a bid-ask spread of \$0.02 and a mid of \$10.

1. Imbalance = 1 (pressure for upward price move). $[NPV = 0]$
2. Bid/ask goes up to \$10.00/\$10.02. $[NPV = 0]$
3. Imbalance = 1. We predict another > 0 price change. $[NPV = 0]$
4. We buy 1 share (at \$10.02). $[NPV = -0.01]$
5. Bid/ask goes up to \$10.01/\$10.03. $[NPV = 0]$
6. Imbalance = -1 (pressure for a downward move). $[NPV = 0]$

7. Bid/ask goes down to \$10.00/\$10.02. [$NPV = -0.01$]

8. Imbalance = -1. We predict another < 0 price change.

9. We sell 1 share (at \$10.00). [$NPV = -0.02$]

10. Bid/ask goes down to \$9.99/\$10.01. [$NPV = -0.02$]

In this example the price goes up and back down by two cents to return to where it started, and in the process we lost \$0.02. Now imagine what happens if we price goes up by one cent, up by one cent, then down by ten cents, down by one cent. In this case we lose \$0.11. We're unable to predict that initial upward or downward price change, and only react to it.

Ideas to Explore and Next Steps

- Model the mid price instead of the bid or ask, hold the bid-ask spread as a constant (average observed), and predict price changes at least as great as the spread, instead of simply non-zero.
- Calculate imbalance using a weighted average of the best n bid (resp. ask) prices. This may reduce noise in the signal, have an effect on the size of the imbalance averaging window, and be a stronger predictor.
- Transition to exploring the stochastic control problem.

Chapter 3

Stochastic Optimal Control

Hello and welcome to this limited edition chapter on stochastic optimal control. Read on if you dare subject yourself to the infinite wisdom contained herein.

3.1 Continuous Time

Below we list the processes involved in the optimization problem:

Imbalance & Midprice Change	$\boldsymbol{Z}_t = (\rho_t, \Delta_t)$	CTMC with generator G
Imbalance	$\rho_t = \boldsymbol{Z}_t^{(1)}$	LOB imbalance at time t
Midprice	S_t	evolves according to CTMC
Midprice Change	$\Delta_t = \boldsymbol{Z}_t^{(2)} = S_t - S_{t-s}$	s a pre-determined interval
Bid-Ask half-spread	ξ_t	constant?
LOB Shuffling	N_t	Poisson with rate $\lambda(\boldsymbol{Z}_t)$
ΔPrice: LOB shuffled	$\{\eta_{0,z}, \eta_{1,z}, \dots\} \sim F_z$	i.i.d. with $z = (k, l)$, where $k \in \{\#\text{bins}\},\ l \in \{\Delta\$\}$
Other Agent MOs	$K_t^{\pm}$	Poisson with rate $\mu^{\pm}(\boldsymbol{Z}_t)$
LO posted depth	$\delta_t^{\pm}$	our $\mathcal{F}$-predictable controlled processes
Our LO fill count	$L_t^{\pm}$	$\mathcal{F}$-predictable, non-Poisson
Our MOs	$M_t^{\pm}$	our controlled counting process
Our MO execution times	$\boldsymbol{\tau}^{\pm} = \{\tau_k^{\pm} : k = 1, \dots\}$	increasing sequence of $\mathcal{F}$-stopping times
Cash	$X_t^{\boldsymbol{\tau},\delta}$	depends on our processes M and δ
Inventory	$Q_t^{\boldsymbol{\tau},\delta}$	depends on our processes M and δ

$L_t^{\pm}$ are counting processes (not Poisson) satisfying the relationship that if at time t we have a sell limit order posted at a depth δ_t^-, then our fill probability is $e^{-\kappa\delta_t^-}$ conditional on a buy market order arriving; namely:

$$\mathbb{P}[\,\mathrm{d}L_t^- = 1 \,|\, \mathrm{d}K_t^+ = 1] = e^{-\kappa\delta_t^-} \tag{3.1}$$
$$\mathbb{P}[\,\mathrm{d}L_t^+ = 1 \,|\, \mathrm{d}K_t^- = 1] = e^{-\kappa\delta_t^+} \tag{3.2}$$

The midprice S_t evolves according to the Markov chain and hence is Poisson with rate λ and jump size η, both of which depend on the state of the Markov chain. This Poisson process is all-inclusive in the sense that it accounts for any midprice change, be it from executions, cancellations, or order modifications with the LOB. Thus, the stock midprice S_t evolves according to the SDE:

$$\mathrm{d}S_t = \eta_{N_{t^-}, Z_{t^-}}\,\mathrm{d}N_t \tag{3.3}$$

and additionally satisfies:

$$S_t = S_{t_0} + \int_{t_0+s}^{t} \Delta_u \,\mathrm{d}u \tag{3.4}$$

In executing market orders, we assume that the size of the MOs is small enough to achieve the best bid/ask price, and not walk the book. Hence, our cash process evolves according to:

$$\begin{aligned}\mathrm{d}X_t^{\boldsymbol{\tau},\delta} = &\underbrace{(S_t + \xi_t + \delta_t^-)\,\mathrm{d}L_t^-}_{\text{sell limit order}} - \underbrace{(S_t - \xi_t - \delta_t^+)\,\mathrm{d}L_t^+}_{\text{buy limit order}} \\ &+ \underbrace{(S_t - \xi_t)\,\mathrm{d}M_t^-}_{\text{sell market order}} - \underbrace{(S_t + \xi_t)\,\mathrm{d}M_t^+}_{\text{buy market order}}\end{aligned} \tag{3.5}$$

Based on our execution of limit and market orders, our inventory satisfies:

$$Q_0^{\boldsymbol{\tau},\delta} = 0, \qquad Q_t^{\boldsymbol{\tau},\delta} = L_t^+ + M_t^+ - L_t^- - M_t^- \tag{3.6}$$

We define a new variable for our net present value (NPV) at time t, call it $W_t^{\boldsymbol{\tau},\delta}$, and hence $W_T^{\boldsymbol{\tau},\delta}$ at terminal time T is our 'terminal wealth'. In algorithmic trading, we want to finish the trading day with zero inventory, and assume that at the terminal time T we will submit a market order (of a possibly large volume) to liquidate remaining stock. Here we do not assume that we can receive the best bid/ask price - instead, the price achieved will be $(S - \mathrm{sgn}(Q)\xi - \alpha Q)$, where

$\operatorname{sgn}(Q)\xi$ represents crossing the spread in the direction of trading, and αQ represents receiving a worse price linearly in Q due to walking the book. Hence, $W_t^{\boldsymbol{\tau},\delta}$ satisfies:

$$W_t^{\boldsymbol{\tau},\delta} = \underbrace{X_t^{\boldsymbol{\tau},\delta}}_{\text{cash}} + \underbrace{Q_t^{\boldsymbol{\tau},\delta}\left(S_t - \operatorname{sgn}(Q_t^{\boldsymbol{\tau},\delta})\xi_t\right)}_{\text{book value of assets}} - \underbrace{\alpha\left(Q_t^{\boldsymbol{\tau},\delta}\right)^2}_{\text{liquidation penalty}} \tag{3.7}$$

The set of admissible trading strategies is the product of the sets $\mathcal{T}$, the set of all $\mathcal{F}$-stopping times, and $\mathcal{A}$, the set of all $\mathcal{F}$-predictable, bounded-from-below depths δ. We only consider $\delta^{\pm} \geq 0$, since at $\delta = 0$ our fill probability is $e^{-\kappa\delta} = 1$, so we cannot increase the chance of our limit order being filled by posting any lower than at-the-touch; doing so would only diminish our profit.

