

Neptune Money Market:

A next generation lending platform with in built leveraged investing and a PID transformed interest rate curve

E. Woolsey

Summary

Typical DeFi money markets rely on a static curve which determines interest rates as a function of the current market utilization. This approach has been shown to work reliably, but is unable to consistently achieve a stable market utilization. Neptune is a next generation money market that is able to dynamically adjust the interest rate curve in order to achieve consistent utilization, even throughout rapid market volatility. This is achieved through the use of a PID control algorithm that vertically stretches the interest rate curve in a way which counteracts destabilizing external forces. In addition to this, Neptune is able to offer leveraged investing by integrating with local DEXs and allowing users to swap locked collateral held in their accounts.

1 Assumptions and Scope

Neptune consists of three main pieces of infrastructure. The first is a set of smart contracts that exist within a cosmos blockchain. These smart contracts form the core of the protocol and handle all logic related to borrowing, lending, interest rates, and liquidations. The second piece of infrastructure is the off chain software (OCS) which is responsible for executing various heart beat functions that take place on chain. These include updating oracles, triggering control system updates, and executing liquidations. The last piece of infrastructure is the web app, which serves as the interaction point between the users and the smart contracts themselves. It provides all of the necessary information and functionality for users to see their balances, pertinent financial information, and to interact with the blockchain.

2 Smart Contract Architecture

This section will focus on the overall structure and responsibilities of Neptune's core smart contracts; The Price Oracle, The Interest Model, The Market, and the Querier. The NEPT Token contract is discussed separately in section 11.

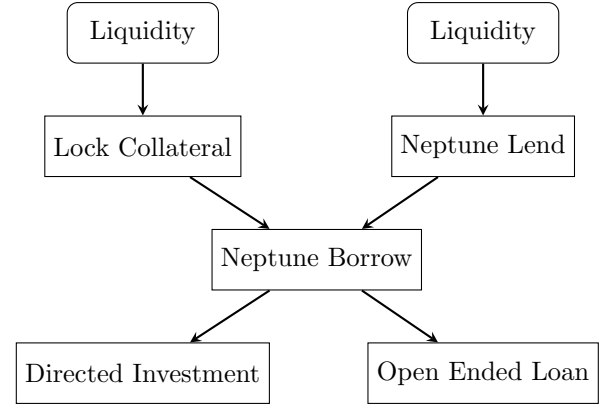


Figure 1: A visualization of the flow of liquidity through the Neptune Money Market

2.1 The Price Oracle Contract

The Price Oracle is responsible for providing the source of truth that the Neptune Money Market uses to obtain asset prices and it determines the price at which users' assets are put up for liquidation. Prices are obtained by the Price Oracle Feeder which will be discussed in depth in section 10. The Price Oracle keeps track of the timestamp of the last price update and returns this timestamp when queried. This allows the Market contract to determine if the price is stale. Any price older than a configurable time window is rejected outright, and any action which depends on that price will fail until a new update is received. This ensures that the market is never able to open, modify, or liquidate a position using an outdated price.

The prices of receipt tokens are not fed to the oracle directly. Instead, they are derived on demand from the price of the underlying asset and the current exchange ratio of the receipt token, as described by equation 1.

2.2 The Interest Model Contract

The Interest Model is responsible for calculating the borrow rate for each individual market. It contains the logic for the PID control system and for the polynomial interest rate curve which will be discussed in depth in section 9. The Interest Model relies on being executed at regular intervals by the PID Updater

which is discussed in section 10.

2.3 The Querier Contract

The querier contract serves simply to provide additional queries that are unnecessary for the operation of the system, but provide a convenience for users and developers.

2.4 The Market Contract

The market contract is the heart of the Neptune Money Market. Its primary responsibilities are to manage lending, borrowing, flash loans, leveraged swaps, and collateral liquidations. It is the point of interaction for all external execute messages and contains the logic to keep track of individual account balances and the distribution of interest. It also serves as the hub or connection point which connects the entire set of Neptune smart contracts together. Before any action is performed on a given market, the market contract first distributes all interest that has accrued since the previous interaction, ensuring that every action is performed against an up to date state.

