Limit Order Book Dynamics

Our goal is to use the dynamics of the Limit Order Book (LOB) as an indicator for high-frequency stock price movement, thus enabling statistical arbitrage. Formally, we will the study limit order book imbalance process, $I(t)$, and the stock price process, $S(t)$, and attempt to establish a stochastic relationship $\dot{S} = f(S, I, t)$. We will then attempt to derive an optimal trading strategy based on the observed relationship.

Recap Next Steps

1. Complete in-sample backtesting of the 'naive' trading strategies.
2. Formulate stochastic control problem
3. Extra Reading: Bellman Equations, MDP, Partially Observable MDP

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:

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

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:

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

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:

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
```

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 excuted, 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.

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 $\mathbb{P}\left[\Delta S_{curr} = 0\right] > 0.5$ 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.

Algorithm 2 Naive+ Trading Strategy

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

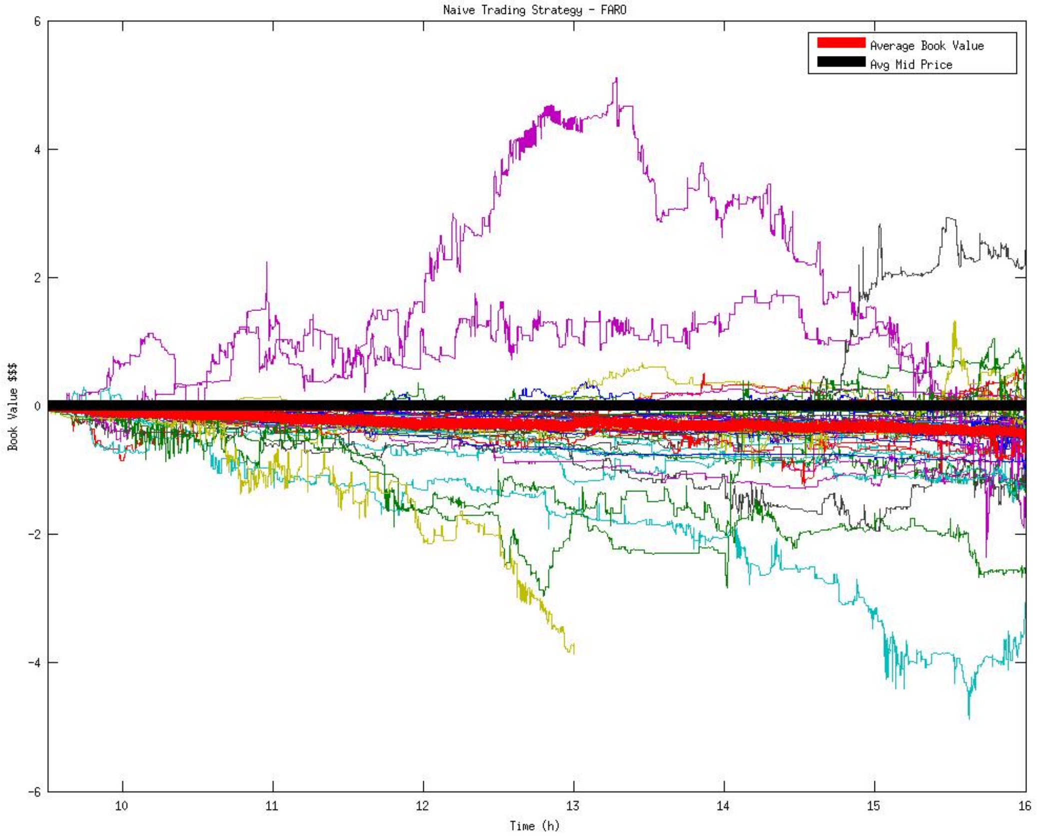


Figure 1: FARO: Bookvalue against time of trading day.