For deriving an optimal trading strategy via dynamic programming, I will consider the performance criteria that maximizes terminal wealth. With the above notation, the performance criteria function can be written

$$H^{\boldsymbol{\tau},\delta}(t,x,s,\boldsymbol{z},q) = \mathbb{E}\left[W_T^{\boldsymbol{\tau},\delta}\right] \tag{3.8}$$

And the value function, in turn, is given by

$$H(t,x,s,\boldsymbol{z},q) = \sup_{\boldsymbol{\tau}\in\mathcal{T}_{[t,T]}} \sup_{\delta\in\mathcal{A}_{[t,T]}} H^{\boldsymbol{\tau},\delta}(t,x,s,\boldsymbol{z},q) \tag{3.9}$$

The following theorems establish the dynamic programming method we will utilize to solve this type of problem:

Theorem 1 ([Cartea et al., 2015])***. Dynamic Programming Principle for Optimal Stopping and Control.*** *If an agent's performance criteria for a given admissible control $\boldsymbol{u}$ and admissible stopping time τ are given by*

$$H^{\tau,\boldsymbol{u}}(t,\boldsymbol{x}) = \mathbb{E}_{t,\boldsymbol{x}}[G(X_\tau^{\boldsymbol{u}})]$$

and the value function is

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

then the value function satisfies the Dynamic Programming Principle

$$H(t,\boldsymbol{x}) = \sup_{\tau\in\mathcal{T}_{[t,T]}} \sup_{\boldsymbol{u}\in\mathcal{A}_{[t,T]}} \mathbb{E}_{t,\boldsymbol{x}}\left[G(X_\tau^{\boldsymbol{u}})\mathbb{1}_{\tau<\theta} + H(\theta, X_\theta^{\boldsymbol{u}})\mathbb{1}_{\tau\geq\theta}\right] \tag{3.10}$$

for all $(t, \boldsymbol{x}) \in [0, T] \times \mathbb{R}^m$ *and all stopping times* $\theta \le T$.

Theorem 2 ([Cartea et al., 2015]). ***Dynamic Programming Equation for Optimal Stopping and Control.*** *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*

$$0 = \max\left\{\partial_t H + \sup_{\boldsymbol{u} \in \mathcal{A}_t} \mathcal{L}_t^{\boldsymbol{u}} H \; ; \; G - H\right\} \tag{3.11}$$

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

3.1.1 Maximizing Terminal Wealth (Continuous)

In this section we solve the DPE that results from using the maximal terminal wealth performance criteria. The quasi-variational inequality in equation 3.11 can be interpreted as follows: the max operator is choosing between posting limit orders or executing market orders; the second term, $G - H$, is the stopping region and represents the value derived from executing a market order; and the first term is the continuation region, representing the value of posting limit orders. We'll use the shorthand $H(\cdot) = H(t, x, s, \boldsymbol{z}, q)$ and solve for $\mathrm{d}H$ inside the continuation region, hence $\mathrm{d}M^{\pm} = 0$, in order to then extract out the infinitesimal generator.

$$\begin{aligned}
\mathrm{d}H(t, x, s, \boldsymbol{z}, q) &= \sum_i \partial_{x_i} H \, \mathrm{d}x_i && (3.12) \\
&= \partial_t H \, \mathrm{d}t + \partial_{K^{\pm}} H \, \mathrm{d}K^{\pm} + \partial_{\boldsymbol{Z}} H \, \mathrm{d}\boldsymbol{Z} && (3.13) \\
&= \partial_t H \, \mathrm{d}t + \left\{ e^{-\kappa\delta^-} \mathbb{E}\big[H(t, x + (s + \xi + \delta^-), s, \boldsymbol{z}, q - 1) - H(\cdot) \big] \right\} \mathrm{d}K^+ \\
&\qquad + \left\{ e^{-\kappa\delta^+} \mathbb{E}\big[H(t, x - (s - \xi - \delta^+), s, \boldsymbol{z}, q + 1) - H(\cdot) \big] \right\} \mathrm{d}K^- && (3.14) \\
&\quad + \sum_{\mathbf{j}} \mathbb{E}\left[H(t, x, s + \eta_{0,\mathbf{j}}, \mathbf{j}, q) - H(\cdot) \right] \mathrm{d}Z_{\boldsymbol{z},\mathbf{j}}
\end{aligned}$$

TODO: section on compensated processes, especially for Markov Chain

Substitute in the following identities for the compensated processes

$$\mathrm{d}M^{\pm} = \mathrm{d}\tilde{K}^{\pm} + \mu^{\pm}(\boldsymbol{z}) \, \mathrm{d}t \tag{3.15}$$

$$\mathrm{d}Z_{\boldsymbol{z},\mathbf{j}} = \mathrm{d}\tilde{Z}_{\boldsymbol{z},\mathbf{j}} + G_{\boldsymbol{z},\mathbf{j}} \, \mathrm{d}t \tag{3.16}$$

$$
\begin{aligned}
&= \partial_t H \,\mathrm{d}t + \Big\{ \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} \mathbb{E}\big[H(t, x + (s + \xi + \delta^-), s, \boldsymbol{z}, q-1) - H(\cdot)\big] \\
&\qquad\qquad + \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} \mathbb{E}\big[H(t, x - (s - \xi - \delta^+), s, \boldsymbol{z}, q+1) - H(\cdot)\big] \\
&\quad + \sum_{\mathbf{j}} G_{\boldsymbol{z},\mathbf{j}} \mathbb{E}\left[H(t, x, s + \eta_{0,\mathbf{j}}, \mathbf{j}, q) - H(\cdot)\right] \Big\} \,\mathrm{d}t \\
&\qquad\qquad + \left\{ e^{-\kappa\delta^-} \mathbb{E}\big[H(t, x + (s + \xi + \delta^-), s, \boldsymbol{z}, q-1) - H(\cdot)\big] \right\} \mathrm{d}\tilde{K}^+ \\
&\qquad\qquad + \left\{ e^{-\kappa\delta^+} \mathbb{E}\big[H(t, x - (s - \xi - \delta^+), s, \boldsymbol{z}, q+1) - H(\cdot)\big] \right\} \mathrm{d}\tilde{K}^- \\
&\quad + \sum_{\mathbf{j}} \mathbb{E}\left[H(t, x, s + \eta_{0,\mathbf{j}}, \mathbf{j}, q) - H(\cdot)\right] \mathrm{d}\tilde{Z}_{\boldsymbol{z},\mathbf{j}}
\end{aligned}
\tag{3.17}
$$

From which we can see that the infinitesimal generator is given by

$$
\begin{aligned}
\mathcal{L}_t^{\delta} H = {} & \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} \mathbb{E}\big[H(t, x + (s + \xi + \delta^-), s, \boldsymbol{z}, q-1) - H(\cdot)\big] \\
& + \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} \mathbb{E}\big[H(t, x - (s - \xi - \delta^+), s, \boldsymbol{z}, q+1) - H(\cdot)\big] \\
& + \sum_{\mathbf{j}} G_{\boldsymbol{z},\mathbf{j}} \mathbb{E}\left[H(t, x, s + \eta_{0,\mathbf{j}}, \mathbf{j}, q) - H(\cdot)\right]
\end{aligned}
\tag{3.18}
$$

Now, our DPE has the form

$$
\begin{aligned}
0 = \max\Big\{ & \partial_t H + \sup_{\boldsymbol{u} \in \mathcal{A}_t} \mathcal{L}_t^{\boldsymbol{u}} H \,;\, H(t, x - (s + \xi), s, \boldsymbol{z}, q+1) - H(\cdot) \,; \\
& \qquad\qquad\qquad H(t, x + (s - \xi), s, \boldsymbol{z}, q-1) - H(\cdot) \Big\}
\end{aligned}
\tag{3.19}
$$

with boundary conditions

$$
H(T, x, s, \boldsymbol{z}, q) = x + q(s - \operatorname{sgn}(q)\xi) - \alpha q^2 \tag{3.20}
$$

$$
H(t, x, s, \boldsymbol{z}, 0) = x \tag{3.21}
$$

The three terms over which we are maximizing represent the continuation regions and stopping regions of the optimization problem. The first term, the continuation region, represents the limit order controls; the second and third terms, each a stopping region, represent the value gain from executing a buy market order and a sell market order, respectively.

