A Modular Framework for RL Optimal Execution

ZHAW School of Engineering

Fernando de Meer Pardo & Alexander Posth

September 30, 2022

Innosuisse Project

Strengthening Swiss Financial SMEs through Applicable Reinforcement Learning

Auf einen Blick

Projektleiter/in : Dr. Jan-Alexander Posth

Co-Projektleiter/in : Christoph Auth

Stellv. Projektleiter/in : Prof. Dr. Jörg Osterrieder

Projektteam : Ionut Florin Dascalu, Fernando De Meer Pardo, Dr. Thom Misheva, David Jaggi, Piotr Kamil Kotlarz

Projektvolumen : CHF 300'000

Projektstatus : laufend

Drittmittelgeber : Innosuisse (Innovationsprojekt / 47959.1 IP-SBM)

Kontaktperson : Jan-Alexander Posth

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

ARAMIS

Startseite Datenausgabe Partner Hilfe Impressum Projektsuche

Projektsuche
Aktuelles Projekt
Neuste Projekte
Grafiken
Login

Forschungsstelle	INNOSUISSE
Projektnummer	47959.1 IP-SBM
Projekttitel	Strengthening Swiss Financial SMEs through Applicable Reinforcement Learning
Projekttitel Englisch	Strengthening Swiss Financial SMEs through Applicable Reinforcement Learning

Grunddaten Texte Beteiligte

Projektstatus	In Bearbeitung / verpflichtet
Startdatum	15.04.2021
Endedatum	15.10.2022
Gesamtkosten bewilligt	312'315.00 CHF
Bereich	90 Social Sciences & Business Management
Forschungsart	Angewandte Forschung und Entwicklung
NABS-Klassifikation	Industrielle Produktivität und Technologie
Fachbereiche	100 % S181 Finanzwissenschaft

Letzte Änderung Projekt

Strengthening Swiss Financial SMEs through Applicable Reinforcement Learning

Mission

Our goal is to overcome the hurdles practitioners face in implementing RL models in Finance by identifying and addressing outstanding practical challenges in the application of RL and by creating exemplary and openly accessible use cases that demonstrate the enormous potential of RL applications in finance.

The Team

- Fernando de Meer Pardo
- David Jaggi
- Linus Groß
- Thomas Gramespacher
- Marc Weibel
- Jan-Alexander Posth
- (Christoph Auth,Florin Dascalu, Jörg Osterrieder, Branka Hadji Misheva)

The Use Cases

- Optimal execution via RL
- Factor Rotation
- (Recommender systems)
- (Multi-agent trading systems)

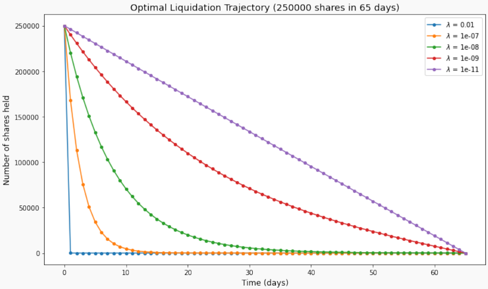
The Specs

- Start / end date: 15.04.2021 / 15.02.2023
- Total budget: 312’315.00 CHF

Optimal Execution

Trade Execution

- An **execution** in the context of a electronic financial market is the process in which a number of units of a publicly traded financial asset are bought or sold through said market.
- Optimal Execution algorithms aim to carry out this process by dividing the transaction into child orders in such a way that a **series of metrics are optimized.**



Optimal Execution

Optimal Execution

- Since the profitability of any investment **heavily depends on its trades executions**, in most jurisdictions explicit rulings regulate the obligation of financial service providers to **perform optimal/best execution** for their clients.
- The main aim of these rulings is to prevent financial service providers from collecting rebates on transactions at the clients' expense by executing client orders at more unfavorable prices that those present in the public markets, but they also force providers to **choose a method of execution** if the client has not explicitly done so.

Optimal Execution

- The monitoring of performed executions has to be carried out via comparison against **market benchmarks such as the Volume Weighted Average Price (VWAP)**. This has lead to the domination of a small number of well-known easily explainable algorithms over the commercial offering of execution services.

Optimal Execution

Classical Algorithms

Examples of popular algorithms include:

- Algorithms that attempt to directly capture the **VWAP (Volume Weighted Average Price)** benchmark such as Percentage-Of-Volume.
- **Time Weighted Average Price (TWAP)** algorithms that submit child orders of the same volume at a constant rate.
- Algorithms that interact with the **bid/ask spread**.

All of these algorithms incorporate a relatively **limited amount of market information** in their decision-making process and decide order submissions based on **easily explainable rules**.

The Deep RL Paradigm

- Conversely, Deep RL Algorithms can incorporate **arbitrary market data** in their decision-making. They are able to obtain, by **learning from experience** on historical market data, adaptable execution policies.

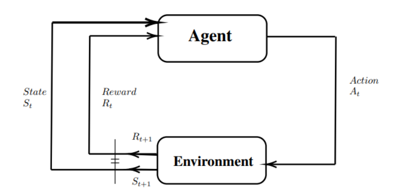


Figure: Components and flow of the RL framework.

RL in Finance – Challenges

- In other areas of the RL literature where the Environment can be easily interacted with (for example in physical simulations, games etc.) the usual focus of research is on the **Agents and Optimization Algorithms** used. This has also been the case with most previous works on RL Optimal Execution.
- One crucial characteristic of RL applied in market applications is that we do not have access to the true environment and thus have to decide on how to **simulate its behaviour.**

Optimal Execution

RL in Finance – Challenges

- Performance under one simulation setup **does not necessarily generalize** to other setups and ranking simulation setups by their quality is challenging (or outright impossible). It is thus necessary to be able to implement **multiple simulation setups in a consistent way**.

