A Stochastic Control Based High-Frequency Trading Algorithm

Robert Sabum

North Central College

Abstract

This paper presents a stochastic control based algorithm designed to optimize bid-ask quotes in a high-frequency trading environment. The primary objective is to maximize the net profit and loss (P&L) by effectively trading the spread, while minimizing risk through penalties on both running and terminal inventory. The model utilizes a Bachelier process to simulate mid price movements, along with Poisson processes for incoming buy and sell orders. The Hamilton-Jacobi-Bellman equation is applied to determine optimal trading strategies at each time step. Results from a simulated market show that the algorithm efficiently balances profitability and risk, achieving a 96% probability of positive returns.

1 Problem Formulation

The algorithm seeks to optimize the bid ask quotes of a market maker for each time $t < T$ such that the net Profit and Loss of trading the spread is maximized at the end of the trading session time T. We also want to control the algorithm for excessive risk taking during the trading session. Thus, we impose two penalties: (1) running inventory penalty for trading too large exposures during the trading session, and (2) terminal inventory penalty i.e. holding non-zero inventory at the end of the trading day. Mathematically, the maximization of stochastic system subject to the penalties is equivalent to maximizing the following objective:

$$V(t, s_t, x_t, q_t) = \max_{\delta^{\mathrm{b}}, \delta^{\mathrm{a}}} \mathbb{E}\left[x_T + q_T(s_T - \alpha q_T) - \phi \int_t^T q_s^2 ds \mid x_t\right]$$

where

- s_t is the mid-price of the asset at time t
- x_t is the market maker's cash at time t
- q_t is the market maker's inventory at time t
- δ^{b} is how deep in the market we quote our bid price

- δ^{a} is how deep in the market we quote our ask price
- $\alpha > 0$ is terminal inventory penalty parameter
- $\phi > 0$ is running inventory penalty parameter

Additionally, very trade pays spread and rebate of ϵ:

- If a MO hits the ask-quote of the MM: $q \leftarrow q - 1$ and $x \leftarrow x + (S + \delta^{\mathrm{a}}) + \epsilon$
- If a MO hits the bid-quote of the MM: $q \leftarrow q + 1$ and $x \leftarrow x - (S - \delta^{\mathrm{b}}) + \epsilon$

2 System Dynamics

2.1 Mid Price

The mid price is modeled by a Bachelier process, a continuous-time stochastic process characterized by a normal distribution of price changes. Specifically, the mid price S_t evolves according to the stochastic differential equation:

$$dS_t = \mu\, dt + \sigma\, dW_t$$

where μ represents the drift, or the average rate of price change over time, σdenotes the volatility, capturing the standard deviation of price fluctuations, and $W_t$is a standard Wiener process modeling the random shocks.

2.2 Incoming Order Flow

Incoming market orders at the mid price are represented via two poisson processes with parameters λ^{b} and λ^{s}. Additionally, the further away we quote from the mid price, the less likely we are to find incoming orders. Therefore, we use the following functions to model orderflow decay as we move further away from the mid price:

$$\lambda_t^{\mathrm{b}}(\delta_t^{\mathrm{b}}) = \lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}} \delta_t^{\mathrm{b}}}$$
$$\lambda_t^{\mathrm{a}}(\delta_t^{\mathrm{a}}) = \lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}} \delta_t^{\mathrm{a}}}$$

with κ^{b} and κ^{a} being positive constants indicating the orderflow decay as we move further away from the mid price. as we can see this function is decreasing with an increase in δ.

2.3 Inventory Process

The inventory process is modeled by two independent Poisson processes representing the arrival of buy and sell orders. The change in inventory at any time t, denoted by dQ_t, evolves as the difference between the cumulative number of buy orders and sell orders at time t:

$$dQ_t = N_t^{\mathrm{b}} - N_t^{\mathrm{a}}$$

where $N_t^{\mathrm{b}}$and $N_t^{\mathrm{a}}$are the respective Poisson processes for buy and sell orders whose intensities are λ_t^{b} and λ_t^{s} as defined in the previous section.