Let's introduce the ansatz $H(\cdot) = x + q(s - \operatorname{sgn}(q)\xi) + h(t, \boldsymbol{z}, q)$. The first two terms are the wealth plus book value of assets, hence a mark-to-market of the current position, whereas the $h(t, \boldsymbol{z}, q)$ captures value due to the optimal trading strategy. The corresponding boundary conditions on

h are

$$h(T, \boldsymbol{z}, q) = -\alpha q^2 \tag{3.22}$$

$$h(t, \boldsymbol{z}, 0) = 0 \tag{3.23}$$

Substituting this ansatz into equation 3.18, we get:

$$\begin{aligned}
\mathcal{L}_t^\delta H = \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} \big[\delta^- &+ \xi[1 + \operatorname{sgn}(q-1) + q(\operatorname{sgn}(q) - \operatorname{sgn}(q-1))] \\
&+ h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q)\big] \\
+ \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} \big[\delta^+ &+ \xi[1 - \operatorname{sgn}(q+1) + q(\operatorname{sgn}(q) - \operatorname{sgn}(q+1))] \\
&+ h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big] \\
+ \sum_{\mathbf{j}} G_{\boldsymbol{z},\mathbf{j}} &\left[q\mathbb{E}[\eta_{0,\mathbf{j}}] + h(t, \mathbf{j}, q) - h(t, \boldsymbol{z}, q)\right]
\end{aligned} \tag{3.24}$$

We can further simplify the factors of ξ; for example, in the case of the δ^+ term, we can write

$$\begin{aligned}
1 - \operatorname{sgn}(q+1) + q(\operatorname{sgn}(q) - \operatorname{sgn}(q+1)) &= 1 - (-\mathbb{1}_{q\leq-2} + \mathbb{1}_{q\geq0}) + \mathbb{1}_{q=-1} \\
&= 1 + (\mathbb{1}_{q\leq-1} - \mathbb{1}_{q\geq0}) \\
&= 2 \cdot \mathbb{1}_{q\leq-1}
\end{aligned}$$

This gives us the simplified infinitesimal generator term

$$\begin{aligned}
\mathcal{L}_t^\delta H = \mu^+(\boldsymbol{z}) e^{-\kappa\delta^-} &\big[\delta^- + 2\xi \cdot \mathbb{1}_{q\geq1} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q)\big] \\
+ \mu^-(\boldsymbol{z}) e^{-\kappa\delta^+} &\big[\delta^+ + 2\xi \cdot \mathbb{1}_{q\leq-1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big] \\
+ \sum_{\mathbf{j}} G_{\boldsymbol{z},\mathbf{j}} &\left[q\mathbb{E}[\eta_{0,\mathbf{j}}] + h(t, \mathbf{j}, q) - h(t, \boldsymbol{z}, q)\right]
\end{aligned} \tag{3.25}$$

In the DPE, the first term requires finding the supremum over all $\delta^\pm$ of the infinitesimal generator. For this we can set the partial derivatives with respect to both δ^+ and δ^- equal to zero to solve for the optimal posting depth, which we denote with a superscript asterisk. For δ^+ we get:

$$0 = \partial_{\delta^+}\left[e^{-\kappa\delta^{+*}}\big[\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq-1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big]\right] \tag{3.26}$$

$$= -\kappa e^{-\kappa\delta^{+*}}\big[\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq-1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big] + e^{-\kappa\delta^{+*}} \tag{3.27}$$

$$= e^{-\kappa\delta^{+}}\big[-\kappa\big(\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq-1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big) + 1\big] \tag{3.28}$$

Since $e^{-\kappa\delta^{+*}} > 0$, the term inside the square braces must be equal to zero:

$$0 = -\kappa\big(\delta^{+*} + 2\xi \cdot \mathbb{1}_{q\leq -1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q)\big) + 1 \tag{3.29}$$

$$\delta^{+*} = \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q\leq -1} - h(t, \boldsymbol{z}, q+1) + h(t, \boldsymbol{z}, q) \tag{3.30}$$

Recalling that our optimal posting depths are to be non-negative, we thus find that the optimal buy limit order posting depth can be written in feedback form as

$$\delta^{+*} = \max\left\{0\ ;\ \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q\leq -1} - h(t, \boldsymbol{z}, q+1) + h(t, \boldsymbol{z}, q)\right\} \tag{3.31}$$

We can follow similar steps to solve for the optimal sell limit order posting depth

$$\delta^{-*} = \max\left\{0\ ;\ \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q\geq 1} - h(t, \boldsymbol{z}, q-1) + h(t, \boldsymbol{z}, q)\right\} \tag{3.32}$$

Turning our attention to the stopping regions of the DPE, we can use the ansatz to simplify the expressions:

$$\begin{aligned}
&H(t, x-(s+\xi), s, \boldsymbol{z}, q+1) - H(\cdot) \\
&\qquad = x - s - \xi + (q+1)(s - \operatorname{sgn}(q+1)\xi) + h(t, \boldsymbol{z}, q+1) && (3.33)\\
&\qquad\quad - \big[x + q(s - \operatorname{sgn}(q)\xi) + h(t, \boldsymbol{z}, q)\big] \\
&\qquad = -\xi\big[(q+1)\operatorname{sgn}(q+1) - q\operatorname{sgn}(q) + 1\big] + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) && (3.34)\\
&\qquad = -2\xi \cdot \mathbb{1}_{q\geq 0} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) && (3.35)
\end{aligned}$$

and similarly,

$$H(t, x+(s-\xi), s, \boldsymbol{z}, q-1) - H(\cdot) = -2\xi \cdot \mathbb{1}_{q\leq 0} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q) \tag{3.36}$$

Substituting all these results and simplifications into the DPE, we find that h satisfies

$$
\begin{aligned}
0 = \max\Bigg\{ & \partial_t h + \mu^+(\boldsymbol{z}) e^{-\kappa \delta^{-*}} \left(\delta^{-*} + 2\xi \mathbb{1}_{q \geq 1} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q) \right) \\
& \quad + \mu^-(\boldsymbol{z}) e^{-\kappa \delta^{+*}} \left(\delta^{+*} + 2\xi \cdot \mathbb{1}_{q \leq -1} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) \right) \\
& \quad + \sum_{\mathbf{j}} G_{\boldsymbol{z},\mathbf{j}} \left[q l \mathbb{E}\left[\eta_{0,\mathbf{j}}\right] + h(t, \mathbf{j}, q) - h(t, \boldsymbol{z}, q) \right] \ ; \\
& - 2\xi \cdot \mathbb{1}_{q \geq 0} + h(t, \boldsymbol{z}, q+1) - h(t, \boldsymbol{z}, q) \ ; \\
& - 2\xi \cdot \mathbb{1}_{q \leq 0} + h(t, \boldsymbol{z}, q-1) - h(t, \boldsymbol{z}, q) \Bigg\}
\end{aligned} \tag{3.37}
$$

Looking at the simplified feedback form in the stopping region, we see that a buy market order will be executed at time τ_q^+ whenever

$$
h(\tau_q^+, \boldsymbol{z}, q+1) - h(\tau_q^+, \boldsymbol{z}, q) = 2\xi \cdot \mathbb{1}_{q \geq 0} \tag{3.38}
$$

and a sell market order whenever

$$
h(\tau_q^+, \boldsymbol{z}, q-1) - h(\tau_q^+, \boldsymbol{z}, q) = 2\xi \cdot \mathbb{1}_{q \leq 0} \tag{3.39}
$$

Consider than when our inventory is positive, we can purchase a stock at $s + \xi$, but it will be marked-to-market at $s - \xi$, resulting in a value difference of 2ξ. With negative inventory, we will still purchase at s_ξ, but will now also value at $s + \xi$ because our overall position is still negative, producing no value difference. In particular, with negative inventory, we will execute a buy market order so long as it does not change our value function; and with zero or positive inventory, only if it increases the value function by the value of the spread. The opposite holds for sell market orders. Together, these indicate a penchant for using market orders to drive inventory levels back toward zero when it has no effect on value, and using them to gain extra value only when the expected gain is equal to the size of the spread. This is reminiscent of what we saw in the exploratory data analysis: if a stock is worth S, we can purchase it at $S + \xi$ and immediately be able to sell it at $S - \xi$, at a loss of 2ξ; this was the most significant source of loss in the naive trading market order strategy. Hence we need to expect our value to increase by at least 2ξ when executing market orders for gain.