Optimal Execution as a RL problem

In order to implement an Optimal Execution RL Environment we need a series of functionalities:

- Data fetching and pre-processing
- Construction of observations
- Action processing
- Child order execution
- Simulation of benchmarks
- Reward calculations

Proposed RL Environment Setup

- We introduce a **modular framework** that allows for the separate implementation of all the different functionalities and internal processes that an Optimal Execution RL Environment needs to incorporate. The core modules we introduce are the *Data Feed*, the *Broker* and the *Execution Algo*

Class Name	Functionalities	Required Methods	Dependency
`Data Feed`	Data retrieval and pre-processing	`reset()`	`Broker` attribute
`Execution Algo`	Execution Logic	`reset()`	`Broker` attribute
`Broker`	Order Execution Simulation	`reset()` & `simulate_order()`	Environment attribute
gym Environment	RL logic	`reset()` & `step()`	*None*

Table 1: Proposed modules of the Optimal Execution RL framework.

Proposed Implementation

Proposed RL Environment Setup

- We present an example implementation of a particular Optimal Execution setup through our modular framework. We share the code in a public Github Repository: `https://github.com/FernandoDeMeer/RL_Optimal_Execution`

GitHub

Proposed Implementation

Example RL Environment Setup

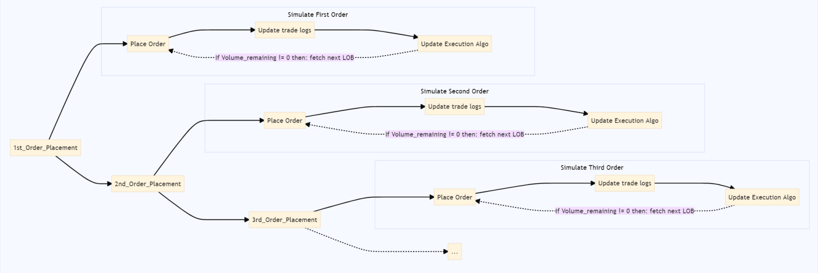


Figure: Illustration of the Simulation of Order Executions.

Proposed Implementation

Example RL Environment Setup

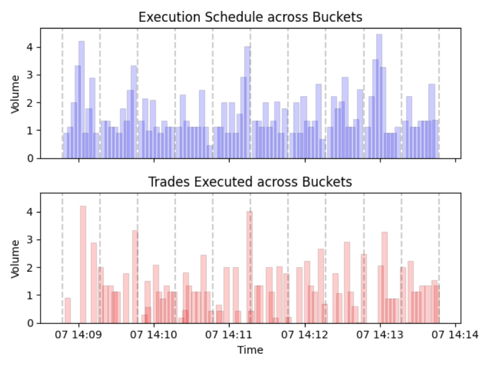


Figure: Illustration of an order submission schedule vs executed trades following the previous Algorithms.

Experiments

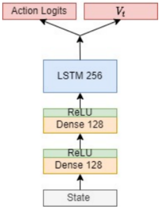


Figure 7: LSTM agent Architecture.

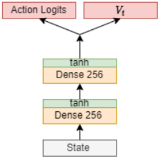


Figure 8: FCN agent Architecture.

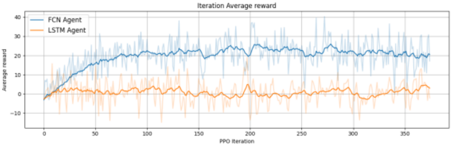


Figure: We showcase how to benchmark two different agents over a training period. Observe that the FCN Agent manages to outperform consistenly in-sample.

Results Out-Of-Sample

	Mean ± std		
Training Period	01/06/2021-05/06/2021	06/06/2021-10/06/2021	11/06/2021-16/06/2021
Evaluation Period	06/06/2021-10/06/2021	11/06/2021-16/06/2021	16/06/2021-20/06/2021
Agent Name			
LSTM agent	**0.0616 ± 0.48**	-0.0080 ± 0.33	0.0418 ± 0.34
FCN agent	0.0506 ± 0.46	**0.0785 ± 0.39**	**0.1087 ± 0.40**

Table 2: Out-Of-Sample performance of the iteratively trained models measured as the mean of the $ difference between the RL achieved price (VWAP) vs that of the TWAP benchmark. Dates are in *dd/mm/yyyy* format.

Conclusions

Conclusions

- Our work can be of interest to both practitioners and academics since it provides a basis that can vertebrate and serialize the implementation of all the aspects of RL Optimal Execution, allowing for a more uniform way to **implement, benchmark and monitor** different simulation setups.
- Our paper is currently under review for publication in the Springer Open Journal of Financial Innovation and can be accessed on arxiv:

`https://arxiv.org/abs/2208.06244`

Visit our project website for more info:
`https://github.zhaw.ch/pages/IWA/rlif/`

Fernando de Meer Pardo

- zhaw.ch/en/about-us/person/demp/
- LinkedIN: linkedin.com/in/fernando-de-meer-pardo-15431b154
- ORCiD: 0000-0003-2233-1858

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw
School of
Engineering

Alexander Posth

- zhaw.ch/en/about-us/person/posh
- LinkedIN: linkedin.com/in/dr-jan-alexander-posth
- ORCiD: 0000-0002-5898-0676

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw
School of
Management and Law