Limit Order Book Simulation

Andrew Braun, Evan Strasdin

December 2025

1 Problem Description and Goal of the Simulation Study

Modern electronic financial markets match buy and sell interest through a *limit order book* (LOB), which aggregates outstanding limit orders at discrete price levels and executes trades whenever compatible buy and sell orders meet. The microscopic dynamics of the LOB—arrival of new limit and market orders, cancellations of resting orders, and the quoting behaviour of liquidity providers—jointly determine key measures of market quality such as bid–ask spreads, execution latency, depth, and price volatility. Analytical models for these dynamics are typically intractable once realistic features such as price–time priority, stochastic order sizes, and strategic market making are introduced, which motivates the use of discrete-event simulation.

In this project we construct a continuous-time discrete-event simulation of a single-asset LOB with a single quoting market maker. Order flow on both sides of the book is driven by independent Poisson processes for limit orders, market orders, and cancellations, and the LOB is represented as two priority queues that enforce price–time priority. On top of this background order flow, a market maker posts bid and ask quotes at regular intervals according to a simple inventory-based strategy: quotes are placed symmetrically around the midprice with a configurable base spread and are skewed as a linear function of the current inventory.

The simulation study focuses on three structural components of the system:
(i) an *order strategy* parameter controlling the mean size of incoming orders,
(ii) a *market structure* parameter controlling how tightly limit order prices are distributed around the prevailing midprice, and
(iii) a *market maker strategy* parameter controlling the aggressiveness of the market maker's quotes.
By varying these components over a factorial grid of scenarios and running multiple independent replications in each scenario, the goal is to quantify how they affect market quality and market maker performance. Concretely, the study aims to estimate and compare average spreads, execution-time distributions, trading activity, LOB depth profiles, price volatility, and market maker profit and inventory risk across scenarios, and to draw data-driven conclusions about

which combinations of structural parameters lead to more efficient and stable markets.

2 Simulation Model

The simulated market is a continuous-time, double-auction system in which all state changes arise from discrete events: arrivals of limit orders, arrivals of market orders, cancellations of resting orders, and periodic submissions of market-maker quotes. The core of the model is the limit order book (LOB), which we represent explicitly as a pair of priority queues. This representation provides a clean mapping from market microstructure rules-specifically, price–time priority-to well-understood queueing constructs.

2.1 Mapping to a Priority Queueing Model

At any time t, the LOB maintains two ordered sets of outstanding limit orders:

$$\mathcal{B}(t) = \{\text{resting buy orders}\}, \qquad \mathcal{A}(t) = \{\text{resting sell orders}\}.$$

Each order is defined by its side (buy or sell), limit price, creation time, remaining quantity, and an identifier labeling the trader who submitted it. Because execution in electronic markets follows price–time priority, orders at more competitive prices must be considered first, and ties at the same price are broken by arrival time.

To encode these priorities efficiently, we treat $\mathcal{B}(t)$ as a *max-heap* keyed lexicographically by

$$(-\text{price},\ \text{time},\ \text{order\_id}),$$

so that the order with the highest price and earliest creation time always sits at the top. Analogously, $\mathcal{A}(t)$ is a *min-heap* keyed by

$$(\text{price},\ \text{time},\ \text{order\_id}),$$

ensuring that the lowest-priced, earliest-arriving sell order has highest priority. This mapping converts the LOB into a two-server priority queueing system in which the "service" operation corresponds to trade execution: whenever the book becomes crossed, meaning

$$P_{\text{best bid}}(t) \ \geq \ P_{\text{best ask}}(t),$$

the simulation executes one unit of quantity by removing one unit from the highest-priority buy and sell orders. Because the events are processed in continuous time, the queues evolve dynamically as orders arrive, are partially or fully executed, or are cancelled.

2.2 Order Dynamics and Event Types

The system evolves through five independent Poisson arrival processes:

- limit buy arrivals,
- limit sell arrivals,
- market buy arrivals,
- market sell arrivals,
- cancellations of resting orders.

If the next event at time t is a limit-order arrival, the simulation draws a limit price by sampling an exponentially distributed distance from the prevailing midprice and rounding it to the nearest tick. Market-order arrivals are modeled as sequences of one-unit executions against the opposite queue, again following priority rules.

Cancellations are modeled as abandoning the current best order on a randomly selected side of the book. Although simple, this abstraction corresponds to impatient traders withdrawing orders and has the practical effect of preventing the book from accumulating stale liquidity that might distort spread or execution-time statistics.