2.4 Cash Process

The cash process tracks the net earnings from executed trades on both the ask and bid sides, incorporating both the revenue from selling and the costs from buying, along with a rebate for each executed trade. Let the price at which a market maker sells (ask) be denoted by $S_t + \delta^{\mathrm{a}}$ and the price at which they buy (bid) by $S_t - \delta^{\mathrm{b}}$. The change in the cash process at time t, denoted by dX_t, is given by:

$$dX_t = (S_t + \delta^{\mathrm{a}} + \epsilon)dN_t^{\mathrm{a}} - (S_t - \delta^{\mathrm{b}} + \epsilon)dN_t^{\mathrm{b}}$$

with $N_t^{\mathrm{a}}$and N_t^{b}representing the number of sell and buy orders, respectively, and ϵ is the rebate earned for each executed trade. This process accounts for the gains from selling at the ask, the costs from buying at the bid, and the cumulative rebates from providing liquidity.

3 Solving the control problem

The problem above may be solved as a Hamilton-Jacobi-Bellman Quasi Variational Inequality where the market maker decides between (1) market making, (2) sending a market order to buy one contract or (3) sending a market order to sell one contract. Applying standard Dynamic Programming Principle to $V(t, x_t, q_t, s_t)$ tells us that V must satisfy the following HJB-QVI:

$$0 = max\left(\frac{\partial V}{\partial t} + \underset{\delta^b,\delta^a}{max}\left[\mathcal{L}V(t) - \phi q_t^2\right]; \mathcal{S}V - V; \mathcal{B}V - V\right)$$

where the operators $\mathcal{S}$ and $\mathcal{B}$ act on $V(t, x_t, q_t, s_t)$ such that

$$\mathcal{S}V(t) = V(t, x_t - \xi, q_t - 1, s_t)$$
$$\mathcal{B}V(t) = V(t, x_t - \xi, q_t + 1, s_t)$$

In other words, they represent (in abstract sense) the cost of taking liquidity.

3.0.1 HJB Part

We begin by examining the HJB part of the inequality:

$$\frac{\partial V}{\partial t} + \underset{\delta^b,\delta^a}{max}\left[\mathcal{L}V(t) - \phi q_t^2\right]$$

3.0.2 Derivation of $\mathcal{L}V(t,x,s,q)$:

Let's recall the definition of the infinitesimal generator:

$$\mathcal{L}V(t,x,s,q) = \lim_{dt\to 0} \frac{\mathbb{E}\big[\partial V(t,x,s,q)\big]}{dt}$$

We start by applying Ito's lemma to $V(t,x,s,q)$:

$$\begin{aligned}\partial V(t,x,s,q) = \partial_t V dt + \partial_s V dS_t + \frac{1}{2}\partial_{ss} V dS_t^2 \\ + \big[V(t, x-(s-\delta^{\mathrm{b}})+\epsilon, s, q+1) - V(t,x,s,q)\big] dN_t^{\mathrm{b}} \\ + \big[V(t, x+(s+\delta^{\mathrm{a}})+\epsilon, s, q-1) - V(t,x,s,q)\big] dN_t^{\mathrm{a}}\end{aligned}$$

Since we assume Bachelier type model for the spot price, $dS_t = \mu dt + \sigma dW_t$ and $dS_t^2 = \sigma^2 dt$. Substituting these into the above gives:

$$\begin{aligned}\partial V(t,x,s,q) = \partial_t V dt + \partial_s V(\mu dt + \sigma dW_t) + \frac{1}{2}\partial_{ss} V \sigma^2 dt \\ + \big[V(t, x-(s-\delta^{\mathrm{b}})+\epsilon, s, q+1) - V(t,x,s,q)\big] dN_t^{\mathrm{b}} \\ + \big[V(t, x+(s+\delta^{\mathrm{a}})+\epsilon, s, q-1) - V(t,x,s,q)\big] dN_t^{\mathrm{a}}\end{aligned}$$

