Proposal for Cosmos based Automated Market Maker

Lite Paper

Dongsam Byun, Hyung Yeon Lee

January 2021

Contents

1 Version History

Version	Date	Authors	Description
0.1	03-Feb-2021	Dongsam Byun Hyung Yeon Lee	Version for peer review

2 Introduction

Recent enhancements in blockchain technology have allowed developers to build scalable, automated utilities upon trustless infrastructure. One utility class that has evolved significantly is Automated Market Makers (AMM) ("Uniswap V2 Core" 2020). AMMs allow investors without significant capital or financial knowledge to invest in market-making opportunities.

Within the Cosmos Network ("What Is Cosmos?" 2021), a significant advancement is Inter-Blockchain Communication ("Cosmos IBC" 2021) (IBC) protocol. IBC allows the easy transfer of tokens across multiple connected Cosmos based blockchains. Also, IBC pegs will enable users to deposit tokens in one blockchain and withdraw the tokens in another blockchain. Pegs such as ETH Peggy ("Cosmos ETH Peggy" 2021) and BTC Peg allow users to bring Ethereum assets and BTC into the Cosmos Network.

Inspired by these advances, this paper proposes a Decentralized Token Exchange (DEX) on the Cosmos Ecosystem. The module features an AMM model combined with order books, multi-block batches, and limit orders to bring high-quality liquidity to the Cosmos Hub. This marketplace will likely be an essential core utility as IBC and pegzones increase token exchange activity in the Cosmos Ecosystem.

3 Uniswap AMM based models

In the Ethereum ecosystem, there has been significant development of Decentralized Finance (DeFi) utilities. The Uniswap Automated Market Maker (AMM) model("Uniswap V2 Core" 2020) is the most notable.

Automated Market Makers are a decentralized exchange (DEX) mechanism that uses a pre-defined algorithm to allow users to swap tokens. AMM's have brought several advantages:

- They democratize market-making participation: Anyone even without significant capital or financial engineering skills can join the market-making activities by merely depositing tokens into the AMM liquidity pool
- They remove the middle man: AMM's are built upon decentralized blockchain networks, thus allowing users to interact without the necessity of a centralized operator or a custody
- They simplify swap execution: Users can efficiently execute swap transactions as they don't use complex order book interfaces.

Overall, despite their simplicity, AMM's have brought significant benefits and exhibit stable properties under many market conditions (Angeris et al. 2020). However, there are weaknesses in the design, particularly in order execution and price inconsistencies.

3.1 Price inconsistency

The heart of the AMM model is the Constant Product Equation. $R_x R_y = k$ where R_x and R_y represent the reserve amount of different two tokens (x and y) and k is constant such that k > 0. This product remains constant during the token swap process such that for time t+1.

$$R_x(t)R_y(t) = R_x(t+1)R_y(t+1)$$

From this, it is observed that when a user places an order of Δ_x tokens

$$R_x R_y = (R_x + \Delta_x)\left(R_y - \frac{\Delta_x}{p_s}\right)$$

Where the effective swap price p_s is the swap price. Rearranging this gives

$$p_s = \frac{R_x + \Delta_x}{R_y}$$

3

However, the new pool price p_p is

$$p_p = \frac{R_x + \Delta_x}{R_y - \frac{\Delta_x}{p_s}} = p_s + \frac{\Delta_x}{R_y} + \frac{\Delta_x^2}{R_x R_y}$$

The implication of the above equations implies that the post-swap pool price is different from the swap price. This price inconsistency causes several effects:

- Repeated price swing attracts more arbitrage opportunities than what AMM needs for real-time price discovery
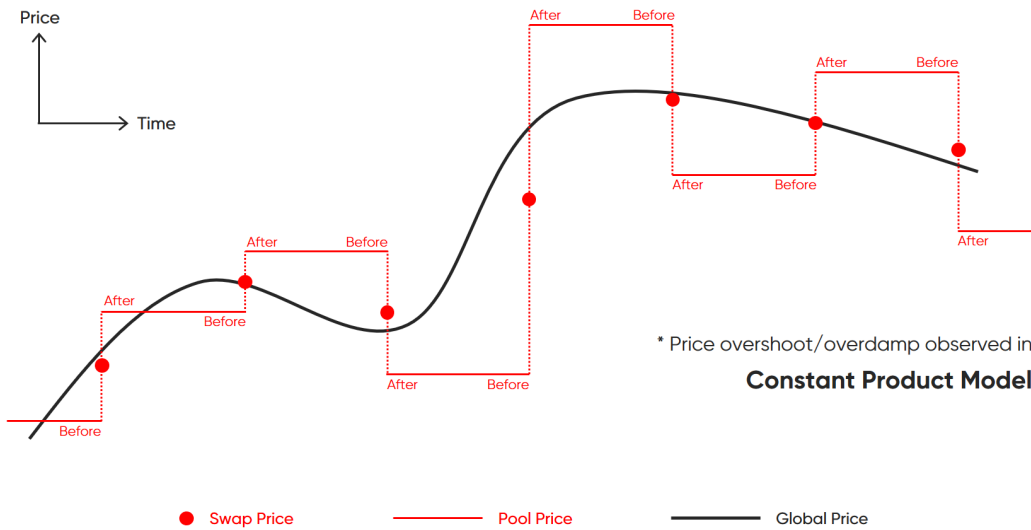- Excessive arbitrage opportunities in AMM will cause more loss to pool investors and traders

Figure 3.1: Inefficient Price Discovery By Constant Product Model

3.2 Order Execution Priority

In Ethereum, miners have absolute power to sort transactions in a block. The transaction order can significantly impact the execution price in Uniswap's AMM model. This problem exists not only in Proof of Work (PoW) consensus networks but also in Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) network environments (Zhou et al. 2020). The combination of price inconsistency described above with transaction ordering can adversely impact execution price.

A second effect of the control of validators over transaction ordering is gas/latency competition. Because traders compete with each other for execution priority, this causes an increase in gas

prices. It can also encourage undesirable behaviors such as collusion between validators/miners and traders and front running.

3.3 Order types

In current Uniswap AMM models, orders cannot be alive for more than one block because they are either immediately executed or failed. This means that after one block, all the liquidity has to be lost and needs to be replenished with the new orders.

Further, AMM's typically do not permit limit orders (although this can be done via third parties). A limit order is an order to buy or sell a token at a specific price or better. Unlike market orders, limit orders allow users to place orders at a price they are prepared to buy/sell. These orders remain in the market until they are satisfied or cancelled. Orders of this type aid price discovery and increase market participation. Limit orders also allow active market participants to use various trading strategies, which in turn adds enhances liquidity and reduces the price impact(slippage) of swap orders, hence reducing trading costs for users

3.4 Fractional Order Execution

When given a larger order, traders will often fill it through a series of smaller trades with different counterparties, at various venues, at different prices over some time. In doing so, they get better prices for clients and can better manage liquidity. This fractional execution (or partial filling) of orders is a use-case that is not possible with current automated market maker models.

4 Proposed Cosmos AMM

Based on the analysis above, we propose a Hybrid Exchange model that combines a batch-based order book ("Traditional Orderbook System" 2021) matching algorithm with an AMM-based methodology such that

Fig.1 Order book model   **Fig.2** Liquidity pool model   **Fig.3** Hybrid model
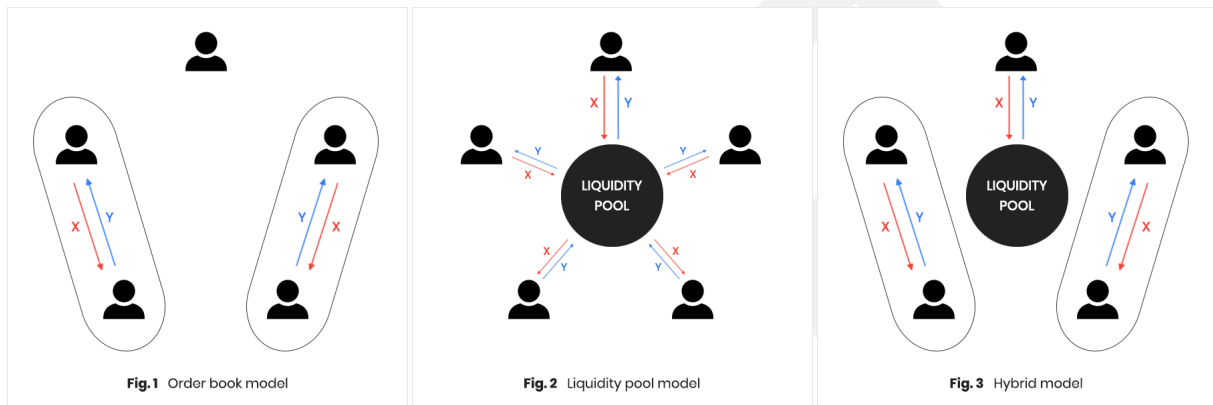
- Orders are accumulated in the order book
- At every batch execution height the order book is processed using a matching engine
- The liquidity pool participates in the matching process using the equivalent swap price model described in section 4.3

Also, the proposed model

- Permits limit orders that stay in the order book until they are filled or cancelled. Indeed, they can remain open over multiple batches
- Allows partial filling of orders in the case where order price is equal to swap price.