2.3 Market Maker Behaviour

A single market maker (MM) participates alongside the stochastic background order flow. At fixed intervals, the MM observes the current midprice,

$$M(t) = \tfrac{1}{2}\big(P_{\text{bid}}(t) + P_{\text{ask}}(t)\big),$$

and submits one-unit buy and sell limit orders at prices

$$M(t) - \delta + S(t) \qquad \text{and} \qquad M(t) + \delta + S(t),$$

where δ is a configurable base spread and $S(t)$ is an inventory-skew term proportional to the MM's current inventory. This state-dependent rule embeds a stabilizing mechanism: excessive inventory induces the MM to place quotes further away from the midprice on the side that reduces exposure.

When MM orders are executed, inventory and cash are updated immediately, and mark-to-market P&L is computed using the most recent trade price. This mechanism allows the simulation to evaluate the profitability and risk of the MM under different market conditions.

2.4 Discrete-Event Representation

All components described above are integrated into a discrete-event simulation (DES). Each event updates the system state and may trigger additional "service completions" (trades) if the book becomes crossed. The priority-queue formulation ensures that each such trade corresponds precisely to serving the highest-priority items in the two queues. Thus, the DES framework provides a faithful and computationally efficient representation of LOB microstructure while allowing controlled experimentation over structural parameters.

3 Simulation Parameters

The simulation operates under a set of fixed environmental parameters and a controlled set of experimental factors. These parameters govern the arrival patterns of orders, the behaviour of the market maker, the pricing rules for limit orders, and the overall duration and granularity of the simulated market. All simulations share the same baseline configuration, ensuring that differences in performance metrics arise only from the structural factors varied in the experimental design.

3.1 Fixed Parameters

Several parameters are held constant across all scenarios in order to establish a well-defined and realistic market environment:

- **Trading horizon.**
  Each simulation run spans 23,400 seconds (6.5 hours), corresponding to a stylized full trading session. This interval is long enough to generate substantial trading volume and to stabilize estimates of spreads, execution times, volatility, and market-maker performance.

- **Price discretization.**
  All prices evolve on a discrete grid with a tick size of 0.01. Limit prices generated by the model's stochastic pricing rule are rounded to the nearest tick, ensuring consistent granularity across scenarios.

- **Initial conditions.**
  At time $t = 0$, the order book is seeded with a minimal buy order at 99.99 and a sell order at 100.01. This guarantees a valid initial midprice and a strictly positive spread, avoiding undefined behaviour before the first stochastic event.

- **Order-flow intensities.**
  The market is driven by five independent Poisson processes with fixed rates:
  $$\lambda_{\text{limit buy}} = 0.6, \quad \lambda_{\text{limit sell}} = 0.6,$$
  $$\lambda_{\text{market buy}} = 0.2, \quad \lambda_{\text{market sell}} = 0.2, \quad \lambda_{\text{cancel}} = 0.1.$$

These values produce an active market in which limit order submissions dominate market orders, and cancellations occur frequently enough to prevent unrealistic liquidity accumulation.

- **Market-maker quoting rules.** The market maker posts new buy and sell quotations every 10 seconds. Quotes are placed symmetrically around the prevailing midprice and include an inventory-based adjustment term governed by a fixed skew coefficient of 0.05. This induces stabilizing behaviour by encouraging the market maker to reduce excess inventory.

Collectively, these fixed parameters define a neutral market environment that allows direct comparison of structural changes introduced by the experimental factors.

3.2 Experimental Factors

The study examines three structural components of the market microstructure, each varied over five levels. Together, these form a full $5 \times 5 \times 5$ factorial design containing 125 distinct scenarios:

Component	Parameter Description	Levels
Order Strategy	Mean submitted order size	$\{1, 5, 10, 20, 30\}$
Market Structure	Decay rate of limit-price distribution	$\{0.05, 0.10, 0.20, 0.30, 0.50\}$
MM Strategy	Base half-spread of market-maker quotes	$\{0.10, 0.25, 0.50, 1.00, 2.00\}$

These factors were selected to span regimes of practical interest:

- Small order-size means represent highly granular, retail-like trading, while larger means create environments with deeper but more volatile liquidity.
- Small decay rates produce wide books with orders dispersed far from the midprice, whereas large decay rates produce very tight books with intense competition for queue priority.
- The market maker's base spread determines the aggressiveness of its liquidity provision strategy, directly affecting spreads, fill rates, and profitability.

The full factorial grid permits the isolation of main effects and interactions, enabling a systematic study of how structural market features interact with strategic quoting behaviour.

3.3 Replication Design

To obtain reliable statistical estimates, each scenario is simulated with five independent replications. This ensures that scenario-level performance metrics (means and confidence intervals) are supported by independent samples rather than single-run outcomes.

