Vega Market Sim: A Protocol Analysis, Parameter Optimisation and Exploration Tool

https://github.com/vegaprotocol/vega-market-sim

Tom McLean[1] and David Šiška[1,2]

[1] Vega Protocol
[2] School of Mathematics, University of Edinburgh

Abstract. Vega Market Sim is a tool allowing simulation of hours or days of interactions with Vega protocol to be run in the span of several seconds together with the ability to pause at arbitrary moments to inspect all state. Vega Protocol itself is a complex system and in any complex system there may be emergent behaviours which may or may not be desirable. This is true across DeFi protocols. As these protocols increase in complexity, the ability to reason out their exact behaviours and interactions under all conditions a priori becomes a significant task. In environments where interactions with other protocols are possible, such as on the Ethereum chain, the task becomes harder still. Vega Market Sim demonstrates the basics of how agent based simulations, combined with reinforcement learning, can be used to explore DeFi protocols and to make better design decisions and parameter choices. While the tool itself is built specifically for Vega Protocol we believe that the design patterns can be replicated to build similar tools for other use cases and thus we believe the tool is interesting to wide audience. We also believe that the codified nature of DeFi which allows for extensive simulations and testing is one area where it truly has an edge over the traditional financial system.

Keywords: Mechanism Design, Agent Based Simulation, Reinforcement Learning

1 Introduction

Vega Protocol [2] is an L1 blockchain protocol built to support derivatives trading. It offers fee-less transactions and a full limit order book to allow for optimal price discovery, on a network supported by a decentralised group of independent node operators. Permissionless market creation allows community members to define markets settling based on a flexible oracle framework, and whilst currently trading instruments are restricted to fixed-expiry-date futures this is anticipated to expand into spot markets, perpetual futures, options and ultimately custom WASM-defined instruments.

This flexibility and power comes at a cost of significantly longer technical specifications, those of the Vega Protocol itself[3] at time of writing are spread across approximately 80 documents detailing various aspects of behaviour, and this for a single protocol with well-defined boundaries. When we wish to test how this protocol truly behaves, we can build extensive system and integration tests to ensure that the code as-implemented conforms to the required designs, but the question of how to test the design itself is a harder task. Scaling out to consider the ecosystem as a whole this difficulty only increases, and whilst specific pathways of interaction and response can be investigated individually, ensuring behaviour is reasonable under unexpected conditions or actions exhaustively swiftly becomes infeasible.

In traditional markets and financial systems this is often where the investigations must cease. A regulatory agency may test an individual bank or market with unexpected theoretical conditions, but these investigations must be limited in scope and frequency due to their very nature. The move to codified systems with defined behaviour, and the ability to run these systems in a controlled manner whenever desired, presents an opportunity to move beyond this barrier and gain a deeper understanding of the behaviour of these complex systems under scenarios both expected and unexpected, allowing both greater confidence in their behaviours and better information for making decisions about their management.

In this paper and presentation we will demonstrate the tools we have built for Vega Protocol to start investigating and answering some of these questions.

[3] https://github.com/vegaprotocol/specs/tree/master/protocol

2 Vega Market Simulator

The core of our tool-set, and what we will be demonstrating, is the Vega Market Simulator. The Market Simulator allows one to instantiate a fully working Vega Protocol instance solely confined to one machine, which accepts all transactions sent to it as signed correctly, and builds a block only when requested. This grants one the ability to simulate hours to days of trading in seconds, followed by freezing the chain and inspecting the state and history. On top of that, we have built a layer of functions for interacting with the markets in terms of financial market primitives (creating orders, monitoring positions and risk, inspecting an order book etc).

On this base we have built a library of agents of varying complexity to provide building blocks for interesting simulations. These fall into three categories:

- *Zero intelligence agents:* are liquidity takers placing market orders of various sizes and at various intensities.
- *Optimising agents:* solve control problems based on fixed parameters [1] e.g. provide liquidity or trade based on trends.
- *Learning agents:* implement reinforcement learning algorithms (e.g. policy gradient [3]) to collect rewards resulting from interacting with the environment.

By including within these agents a combination of those with "controlled" randomisation (random volumes of buy vs sell) and more "free" randomisation (reinforcement learning agents who actively explore the action and state space) one can then control a trade-off between testing the system in novel environments and investigating the behaviours of specific setups.

3 Use Cases

Now that we have the capability to simulate markets with active trading there are a few ways this can be of use in analysing the behaviour of a DeFi protocol.