4.1 Batch Execution

To address the issues related to order execution, it is proposed to use a Batch Execution methodology. This follows a proposal outlined by (Pourpouneh, Nielsen, and Ross 2020), which is called "batch auction"

> An alternative market solution is a DEX with discrete clearing as opposed to continuous clearing – a so-called "batch auction." The frequent batch auction provides all buyers and sellers with the same trading opportunities by removing the randomness from the speed of processing the orders. Instead of focusing on getting first in line to trade at a given price, buyers

> and sellers are given the opportunity to submit numerous contingent bids and asks that all enter the same double auction within a given time window. The batch auction has been put forward to address the front running problem caused by High-Frequency Trading (HFT) on traditional financial exchanges (Budish et al. 2014). In crypto, the batch auction can solve a number of problems. The front running problem is addressed directly as all bids and asks are treated equally. As each batch can include as many bids and asks as needed, performance limitations can be captured by the time window.

In batch execution, orders are accumulated in a liquidity pool for a pre-defined period, which can be one or more blocks in length. Orders are then added to the pool and executed at the end of the batch.

In our model, there are two key features:

- Unexecuted orders from the batch remain in the order book to be executed in a future batch
- The batch period can be changed to reflect market conditions. For example, the batch period can be lengthened when there exists a significant price change due to high order volumes. An extended batch period invites more traders to participate in price discovery, which results in a more balanced and stable process. This is similar to the "dynamic closing" or "extended bidding" models using in many online auction platforms ("Auction Terminology" 2020).

As (Pourpouneh, Nielsen, and Ross 2020) mentions, for a DEX, batch execution prevents front-running and collusion between miners/validators and traders, resulting in a fairer trading environment for all.

4.2 Order Matching Rules

The model allows fractional and full matching of orders based on the following criteria:

- Swap orders from X to Y

    - If order price > swap price then the order must be fully matched
    - If order price == swap price then the order can be fully/fractionally matched
    - If order price < swap price then order must not be matched at all

- Swap orders from Y to X

    - If order price < swap price then the order must be fully matched
    - If order price == swap price then the order can be fully/fractionally matched
    - If order price > swap price then the order must not be matched at all

The liquidity pool contributes liquidity to order matching through the Equivalent Swap Price Model.

4.3 Equivalent Swap Price Model

As seen in section 3.1, the Constant Product Model results in differences between the Swap Price and Pool price after execution. This creates an issue when using an order book with the liquidity pool since there will be a difference between the swap price in the order book and the price offered by the liquidity pool. To address this, the swap price calculation is redefined, so that swap price and post-swap pool price are equivalent:

$$p_p = \frac{R_x + \Delta_x}{R_y - \frac{\Delta_x}{p_s}} = p_s$$

Solving the above equation for SwapPrice p_s yields

$$p_s = \frac{(R_x + 2\Delta_x)}{R_y}$$

Compared to the Constant Product Model, the Equivalent Swap Price Model reduces the arbitrage opportunity because the pool price lands precisely on the last swap price.
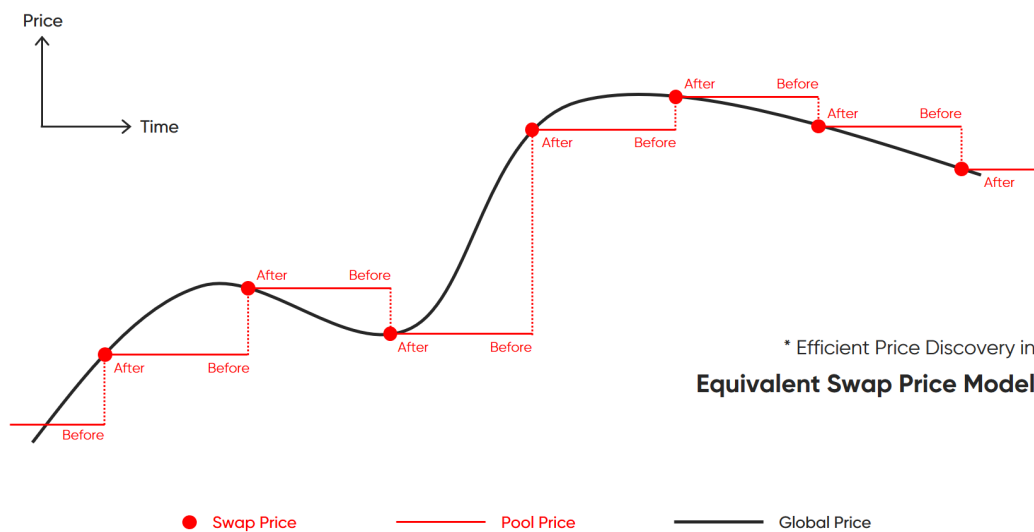

Figure 4.1: Efficient Price Discovery By Equivalent Swap PriceModel

However, it should be noted that the result of this price equivalence is that the Constant Product Formula does not hold. This means that the liquidity pool balance is path-dependent. (Buterin 2017)

This model decides the number of orders provided by the liquidity pool for a given swap price p_s

$$\Delta_y = \Delta_x = 0 \qquad \text{when } p_s = p_p$$

$$\Delta_y = \frac{p_s R_y - R_x}{2p_s} \qquad \text{when } p_s > p_p$$

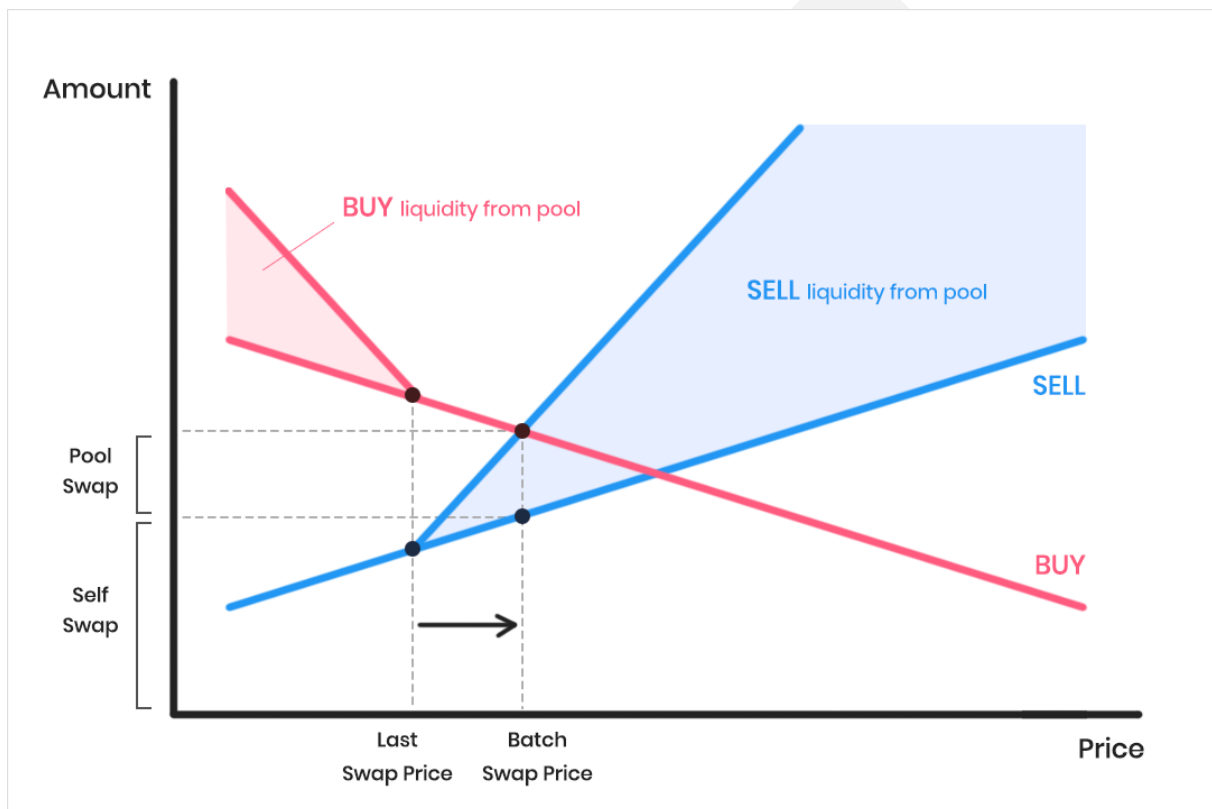$$\Delta_x = \frac{R_x - p_s R_y}{2} \qquad \text{when } p_s < p_p$$