3.4 Common Random Numbers (CRN)

A common random numbers design is employed across the experiment: the same sequence of background randomness is used across all scenarios within a given replication. That is, replication r uses a shared random-number seed for all 125 scenarios. This induces strong pairing between scenarios, sharply reducing variance in scenario comparisons and yielding more sensitive detection of factor effects.

4 Methodology

The simulation is implemented as a continuous-time, discrete-event model in which all market activity is generated endogenously by stochastic event processes and by the quoting behaviour of the market maker. This section describes the structure of the simulation engine, the flow of events, and the procedures used to generate limit prices, track executions, and collect performance statistics. The description corresponds directly to the logic executed in the underlying code, but is presented in conceptual rather than implementation-specific terms.

4.1 Discrete-Event Flow

The simulation maintains a global event clock and advances time by processing whichever event is scheduled to occur next. At the start of each replication, the clock is set to zero, the order book is initialized with a minimal buy and sell order to establish an initial midprice, and candidate arrival times are generated for all stochastic event streams. The simulation then proceeds according to the following loop:

1. **Identify the next event.** Each stochastic event type (limit buys, limit sells, market buys, market sells, cancellations) has an associated next-arrival time generated from an exponential distribution. A separate deterministic timer tracks the next scheduled quoting action of the market maker. The simulation selects the smallest of these times and advances the global clock to that moment.

2. **Execute the event.** Depending on the event type, one of the following actions takes place:

   - *Limit order arrival.* A new limit buy or limit sell order is created. Its limit price is generated by sampling an exponentially distributed distance from the current midprice. The sampled distance is rounded to the nearest tick and applied on the appropriate side of the book (below the midprice for buys, above for sells). The order is inserted into the priority queue corresponding to its side.
   - *Market order arrival.* A market order triggers immediate execution against the opposite side of the book. The simulation repeatedly removes one unit from the best available quote until either the market

order's required quantity is fully executed or no resting liquidity remains. Each one-unit execution is recorded as an individual trade, and the state of the limit order book is updated accordingly.

- *Cancellation.* A random side (buy or sell) is chosen, and the current best order on that side is cancelled. Cancelling only the top-of-book order simplifies the abandonment mechanism while still capturing realistic turnover of stale liquidity.
- *Market-maker quoting.* The market maker observes the best available buy and sell prices and computes the midprice. Based on this midprice, the configured base spread, and an inventory-dependent skew term, the market maker posts new one-unit buy and sell orders. These orders enter the book and are thereafter subject to the same execution and cancellation rules as all other resting orders.

After executing any event, a new next-arrival time is drawn for that event type.

3. **Clear crossed markets.** After processing an event, the simulation checks whether the best buy price is greater than or equal to the best sell price. If so, the book is crossed, and the simulation executes one unit of a trade between the best bid and best ask. This procedure continues in a loop until the book becomes uncrossed. Each trade updates order quantities, removes inactive orders, adjusts market-maker inventory and cash (if the MM participated), and records data needed for execution-time and P&L calculations.

4. **Repeat until horizon is reached.** The event loop continues until the simulation clock reaches the fixed horizon of 23,400 seconds.

This event-based structure ensures that the simulation captures the precise timing, sequencing, and interdependence of order-flow events, executions, and market-maker actions, all while maintaining computational efficiency.

4.2 Generation of Limit Prices

Whenever a new limit order arrives, its price is determined by an exponential distance model reflecting how traders submit orders relative to the current midprice. Specifically:

1. Compute the current midprice as the midpoint of the best standing buy and sell prices.

2. Sample an exponential random variable with rate equal to the scenario's *limit-price decay parameter*. Smaller rates generate larger distances and more dispersed books, while larger rates concentrate orders near the midprice.

3. Convert this distance into integer ticks (based on a tick size of 0.01) and apply it on the appropriate side of the midprice.

This mechanism ensures that shifts in the decay parameter directly affect order placement behaviour and, consequently, spreads, depth, and volatility.

4.3 Tracking Executions

The simulation records detailed information about executions:

- Each one-unit trade is logged with its price, time, and whether the market maker was involved.
- For limit orders, execution time is recorded as the difference between creation time and completion time.
- Aggregate statistics such as total trades, executions per hour, and depth at the end of the run are updated incrementally throughout the simulation.

Market-maker inventory and cash are updated in real time during every execution in which the MM participates, allowing accurate computation of mark-to-market P&L at the end of the run.