3 Lending

The lending model used by Neptune is similar to that of Compound Finance. When a user deposits an asset into the market they receive a receipt token (or n-token) in return, which represents their share of the lending pool. The receipt token is minted by the market contract upon deposit and is burned by the market when the user redeems it to retrieve the underlying asset. The first deposit into a market is issued receipt tokens 1:1, and each subsequent deposit is issued receipt tokens in proportion to its share of the lending pool. As interest accrues, the lending pool balance grows while the receipt token supply remains unchanged, so each receipt token is redeemable for an ever increasing quantity of the underlying asset. Prices for receipt tokens are generated using the price of the underlying asset in conjunction with the lending pool balance and the corresponding receipt token supply.

$$P_{receipt} = P_{underlying} \frac{Balance_{lending}}{Supply_{receipt}} \quad (1)$$

A user's share of the lending pool is represented solely by the quantity of receipt tokens that they hold. When interest is distributed from borrowers to lenders, their individual quantity of shares in either pool remains unchanged, but the balances of both the debt pool and the lending pool are increased.

Lenders may redeem their receipt tokens at any time, provided that there is sufficient unborrowed liquidity remaining in the market. A redemption that would cause the total amount borrowed to exceed the total amount lent is rejected. It is the job of the interest

rate control system described in section 9 to ensure that this situation is rare and short lived.

4 Collateral

In order to borrow on the Neptune Money Market, a user must first deposit collateral into their account to secure their loan. Denominations must specifically be whitelisted for use as collateral, and significant effort and risk analysis must be undertaken before a new collateral type is added to the market. Each collateral denomination has a specific set of parameters that are associated with it including *liquidation ltv*, *allowable ltv*, *min discount*, *max discount*, and *collateral cap*.

The *ltvs* and discounts for a specific collateral are selected based on the denomination's risk profile. Assets with low liquidity or high volatility present a higher risk to the system and therefore require a lower *ltv* and a larger discount. It is imperative that liquidators are offered significant rewards and given sufficient time so that they can effectively perform liquidations when the market demands it. The *collateral cap* is an optional hard limit on the total cumulative amount of a given collateral that can be used to secure margin positions. This total should not approach or exceed the volume of liquidity that is available locally so that efficient liquidations can be performed in any and all market conditions.

4.1 n-Tokens as Collateral

Using n-tokens as collateral allows users to put their collateral to work, passively earning interest while simultaneously securing their margin position. n-tokens used as collateral are not treated any differently than any other form of collateral. The same parameters are set for these tokens and they are turned over directly to the liquidator as-is in the event of a liquidation.

5 Borrowing

After a user has deposited collateral into the market, they may borrow against it. The amount that they can borrow is determined by the allowable loan to value ratio (L_a). The minimum *LTV* that an account may be liquidated at is defined as the liquidation *LTV*, or L_l . The allowable *LTV* is always set below the liquidation *LTV*, so that a newly opened position has a buffer before it becomes eligible for liquidation.

The value of a given collateral denomination n is given by

$$C_n = c_n P_n \quad (2)$$

where:

c_n denotes the balance of the collateral and
 P_n denotes its price.

The value of a given debt denomination m is given by

$$D_m = d_m P_m \quad (3)$$

where:

d_m denotes the balance of the debt, including accrued interest, and

P_m denotes its price.

Each collateral denomination can have a unique L_a and L_l and when multiple collateral denominations are used within a single margin account we combine these concepts into an overall account health. A useful quantity referred to as Scaled Collateral Value is given by

$$C_s = \sum_n L_{ln} C_n \quad (4)$$

where:

$\sum_n$ denotes the sum over all collateral in the account
 L_{ln} denotes the liquidation LTV of collateral n .

Then finally account health is given by:

$$H = \frac{C_s}{D} = \frac{\sum_n L_{ln} C_n}{\sum_m D_m} \quad (5)$$

Accounts with a health of less than 1 are considered under-collateralized and are at risk of liquidation. The responsibility is on the user's shoulders to maintain sufficient account health by ensuring there is sufficient collateral value in their accounts at all times.