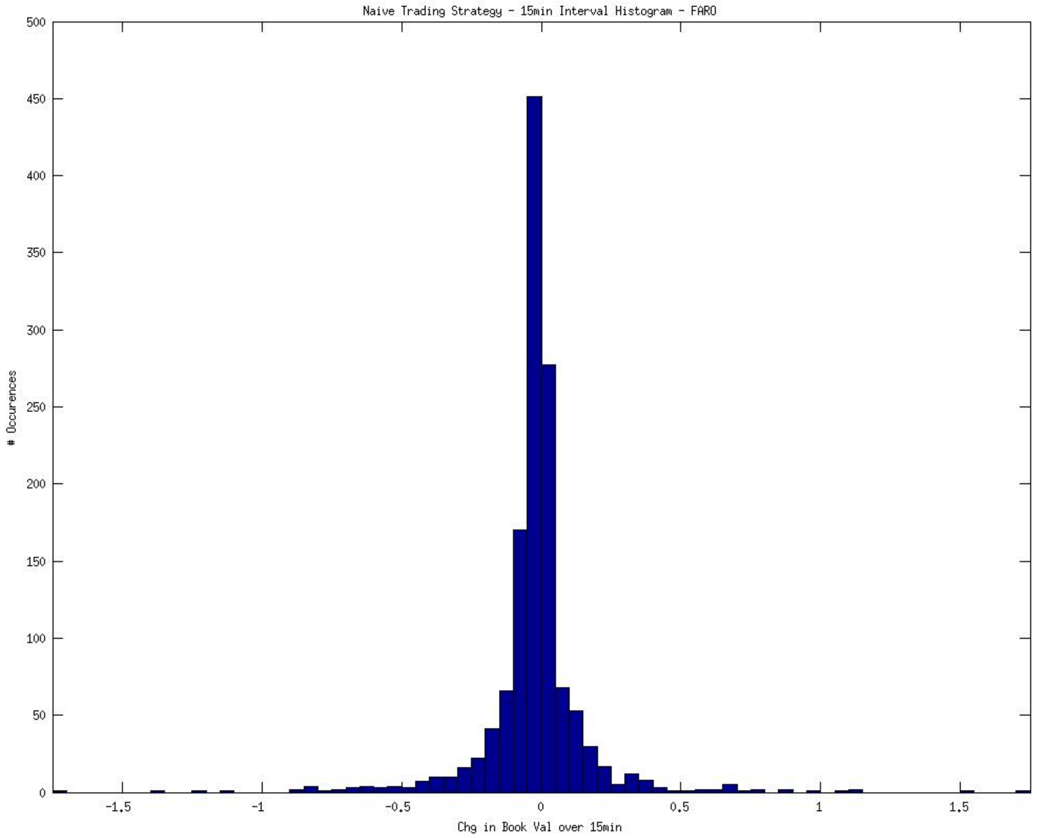


Figure 2: FARO: Histogram of 15min bookvalue changes.

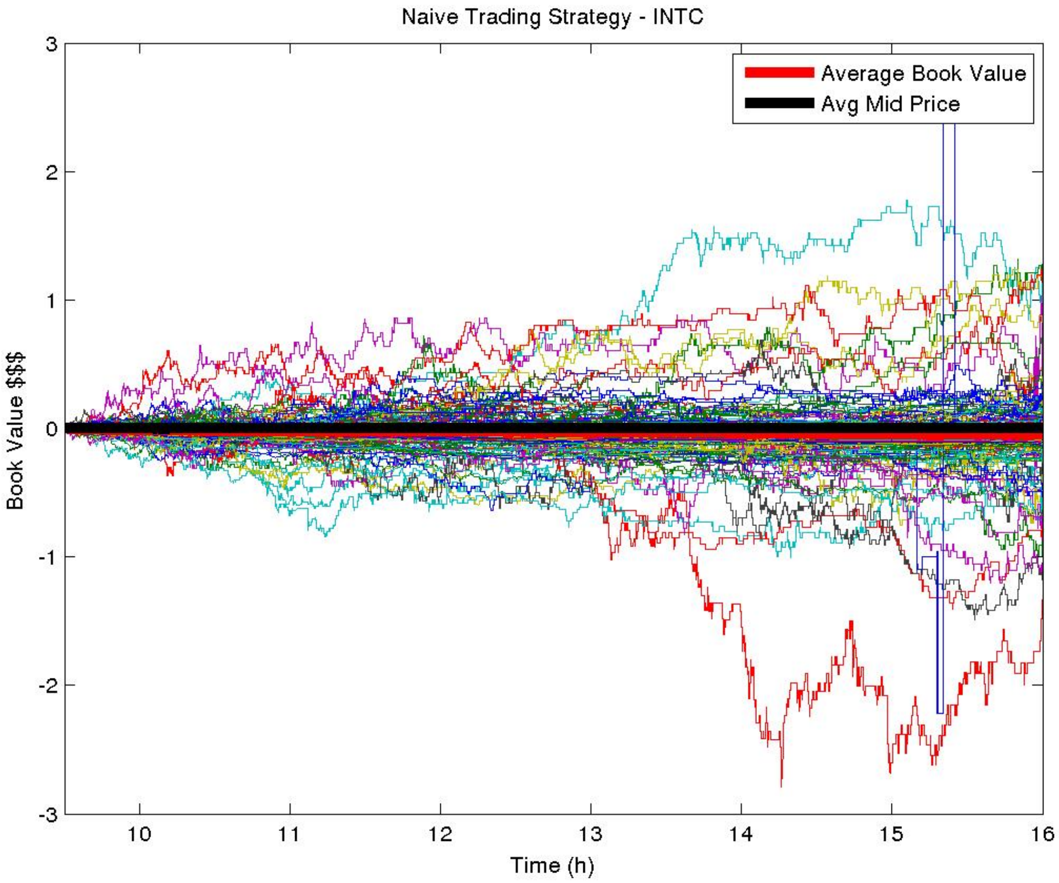


Figure 3: INTC: Bookvalue against time of trading day.