4.4 Collection of Performance Metrics

At the end of each simulation run, a comprehensive set of performance metrics is computed and stored:

- average bid–ask spread over the entire run,
- mean, median, and 99th percentile execution times,
- trade counts and execution rates,
- midprice volatility based on sampled returns,
- depth within fixed tick bands around the midprice,
- market-maker inventory statistics and final mark-to-market P&L,
- total wall-clock runtime of the simulation.

Each individual run outputs a complete row containing these statistics. With five replications per scenario, these data support the estimation of means and confidence intervals for all performance measures at the scenario level.

4.5 Rationale

The methodology is designed to isolate how structural parameters - order-size distribution, limit-price dispersion, and market-maker quoting aggressiveness - affect market quality and liquidity provision. The discrete-event structure ensures that all dynamics are captured at high temporal resolution, while the factorial design and CRN replication strategy provide the statistical power and variance reduction necessary for reliable comparison of scenarios.

5 Analysis

This section summarizes the empirical behaviour of the simulated limit order book across the $5 \times 5 \times 5 = 125$ scenarios defined by the three structural factors: (i) the mean submitted order size `order_size_mean`, (ii) the decay rate of the limit-price distribution `limit_price_decay_rate`, and (iii) the market maker's base half-spread `mm_base_spread`. For each scenario, we ran 5 independent replications using distinct random seeds and recorded a rich set of performance metrics per run. We then computed scenario-level point estimates and 95% confidence intervals based on the cross-replication data.

5.1 Metrics and Aggregation Procedure

For each replication of each scenario, the simulation produced a single row of summary statistics, including:

- **Market-quality metrics:** average bid–ask spread (`avg_spread`), mean execution time (`mean_exec_time`), median and 99$^{\text{th}}$ percentile execution times, trade count and trades per hour, and midprice volatility (`price_volatility`).
- **Market-maker metrics:** final mark-to-market profit and loss (`mm_final_pnl`), P&L per hour, P&L per 1,000 trades, and inventory statistics.
- **Liquidity metrics:** resting and market execution counts, total executions, and depth at the first five price levels on each side of the book.

In addition, we constructed a *fill-rate* measure on a per-run basis,

$$\texttt{avg\_fill\_rate} = \frac{\texttt{trade\_count}}{\texttt{total\_executions}},$$

which captures how often resting orders actually result in trades. For each scenario, we then computed cross-replication means and 95% confidence intervals for the four primary metrics used in the analysis: `avg_spread`, `avg_fill_rate`, `price_volatility`, and `mm_final_pnl`. These were obtained via standard t-intervals: if $X_1, \ldots, X_R$ denote the $R = 5$ replication values for a given metric in a fixed scenario, then

$$\hat{\mu} = \frac{1}{R} \sum_{r=1}^{R} X_r, \qquad \hat{\sigma}^2 = \frac{1}{R-1} \sum_{r=1}^{R} (X_r - \hat{\mu})^2,$$

and the 95% confidence interval is

$$\hat{\mu} \pm t_{0.975,\, R-1}\, \frac{\hat{\sigma}}{\sqrt{R}}.$$

The simulation code writes these scenario-level estimates directly into the `simulation_results_full.csv` file as columns with suffixes `_Mean` and `_CI_95` (for example, `avg_spread_Mean` and `avg_spread_CI_95`). The relative half-widths of the 95% intervals are small for most metrics. Across all 125 scenarios, the median ratio of `avg_spread_CI_95` to `avg_spread_Mean` is about 1.4%, and the corresponding ratios for `price_volatility` and `mm_final_pnl` are roughly 11% and 6% at the median. This indicates that differences between scenarios are not driven by sampling noise alone.

Finally, to check for anomalies in the replication behaviour, we plotted per-seed boxplots for the main metrics (e.g., `avg_spread`, `mean_exec_time`, and `mm_final_pnl`) across all scenarios. These boxplots (Figures 1–3) show that all seeds yield comparable dispersion and that there are no outlier seeds with systematically different behaviour, providing a sanity check for the simulation and the random-number usage.

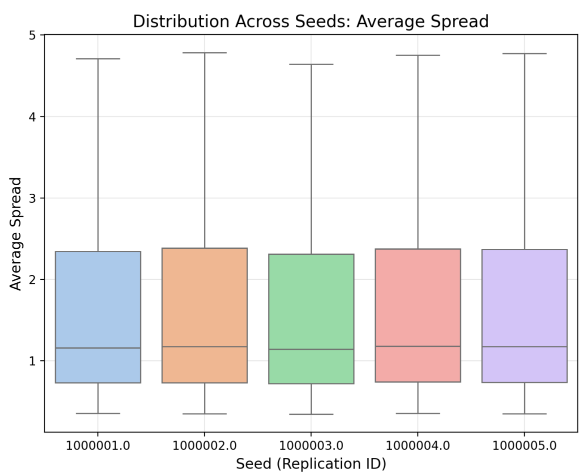

Figure 1: Distribution of replication-level average spread across seeds.