New borrows and collateral withdrawals are only permitted if the account would remain within its allowable LTV after the action is performed, that is

$$\sum_m D_m \leq \sum_n L_{an} C_n \quad (6)$$

Each borrowable asset additionally has a *borrow halt utilization* and an optional *borrow cap*. New borrows that would push the market utilization above the borrow halt utilization, or the total amount borrowed above the borrow cap, are rejected. This guarantees that a minimum amount of liquidity is always left available for lenders to withdraw.

5.1 Cross Margin

When depositing collateral for the first time in the Neptune Money Market, a new account is created for that address with an index of 0. An individual wallet may open numerous margin accounts, each with a unique index and may borrow any number of distinct debt denominations against any number of distinct collateral denominations within a single margin account. Each margin account is isolated from the others allowing users to maintain different risk profiles in different accounts without worrying that a poorly performing asset in one account could potentially affect the liquidation risk of another.

6 Leveraged Investing

Borrowed funds on a conventional money market leave the protocol entirely. A user wishing to take a leveraged position must borrow an asset, withdraw it, swap it on a DEX, deposit the result as collateral, and repeat this process until the desired leverage is reached. Each loop costs gas and exposes the user to price movements between transactions. Neptune removes this friction by integrating directly with local DEXs, allowing users to swap assets that remain locked within their margin accounts. This is the *Directed Investment* path shown in figure 1.

6.1 Exchanges

For each pair of assets, governance may whitelist a single liquidity pool to be used by the market when swapping between them. Only whitelisted pools are ever used, which ensures that swaps are routed through venues with sufficient depth and that a malicious pool cannot be substituted by a user. Pairs may be added, updated, or removed at any time.

6.2 Margin Account Operations

The following operations are available on any margin account. Each operation executes atomically, and the account must satisfy the allowable LTV condition of equation 6 once it is complete, otherwise the entire operation is reverted.

Over Leverage Borrows an asset and immediately swaps it into a different collateral asset, which is deposited back into the same account. This allows a user to open a leveraged position in a single transaction.

Swap Collateral Swaps one collateral asset held in an account for another, without the collateral ever leaving the protocol. This allows a user to rebalance or rotate a position while their debt remains open.

Sell Collateral to Repay Debt Swaps collateral directly into a debt asset and uses the proceeds to repay the debt. This allows a user to reduce their leverage, or to close a position entirely, without needing to source the debt asset externally.

Lend Collateral Lends collateral held in an account to the market, and replaces it in the account with the corresponding n-tokens. The collateral continues to secure the account while also earning interest.

Redeem Lent Collateral The reverse of the above, redeeming n-tokens held as collateral back into the underlying asset.

Over Leverage and Swap Collateral accept a minimum receive amount, and are reverted if the swap would return less than this amount. This protects users from excessive slippage and from being sandwiched by other actors on chain. The destination asset must be whitelisted as collateral and must not exceed its collateral cap, and any new borrow must respect the borrow halt utilization and borrow cap of its market.

6.3 Maximum Leverage

Consider a user who deposits collateral of value C and uses Over Leverage to borrow a value D which is swapped into more of the same collateral. Ignoring swap fees and slippage, the account remains within its allowable LTV provided that

$$D \leq L_a(C + D) \quad (7)$$

which gives the maximum debt that can be taken against the initial deposit,

$$D_{max} = \frac{L_a}{1 - L_a} C \quad (8)$$

The total exposure of the position is then $C + D_{max}$, so the maximum leverage available for a given collateral is

$$\lambda_{max} = \frac{1}{1 - L_a} \quad (9)$$

For example, a collateral with an allowable LTV of 0.75 supports up to $4\times$ leverage. Because leverage is bounded by the same L_a used for ordinary borrowing, leveraged positions require no special treatment by the liquidation system. They are simply margin accounts like any other, and are liquidated in exactly the same way.

7 Liquidation

In order to protect the protocol from loans that become under-collateralized during a market downturn, Neptune must be able to effectively liquidate positions to recoup capital owed to lenders. The Neptune team has developed a novel approach to create capital efficient outsourced liquidations using a dynamic liquidation discount.