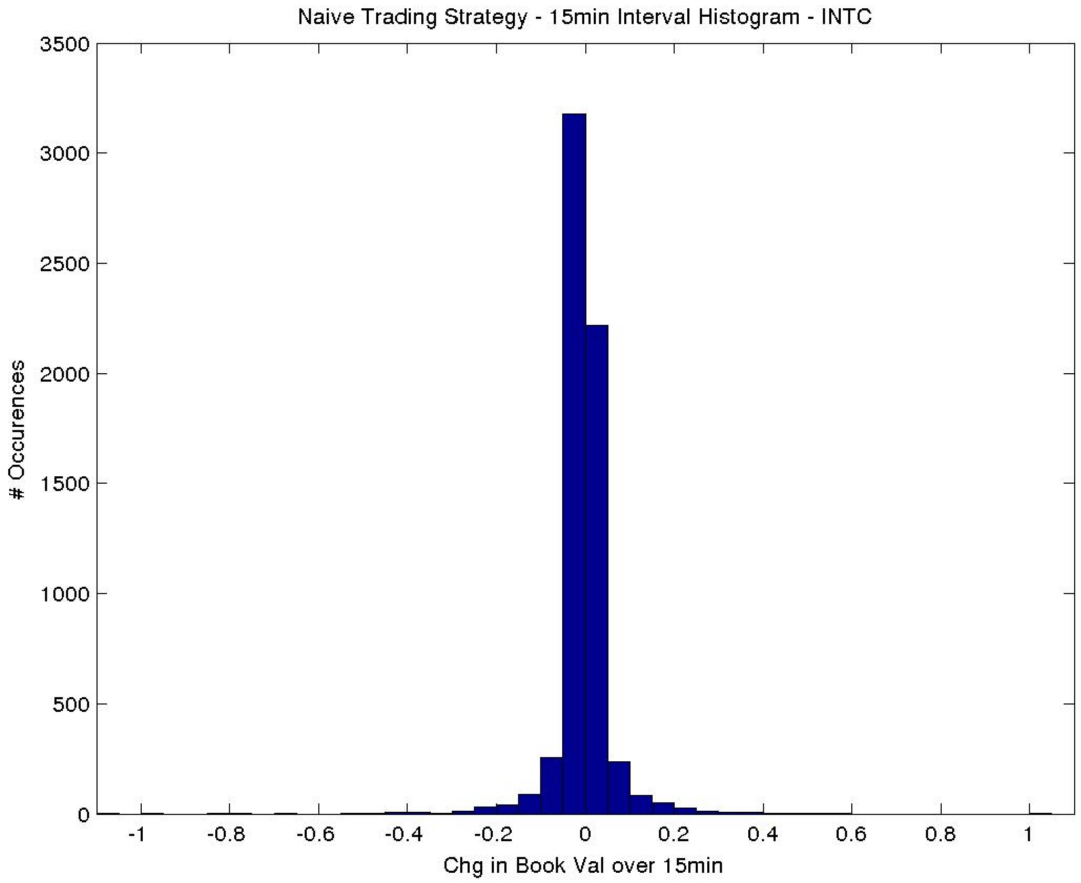


Figure 4: INTC: Histogram of 15min bookvalue changes.