The variational inequality in Equation (3.37) yields that whilst in the continuation region, we instead have

$$
h(\tau_q^+, \boldsymbol{z}, q+1) - h(\tau_q^+, \boldsymbol{z}, q) \leq 2\xi \cdot \mathbb{1}_{q \geq 0} \tag{3.40}
$$

$$h(\tau_q^+, \boldsymbol{z}, q-1) - h(\tau_q^+, \boldsymbol{z}, q) \leq 2\xi \cdot \mathbb{1}_{q \leq 0} \tag{3.41}$$

Taken together, these inequalities yield

$$-2\xi \cdot \mathbb{1}_{q \geq 0} \leq h(t, \boldsymbol{z}, q) - h(t, \boldsymbol{z}, q+1) \leq 2\xi \cdot \mathbb{1}_{q \leq -1} \tag{3.42}$$

$$-2\xi \cdot \mathbb{1}_{q \leq 0} \leq h(t, \boldsymbol{z}, q) - h(t, \boldsymbol{z}, q-1) \leq 2\xi \cdot \mathbb{1}_{q \geq 1} \tag{3.43}$$

or alternatively,

$$h(t, \boldsymbol{z}, q) \leq h(t, \boldsymbol{z}, q+1) \leq h(t, \boldsymbol{z}, q) + 2\xi, \qquad q \geq 0 \tag{3.44}$$

$$h(t, \boldsymbol{z}, q) \leq h(t, \boldsymbol{z}, q-1) \leq h(t, \boldsymbol{z}, q) + 2\xi, \qquad q \leq 0 \tag{3.45}$$

TODO: insert the little bubbles with sell and buy at the inequatlity signs. sell, buy, buy, sell left right top down.

Recalling the boundary condition $h(t, \boldsymbol{z}, 0) = 0$, this tells us that the function h is non-negative everywhere. Furthermore, noting the feedback form of our optimal buy limit order depth given in equation Equation (3.31), together with the inequalities in Equation (3.42) and Equation (3.43), we obtain bounds on our posting depths given by

$$\delta^{+*} = \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q \leq -1} - h(t, \boldsymbol{z}, q+1) + h(t, \boldsymbol{z}, q) \tag{3.46}$$

$$\geq \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q \leq -1} - 2\xi \cdot \mathbb{1}_{q \geq 0} \tag{3.47}$$

$$= \frac{1}{\kappa} - 2\xi \tag{3.48}$$

$$\delta^{+*} \leq \frac{1}{\kappa} - 2\xi \cdot \mathbb{1}_{q \leq -1} + 2\xi \cdot \mathbb{1}_{q \leq -1} \tag{3.49}$$

$$= \frac{1}{\kappa} \tag{3.50}$$

Combined with the identical conditions on the sell depth, we have the conditions

$$\boxed{\frac{1}{\kappa} - 2\xi \leq \delta^{\pm *} \leq \frac{1}{\kappa}} \tag{3.51}$$

A possible interpretation of the unexpected upper bound on the posting depth is that if the calculated buy (resp. sell) depth is 'sufficiently' large so as to indicate a disposition against buying (resp. selling), then it is actually optimal to sell (resp. buy) instead.

3.2 Discrete Time

Reminder of our processes (a little bit of abuse of notation going on):

$\boldsymbol{z}_k = (\rho_k, \Delta_k)$ - 2-D time-homogenous Markov Chain with transition probabilities $\boldsymbol{P}_{ij}$, where $\rho_k \in \Gamma$ and Γ represents the set of imbalance bins, and $\Delta_k = \operatorname{sgn}(s_k - s_{k-1}) \in \{-1, 0, 1\}$.

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

$$\text{Control } \vec{u}_k = \begin{pmatrix} \delta_k^+ \\ \delta_k^- \\ M_k^+ \\ M_k^- \end{pmatrix} \qquad \begin{array}{c} \text{bid posting depth} \\ \text{ask posting depth} \\ \text{buy MO - binary control} \\ \text{sell MO - binary control} \end{array}$$

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

Following [Kwong, 2015], we'll write the evolution of the Markov chain as a function of the current state and a uniformly distributed random variable ω:

$$\boldsymbol{z}_{k+1} = T(\boldsymbol{z}_k, \omega_k) = \sum_{i=0}^{|\Gamma|} i \cdot \mathbb{1}_{\left(\sum_{j=0}^{i-1} \boldsymbol{P}_{\boldsymbol{z}_k,j}, \sum_{j=0}^{i} \boldsymbol{P}_{\boldsymbol{z}_k,j}\right]}(\omega_k) \tag{3.52}$$

Here $\mathbb{1}_A(\omega) = \begin{cases} 1 & \text{if } \omega \in A \\ 0 & \text{if } \omega \notin A \end{cases}$, and hence Z_{k+1} is assigned to the value i for which ω_k is in the indicated interval of probabilities.

Our Markovian state evolution function f, given by $\vec{x}_{k+1} = f\left(\vec{x}_k, \vec{u}_k, \vec{w}_k\right)$, can be written explicitly as

$$\begin{pmatrix} x_{k+1} \\ s_{k+1} \\ \boldsymbol{z}_{k+1} \\ q_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ s_k + \eta_{k+1, T(\boldsymbol{z}_k, \omega_k)} \\ T(\boldsymbol{z}_k, \omega_k) \\ q_k \end{pmatrix} + \begin{pmatrix} s_k + \xi + \delta_k^- \\ 0 \\ 0 \\ -1 \end{pmatrix} L_k^- + \begin{pmatrix} -(s_k - \xi - \delta_k^+) \\ 0 \\ 0 \\ 1 \end{pmatrix} L_k^+ \tag{3.53}$$

The cash process at a subsequent timestep is equal to the cash at the previous step, plus the profits and costs of executing market and/or limit orders. At time k, if the agent posts a sell limit order that gets filled "between timesteps" k and $k+1$ (depending on the binary random variable L_k^-, itself depending on the binary random variable K_k^+), the revenue depends on the stock price at k. This is consistent with reality as with backtesting: while we are choosing to model the posting *depth*, in reality a submitted limit order has a specific price specified - thus once the order is submitted at k, the potential cash received is fixed.

Our impulse control at every time step is given by

$$\begin{pmatrix} x_k \\ s_k \\ \boldsymbol{z}_k \\ q_k \end{pmatrix} = \begin{pmatrix} x_k \\ s_k \\ \boldsymbol{z}_k \\ q_k \end{pmatrix} + \begin{pmatrix} s_k - \xi \\ 0 \\ 0 \\ -1 \end{pmatrix} M_k^- + \begin{pmatrix} -(s_k + \xi) \\ 0 \\ 0 \\ 1 \end{pmatrix} M_k^+ \tag{3.54}$$

Our market orders assume immediate execution, and are assumed to be sufficiently small in volume so as to not affect order imbalance or the midprice.