5.2 Effects on Market Quality

Average Spread

Figure 4 (the `main_effects_avg_spread_Mean.png` plot in `figs_out/`) shows the main effects of each factor on the average spread. Three patterns stand out:

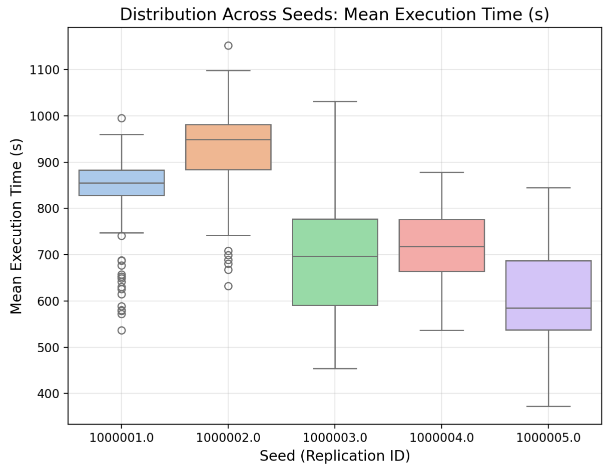

Figure 2: Distribution of replication-level mean execution time across seeds.

1. The **limit-price decay rate** is the dominant driver of spreads. When the decay rate is low (0.05), limit orders are dispersed far from the midprice and the average spread is wide, with a scenario-averaged mean of about 3.94 and an average 95% half-width of approximately 0.08. As the decay rate increases to 0.50, orders cluster near the midprice and the average spread falls to about 0.55 with a very narrow confidence interval (average half-width below 0.01). This represents a reduction in spread of more than a factor of 7 across the range of decay rates.

2. The **market-maker base spread** has a secondary but clear effect. Averaging over the other two factors, the scenario-level mean of `avg_spread` increases from roughly 1.51 at `mm_base_spread` = 0.10 to about 2.12 at `mm_base_spread` = 2.00. This reflects the direct mechanical contribution of wider MM quotes to the overall bid–ask spread.

3. The **mean order size** has only a modest impact on the spread. The average spread increases from approximately 1.51 at `order_size_mean` = 1 to about 1.80 for larger order-size means. The effect is monotone but small relative to the changes induced by the decay rate and the MM base spread.

Interaction heatmaps for `avg_spread_Mean` (generated as `interaction_avg_spread_Mean_....png`) confirm that the decay rate dominates. In particular, Figure 5 shows the interaction between the decay rate and the MM base spread. Even with a small MM base spread, scenarios with a low decay rate (0.05) exhibit much wider spreads than those with a high decay rate (0.50). The MM base spread mainly shifts the level of the spread within each decay regime, rather than changing the overall shape of the response.

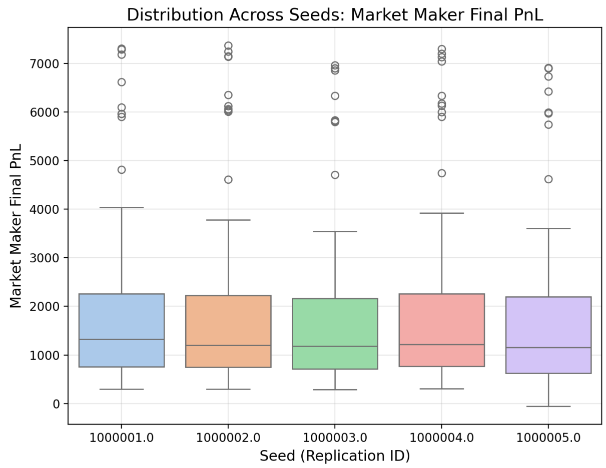

Figure 3: Distribution of replication-level market-maker final P&L across seeds.

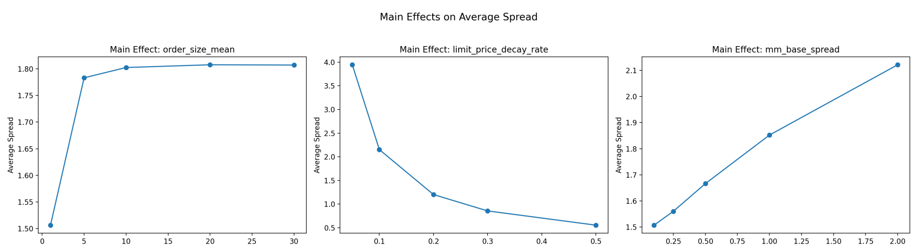

Figure 4: Main effects of the three factors on average spread.

