On Dodoex

Taylor Hulsmans

June 2021

Abstract

The decentralized Exchange, DODOEX was analyzed and reflected on for improvement. Its greatest strength is the removal of impermanent loss and the ability to asymmetrically add liquidity without losing path independence. Its greatest challenge is (permanent) arbitrage loss from volatile pairs and the requirement to manually fine tune parameters ex post without a coherent economic model to reason about it. An improvement on the protocol is theorized that reduces the risk (actually magnitude) of arbitrage loss by utilizing Chainlinks upcoming implied volatility feed to dynamically adjust the fee rate and liquidity parameter k. As past prices do not predict future prices, the intuition behind this improvement is to increase the pro-activity of the market maker by having it react to inferences on the future state of the market.

1 Introduction

Dodoex is a self described proactive market maker (PMM). In contrast to Automated market makers (AMM) like Uniswap, which rely on arbitragers to bring the pool to price equilibrium, this PMM relies on chainlink oracle price feeds. While this removes the need for symmetric liquidity addition, and by extension the subsuming of dual risk and impermanent loss, the price oracle feed is not instantaneous and is an average from many different markets, and thus exposes liquidity providers to a different sort of risk, called arbitrage loss, which unfortunately is permanent.[figure 1] This risk is characterized by actors (lets not even start on miners) taking advantage of front running the oracle price feed, which only changes when the average price changes by 0.5%, with individual arbitrage moments that are greater than the 0.6% transaction fee. It is during moments of high price volatility that these opportunities emerge. Additionally, the Dodoex claims that liquidity parameter k has an effect on the risk of arbitrage loss, the model of which remains wholly 'holistic'.[figure 2] To overcome these challenges, Dodoex utilizes centralized smart contract architectures and extensive back testing to fine tune transaction fees and the liquidity parameters, alongside fine print warnings to engage in active management when one observes volatility. Given that markets are not strong-form efficient and past

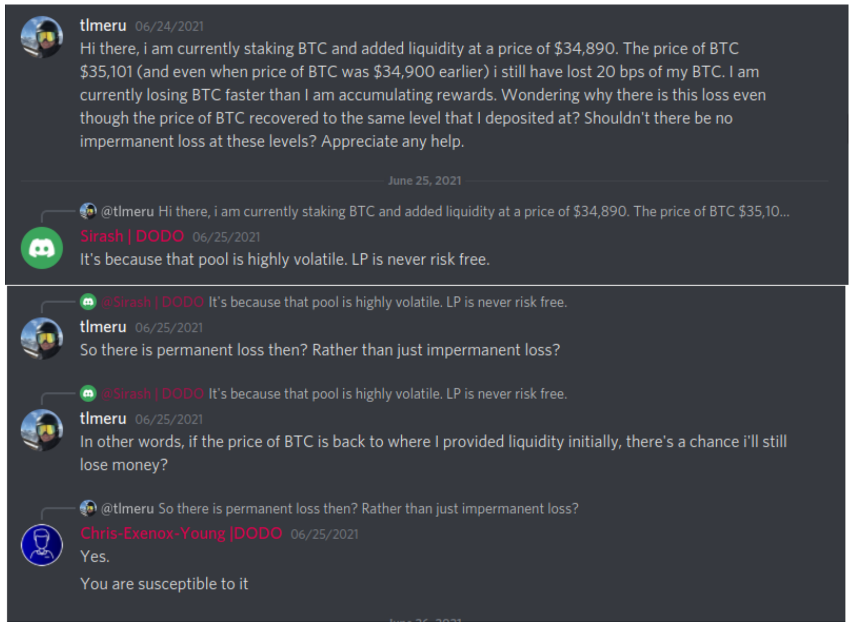

Figure 1: Sorry Timeru, but that loss is permanent, and we don't have much idea how to quantify that risk anyways