Finally, taking expectation and dividing by dt yields:

$$\begin{aligned}\mathcal{L}V(t,x,s,q) = \partial_t V + \partial_s V \mu + \frac{1}{2}\partial_{ss} V \sigma^2 \\ + \lambda_t^{\mathrm{b}}\big[V(t, x-(s-\delta^{\mathrm{b}})+\epsilon, s, q+1) - V(t,x,s,q)\big] \\ + \lambda_t^{\mathrm{a}}\big[V(t, x+(s+\delta^{\mathrm{a}})+\epsilon, s, q-1) - V(t,x,s,q)\big]\end{aligned}$$

Since we made the explicit assumption that $\lambda_t = \lambda e^{-\kappa\delta}$ we get that:

$$\begin{aligned}\mathcal{L}V(t,x,s,q) = \partial_t V + \partial_s V \mu + \frac{1}{2}\partial_{ss} V \sigma^2 \\ + \lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}}\big[V(t, x-(s-\delta^{\mathrm{b}})+\epsilon, s, q+1) - V(t,x,s,q)\big] \\ + \lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}}\big[V(t, x+(s+\delta^{\mathrm{a}})+\epsilon, s, q-1) - V(t,x,s,q)\big]\end{aligned}$$

HJB equation:

$$\begin{aligned}0 = \partial_t V + \partial_s V \mu + \frac{1}{2}\partial_{ss} V \sigma^2 \\ + \max_{\delta^{\mathrm{b}}}\big(\lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}}\big[V(t, x-(s-\delta^{\mathrm{b}})+\epsilon, s, q+1) - V(t,x,s,q)\big]\big) \\ + \max_{\delta^{\mathrm{a}}}\big(\lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}}\big[V(t, x+(s+\delta^{\mathrm{a}})+\epsilon, s, q-1) - V(t,x,s,q)\big]\big) \\ - \phi q^2\end{aligned}$$

Since the terminal cash X_T, is dependent on cumulative trading profits, we can simlify the PDE as follows:

$$\begin{aligned}
0 &= \partial_t V + \partial_s V \mu + \frac{1}{2}\partial_{ss} V \sigma^2 \\
&+ \max_{\delta^{\mathrm{b}}}\left(\lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}}\left[\delta^{\mathrm{b}} + \epsilon - s + V(t,s,q+1) - V(t,s,q)\right]\right) \\
&+ \max_{\delta^{\mathrm{a}}}\left(\lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}}\left[\delta^{\mathrm{a}} + \epsilon + s + V(t,s,q-1) - V(t,s,q)\right]\right) \\
&- \phi q^2
\end{aligned}$$

We can further simplify the PDE by observing that the maximization of δ^{b} and δ^{a} is independent of s_t. Therefore, V becomes a function of only q and t, removing the need of our spatial derivatives with respect to s:

$$\begin{aligned}
0 &= \partial_t V \\
&+ \max_{\delta^{\mathrm{b}}}\left(\lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}}\left[\delta^{\mathrm{b}} + \epsilon + V(t,q+1) - V(t,q)\right]\right) \\
&+ \max_{\delta^{\mathrm{a}}}\left(\lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}}\left[\delta^{\mathrm{a}} + \epsilon + V(t,q-1) - V(t,q)\right]\right) \\
&- \phi q^2
\end{aligned}$$

with boundry condition $V(T, q_T) = -\alpha q_T^2$.

3.1 Optimal Bid and Ask Depths

The optimal bid and ask quotes are calculated by solving the following optimization problems at each time step t:

$$\begin{aligned}
&\operatorname*{argmax}_{\delta^{\mathrm{b}}}\left[\lambda^{\mathrm{b}} e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}}\left[\delta^{\mathrm{b}} + \epsilon + V(t,q+1) - V(t,q)\right]\right] \\
&\operatorname*{argmax}_{\delta^{\mathrm{a}}}\left[\lambda^{\mathrm{a}} e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}}\left[\delta^{\mathrm{a}} + \epsilon + V(t,q-1) - V(t,q)\right]\right]
\end{aligned}$$

we can solve the above by differentiating with respect to δ^{b} and δ^{a} and setting the derivative to zero. Additionally we constrain our bid and ask depths to values between 0 and ∞. This gives:

$$\begin{aligned}
\delta^{\mathrm{b}} &= \max(0, \frac{1}{\kappa^{\mathrm{b}}} - \epsilon - V(t,q+1) + V(t,q)) \\
\delta^{\mathrm{a}} &= \max(0, \frac{1}{\kappa^{\mathrm{a}}} - \epsilon - V(t,q-1) + V(t,q))
\end{aligned}$$