Figure 4.2: Demand and Supply

4.4 Fees

There are two types of fees within the liquidity module: those related to the administration of pools and those that are transaction-related. All fees have some specific economic purpose.

Pool Administration:

- Pool Creation: a fee is charged when a new pool is created. The aim is to prevent excessive pool creation and encourage contributions to existing liquidity pools. These fees are paid in

Atoms into the community fund.

- Pool Withdrawal: investors withdrawing funds from the pool are charged a fee proportional to the amount withdrawn. These fees are accumulated in the pool for the benefit of the other investors. The aim is that this should protect remaining investors from attack vectors that incorporate frequent deposit and withdrawal.

Transaction Fees

Firstly, it should be noted that - as per any other transaction included in a block - gas fees are levied. These fees are

- Paid upon block commit
- Paid even if the order is ultimately not executed
- Fees go to Atom delegators, validators, and also to the community fund. ("Gas and Fees" 2021)

Finally, a Swap fee proportional to the transacted amount is paid on order execution to the Liquidity Pool where that execution occurs. These are accumulated in the pool and distributed pro-rata to each pool investor. This is akin to the Uniswap model.

Initial analysis proposes the following fee structure:

- Swap Fee Rate = 0.003 (0.3%)
- Withdraw Fee Rate = 0.003 (0.3%)
- Pool Creation Fee = 100Atom

5 Conclusion

This paper briefly outlines the proposed design of a Cosmos Hub AMM. We believe that this will bring significant financial utility to the Cosmos Network through the following features:

- Liquidity provision for token exchange without relying on centralized operators,
- High-quality liquidity through the combination of AMM mechanism and traditional order book system,
- Fee earning opportunities for Cosmos Network users through liquidity pool participation,
- Possibility to trade any tokenized assets, including other DeFi investment tokens

In addition, Cosmos Hub AMM liquidity provision plus offers exciting opportunities to put the Cosmos Hub at the center of an inter-blockchain financial ecosystem.

The potential of the Cosmos Hub AMM is inspiring. With IBC's asset transfer capability, it offers Cosmos Hub an opportunity to become the center of the inter-blockchain financial ecosystem.

Module implementation can be found at https://github.com/tendermint/liquidity/tree/develop

References

Angeris, Guillermo, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. 2020. "An Analysis of Uniswap Markets." https://arxiv.org/pdf/2009.14021.pdf.

"Auction Terminology." 2020. Wikipedia. 2020. https://en.wikipedia.org/wiki/Auction#Terminology.

Buterin, Vitalik. 2017. "On Path Independence." 2017. https://vitalik.ca/general/2017/06/22/marketmakers.html.

"Cosmos ETH Peggy." 2021. Althea. 2021. https://blog.althea.net/gravity-bridge/.

"Cosmos IBC." 2021. Cosmos. 2021. https://docs.cosmos.network/v0.40/ibc/overview.html.

"Gas and Fees." 2021. Cosmos. 2021. https://github.com/cosmos/cosmos-sdk/blob/afda62174fe6531b2a40c595a4d9396c28e8b391/docs/basics/gas-fees.md.

Pourpouneh, Mohsen, Kurt Nielsen, and Omri Ross. 2020. "Automated Market Makers." IFRO Working Paper 2020/08. University of Copenhagen, Department of Food; Resource Economics. https://EconPapers.repec.org/RePEc:foi:wpaper:2020_08.

"Traditional Orderbook System." 2021. Wikipedia. 2021. https://en.wikipedia.org/wiki/Order_book.

"Uniswap V2 Core." 2020. Uniswap. 2020. https://uniswap.org/whitepaper.pdf.

"What Is Cosmos?" 2021. Cosmos. 2021. https://cosmos.network/intro.

Zhou, Liyi, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. 2020. "High-Frequency Trading on Decentralized on-Chain Exchanges." https://arxiv.org/pdf/2009.14021.pdf.