Agent based modeling of the RAI system

experience

March 10, 2021

1 Introduction

In this report, RAI's market behavior under different conditions is explored using an agent based model. Agents have their own strategies and decide whether they want to interact with either the RAI system or a Uniswap RAI/ETH pool in discrete steps.

2 Preliminary: simplified model

2.1 Description

The model is initially largely simplified to reflect the current real world parameters of the deployed RAI system and get some intuition on the kind of interactions that can exist between the different agents and the system, and indirectly, between the agents themselves. In particular, the impact of an incentive for RAI/ETH liquidity providers in the form of future FLX tokens is investigated under these conditions:

- The liquidity pool is seeded with some baseline amount of liquidity that cannot be withdrawn
- The agents' initial holdings Ether, annualized return threshold to enter the system, and expected FLX valuation are drawn from a uniform distribution
- The only strategy for agents is to buy RAI and provide liquidity in RAI/ETH with their entire net worth if the annualized return they would get is above their threshold, and remove liquidity and sell RAI for ETH otherwise
- The RAI system controller is a simple proportional controller with parameter K_p

The simulation code used for the present report advances in 1-hour time steps. Every step, the following actions occur:

1. Agents are randomized
2. Each agent sequentially checks if they annualized return they would get (or are getting) for providing liquidity is above their threshold:
    - If yes and the agent already provides liquidity, or if no and the agent doesn't provide liquidity nothing changes

- If yes and the agent doesn't provide liquidty, they buy RAI with the exact amount needed so that they can add liquidity with their entire current net worth.
- If no, and the agent is currently providing liquidity, they redeem all of their liquidity provider tokens and directly sell the obtained RAI for ETH in the pool

3. The price at the end of all agents interactions is saved for the time weighted average price calculation
4. The redemption rate is updated based on the error between the current value of the redemption price and the 16H time weighted average price of the pool
5. The simulation moves to the next timestep

The calculation of agents expected returns is made of two components. First, given an expected fully diluted FLX valuation, total amount of FLX given per day to liquidity providers, and the agent's pool share or potential pool share, an expected amount of rewards per day in USD is calculated and extrapolated to a year if all things stayed equal. Denoting the expected FLX valuation V_{FLX}, the FLX distributed per day to liquidity providers D_{FLX}, the agent's pool share s_{pool}, and the current value of the agent's pool share V_{share} this part of the returns is:

$$\text{FLX rewards expected returns (in \%)} = 100 \times \left(\frac{V_{FLX} \times D_{FLX} \times s_{\text{pool}} \times N_{\text{days}}}{V_{\text{share}}} - 1 \right)$$

Second, it is assumed that the agents believe that the system works, and that the market price of RAI will *eventually* reach the redemption price, so they can interpret the difference between the future redemption price as negative return which adds to the positive return of the rewards. The future redemption price is extrapolated from the current redemption price to which is applied the current redemption rate (compounding). Denoting the current market price p_m, the future redemption price p_r^f and the current redemption rate (allowed to be negative) r, this part of the returns is:

$$\text{Redemption rate driven returns (in \%)} = 100 \times \left| 1 - \frac{p_r^f}{p_m} \right|$$

The size of the RAI order needed to provide liquidity with one's entire ETH wallet content: size $= R_{ETH} \left(\sqrt{1 + \frac{w_{ETH}}{R_{ETH}}} - 1 \right)$ where w_{ETH} is the agent's ETH wallet content and R_{ETH} is the current reserve of the liquidity pool in ETH (see Appendix for proof).

2.2 Results