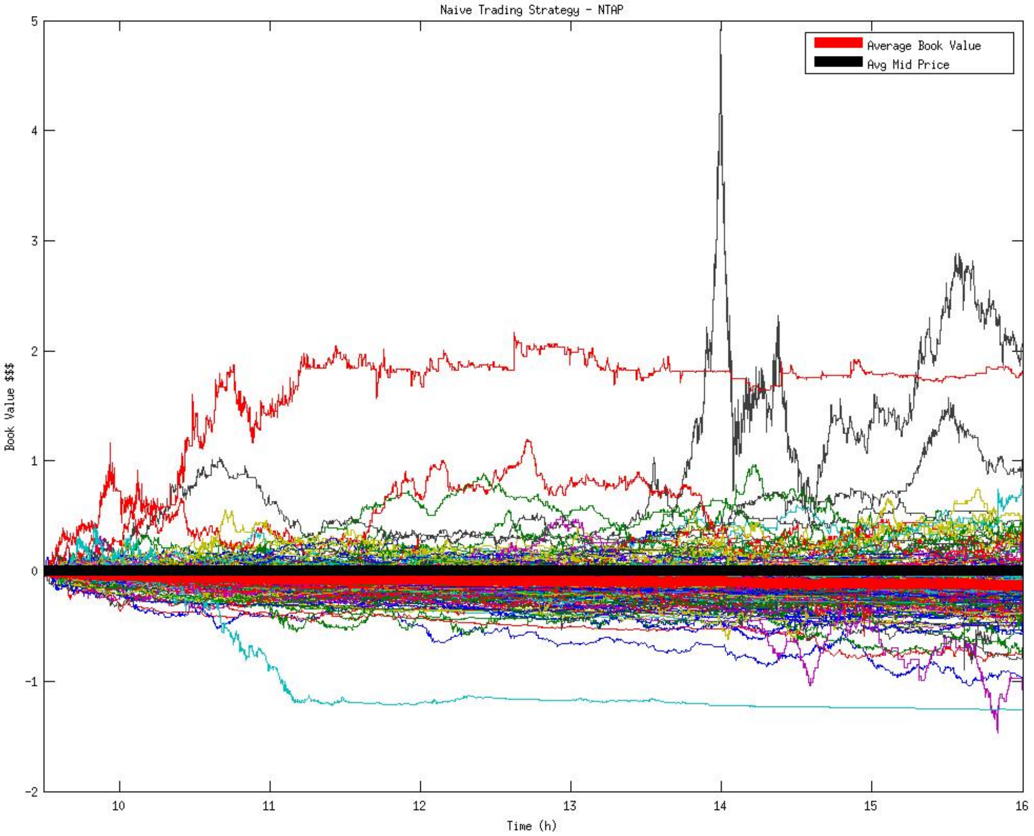


Figure 5: NTAP: Bookvalue against time of trading day.

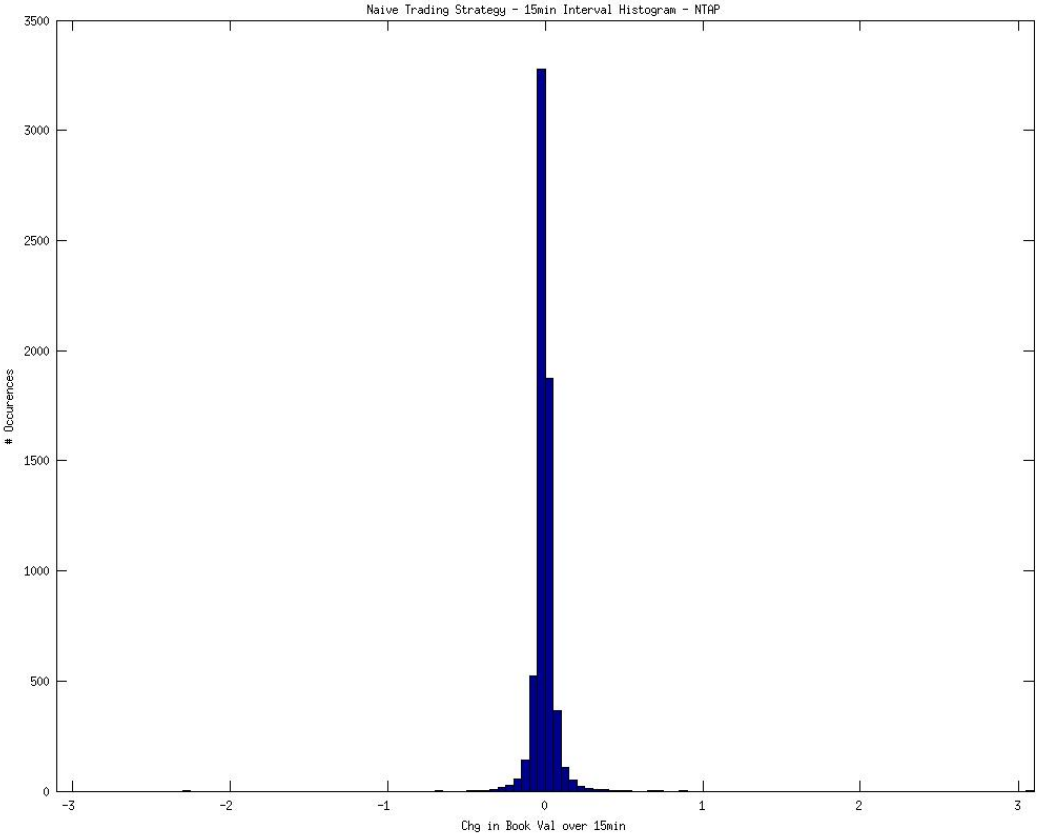


Figure 6: NTAP: Histogram of 15min bookvalue changes.