Each accepted collateral denomination has a minimum δ_{min} and maximum δ_{max} discount, and the market defines a characteristic width ω . The characteristic width describes the magnitude of change in an account's health below 1.0 which will result in the maximum discount being issued to the liquidator. The formula for the chosen discount for a given account health is given by:

$$\delta = \begin{cases} \frac{(1-H)(\delta_{max}-\delta_{min})}{\omega} + \delta_{min} & \text{if } H \geq 1 - \omega \\ \delta_{max} & \text{if } H < 1 - \omega \end{cases} \quad (10)$$

where:

$H < 1$ is the health of the account prior to liquidation.

The intention of this system is to encourage liquidators to compete with one another by linearly increasing the discount as the health of an account begins to deteriorate. Liquidators must wait long enough until the discount is sufficiently high to make the liquidation profitable, but not *so* long so as to allow another liquidator to execute before them.

The general process for executing a liquidation is as follows:

Secure Funds The Neptune liquidation system requires liquidators to secure the assets required to repay the debt held by the under-collateralized account. This can be done conventionally or by taking out a flash loan from Neptune's Market.

Calculate Liquidation Parameters The Liquidate execute message takes in a number of parameters that must be decided upon by the liquidator. Because the liquidation call is rather complex, it has been built in a *forgiving* manner, meaning it will attempt to liquidate a provided account as much as possible, within the bounds of the parameters provided. For a given liquidation operation the liquidator must provide the following parameters:

- The amount, denomination, and minimum acceptable discount of each collateral they wish to receive.
- The amount and denomination of each debt to be repaid.

Execute The Call The Market contract will attempt to liquidate the account, returning the liquidated collateral to the liquidator, as well as any unused assets that were sent in excess of what was required.

When an under-collateralized account is liquidated it will be flagged for complete liquidation if the total value of its collateral is below the partial liquidation threshold. The purpose of the partial liquidation threshold is to incentivize liquidators to liquidate small accounts. If the reward of a liquidation is not high enough to cover the cost of gas, these small accounts may be ignored and eventually become delinquent. If an account is above the threshold the market will attempt to partially liquidate the account, bringing its health back up to the liquidation health target.

During a partial liquidation, the maximum amount of collateral denomination n offered to the liquidator is given by:

$$\Delta c_n = \frac{H_T D - C_s}{P_n[H_T(1 - \delta)(1 - f) - L_{ln}]} \quad (11)$$

where:

H_T denotes the target account health for the

liquidation

δ represents the liquidation discount

f represents the fixed rate protocol liquidation fee

If the denominator of equation 11 is not positive, the target health cannot be reached by liquidating collateral n , and the entire balance of that collateral is made available instead. Of the collateral that is liquidated, a fraction f is retained by the protocol and sent to the war chest, and the remainder is transferred to the liquidator.

7.1 Bad Debt

In extreme market conditions it is possible for an account's collateral to lose value faster than it can be liquidated. If an account has had all of its collateral liquidated but still holds outstanding debt, that debt can no longer be recovered. In this case the remaining debt is removed from the debt pool and the corresponding amount is deducted from the lending pool. The loss is therefore spread evenly across all lenders of that asset in proportion to their share, rather than leaving the market permanently insolvent. Conservative *LTVs*, collateral caps, and a healthy community of liquidators exist to ensure that this scenario is never encountered in practice.

8 Flash Loans

Flash loans allow any user to borrow assets from the market without collateral, provided that the loan is repaid in full within the same transaction. When a flash loan is requested, the market transfers the requested assets to a receiver contract and executes the user provided messages from that contract. Once these messages have executed, the market asserts that the loan, along with a fixed flash loan fee, has been fully repaid. If it has not, the entire transaction is reverted, and it is as if the loan never took place. Because of this, flash loans present no risk of loss to lenders.

The primary purpose of flash loans within Neptune is to enable capital efficient liquidations. A liquidator does not need to hold the assets required to repay an under-collateralized account's debt. They may instead borrow those assets via a flash loan, liquidate the account, swap the received collateral on a local DEX, and repay the loan, all within a single atomic transaction. Access to flash loans is gated by the quantity of staked NEPT held by the borrower, as discussed in section 11. Flash loan fees are sent directly to the war chest.