3.2.1 Dynamic Programming

The system formulation allows both continuous and impulse control to mimic what was done in the continuous time section, though in discrete time there is no *a priori* distinction between the two [Bensoussan, 2008]. The following theorem shows that in this case a quasi-variational inequality formulation does exist, and that it is equivalent to the standard dynamic programming formulation. The result is a simplified expression that mirrors the continuous time analysis.

Theorem 3 ([Bensoussan, 2008])**. *Dynamic Programming with Impulse Control in Discrete Time.*** *Consider a controlled Markov Chain with state space $X = \mathbb{R}^d$, transition probability $\pi(x, v, \mathrm{d}\eta)$, and positive, bounded, uniformly continuous cost function $l(x, v)$.*

Introduce an impulse control w. Define the extended cost function by $l(x, v, w) = l(x+w, v)+c(w)$, the extended transition probability by $\pi(x, v, w, \mathrm{d}\eta) = \pi(x+w, v, \mathrm{d}\eta)$ with the associated operator $\Phi^{v,w} f(x) = \int_{\mathbb{R}^d} f(\eta)\pi(x, v, w, \mathrm{d}\eta) = \Phi^v f(x + w)$.

Consider a decision rule V, W with associated probability $\mathbb{P}^{V,W,x}$ on $\Omega, \mathcal{A}$ for which $y_1 = x$ a.s.

Consider the pay-off function

$$J_x(V, W) = \mathbb{E}^{V,W,x}\left[\sum_{n=1}^{\infty} \alpha^{n-1} l(y_n, v_n, w_n)\right] \tag{3.55}$$

and the corresponding Bellman equation

$$u(x) = \inf_{\substack{v \in U \\ w \geq 0}} [l(x + w, v) + c(w) + \alpha \Phi^v u(x + w)] \tag{3.56}$$

Assume:

1. $\Phi^V \phi_v(x)$ *is continuous in* v, x *if* $\phi_v(x) = \phi(x, v)$ *is uniformly continuous and bounded in* x, v*;*

2. $c(w) = K\mathbb{1}_{w=0} + c_0(w), \quad c_0(0) = 0, \qquad c_0(w) \to \infty$ *as* $|w| \to \infty$*,* $c_0(w)$ *is sub-linear positive continuous;*

3. U *is compact.*

Then there exists a unique, positive, bounded solution of Equation (3.56) belonging to the space of uniformly continuous and bounded functions. Further, this solution is identical to that of

$$u(x) = \min\left\{K + \inf_{w \geq 0}[c_0(w) + u(x + w)]\ ;\ \inf_{v \in U}[l(x, v) + \alpha \Phi^v u(x)]\right\} \tag{3.57}$$

3.2.2 Maximizing Terminal Wealth (Discrete)

Following the dynamic programming with impulse control programme, we introduce the value function $V_k^{\delta\pm}$. Here, as in the continuous-time formulation, our objective is to maximize the terminal wealth performance criteria given by

$$V_k^{\delta\pm}(x, s, \boldsymbol{z}, q) = \mathbb{E}\left[W_T^{\delta\pm}\right] = \mathbb{E}_{k,x,s,\boldsymbol{z},q}\left[X_T^{\delta\pm} + Q_T^{\delta\pm}(S_T - \operatorname{sgn}(Q_T^{\delta\pm})\xi) - \alpha(Q_T^{\delta\pm})^2\right] \tag{3.58}$$

where, as before, the notation $\mathbb{E}_{k,x,s,\boldsymbol{z},q}[\ \cdot\ ]$ represents the conditional expectation

$$\mathbb{E}[\ \cdot \mid X_k = x, S_k = s, \boldsymbol{Z}_k = \boldsymbol{z}, Q_k = q]$$

In this case, our dynamic programming equations (DPEs) are given by

$$V_T(x,s,\boldsymbol{z},q) = x + q(s - \mathrm{sgn}(q)\xi) - \alpha q^2 \tag{3.59}$$

$$\begin{aligned} V_k(x,s,\boldsymbol{z},q) = \max\Bigg\{ & \sup_{\delta^\pm}\Big\{\mathbb{E}_{\mathbf{w}}\left[V_{k+1}(f((x,s,\boldsymbol{z},q),\boldsymbol{u},\mathbf{w}_k)\right]\Big\} \ ; \\ & V_k(x + s_k - \xi, s_k, \boldsymbol{z}_k, q_k - 1) \ ; \\ & V_k(x - s_k - \xi, s_k, \boldsymbol{z}_k, q_k + 1)\Bigg\} \end{aligned} \tag{3.60}$$

where expectation is with respect to the random vector $\mathbf{w}_k$. Note that in this formulation we do not have per stage costs, as the cost of execution is bundled into the state x. Nevertheless, it is rather immediate that the execution costs could be disentangled from the system state and seen to satisfy the theorem assumptions. Hypothetically we could add the fourth case where $M^+ = M^- = 1$, though a quick substitution shows that it is always strictly 2ξ less in value than the case of only limit orders, where $M^+ = M^- = 0$. This should be evident, as buying and selling with market orders in a single timestep yields a guaranteed loss as the agent is forced to cross the spread.

To simplify the DPEs, we introduce a now familiar ansatz:

$$V_k(x,s,\boldsymbol{z},q) = x + q(s - \mathrm{sgn}(q)\xi) + h_k(\boldsymbol{z},q) \tag{3.61}$$

with boundary condition $h_k(\boldsymbol{z},0) = 0$ and terminal condition $h_T(\boldsymbol{z},q) = -\alpha q^2$. Substituting this ansatz into the Equation (3.60), we obtain

$$\begin{aligned} 0 = \max\Bigg\{ & \sup_{\delta^\pm}\Big\{\mathbb{E}_{\mathbf{w}}\left[V_{k+1}(f((x,s,\boldsymbol{z},q),\boldsymbol{u},\mathbf{w}_k)\right] - V_k(x,s,\boldsymbol{z},q)\Big\} \ ; \\ & V_k(x + s_k - \xi, s_k, \boldsymbol{z}_k, q_k - 1) - V_k(x,s,\boldsymbol{z},q) \ ; \\ & V_k(x - s_k - \xi, s_k, \boldsymbol{z}_k, q_k + 1) - V_k(x,s,\boldsymbol{z},q)\Bigg\} \end{aligned} \tag{3.62}$$

$$\begin{aligned} 0 = \max\Bigg\{ & \sup_{\delta^\pm}\Big\{\mathbb{E}_{\mathbf{w}}\Big[(s + \xi + \delta^-)L_k^- - (s - \xi - \delta^+)L_k^+ \\ & \qquad + (L_k^+ - L_k^-)\big(s + \eta_{0,T(\boldsymbol{z},\omega)} - \mathrm{sgn}(q + L_k^+ - L_k^-)\xi\big) \\ & \qquad + q\left(\eta_{0,T(\boldsymbol{z},\omega)} - \big(\mathrm{sgn}(q + L_k^+ - L_k^-) - \mathrm{sgn}(q)\big)\,\xi\right) \\ & \qquad + h_{k+1}(T(\boldsymbol{z},\omega), q + L_k^+ - L_k^-) - h_k(\boldsymbol{z},q)\Big]\Big\} \ ; \\ & -2\xi \cdot \mathbb{1}_{q\geq 0} + h_k(\boldsymbol{z},q+1) \ ; \\ & -2\xi \cdot \mathbb{1}_{q\leq 0} + h_k(\boldsymbol{z},q-1)\Bigg\} \end{aligned} \tag{3.63}$$