Execution Times and Volatility

The main-effect plot for `mean_exec_time_Mean` (Figure 6) shows that:

- Increasing the **mean order size** lengthens execution times. The scenario-averaged mean execution time rises from about 688 seconds at `order_size_mean` = 1 to around 794 seconds at the largest order-size setting. Larger orders sit in the book longer on average before completing.

- Higher **limit-price decay rates** also lead to longer execution times. As the decay rate increases from 0.05 to 0.50, the average mean execution time increases from roughly 730 to 780 seconds. With tighter books, orders that are not immediately competitive may still be far enough from the midprice to experience non-trivial waiting times before execution.

- The **MM base spread** has a relatively minor effect on execution times compared with the other two factors.

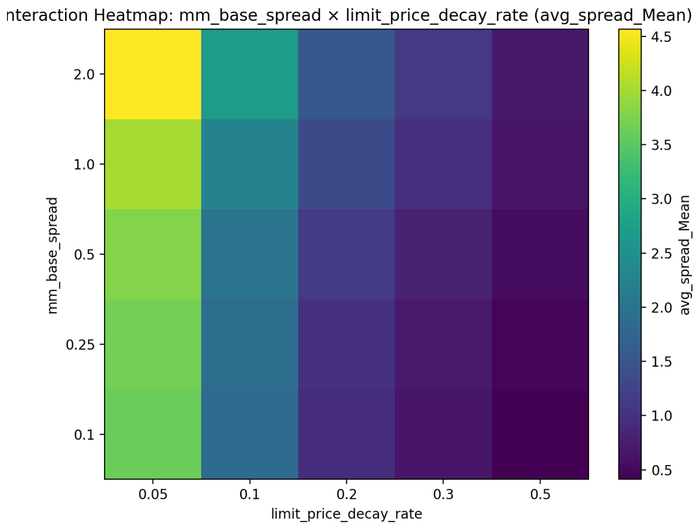

Figure 5: Interaction between limit-price decay rate and MM base spread for average spread.

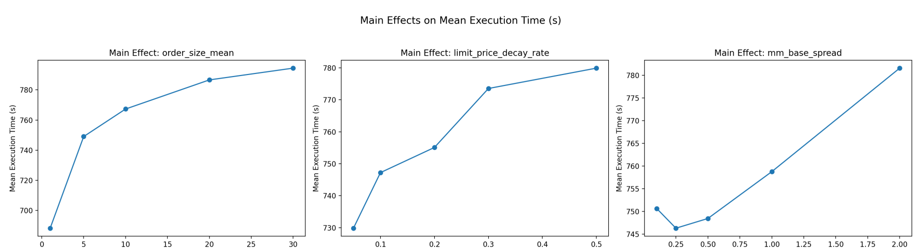

Figure 6: Main effects of the three factors on mean execution time.

For **price volatility**, the decay rate is again the main driver. At decay 0.05, the scenario-averaged volatility is on the order of 3.6×10^{-5}, whereas at decay 0.50 it falls to roughly 7.8×10^{-7}. Tighter books (higher decay) concentrate liquidity near the midprice, dampening price movements. The order-size and MM-spread effects on volatility are small in comparison.

Fill Rate

The average fill rate `avg_fill_rate` is high in all scenarios, reflecting an active market with frequent executions. Across all 125 scenarios, the scenario-level means range from about 0.99 to nearly 1.00, with 95% confidence interval half-widths typically below 0.006. Main-effect summaries show that:

- Larger **order-size means** slightly increase the fill rate (from roughly 0.99 at the smallest mean to about 0.999 at the largest).
- Higher **MM base spreads** also increase the fill rate (from about 0.991 at 0.10 to about 0.9995 at 2.00), consistent with wider quotes being more likely to be "hit" by market orders.
- Higher **limit-price decay rates** modestly increase the fill rate as more liquidity concentrates near the midprice.

Because fill rates are already close to one, these effects are small in absolute terms but statistically clear given the narrow confidence intervals.

5.3 Market Maker Performance

We now focus on the market maker's final mark-to-market P&L (`mm_final_pnl_Mean`) and its 95% confidence intervals.