4 Numerical Computation

We may approximate the solution to the HJB equation numerically using a backwards finite difference scheme by discretizing the time derivative:

$$\partial_t V(t,q) \approx \frac{V(t,q) - V(t-\Delta t, q)}{\Delta t}$$

Substituting this into the HJB equation gives:

$$\begin{aligned}
0 = &\frac{V(t,q) - V(t-\Delta t, q)}{\Delta t} \\
&+ \lambda_0^{\mathrm{b}} \max_{\delta^{\mathrm{b}}} \left(e^{-\kappa^{\mathrm{b}} \delta^{\mathrm{b}}} \left[\delta^{\mathrm{b}} + \epsilon + V(t, q_t + 1) - V(t, q_t) \right] \right) \\
&+ \lambda_0^{\mathrm{a}} \max_{\delta^{\mathrm{a}}} \left(e^{-\kappa^{\mathrm{a}} \delta^{\mathrm{a}}} \left[\delta^{\mathrm{a}} + \epsilon + V(t, q_t - 1) - V(t, q_t) \right] \right) \\
&- \phi q_t^2
\end{aligned}$$

Solving for $V(t - \Delta t, q)$ gives us:

$$\begin{aligned}
V(t-\Delta t, q) = &V(t,q) \\
&+ \lambda^{\mathrm{b}} \max_{\delta^{\mathrm{b}}} \left(e^{-\kappa^{\mathrm{b}} \delta^{\mathrm{b}}} \left[\delta^{\mathrm{b}} + \epsilon + V(t, q_t + 1) - V(t, q_t) \right] \right) \Delta t \\
&+ \lambda^{\mathrm{a}} \max_{\delta^{\mathrm{a}}} \left(e^{-\kappa^{\mathrm{a}} \delta^{\mathrm{a}}} \left[\delta^{\mathrm{a}} + \epsilon + V(t, q_t - 1) - V(t, q_t) \right] \right) \Delta t \\
&- \phi q_t^2 \Delta t
\end{aligned}$$

Now, we define q-grid such that $q_j \in \{q_{\min}, q_{\min} + 1, ..., q_{\max} - 1, q_{\max}\}$ and a t-grid where $t_i \in \{0, \Delta t, ..., T - 2\Delta t, T - \Delta t, T\}$. The following solves $V(t_i, q_j)$ over the grid:

$$\begin{aligned}
V(t_{i-1}, q_j) = &V(t_i, q_j) \\
&+ 1_{q_j < q_{\max}} \lambda^{\mathrm{b}} \max_{\delta^{\mathrm{b}}} \left(e^{-\kappa^{\mathrm{b}} \delta^{\mathrm{b}}} \left[\delta^{\mathrm{b}} + \epsilon + V(t_i, q_j + 1) - V(t_i, q_j) \right] \right) \Delta t \\
&+ 1_{q_j > q_{\min}} \lambda^{\mathrm{a}} \max_{\delta^{\mathrm{a}}} \left(e^{-\kappa^{\mathrm{a}} \delta^{\mathrm{a}}} \left[\delta^{\mathrm{a}} + \epsilon + V(t_i, q_j - 1) - V(t_i, q_j) \right] \right) \Delta t \\
&- \phi q_j^2 \Delta t
\end{aligned}$$

Additionally at each time step t_i, we need to consider the maximum between our three options:

$$V(t_i, q_t) = \max\{V(t_i, q_j); \mathcal{S}V(t_i, q_j); \mathcal{B}V(t_i, q_j)\}$$

Due to the overlapping nature of different inventory levels, we will solve the inequality using an iterative scheme where we repeatedly update the value of $V(t_i, q_j)$ at each point until convergence.