We'll begin by concentrating on the first term in the quasi-variational inequality. Thus, we want to solve

$$\begin{aligned}\sup_{\delta^\pm}\Bigg\{\mathbb{E}_{\mathbf{w}}\Bigg[&(s+\xi+\delta^-)L_k^- - (s-\xi-\delta^+)L_k^+ \\ &+ (L_k^+ - L_k^-)\big(s+\eta_{0,T(\boldsymbol{z},\omega)} - \operatorname{sgn}(q+L_k^+-L_k^-)\xi\big) \\ &+ q\left(\eta_{0,T(\boldsymbol{z},\omega)} - \big(\operatorname{sgn}(q+L_k^+-L_k^-) - \operatorname{sgn}(q)\big)\,\xi\right) \\ &+ h_{k+1}(T(\boldsymbol{z},\omega), q+L_k^+-L_k^-) - h_k(\boldsymbol{z},q)\Bigg]\Bigg\}\end{aligned} \tag{3.64}$$

As other agents' market orders as Poisson distributed, we have that $\mathbb{P}[K_k^+ = 0] = \frac{e^{-\mu^+(\boldsymbol{z})\Delta t}(\mu^+(\boldsymbol{z})\Delta t)^0}{0!} = e^{-\mu^+(\boldsymbol{z})\Delta t}$, and so the probability of seeing some positive number of market orders is

$$\mathbb{P}[K_k^+ > 0] = 1 - e^{-\mu^+(\boldsymbol{z})\Delta t} \tag{3.65}$$

Now we make the simplified assumption that the *aggregate* of the orders walks the limit order book to a depth of p_k, and if $p_k > \delta^-$, then our sell limit order is lifted. As in the continuous time section, we will assume that the probability of our order being lifted is $e^{-\kappa\delta^-}$. Thus we have the following preliminary results:

$$\mathbb{P}[L_k^- = 1 | K_k^+ > 0] = e^{-\kappa\delta^-} \tag{3.66}$$

$$\mathbb{P}[L_k^- = 0 | K_k^+ > 0] = 1 - e^{-\kappa\delta^-} \tag{3.67}$$

$$\mathbb{E}[L_k^-] = \mathbb{P}[L_k^- = 1 | K_k^+ > 0] \cdot \mathbb{P}[K_k^+ > 0] \tag{3.68}$$

$$= (1 - e^{-\mu^+(\boldsymbol{z})\Delta t})e^{-\kappa\delta^-} \tag{3.69}$$

For ease of notation, we'll write the probability of the $L_k^- = 1$ event as $p(\delta^-)$. This gives us the additional results:

$$\mathbb{P}[L_k^- = 1] = p(\delta^-) = \mathbb{E}[L_k^-] \tag{3.70}$$

$$\mathbb{P}[L_k^- = 0] = 1 - p(\delta^-) \tag{3.71}$$

$$\partial_{\delta^-}\mathbb{P}[L_k^- = 1] = -\kappa p(\delta^-) \tag{3.72}$$

$$\partial_{\delta^-}\mathbb{P}[L_k^- = 0] = \kappa p(\delta^-) \tag{3.73}$$

Let's pre-compute some of the terms that we'll encounter in the supremum, namely the expectations of the random variables. To each we will assign an uppercase Greek letter as shorthand, as will be evident from the analysis.

$$\begin{aligned}
\mathbb{E}[\operatorname{sgn}(q + L_k^+ - L_k^-)] &= \mathbb{P}[L_k^- = 1] \cdot \mathbb{P}[L_k^+ = 1] \cdot \operatorname{sgn}(q) \\
&\quad + \mathbb{P}[L_k^- = 1] \cdot \mathbb{P}[L_k^+ = 0] \cdot \operatorname{sgn}(q-1) \\
&\quad + \mathbb{P}[L_k^- = 0] \cdot \mathbb{P}[L_k^+ = 1] \cdot \operatorname{sgn}(q+1) \\
&\quad + \mathbb{P}[L_k^- = 0] \cdot \mathbb{P}[L_k^+ = 0] \cdot \operatorname{sgn}(q)
\end{aligned} \tag{3.74}$$

$$\begin{aligned}
&= p(\delta^-)p(\delta^+) \operatorname{sgn}(q) \\
&\quad + p(\delta^-)(1 - p(\delta^+)) \operatorname{sgn}(q-1) \\
&\quad + (1 - p(\delta^-))p(\delta^+) \operatorname{sgn}(q+1) \\
&\quad + (1 - p(\delta^-))(1 - p(\delta^+)) \operatorname{sgn}(q)
\end{aligned} \tag{3.75}$$

$$\begin{aligned}
&= \operatorname{sgn}(q)\big[1 - p(\delta^+) - p(\delta^-) + 2p(\delta^+)p(\delta^-)\big] \\
&\quad + \operatorname{sgn}(q-1)\big[p(\delta^-) - p(\delta^+)p(\delta^-)\big] \\
&\quad + \operatorname{sgn}(q+1)\big[p(\delta^+) - p(\delta^+)p(\delta^-)\big]
\end{aligned} \tag{3.76}$$

$$= \begin{cases} 1 & q \geq 2 \\ 1 - p(\delta^-)(1 - p(\delta^+)) & q = 1 \\ p(\delta^+) - p(\delta^-) & q = 0 \\ -\big[1 - p(\delta^+)(1 - p(\delta^-))\big] & q = -1 \\ -1 & q \leq -2 \end{cases} \tag{3.77}$$

$$= \Phi(q, \delta^+, \delta^-) \tag{3.78}$$

Similarly:

$$\begin{aligned}
\mathbb{E}[L_k^+ \operatorname{sgn}(q + L_k^+ - L_k^-)] &= \mathbb{P}[L_k^- = 1] \cdot \mathbb{P}[L_k^+ = 1] \cdot \operatorname{sgn}(q) \\
&\quad + \mathbb{P}[L_k^- = 1] \cdot \mathbb{P}[L_k^+ = 0] \cdot 0 \operatorname{sgn}(q-1) \\
&\quad + \mathbb{P}[L_k^- = 0] \cdot \mathbb{P}[L_k^+ = 1] \cdot \operatorname{sgn}(q+1) \\
&\quad + \mathbb{P}[L_k^- = 0] \cdot \mathbb{P}[L_k^+ = 0] \cdot 0 \operatorname{sgn}(q)
\end{aligned} \tag{3.79}$$

$$= p(\delta^+)\big[p(\delta^-) \operatorname{sgn}(q) + (1 - p(\delta^-) \operatorname{sgn}(q+1)\big] \tag{3.80}$$

$$= p(\delta^+) \begin{cases} 1 & q \geq 2 \\ 1 & q = 1 \\ (1 - p(\delta^-)) & q = 0 \\ -p(\delta^-) & q = -1 \\ -1 & q \leq -2 \end{cases} \tag{3.81}$$

$$= p(\delta^+)\Psi(q,\delta^-) \tag{3.82}$$

and

$$\mathbb{E}[L_k^- \operatorname{sgn}(q + L_k^+ - L_k^-)] = p(\delta^-)\big[p(\delta^+)\operatorname{sgn}(q) + (1-p(\delta^+))\operatorname{sgn}(q-1)\big] \tag{3.83}$$

$$= p(\delta^-)\begin{cases} 1 & q \geq 2 \\ p(\delta^+) & q = 1 \\ -(1-p(\delta^+)) & q = 0 \\ -1 & q = -1 \\ -1 & q \leq -2 \end{cases} \tag{3.84}$$

$$= p(\delta^-)\Upsilon(q,\delta^+) \tag{3.85}$$

We'll also require the partial derivatives of these expectations, which we can easily compute. Below we'll use the simplified notation Φ_+ to denote the function closely associated with the partial derivative of Φ with respect to δ^+.