Main Effects on P&L

The main-effect plot for `mm_final_pnl_Mean` (Figure 7) shows that:

1. The **MM base spread** has a strong positive effect on P&L. Averaging over all order-size means and decay rates, the scenario-level mean of `mm_final_pnl_Mean` increases roughly linearly from about $360 at `mm_base_spread` = 0.10 to about $3,640 at `mm_base_spread` = 2.00. The average 95% half-widths at each base-spread level are on the order of $100–$150, so the differences in P&L between base-spread settings are large relative to sampling uncertainty.
2. The **limit-price decay rate** has a clear negative effect on P&L. At decay 0.05, the scenario-averaged mean P&L is about $2,790, decreasing steadily to roughly $740 at decay 0.50. When the book is wide and liquidity is dispersed (low decay), the market maker earns more per trade and faces less intense competition for order priority.
3. The **order-size mean** has a relatively mild effect on P&L compared with the other two factors. Scenario-averaged P&L increases from about $1,540 at the smallest order-size mean to roughly $1,880 at intermediate sizes, with only small variations thereafter.

In terms of individual scenarios, the largest P&L values occur when the book is wide and the MM quotes with a large base spread. For example, scenarios with `limit_price_decay_rate` = 0.05 and `mm_base_spread` = 2.00 yield scenario-level mean P&L values above $7,000, with 95% half-widths around $200–$300. Conversely, the smallest P&L values—around $300 with tight confidence intervals—occur when the decay rate is high (0.3–0.5) and the MM base spread is very small (0.10).

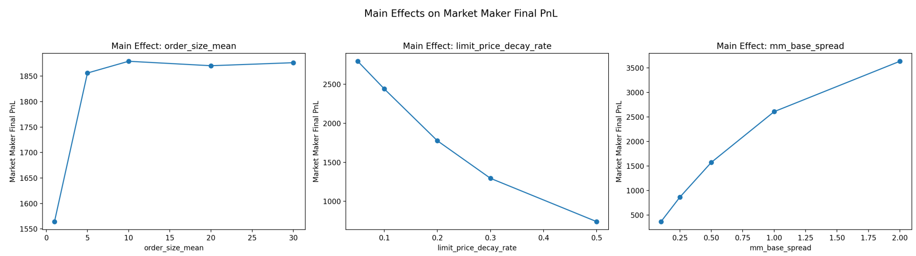

Figure 7: Main effects of the three factors on market-maker final P&L.

P&L versus Market Quality

The combination of spread and P&L results highlights a familiar trade-off:

- From a **market-quality** perspective, higher decay rates and smaller MM base spreads are desirable because they produce narrower spreads and lower volatility.
- From the **MM's perspective**, low decay rates and large base spreads are more profitable, but they generate wider spreads and higher transaction costs for other market participants.

Thus, the simulation reveals a clear interaction between market structure and MM strategy: improving one side of the market (tight spreads) tends to reduce MM profitability, while maximizing MM profit tends to widen spreads.

5.4 Factor Interactions

The interaction heatmaps generated by the second block of plotting code (the `interaction_{...}_{...}_vs_{...}.png` figures) provide a more detailed view of how pairs of factors jointly shape performance.

For **average spread**, the most pronounced interaction is between the limit-price decay rate and the MM base spread. At low decay rates, increasing the MM base spread from 0.10 to 2.00 substantially widens the spread. At high decay rates, spreads are already very tight, and the incremental effect of the MM base spread on the spread is smaller in absolute terms. The interaction between order-size mean and the other two factors is weak: changing order size shifts spreads slightly but does not qualitatively alter the patterns induced by decay and MM spreading.

For **MM P&L**, the decay–spread interaction is again central. High P&L arises when the book is wide (low decay) and the MM quotes aggressively wide spreads; tight books combined with narrow MM spreads produce low P&L. Figure 8 shows the interaction surface for P&L over the decay–spread plane.

For **execution times** and **volatility**, interactions are mild. Execution times tend to be longer when either orders are large or liquidity is concentrated near the midprice, but the patterns are smooth and mostly additive. Volatility is