Algorithm 1 Optimize trading strategy by solving the HJB-QVI

procedure OPTIMIZE($T, N, q_{\min}, q_{\max}, \lambda^{\mathrm{b}}, \lambda^{\mathrm{a}}, \kappa^{\mathrm{b}}, \kappa^{\mathrm{a}}, \epsilon, \xi, \phi, \alpha$)

  $\Delta t \leftarrow TN^{-1}$

  $T \leftarrow \{0, \Delta t, 2\Delta t, \ldots, N\Delta t\}$

  $Q \leftarrow \{q_{\min}, q_{\min}+1, \ldots, q_{\max}\}$

  $V(t,q) \leftarrow 0$

  $U(t,q) \leftarrow 0$

  $V(T, q_j) \leftarrow -\phi q_j^2$ for $q_j \in Q$

  for $i \leftarrow N, 1$ **do**

    $t_i \leftarrow i\Delta t$

    $t_{i-1} \leftarrow (i-1)\Delta t$

    $\beta \leftarrow \infty$

    while $\beta > \beta_{\max}$ **do**

      $V'(t,q) \leftarrow 0$

      for all $q_j \in Q$ **do**

        $\delta^{\mathrm{b}} \leftarrow \max\left(0, (\kappa^{\mathrm{b}})^{-1} - \epsilon - V(t_i, q_j+1) + V(t_i, q_j)\right)$

        $\delta^{\mathrm{a}} \leftarrow \max\left(0, (\kappa^{\mathrm{a}})^{-1} - \epsilon - V(t_i, q_j+1) + V(t_i, q_j)\right)$

        $V_{\mathcal{B}} \leftarrow V(t_{i-1}, q_j+1) - \xi$

        $V_{\mathcal{S}} \leftarrow V(t_{i-1}, q_j-1) - \xi$

        $V_{\mathcal{M}} \leftarrow V(t_i, q_j) - \phi q_j^2 \Delta t$

        $V_{\mathcal{M}} \leftarrow V_{\mathcal{M}} + 1_{q_j < q_{\max}} \lambda^{\mathrm{b}} \left(e^{-\kappa^{\mathrm{b}}\delta^{\mathrm{b}}} \left[\delta^{\mathrm{b}} + \epsilon + V(t_i, q_j+1) - V(t_i, q_j)\right]\right) \Delta t$

        $V_{\mathcal{M}} \leftarrow V_{\mathcal{M}} + 1_{q_j > q_{\min}} \lambda^{\mathrm{a}} \left(e^{-\kappa^{\mathrm{a}}\delta^{\mathrm{a}}} \left[\delta^{\mathrm{a}} + \epsilon + V(t_i, q_j-1) - V(t_i, q_j)\right]\right) \Delta t$

        $V'(t_{i-1}, q_j) \leftarrow \max(V_{\mathcal{B}}, V_{\mathcal{S}}, V_{\mathcal{M}})$

        $U(t_{i-1}, q_j) \leftarrow \underset{\mathcal{B},\mathcal{S},\mathcal{M}}{\operatorname{argmax}}(V_{\mathcal{B}}, V_{\mathcal{S}}, V_{\mathcal{M}})$