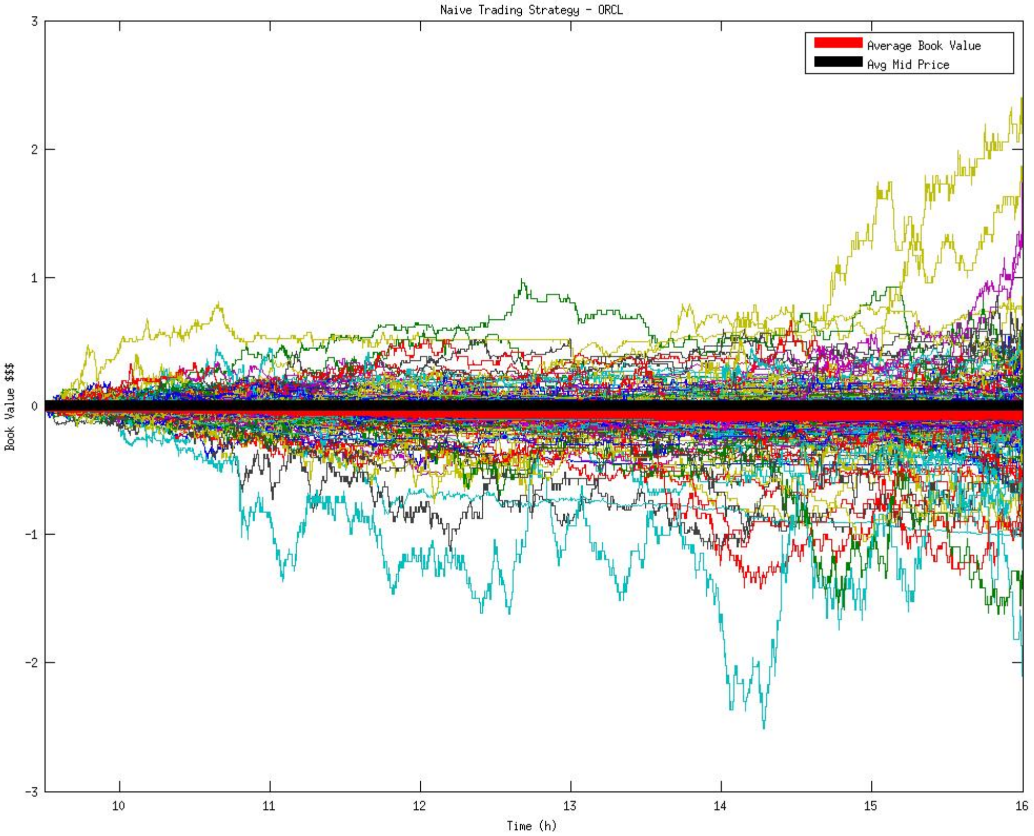


Figure 7: ORCL: Bookvalue against time of trading day.

