Reinforcement Learning for Optimal Execution with the Queue Reactive Model

Edoardo Vittori

Based on a joint work with Tomas Espana, Yadh Hafsi and Fabrizio Lillo

October 18, 2025

Agenda

1. Introduction
2. Market simulation
    - Stochastic models
    - Agent based models
    - Queue reactive model
3. RL for optimal execution
    - Intro to RL
    - Embedding optimal execution
    - Experimental results
4. Conclusions

1. Introduction

Problem statement

Assume the trader must buy (sell) X units of a security over $[0, T]$. The order is completed in N trades at times $t_0, t_1, \ldots, t_{N-1}$, with $t_0 = 0$ and $t_{N-1} = T$. Let v_{t_n} denote the trade size at time t_n, then we have: $\sum_{n=0}^{N-1} v_{t_n} = X$. For a buy problem, $X > 0$, and for a sell problem, $X < 0$.

Execution cost

Assume P_0 is the arrival price, and $\bar{P}_k$ is the execution price for trade v_k, then the execution cost is given by:

$$C(\boldsymbol{v}) = \sum_{k=0}^{N-1} v_k \bar{P}_k - XP_0 = \sum_{k=0}^{N-1} v_k (\bar{P}_k - P_0)$$

This expression is also referred to as implementation shortfall

Objective

$$\arg\min_{v} \ \mathbb{E}[C(\boldsymbol{v})]$$

2. Market Simulation

Stochastic models

Price simulation:

- $P_k = P_{k-1} + \theta v_{k-1} + \eta_{k-1}$
- $\bar{P}_k = P_k + \rho v_k + \textbf{sign}(v_k)\frac{S}{2}$

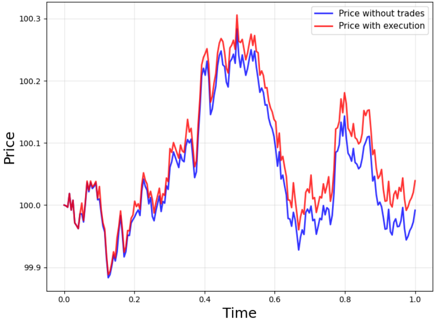
where:

- $\eta_{k-1} \sim$ i.i.d. $\mathcal{N}(0, \sigma^2)$
- θ is the permanent impact coefficient
- ρ is the temporary impact coefficient
- S is the constant bid-ask spread

Considerations

- Further realism can be added by using a transient impact model like in [Obizhaeva and Wang, 2005]
- These models can be calibrated to real data
- **These models only simulate the price, not the limit order book**

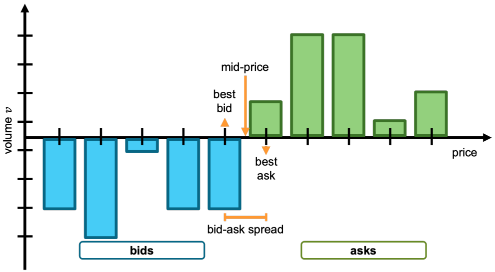
Figure 1: Example of price simulation with a buy execution schedule



Order types

- **Market order** is an order to execute immediately at the best available price in the order book
- **Limit order** is an order that specifies both the price and volume of a trade
- A limit order sits in the order book until it is either **executed** against a matching market order or **canceled**

Figure 2: Illustration of Limit Order Book



Features of the LOB

- Volume imbalance $\frac{v_b - v_a}{v_b + v_a}$
- Volume at best bid and best ask

Agent based models

ABIDES

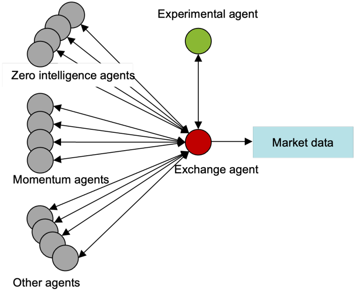
The Agent-Based Interactive Discrete Event Simulation (ABIDES) [Byrd et al., 2019] realistically replicates characteristics of electronic markets such as:

- Continuous double-auction trading
- Network latency and agent computation delays
- Communication solely by means of standardized message protocols

The price process can be described by a **fundamental value** or by using **historical data**. It is possible to create a multi-agent composition using pre-defined agents such as:

- **exchange** agent
- **value** agents
- **momentum** agents
- **noise** agents
- **market maker** agents
- **custom made** agents

Figure 3: Illustration of agent based models



Considerations

- It is not possible to calibrate this simulator to real data
- It is not possible to generate a consistent and realistic transient impact model

Queue reactive model [Huang et al., 2015]

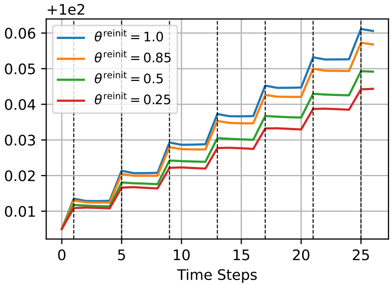
Core Components:

- LOB simulation model for **large tick assets**
- **Queue dynamics** (at fixed price) modeled as a continuous-time Markov process
- State: queue sizes at bid/ask levels
- For queue i:
  - Insertions (limit orders) with intensity $f_i(q)$
  - Removals (due to cancellations or market orders) with intensity $g_i(q)$
  - Queue sizes change by ± 1 at each event
- f_i, g_i calibrated on LOB data

Price Dynamics:

- Mid price updates occur when the best bid or ask queue is depleted
- Post-move queue shapes sampled from empirical distributions
- With the θ^{reinit} parameter you can control the market impact behavior

Figure 4: Average mid-price across 20,000 simulations in which a trader systematically buys the entire best ask at fixed time intervals (vertical dashed lines).