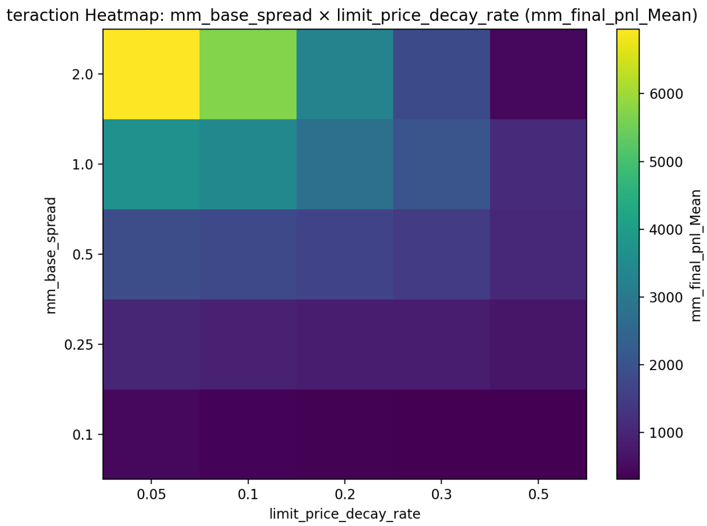

Figure 8: Interaction between limit-price decay rate and MM base spread for market-maker final P&L.

driven primarily by the decay rate, with only small modulations from the other factors.

5.5 Summary of Quantitative Findings

Quantitatively, the simulation yields the following main conclusions:

- The decay rate of the limit-price distribution is the key determinant of market quality. Moving from the lowest to the highest decay level reduces the average spread from about 3.9 to 0.55 and lowers volatility by more than an order of magnitude.
- The MM base spread exerts a strong effect on both spreads and MM P&L. Larger base spreads widen the spread but increase P&L from a few hundred dollars to several thousand dollars over a trading day.
- The mean order size affects execution times and fill rates, but its influence on spreads, volatility, and P&L is comparatively modest.
- Across all scenarios, 95% confidence intervals are generally narrow relative to between-scenario differences, indicating that the observed trends are statistically well supported by the cross-replication data.

These findings set the stage for the concluding section, where we will interpret the results in terms of market design and market-maker strategy, and discuss possible extensions and refinements of the model.

6 Conclusion

This project developed and evaluated a detailed discrete-event simulation of a limit order book with stochastic background order flow and an active quoting market maker. The simulation enabled controlled experimentation over three structural dimensions: (i) the mean submitted order size, (ii) the decay rate governing the dispersion of limit-order prices, and (iii) the market maker's base quoting spread. Across the full $5 \times 5 \times 5$ factorial design and five replications per scenario, we obtained statistically stable estimates of market-quality and market-maker performance metrics, supported by narrow confidence intervals throughout.

The results show that the **limit-price decay rate** is the dominant determinant of overall market quality. Increasing the decay rate from 0.05 to 0.50 contracts the distribution of limit prices around the midprice, reducing the scenario-averaged bid–ask spread from approximately 3.9 to 0.55 and lowering midprice volatility by more than an order of magnitude. Execution-time distributions also shift with the decay rate, with mean execution times increasing by roughly 50 seconds across the same range. These effects are large relative to replication-level uncertainty and remain consistent across all order-size and market-maker settings.

The **market maker's base spread** strongly influences its profitability: scenario-level mean P&L increases from roughly \$360 at a base spread of 0.10 to approximately \$3,600 at a base spread of \$2.00, with the largest profits (over \$7,000) occurring when wide spreads are combined with low decay rates. This behaviour reflects the classical trade-off between liquidity provision and price competitiveness: wide quotes increase per-trade revenue but degrade market quality by widening spreads and reducing price efficiency. Conversely, when decay rates are high and spreads are tight, the market maker earns much less—typically below \$1,000—even with moderately wide quotes.

The **mean order size** plays a secondary role in determining overall market behaviour. Larger submitted orders increase mean execution times by roughly 15% and shift fill rates closer to one, but have comparatively small effects on spreads, volatility, or P&L. This indicates that, within the parameter ranges studied, book shape and pricing structure matter far more than order-size granularity.

Taken together, the results highlight a clear structural tension: parameter settings that yield *high-quality markets* (tight spreads, low volatility, fast executions) coincide with regimes in which the market maker's profitability is lowest. Parameter settings that maximize *market-maker profitability*, in contrast, produce wide spreads and increased transaction costs for other participants. This tension is visible both in the main effects and in the decay–spread interaction surfaces, suggesting that market-design choices such as tick size, order-placement incentives, and MM quoting rules directly influence the balance between liquidity provision and market efficiency.

Several natural extensions follow from this study. Introducing multiple interacting market makers would allow competition effects to be examined explicitly; incorporating richer order-flow models (e.g., correlated arrivals, clustered volatility, adaptive traders) would enable comparison against empirical stylized facts; and experimenting with alternative inventory-based quoting rules could clarify which MM behaviours lead to stable or unstable market conditions. Nonetheless, the current results already illustrate the usefulness of discrete-event simulation for understanding microstructural interactions and for quantifying design trade-offs in automated financial markets.