$$\partial_{\delta^-}\mathbb{E}[\operatorname{sgn}(q + L_k^+ - L_k^-)] = \partial_{\delta^-}\Phi(q,\delta^+,\delta^-) = \kappa p(\delta^-)\begin{cases} 0 & q \geq 2 \\ (1-p(\delta^+)) & q = 1 \\ 1 & q = 0 \\ p(\delta^+) & q = -1 \\ 0 & q \leq -2 \end{cases} \tag{3.86}$$

$$= \kappa p(\delta^-)\Phi_-(q,\delta^+) \tag{3.87}$$

$$\partial_{\delta^+}\mathbb{E}[\operatorname{sgn}(q + L_k^+ - L_k^-)] = \partial_{\delta^+}\Phi(q,\delta^+,\delta^-) = \kappa p(\delta^+)\begin{cases} 0 & q \geq 2 \\ -p(\delta^-) & q = 1 \\ -1 & q = 0 \\ -(1-p(\delta^-)) & q = -1 \\ 0 & q \leq -2 \end{cases} \tag{3.88}$$

$$= \kappa p(\delta^+)\Phi_+(q,\delta^-) \tag{3.89}$$

$$\partial_{\delta^-}\mathbb{E}[L_k^+ \operatorname{sgn}(q+L_k^+-L_k^-)] = \partial_{\delta^-}p(\delta^+)\Psi(q,\delta^-) = \kappa p(\delta^+)p(\delta^-)\begin{cases} 0 & q\geq 2 \\ 0 & q=1 \\ 1 & q=0 \\ 1 & q=-1 \\ 0 & q\leq -2 \end{cases} \tag{3.90}$$

$$= \kappa p(\delta^+)p(\delta^-)\Psi_-(q) \tag{3.91}$$

$$\partial_{\delta^+}\mathbb{E}[L_k^+ \operatorname{sgn}(q+L_k^+-L_k^-)] = \partial_{\delta^+}p(\delta^+)\Psi(q,\delta^-) = -\kappa p(\delta^+)\Psi(q,\delta^-) \tag{3.92}$$

$$\partial_{\delta^-}\mathbb{E}[L_k^- \operatorname{sgn}(q+L_k^+-L_k^-)] = \partial_{\delta^-}p(\delta^-)\Upsilon(q,\delta^+) = -\kappa p(\delta^-)\Upsilon(q,\delta^+) \tag{3.93}$$

$$\partial_{\delta^+}\mathbb{E}[L_k^- \operatorname{sgn}(q+L_k^+-L_k^-)] = \partial_{\delta^+}p(\delta^-)\Upsilon(q,\delta^+) = \kappa p(\delta^+)p(\delta^-)\begin{cases} 0 & q\geq 2 \\ -1 & q=1 \\ -1 & q=0 \\ 0 & q=-1 \\ 0 & q\leq -2 \end{cases} \tag{3.94}$$

$$= \kappa p(\delta^+)p(\delta^-)\Upsilon_+(q) \tag{3.95}$$

Recalling that we have $\boldsymbol{P}$ the transition matrix for the Markov Chain $\boldsymbol{Z}$, with $\boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} = \mathbb{P}[\boldsymbol{Z}_{k+1} = \mathbf{j}|\boldsymbol{Z}_k = \boldsymbol{z}]$, then we can also write:

$$\begin{aligned}\mathbb{E}[h_{k+1}(T(\boldsymbol{z},\omega), q+L_k^+-L_k^-)] = \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\Big[& h_{k+1}(\mathbf{j},q)\big[1-p(\delta^+)-p(\delta^-)+2p(\delta^+)p(\delta^-)\big] \\ & + h_{k+1}(\mathbf{j},q-1)\big[p(\delta^-)-p(\delta^+)p(\delta^-)\big] \\ & + h_{k+1}(\mathbf{j},q+1)\big[p(\delta^+)-p(\delta^+)p(\delta^-)\big]\Big]\end{aligned} \tag{3.96}$$

and its partial derivatives as

$$\begin{aligned}\partial_{\delta^-}\mathbb{E}[h_{k+1}(T(\boldsymbol{z},\omega), q+L_k^+-L_k^-)] = \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\Big[& h_{k+1}(\mathbf{j},q)\big[\kappa p(\delta^-)-2\kappa p(\delta^+)p(\delta^-)\big] \\ & + h_{k+1}(\mathbf{j},q-1)\big[-\kappa p(\delta^-)+\kappa p(\delta^+)p(\delta^-)\big] \\ & + h_{k+1}(\mathbf{j},q+1)\big[\kappa p(\delta^+)p(\delta^-)\big]\Big]\end{aligned} \tag{3.97}$$

$$
\begin{aligned}
&= \kappa p(\delta^-) \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} \Big[h_{k+1}(\mathbf{j}, q) \big[1 - 2p(\delta^+)\big] \\
&\qquad + h_{k+1}(\mathbf{j}, q-1) \big[-1 + p(\delta^+)\big] \\
&\qquad + h_{k+1}(\mathbf{j}, q+1) \big[p(\delta^+)\big] \Big]
\end{aligned} \tag{3.98}
$$

$$
\begin{aligned}
\partial_{\delta^+} \mathbb{E}[h_{k+1}(T(\boldsymbol{z}, \omega), q + L_k^+ - L_k^-)] &= \kappa p(\delta^+) \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} \Big[h_{k+1}(\mathbf{j}, q) \big[1 - 2p(\delta^-)\big] \\
&\qquad + h_{k+1}(\mathbf{j}, q-1) \big[p(\delta^-)\big] \\
&\qquad + h_{k+1}(\mathbf{j}, q+1) \big[-1 + p(\delta^-)\big] \Big]
\end{aligned} \tag{3.99}
$$

Now we tackle solving the supremum in equation 3.64 and thus finding the optimal posting depths, again denoted by a subscript asterisk. First we consider the first-order condition on δ^-, namely that the partial derivative with respect to it must be equal to zero.

$$
\begin{aligned}
0 = \partial_{\delta^-} \Big\{ &(s + \xi + \delta^{-*})\mathbb{E}[L_k^-] - (s - \xi - \delta^+)\mathbb{E}[L_k^+] \\
&+ \mathbb{E}[L_k^+] \left(s + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}]\right) - \xi \mathbb{E}\left[L_k^+ \operatorname{sgn}(q + L_k^+ - L_k^-)\right] \\
&- \mathbb{E}[L_k^-] \left(s + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}]\right) + \xi \mathbb{E}\left[L_k^- \operatorname{sgn}(q + L_k^+ - L_k^-)\right] \\
&+ q\mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - q\xi\mathbb{E}[\operatorname{sgn}(q + L_k^+ - L_k^-)] + q\xi \operatorname{sgn}(q) \\
&+ \mathbb{E}\left[h_{k+1}(T(\boldsymbol{z}, \omega), q + L_k^+ - L_k^-)\right] - h_k(\boldsymbol{z}, q) \Big\}
\end{aligned} \tag{3.100}
$$

$$
\begin{aligned}
= \partial_{\delta^-} \Big\{ &(s + \xi + \delta^{-*})\mathbb{E}[L_k^-] - \xi \mathbb{E}\left[L_k^+ \operatorname{sgn}(q + L_k^+ - L_k^-)\right] \\
&- \mathbb{E}[L_k^-] \left(s + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}]\right) + \xi \mathbb{E}\left[L_k^- \operatorname{sgn}(q + L_k^+ - L_k^-)\right] \\
&- q\xi\mathbb{E}[\operatorname{sgn}(q + L_k^+ - L_k^-)] + \mathbb{E}\left[h_{k+1}(T(\boldsymbol{z}, \omega), q + L_k^+ - L_k^-)\right] \Big\}
\end{aligned} \tag{3.101}
$$

$$
\begin{aligned}
&= p(\delta^{-*}) - \kappa p(\delta^{-*})(s + \xi + \delta^{-*}) - \xi\kappa p(\delta^+) p(\delta^{-*}) \Psi_-(q) \\
&\quad + \kappa p(\delta^{-*}) \left(s + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}]\right) - \xi\kappa p(\delta^{-*}) \Upsilon(q, \delta^+) - q\xi\kappa p(\delta^{-*}) \Phi_-(q, \delta^+) \\
&\quad + \kappa p(\delta^{-*}) \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} \Big[h_{k+1}(\mathbf{j}, q) \big[1 - 2p(\delta^+)\big] + h_{k+1}(\mathbf{j}, q-1) \big[-1 + p(\delta^+)\big] \\
&\qquad + h_{k+1}(\mathbf{j}, q+1) \big[p(\delta^+)\big] \Big]
\end{aligned} \tag{3.102}
$$