3. RL for Optimal Execution

Reinforcement Learning for Optimal Execution

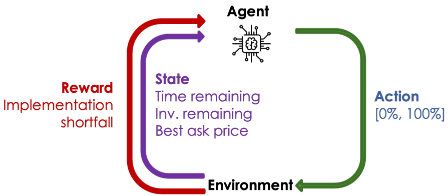
Reinforcement learning basics

- **MDP**: the Markov decision process describes the interaction between agent and environment
- **Objective**: find the policy π which maximizes the discounted sum of the rewards
- $J_\pi = \mathbb{E}_\pi[\sum_t \gamma^t r_t]$ with the reward at time t as r_t

Optimal Execution MDP

- **State**: time remaining, inventory remaining, best ask price
- **Action**: do nothing, market order for volume present in first level of lob:
- **Reward**: $r_t = v_t(P_0 - \bar{P}_t)$ with a terminal penalty

Figure 5: Illustration of MDP flow



Q-learning

- Q-function

$$Q_\pi = \mathbb{E}_\pi\left[\sum \gamma^t R_t \mid s_0, a_0\right]$$

- Bellman Equation

$$Q_\pi = r(s,a) + \gamma \mathbb{E}_{s',a'}\left[Q_\pi(s',a')\right]$$

- Q-learning algorithm

$$Q_t(s,a) = r(s,a) + \gamma \max_{a'} Q_t(s',a')$$

- Q-learning is a tabular algorithm which can be generalized using function approximators

Algorithm examples

- DQN [Hasselt, 2010]
- DDQN [Hasselt, 2010]
- FQI [Ernst et al., 2005]

Experimental setting

MDP

- **State**: time remaining, inventory remaining, best ask price
- **Action**: do nothing, market order for volume present in first level of lob
- **Reward**: $r_t = v_t(P_0 - \bar{P}_t)$ with a terminal penalty

Execution setup

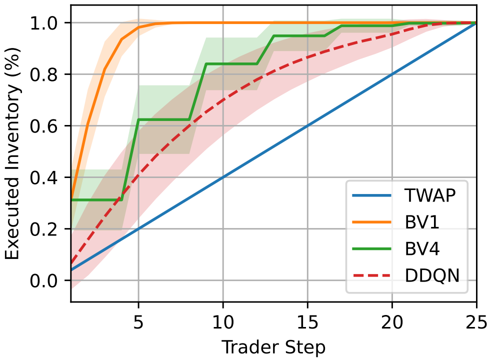
- **Market simulator:** QRM
- **Target:** Buy 25 shares
- **Horizon:** 600 seconds
- **Timestep:** 25-second intervals
- **RL algorithm:** DDQN

Benchmark execution algorithms

- **TWAP:** Time-weighted average price — execute 1 share at each timestep
- **BV1**: execute entire best ask volume at each step (frontloading)
- **BV4**: ???

Figure 6: Learning curves

Figure 7: Execution trajectory



Implementation shortfall and spread distribution

Figure 8: Distribution of the implementation shortfall

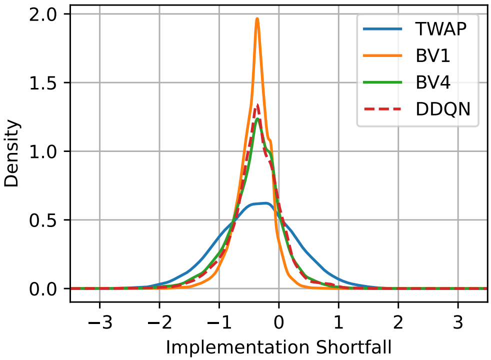
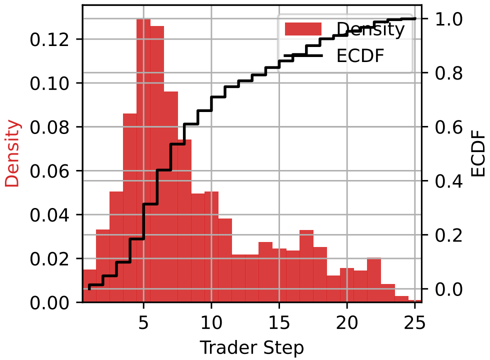


Figure 9: Episode length distribution



4. Conclusions

Conclusions

- In-depth analysis of the QRM model verified market impact is simulated realistically
- Trained RL algorithms to learn an optimal execution strategy
- Obtained execution strategies with a superior performance with respect to the benchmarks

References i

[Byrd et al., 2019] Byrd, D., Hybinette, M., and Balch, T. H. (2019).
Abides: Towards high-fidelity market simulation for ai research.
arXiv preprint.

[Ernst et al., 2005] Ernst, D., Geurts, P., and Wehenkel, L. (2005).
Tree-based batch mode reinforcement learning.
JMLR, 6(Apr):503–556.

[Hasselt, 2010] Hasselt, H. (2010).
Double q-learning.
Advances in neural information processing systems, 23:2613–2621.

[Huang et al., 2015] Huang, W., Lehalle, C.-A., and Rosenbaum, M. (2015).
Simulating and analyzing order book data: The queue-reactive model.
Journal of the American Statistical Association, 110(509):107–122.

[Obizhaeva and Wang, 2005] Obizhaeva, A. A. and Wang, J. (2005).
Optimal trading strategy and supply/demand dynamics.
Journal of Financial markets, 16(1):1–32.