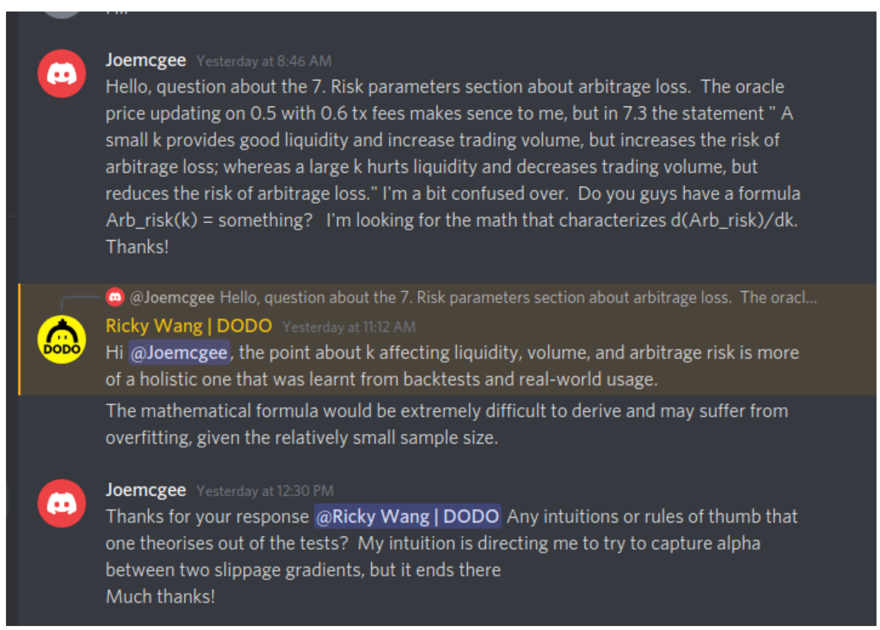

Figure 2: DODOex doesn't really have an idea how their liquidity parameter responds to volatility, only that it does

prices do not predict futures prices, It is the view of this author that the majority of risk in the DODOex platform is systemic, and not measurable until its too late. To overcome this limitation, and to improve the efficiency of the dodoex platform, this paper recommends a self-tuning algorithm for the transaction fee and liquidity parameter that responds automatically not only to the oracle price feed, but also its implied volatility feed as well. Implied volatility is a forecast, an estimate derived from the options market using the Black-Scholes equation. When implied volatility increases the algorithm raises transaction fees and decompresses the liquidity concentrated at the market price, when implied volatility is low, transaction fees are reduced and liquidity becomes more concentrated. The desire is to really put the proactive in its PMM protocol by giving it some cognizance about future volatility expectations.

2 Derivation

2.1 The Problem

Dodex inherits from a volume average decentralized price oracle of 21 different aggregated price feeds. For simplicity let us assume the price points are normally distributed about the mean price of 100, with a very low volatility of 0.125. Note that a 0.5% change in price corresponds to the average price change from 100 to 100.5. We can see here for stable assets an individual arbitrage opportunity

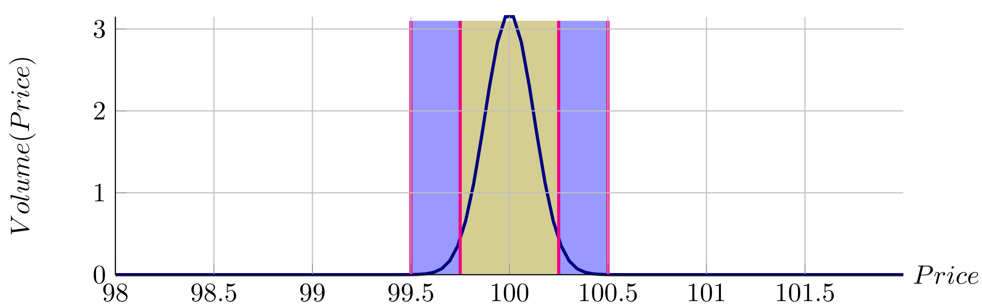

Figure 3: An asset with an average price of 100 and very low volatility ($\sigma = 0.125$). The olive region is plus and minus 1 standard deviation away, while the blue indicates the plus and minus 0.5% price change

would require plus or minus 4 standard deviations away from the mean. An occurrence of only 1 / 15 787! Or by frequency of daily event, twice in ones lifetime. We can see for stable pairs, this liquidity algorithm shows very little potential for arbitrage loss by front running. However, the issue arises when the volatility of the pair increases. When the volatility begets a standard deviation of 0.5, our 0.5% price change only falls within a single standard deviation. Or about 1 in every 3 bits of liquidity falling outside the tolerance of the system

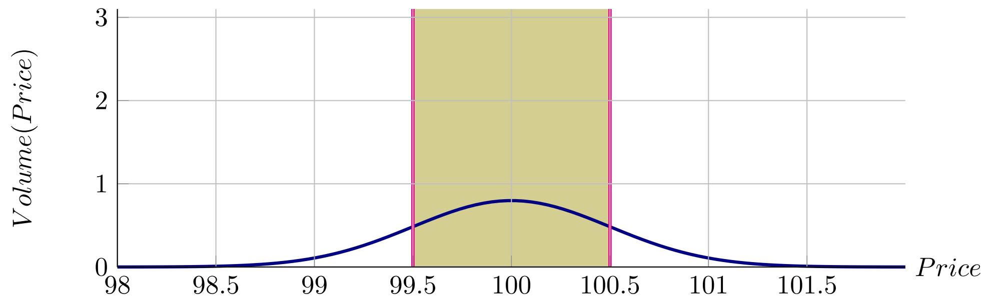

Figure 4: An asset with an average price of 100 and a higher volatility ($\sigma = 0.5$). The olive region is plus and minus 1 standard deviation away, while the blue (completely overlapped) indicates the plus and minus 0.5% price change