      end for

      $\beta \leftarrow \sqrt{\sum_{q_j} (V'(t_{i-1}, q_j) - V(t_{i-1}, q_j))^2}$

      $V(t,q) \leftarrow V'(t,q)$

    end while

  end for

  return $V(t,q), U(t,q)$

end procedure

5 Results

The algorithm was tested within a simulated market environment using specific parameters. The simulation ran for a total time of $T = 60$ minutes, with the number of time steps N set to $T \times 60$, representing the number of seconds. The initial stock price, $s_0 = 100$, followed a Bachelier process with a drift of $\mu = 0$ and volatility $\sigma = 0.01$. The market maker's initial inventory $q_0 = 0$, constrained between $q_{\min} = -50$ and $q_{\max} = 50$. The arrival rates for buy and sell orders at the mid-price were set to $\lambda^{\mathrm{b}} = 50$ and $\lambda^{\mathrm{a}} = 50$, with decay rates of $\kappa^{\mathrm{b}} = 100$ and $\kappa^{\mathrm{a}} = 100$ as quotes moved further from the mid-price. Additional parameters included the liquidity rebate rate $\epsilon = 0.0025$, the transaction cost for submitting market orders $\xi = 0.005$, the running inventory penalty rate $\phi = 0.01$, and the terminal inventory penalty rate $\alpha = 0$.

5.1 Optimal Value Function

The optimal value function exhibits a linear relationship with both time and inventory. As time progresses towards the terminal point, the value function decreases, likely because there is less time available for trading, which reduces the opportunities for generating profits. Regarding inventory, the optimal value remains relatively constant across different levels, despite the running inventory penalty. This constancy can be attributed to the high penalty imposed for deviating from zero inventory, which incentivizes the market maker to submit market orders when inventory strays too far from zero, quickly reverting to a neutral position before continuing to market make.

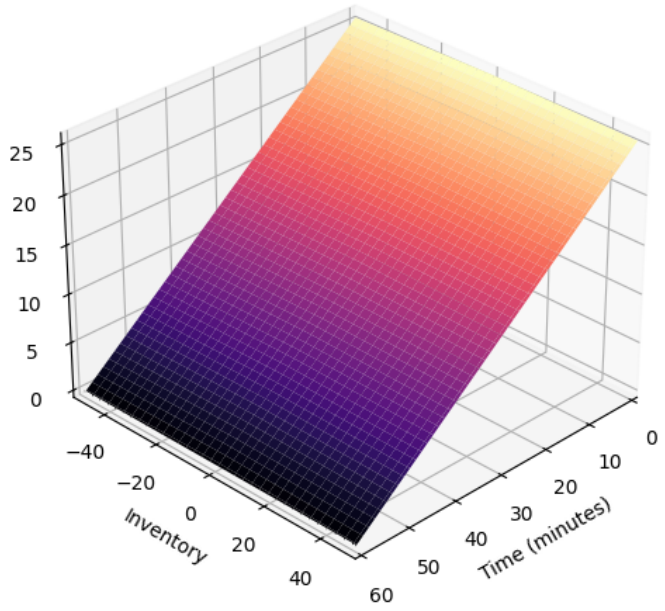
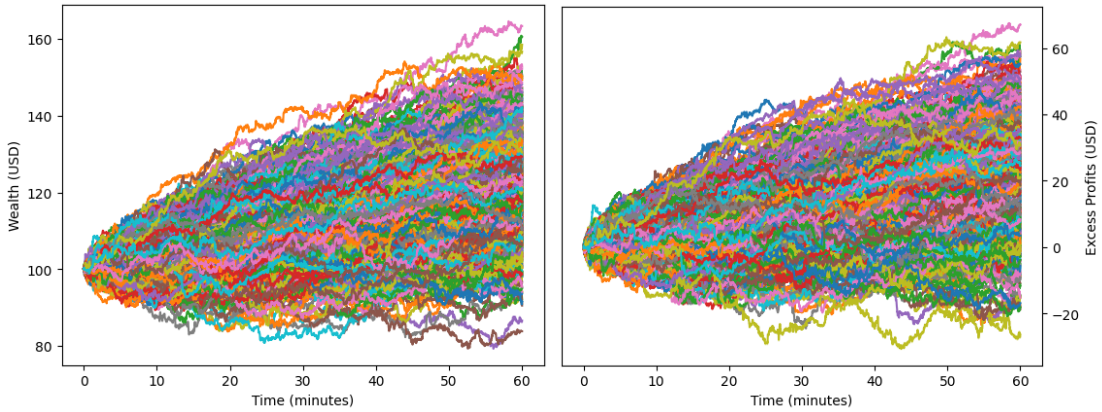