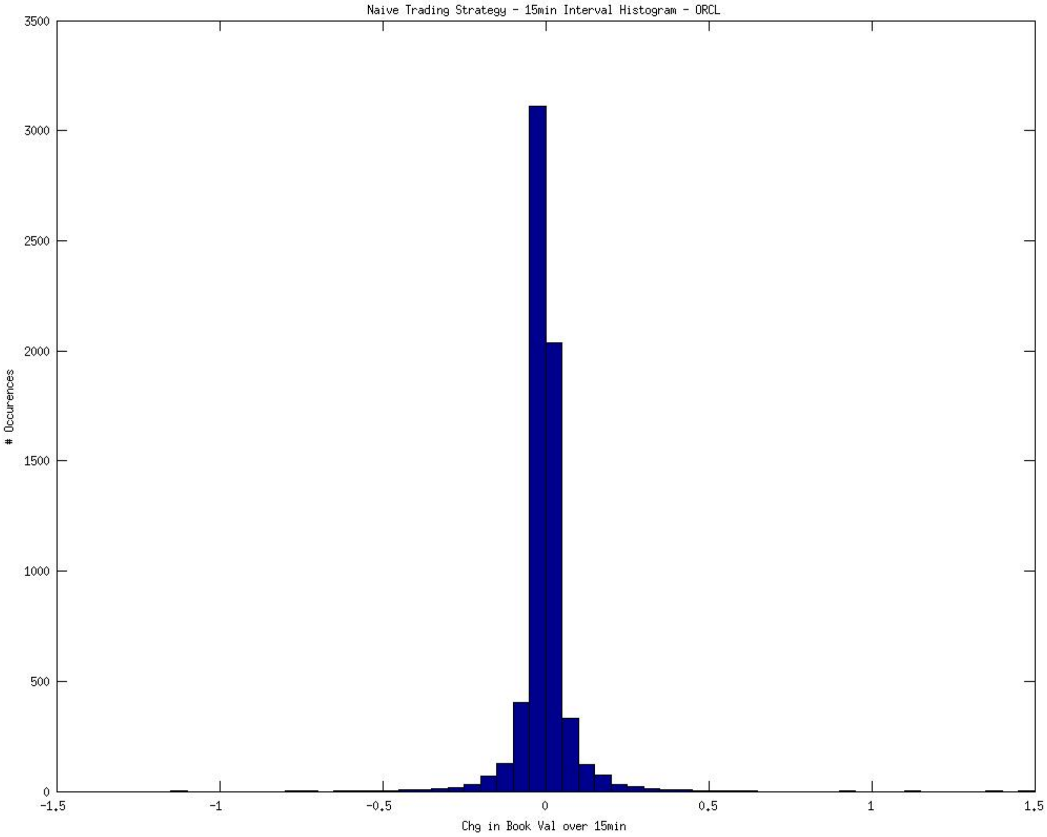


Figure 8: ORCL: Histogram of 15min bookvalue changes.

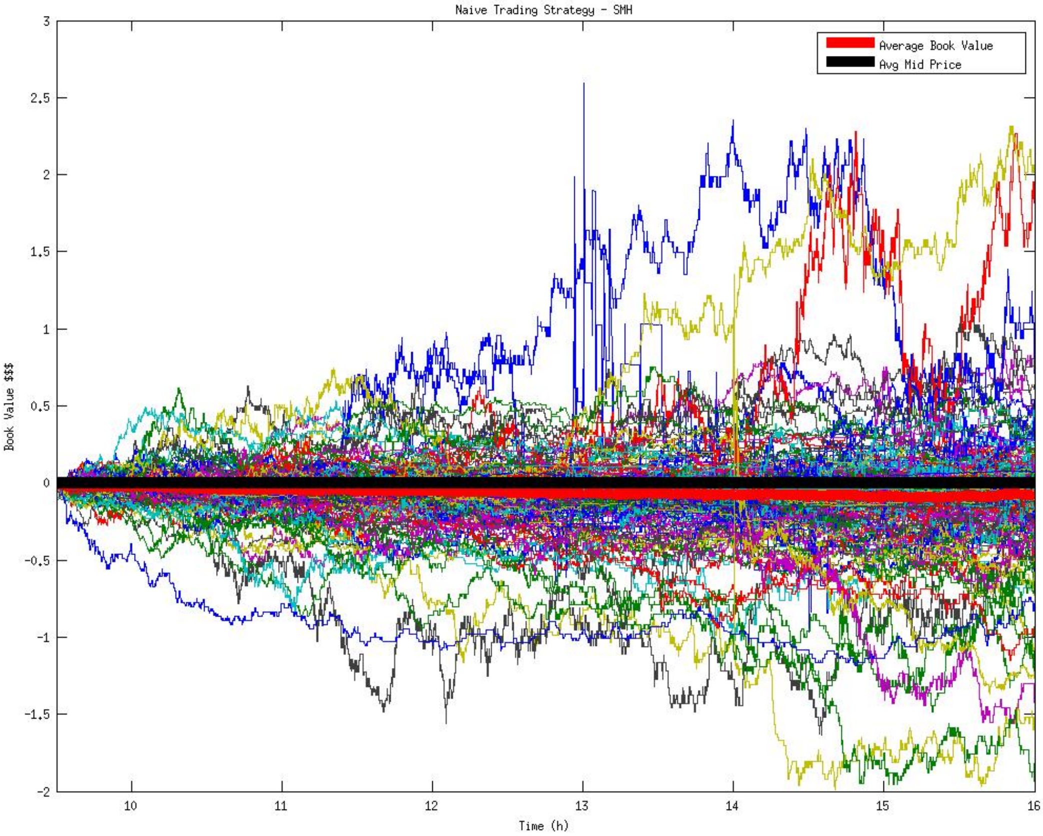


Figure 9: SMH: Bookvalue against time of trading day.