to negate arbitrage loss. That is alot of alpha with a bot that can touch a high volume of different exchanges. Clearly the game of DODOex is not appropriate for pairs with a basic amount of volatility. The statement in their white paper "Please note that arbitrage loss due to front running will increase significantly during drastic market fluctuations. The DODO team recommends withdrawing your assets during fluctuations to avert the risk and proceeding with caution depending on your risk profile." should be pasted in a pop-over div on every endpoint of their website.

2.2 Adding Implied Volatility

The implied volatility of an asset can be derived through the options market for that asset through the black-scholes equation.

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{1}$$

Which, can be simplified with a formula that is based on the at the money price under normal model approximation.

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)} = \left[N\left(\frac{1}{2}\sigma\sqrt{T-t}\right) - N\left(-\frac{1}{2}\sigma\sqrt{T-t}\right)\right]S. \qquad (1)$$

(2)

Which derives an approximation of implied volatility as

$$IV \approx \sqrt{\frac{2\pi}{T}} \cdot \frac{C}{S} \tag{3}$$

Where S is the price of the underlying asset, V is the price of the call option, IV is the implied volatility, and r is the risk free interest rate. Thankfully, Chainlink oracles in their upcoming V2 will be providing this data through their API, making it consistent, reliable and easy to use without advanced quantitative financial methods.

Given their implied volatility is not out yet, it is hard to determine what form they will present their metric. Specifically, how long in the future the option expires. What ever the length of time, log returns, or percent, we can map this value to the normal distribution to dynamically calibrate transaction fees to reduce arbitrage loss to the 3 σ range (or set this as a governance parameter)

$$Calibration(P_{chainlink}, IV_{chainlink}) \mapsto \sigma_{system} \mapsto (tx_{fee}, k) \tag{4}$$

$$\frac{\partial Calibration}{\partial \sigma} > 0 \mapsto tx_{feet1} > tx_{feet0} \tag{5}$$

$$\frac{\partial Calibration}{\partial \sigma} > 0 \mapsto k_{t1} < k_{t0} \tag{6}$$

Ultimately it would be optimal to have some sort on value change to respond to, like the 0.5% price movement update, but until the API is out the exact nature of this variable will be unknown.

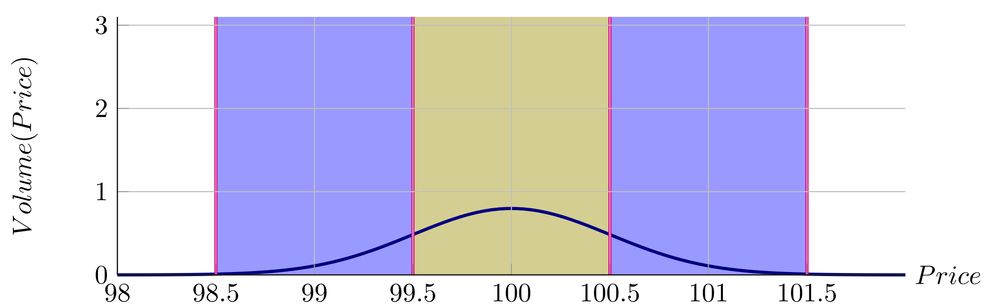

Figure 5: An asset with an average price of 100 and a higher volatility ($\sigma = 0.5$). Can now use the implied volatility feed to re calibrate its transaction fee to minimize the possible arbitrage opportunities, no matter the volatility

3 Closing Notes

DODOex PMM model was analyzed and an improvement was theorized. The major challenge of the algorithm is permanent arbitrage loss arising from changing volatility that it cannot respond too with their manual tool set and current data access. As the platform becomes more popular, trading bots will have higher incentives to capture arbitrage loss and it will become more apparent that their algorithm is always leaky, the only question is how much. To compensate for this, this author proposes to use chainlinks upcoming implied volatility feed alongside its price feed to not only re-calibrate the equilibrium price, but the transaction fee as well and liquidity parameter as well. The big red flag for this protocol is its interface with Miner Extracted Value and their ability to insert the oracle update anywhere in the block they so choose. Thankfully, Chainlink will also be coming out with the fair ordering of transactions at this time as well. In order to bring this protocol to full life, we would likely want to out do their backtesting with a multiagent GAN tasked with attempting to arbitrage each other in a sandbox environment to better understand the relationship of the liquidity parameter and the volatility.