Figure 1: Optimal Value Function

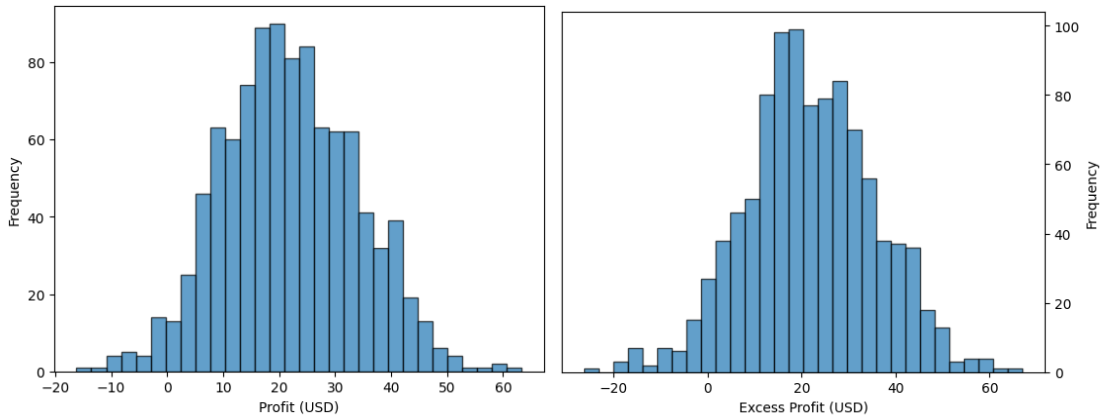
5.2 Monte Carlo Simulations

We ran 1,000 Monte Carlo simulations in our simulated trading environment. The figures below show the raw performance of the market maker's strategy as well as its performance relative to the performance of the underlying asset.



(a) Sampled Wealth Trajectories (b) Sampled Excess Wealth Trajectories

Figure 2: Sampled wealth trajectories



(a) Final Total Profit Distribution (b) Final Total Excess Profit Distribution

Figure 3: Final total profit distributions

We can see that our total and excess final profit have a mean of about $21 over a one hour time frame, both 20% higher than the performance of the underlying asset. Further more the standard deviations were $12.01 and $13.85 respectively. This risk could be further reduced by increasing the running inventory penalty, however this would come at a cost of decreased profitability due to sending frequent market orders. Based on the mean and standard deviations in profits we can see that the strategy is profitable with a probability of approximately 96.5% according to the cumulative density function of the corresponding normal distribution.

6 Conclusion

In conclusion, the market-making algorithm developed in this paper demonstrates its effectiveness in a simulated high-frequency trading environment. By leveraging stochastic models for price dynamics and order arrivals, the algorithm optimizes bid-ask quotes while penalizing excessive inventory holdings. The balance between profitability and risk management is achieved through careful parameterization of inventory penalties and liquidity rebates. Monte Carlo simulations further confirm that the strategy generates consistent profits with a high degree of certainty, highlighting its potential applicability in real-world trading scenarios.

An exciting avenue for a future project is a more sophisticated models for price dynamics, such as jump-diffusion or stochastic volatility processes, could be explored to better capture real-world market behaviors and extreme price movements. Another potential direction involves incorporating adaptive, real-time learning mechanisms into the trading algorithm, allowing it to respond dynamically to market conditions such as volatility spikes or liquidity shifts. The use of machine learning to approximate solutions to the Hamilton-Jacobi-Bellman equation could also be pursued further, with a focus on scalability and eliminating the curse of dimensionality from grid-based methods. Additionally, future work could integrate richer market dynamics, including detailed order book modeling and interactions between traders, potentially by leveraging mean field control theory to simulate diverse market participant behaviors. Finally, expanding the risk management framework beyond inventory penalties to include stop-loss mechanisms or dynamic hedging strategies could improve the algorithm's robustness under various market conditions, further enhancing its applicability to high-frequency trading environments.