Dividing through by $\kappa p(\delta^{-*})$, which is nonzero, and re-arranging, we find that the optimal sell posting depth is given by

$$\delta^{-*} = \frac{1}{\kappa} + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - \xi\left(1 + p(\delta^+)\Psi_-(q) + \Upsilon(q,\delta^+) + q\Phi_-(q,\delta^+)\right) + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\Big[h_{k+1}(\mathbf{j},q)\big[1-2p(\delta^+)\big] + h_{k+1}(\mathbf{j},q-1)\big[-1+p(\delta^+)\big] + h_{k+1}(\mathbf{j},q+1)\big[p(\delta^+)\big]\Big] \tag{3.103}$$

$$= \frac{1}{\kappa} + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\left(\mathbb{1}_{q\geq 1} + p(\delta^+)\mathbb{1}_{q=0}\right) + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\Big[h_{k+1}(\mathbf{j},q)\big[1-2p(\delta^+)\big] + h_{k+1}(\mathbf{j},q-1)\big[-1+p(\delta^+)\big] + h_{k+1}(\mathbf{j},q+1)\big[p(\delta^+)\big]\Big] \tag{3.104}$$

Recalling that we want $\delta^{\pm} \geq 0$, we find:

$$\delta^{-*} = \max\Bigg\{0\ ;\ \frac{1}{\kappa} + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\geq 1} + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\big[h_{k+1}(\mathbf{j},q) - h_{k+1}(\mathbf{j},q-1)\big] - p(\delta^+)\left(2\xi\mathbb{1}_{q=0} - \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\big[h_{k+1}(\mathbf{j},q-1) + h_{k+1}(\mathbf{j},q+1) - 2h_{k+1}(\mathbf{j},q)\big]\right)\Bigg\} \tag{3.105}$$

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

$$\delta^{+*} = \max\Bigg\{0\ ;\ \frac{1}{\kappa} - \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\leq -1} + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\big[h_{k+1}(\mathbf{j},q) - h_{k+1}(\mathbf{j},q+1)\big] - p(\delta^-)\left(2\xi\mathbb{1}_{q=0} - \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\big[h_{k+1}(\mathbf{j},q-1) + h_{k+1}(\mathbf{j},q+1) - 2h_{k+1}(\mathbf{j},q)\big]\right)\Bigg\} \tag{3.106}$$

For ease of notation we'll write $\aleph(q) = \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\left[h_{k+1}(\mathbf{j},q-1) + h_{k+1}(\mathbf{j},q+1) - 2h_{k+1}(\mathbf{j},q)\right]$. Now, assuming we behave optimally on both the buy and sell sides simultaneously, we can substitute equation 3.106 into equation 3.105, while evaluating both at δ^{+*} and δ^{-*} to obtain the optimal posting depth in feedback form:

$$\delta^{-*} = \frac{1}{\kappa} + \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\geq 1} + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\Big[h_{k+1}(\mathbf{j},q) - h_{k+1}(\mathbf{j},q-1)\Big] - (1-e^{\mu^-(\boldsymbol{z})\Delta t})e^{-\kappa\max\left\{0\ ;\ \frac{1}{\kappa} - \mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] - 2\xi\mathbb{1}_{q\leq -1} + \sum_{\mathbf{j}}\boldsymbol{P}_{\boldsymbol{z},\mathbf{j}}\left[h_{k+1}(\mathbf{j},q) - h_{k+1}(\mathbf{j},q+1)\right] - (1-e^{\mu^+(\boldsymbol{z})\Delta t})e^{-\kappa\delta^{-*}}(2\xi\mathbb{1}_{q=0} - \aleph(q))\right\}}\left(2\xi\mathbb{1}_{q=0} - \aleph(q)\right) \tag{3.107}$$

This equation will need to be solved numerically due to the difficulty in isolating δ^{-*} on one side of the equality. Once a solution has been obtained, the value can be substituted back into Equation (3.106) to solve for ${\delta^{+}}^{*}$.

3.2.3 Simplifying the DPE

We now turn to simplifying the DPE in Equation (3.63) by substituting in the optimal posting depths as written in recursive form: Equation (3.106) and Equation (3.105). In doing so we see a incredible amount of cancellation and simplification, and we obtain the rather elegant, and surprisingly simple form of the DPE:

$$
\begin{aligned}
h_k(\boldsymbol{z}, q) = \max \Bigg\{ & q\mathbb{E}[\eta_{0,T(\boldsymbol{z},\omega)}] + \frac{1}{\kappa}(p({\delta^{+}}^{*}) + p({\delta^{-}}^{*})) + \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} h_{k+1}(\mathbf{j}, q) \\
& \qquad + p({\delta^{+}}^{*})p({\delta^{-}}^{*}) \sum_{\mathbf{j}} \boldsymbol{P}_{\boldsymbol{z},\mathbf{j}} \left[h_{k+1}(\mathbf{j}, q-1) + h_{k+1}(\mathbf{j}, q+1) - 2h_{k+1}(\mathbf{j}, q) \right] \; ; \\
& - 2\xi \cdot \mathbb{1}_{q \geq 0} + h_k(\boldsymbol{z}, q+1) \; ; \\
& - 2\xi \cdot \mathbb{1}_{q \leq 0} + h_k(\boldsymbol{z}, q-1) \Bigg\}
\end{aligned}
\tag{3.108}
$$

TODO: Similar commentary to the continuous case. Find the whole inequality thing, bounds on h, bounds on δ. Identical.

At terminal time T, we liquidate our position at a cost of $(s - xi\,\mathrm{sgn}(q) - \alpha q)$ per share, whereas at $T-1$, we can liquidate at the regular cost of $(s - \xi\,\mathrm{sgn}(q)$. It is thus never optimal to wait until maturity to liquidate the position, and instead we force liquidation one step earlier by setting $h(T-1, \boldsymbol{z}, q) = 0 \ \forall q$. This allows us to effectively ignore the terminal condition, and avoids a contradiction with the finding that $h \geq 0$.

We now have an explicit means of numerically solving for the optimal posting depths. Since we know the function h at the terminal timesteps T and $T-1$, we can take one step back to $T-2$ and solve for both the optimal posting depths. With these values we are then able to calculate the value function h_{T-2} using Equation (3.108), and in doing so determine whether to execute market orders in addition to posting limit orders. This process then repeats for each step backward.

Chapter 4

Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Bibliography

[Bensoussan, 2008] Bensoussan, A. (2008). Impulse control in discrete time. *Georgian Mathematical Journal*, 15(3):439–454.

[Cartea et al., 2015] Cartea, A., Jaimungal, S., and Penalva, J. (2015). *Algorithmic and High-Frequency Trading*. Cambridge University Press.

[Kwong, 2015] Kwong, R. (2015). Ece1639 lecture notes. Lecture Notes, University of Toronto, Department of Electrical & Computer Engineering.