Bond Market Making Simulation

July 20, 2025

1 Objective

This report presents a simulation-based market-making framework for a 10-year zero-coupon bond. The model is driven by short-term interest rate dynamics under the CIR model, with synthetic bid/ask pricing and trade arrival modeling.

2 Short Rate Model (CIR)

The short rate $r(t)$ follows the Cox–Ingersoll–Ross (CIR) process:

$$dr = a(b - r)dt + \sigma\sqrt{r}dW_t \tag{1}$$

Bond prices are computed using the closed-form CIR solution:

$$P(t,T) = A(t,T) \cdot e^{-B(t,T) \cdot r(t)} \tag{2}$$

where $A(t,T)$ and $B(t,T)$ are model-specific functions of time to maturity and the CIR parameters, as outlined in John C. Hull's *Options, Futures and Other Derivatives*.

3 Simulation Framework

3.1 Calibrating Parameters

Using the most recent Yield on Zero Coupon Bonds from FRED, we compute observed market bond prices via:

$$P(t,T) = e^{-r(T-t)} \tag{3}$$

We then compute the corresponding model price using the CIR bond pricing function. Calibration is performed by minimizing the squared error:

$$\text{Loss} = \sum(P_{\text{model}} - P_{\text{market}})^2 \tag{4}$$

with respect to the parameters a, b, σ, and r_0.

3.2 Simulating Midpoint Bond Prices

Using the calibrated parameters, we simulate the short rate using Euler discretization at 1-second resolution across 252 trading days (539 minutes per day):

$$r_{t+\Delta t} = r_t + a(b - r_t)\Delta t + \sigma\sqrt{r_t}\sqrt{\Delta t} \cdot z_t \tag{5}$$

where z_t is drawn from a standard normal distribution.

3.3 Bid/Ask Spread

We simulate bid-ask spreads from a uniform distribution:

$$\text{spread} \sim U(0.001, 0.01) \tag{6}$$

Bid and ask prices are then calculated as:

$$\text{bid} = \text{mid} - \frac{\text{spread}}{2} \tag{7}$$

$$\text{ask} = \text{mid} + \frac{\text{spread}}{2} \tag{8}$$

3.4 Simulating Fills

Liquidity sensitivity k is drawn from a normal distribution:

$$k \sim \mathcal{N}(60, 2^2) \tag{9}$$

We fix the baseline activity level $A = 2$ trades per side per second. For each quote, we compute the arrival rates as:

$$\lambda_{\text{bid}} = A \cdot e^{-k \cdot (\text{mid} - \text{bid})} \tag{10}$$

$$\lambda_{\text{ask}} = A \cdot e^{-k \cdot (\text{ask} - \text{mid})} \tag{11}$$

We then simulate the number of trades per side using Poisson random draws:

$$N \sim \text{Poisson}(\lambda) \tag{12}$$

3.5 Final Simulated Data

The resulting dataset includes the following fields for each simulated second:

Timestamp, Midpoint, Bid, Ask, Bid Volume, Ask Volume

4 Market Making Strategy

We simulate a market-making strategy using the above synthetic market environment.

4.1 Estimating k and A from Simulated Data

Every 1200 seconds (20 minutes), we estimate separate liquidity parameters k and A for the bid and ask sides.

We fit the model:

$$\log(\text{volume}) = \log(A) - k \cdot \delta \tag{13}$$

where δ is the distance from the quote to the midprice. Volume is aggregated by binning spreads and summing counts in each bin. This rolling regression allows us to infer time-varying liquidity sensitivity.

4.2 Setting Bid and Ask Prices

We use a base spread of 0.005 and adjust quotes based on inventory using a decay function and sensitivity γ:

$$\text{skew} = \gamma \cdot \ln(\|\text{inventory}\| + 1) \tag{14}$$

$$\text{bid} = \text{mid} - \frac{\text{base\_spread}}{2} \pm \text{skew} \tag{15}$$

$$\text{ask} = \text{mid} + \frac{\text{base\_spread}}{2} \pm \text{skew} \tag{16}$$

When inventory is positive, the bid is reduced to encourage selling, and the ask is lowered to discourage further buying. The opposite is applied when inventory is negative.
We use the natural logarithm as a decay function to ensure that the spread does not grow exponetially or linearly with inventory, but still controls inventory risk.

4.3 Simulating Fills

At each second, we use the estimated $k_{\text{bid}}, A_{\text{bid}}$ and $k_{\text{ask}}, A_{\text{ask}}$ to compute fill probabilities:

$$\lambda_{\text{ask}} = A_{\text{ask}} \cdot e^{-k_{\text{ask}} \cdot (\text{ask} - \text{mid})} \tag{17}$$

$$\lambda_{\text{bid}} = A_{\text{bid}} \cdot e^{-k_{\text{bid}} \cdot (\text{mid} - \text{bid})} \tag{18}$$

These are used to draw the number of fills from a Poisson distribution.

4.4 Flow Toxicity

To introduce more realism into our simulation, we model *toxic flow*—the idea that trades at wide spreads are more likely to come from informed traders who have some advantage or private signal. In real markets, market makers are vulnerable when they quote prices far from fair value, especially if they're being picked off by traders with better information.

We define the probability that an incoming trade is toxic using a sigmoid function:

$$P(t) = \frac{1}{1 + e^{-340\delta}} \tag{19}$$

Here, δ is the distance between our quote and the midprice. As δ increases, the probability of toxicity increases sharply. In other words, the further our quote is from the mid, the more likely we're being targeted by someone who knows more than we do.

To simulate the effect of toxic flow, we only allow these trades to go through if the future price moves against us. For example, if we sell at the ask and the price later goes up, we assume that trade was toxic—we sold too low to someone who knew the price was about to rise. Similarly, if we buy at the bid and the price later drops, we assume we were picked off.

This simple mechanism lets us capture the risk of adverse selection in market making, and shows how quoting too wide can hurt performance over time.

4.5 Tracking Performance

Each executed trade updates both our inventory and cash position:

- Selling at the ask: inventory decreases, cash increases
- Buying at the bid: inventory increases, cash decreases

At every second of the simulation, we log key variables to monitor performance and risk exposure. The following metrics are tracked and saved:

- **Cash:** cumulative cash from executed trades
- **Inventory:** number of bonds held at each time step
- **Volume:** total number of trades (bid and ask) executed per second
- **Toxic P&L:** estimated loss due to adverse selection from toxic flow
- **Bid/Ask/Mid Prices:** quotes and reference price at each second
- **Net Value:** cash plus mark-to-mid valuation of inventory

These values are stored in a time-indexed DataFrame and written to CSV for downstream performance analysis and plotting. The data is timestamped at 1-second resolution over the entire trading horizon.