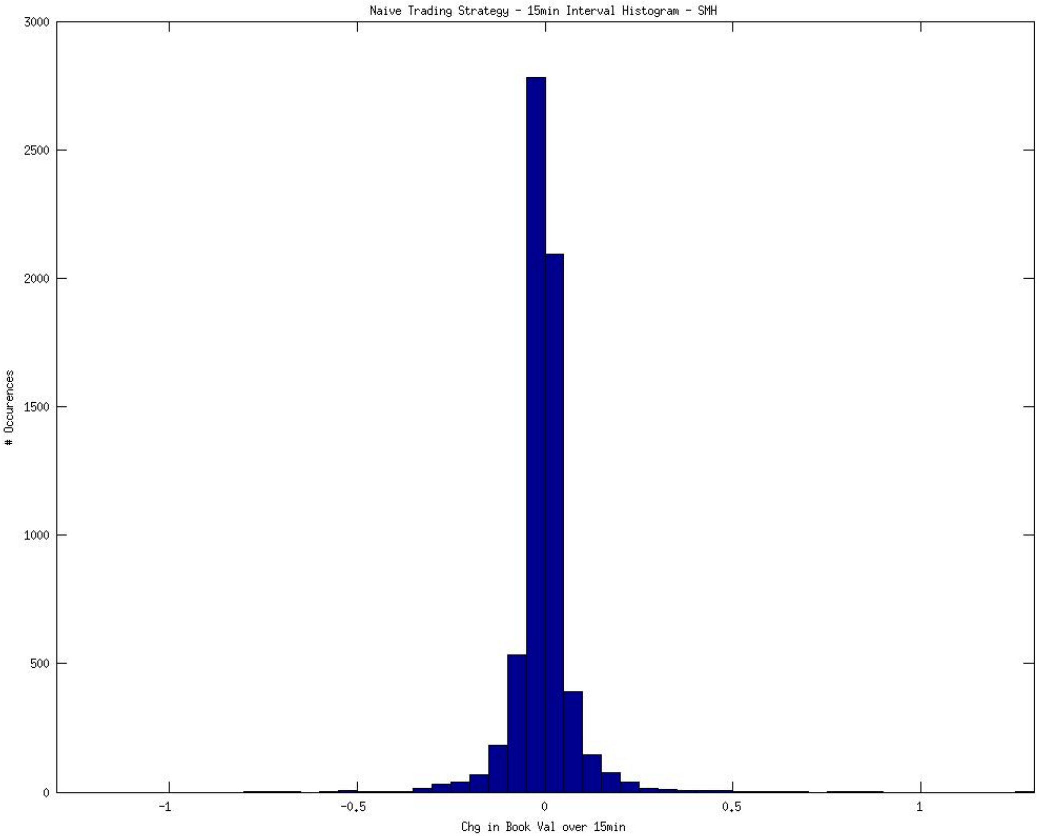


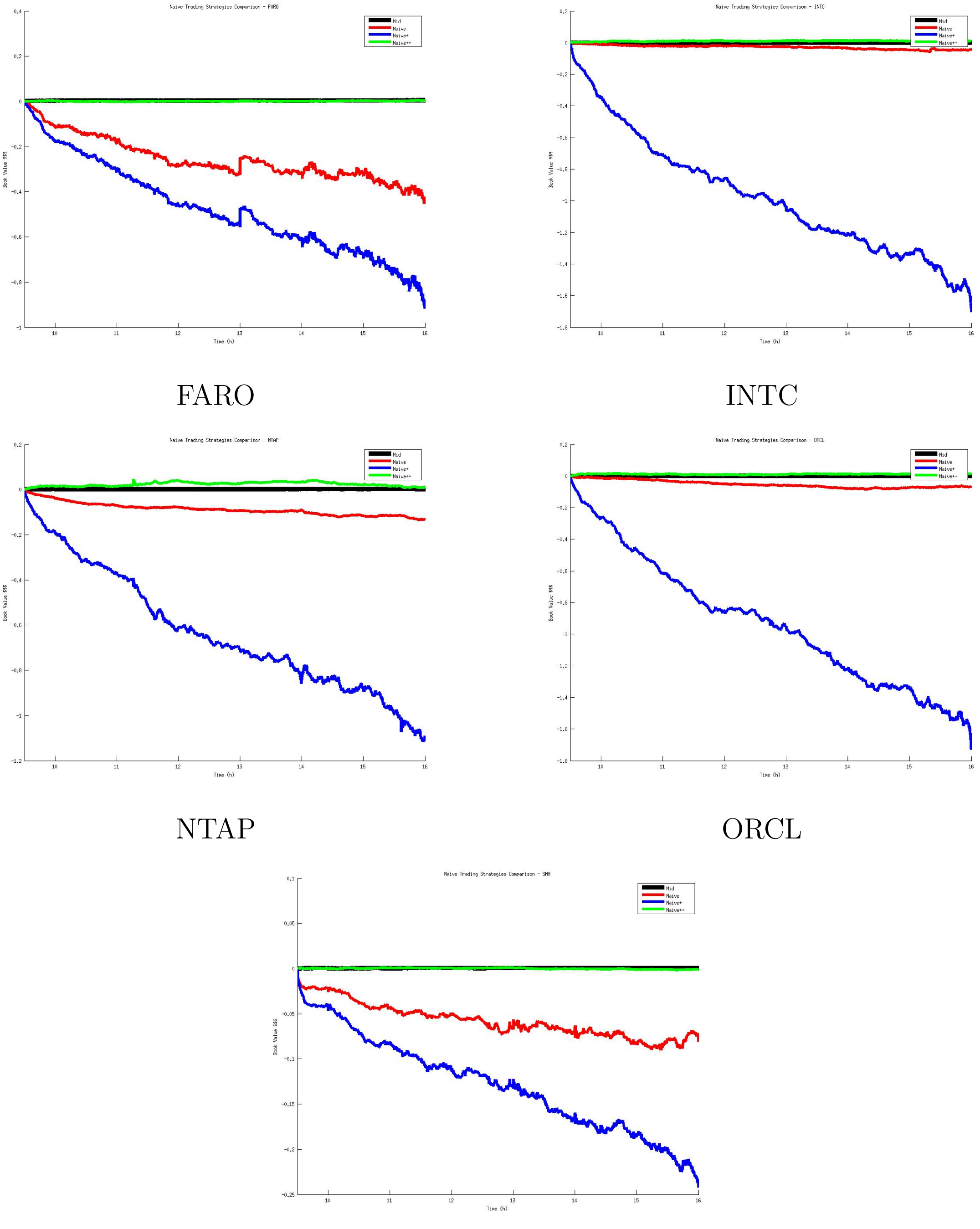
Figure 10: SMH: Histogram of 15min bookvalue changes.

Conclusions from Naive Trading Strategies

As seen in Figure 11, the Naive trading strategy on average underperformed the average mid price, was even further underperformed by the Naive+ strategy which added at-the-touch limit orders when no change was predicted, while the Naive++ strategy of adversely selecting agents that traded against the price change momentum was the only strategy to on average outperform the midprice, even if slightly.

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 intraday 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 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 is the Naive+ strategy underperforming the Naive strategy? Obviously this reduces to: why does maintaining limit orders posted at the touch produce, on average, negative returns when we are predicting no price change. In writing this summary I discovered a code error: I was transacting using the wrong bid/ask prices, which drastically alters all results. Fortunately the Naive++ strategy was did not suffer the same fate.



FARO

INTC

NTAP

ORCL

SMH

Figure 11: Comparison of Naive (red), Naive+ (blue), and Naive++ (green) trading strategies, with benchmark Midprice (black). Plotted are bookvalues against time of trading day, averaged across trading year.