- *Configuring Parameters:* As DeFi protocols grow in complexity, often so too do the number of parameters governing their behaviour. The interaction of these may be non-obvious, or at least difficult to model statically. Dynamic simulations allow one to observe the effect of a parameter change across an entire system.
- *Dynamic Testing:* Whilst static testing can cover known requirements and edge cases, the state space of protocols rapidly grows beyond what can be testable directly. A regular run of a reinforcement learning agent alongside more static policies has helped highlight several edge cases causing unrecoverable errors in the Vega Protocol core.
- *Answering Questions:* Questions affecting design of the system can be more easily answered with a customisable model of the exact current system, answering questions such as "Will a market maker hedging irregularly on a centralised exchange be able to adequately manage their risk without dedicating excess capital to Vega Protocol?" with a known set of assumptions and then a Monte Carlo simulation across the exact production system.
- *User Testing of Trading Strategies:* Vega Protocol users can test their own integrations and training strategies against various testnets[4]. Vega Market Sim allows running of long-term scenarios simulating months of trading activity in the span of mere minutes or hours, allowing e.g. Monte-Carlo investigations of how the user algorithm behaves.
- *Interactive Learning Tool:* Users can use Market Sim to "step through" interactions and inspect state at any point allowing better understanding of various input transaction effects.

4 Discussion & Future Work

Once live trading begins on Vega Protocol itself, and additional complexities arise from further enhancements to the system such as alternative instruments, having a modelling framework with state space exploration beyond specific handmade configurations will become increasingly important for ensuring the system remains robust and safe for use.

We hope that by presenting this modelling framework, and discussing use cases for it, attendees will consider the applicability of similar setups for their own work, alongside the importance of harnessing DeFi systems' unique characteristic of being wholly defined in code to enhance their robustness in the design domain as well as that of the code itself.

[4] e.g. Fairground testnet `https://fairground.wtf/`

References

1. Á. Cartea, S. Jaimungal, and J. Penalva. *Algorithmic and high-frequency trading.* Cambridge University Press, 2015.
2. G. Danezis, D. Hrycyszyn, B. Mannerings, T. Rudolph, and D. Šiška. Vega protocol. `https://vega.xyz/papers/vega-protocol-whitepaper.pdf`, 2018.
3. R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

A Simulation Outputs

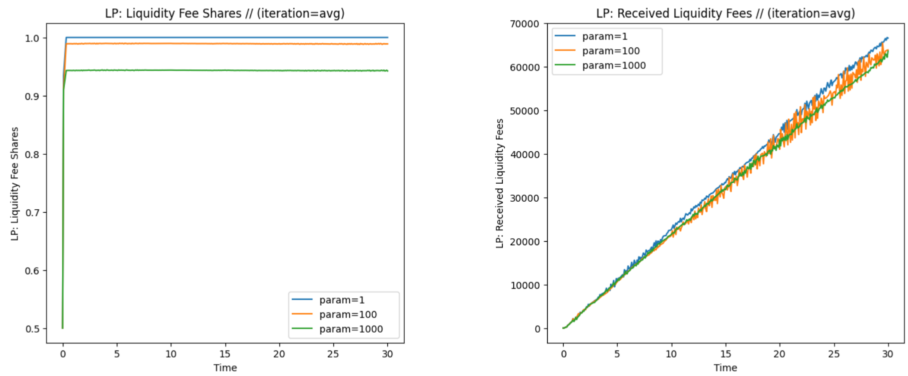

Fig. 1. Running multiple simulations keeping initial parameters static, varying only a single parameter, allows inspection of the impact on values of interest. Fees paid to liquidity providers are a function of how competitive their provided prices are. Figure displays the changing received fees over time with different tolerances for price quality.

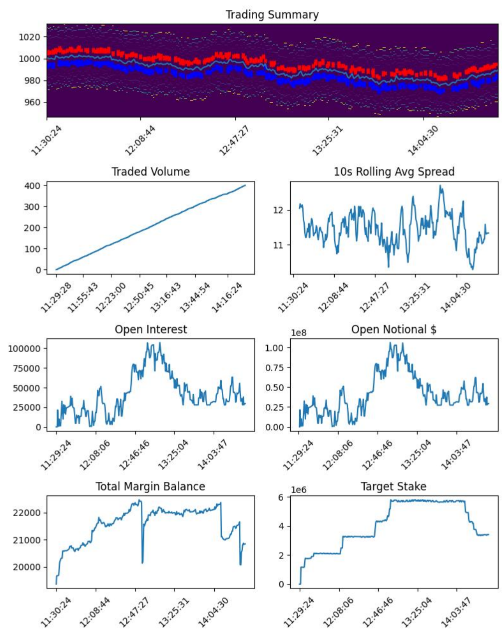

Fig. 2. Simulations of a period of trading which run automatically on each pull request to the Vega Core codebase help visually highlight any unexpected behaviour