9 Interest Rates

The recent collapse of the Terra ecosystem was in part caused by the unnatural subsidizing of interest rates found on Anchor Protocol. A healthy financial ecosystem must stem from the balance of supply and demand, and interest rates must be generated organically, and algorithmically. Anchor's absurdly high interest rate caused a rapid expansion of the money supply that quickly caused UST to become under-collateralized, which played a significant role in the depegging event. Neptune sets interest rates using a PID transformed curve with the aim of stabilizing the deposit utilization ratio (the ratio of capital borrowed vs lent). This solution combines a simple polynomial utilization curve with a PID controller to get the advantages of both technologies. A higher deposit utilization ratio means a more efficient market, but comes with the caveat that lenders may not be able to immediately withdraw liquidity in the event of a significant market volatility. The utilization ratio must then be balanced and stabilized with some sort of decentralized control system.

9.1 Utilization Curve

Traditional DeFi platforms such as Anchor implement a utilization curve to set interest rates. The interest rate is simply a monotonically increasing and concave up function of the utilization ratio $U(\frac{B(t)}{D(t)})$. Utilization curves respond immediately to system disturbances but they fail to precisely maintain stable utilization rates. Natural market forces are constantly changing, and a static curve simply cannot balance these ever changing demands.

9.2 PID Controlled Interest Rates

A new approach to setting interest rates was proposed by Mars Protocol. Put simply, the utilization ratio is used as the process variable in a simple closed loop PID system. PID controllers excel at maintaining stability and precision in complex and unpredictable systems. The general form for a PID control system is given here.

$$\gamma(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (12)$$

where:

K denotes 3 constants which impact the response of the control system.

$e(t)$ denotes the error function.

$\gamma(t)$ denotes the control function.

For a PID controller to be effective it must have sufficiently high input bandwidth so that it can correctly respond to high frequency disturbances. When

operating within a blockchain, this can present a difficult task. An uncharacteristically sudden influx of borrowers could saturate the market's utilization before a purely PID controlled system could even react. Frequent changes in market elasticity further complicate the issue of effectively tuning a PID as well. Neptune has devised a new model that combines these techniques and solves all of these difficulties.

9.3 A Hybrid Model: PID Transformed Curve

Instead of using the control function γ (the output of the PID) to directly set the interest rate, we use it to transform a predefined utilization curve.

$$I_{Borrow} = U(\gamma(t), \frac{B(t)}{D(t)}) \quad (13)$$

This curve takes the form of any arbitrary polynomial. This allows for maximum flexibility as each coefficient can be modified on demand. This polynomial curve is then stretched vertically by the output of the control function, providing dynamic response to changing market conditions.

$$I_{Borrow} = \gamma(t) \sum_{k=0}^n a_k u(t)^k \quad (14)$$

The curve responds instantly to changes in utilization, absorbing high frequency disturbances that the PID would be too slow to react to, while the PID slowly stretches the curve to correct for long term shifts in market demand.

In the case of Neptune, the error function $e(t)$ is derived from the discrepancy between the target utilization ratio and the actual ratio. The control function $\gamma(t)$ is input as a parameter into the utilization curve. The error function is given by

$$e(t) = u_T - u(t) = u_T - \frac{B(t)}{D(t)} \quad (15)$$

9.4 Discrete Implementation

On chain, the PID is evaluated in discrete steps each time it is triggered by the PID Updater. A naive implementation of equation 12 would be highly sensitive to the instantaneous utilization at the moment of each update, which could easily be manipulated by a large borrow or deposit placed immediately before the update. To prevent this, the market maintains a utilization accumulator, which is the integral of the utilization ratio over time. It is updated every time interest is distributed:

$$A(t) = \int_0^t u(\tau) d\tau \quad (16)$$

The measurement used by the PID at step k is then the time weighted average utilization since the previous step,

$$\bar{u}_k = \frac{A(t_k) - A(t_{k-1})}{t_k - t_{k-1}} \quad (17)$$

so that a brief spike in utilization only influences the controller in proportion to how long it persisted.

Several further measures are taken to ensure the controller behaves predictably:

Derivative on Measurement The derivative term is computed from the change in the measured utilization rather than the change in error, $-K_d(\bar{u}_k - \bar{u}_{k-1})$. This prevents a sudden jump in output (derivative kick) when the target utilization u_T is changed.

Integral Accumulation Rather than storing the running sum of the error, the contract stores the running sum of $K_i e_k$. This allows K_i to be retuned at any time without causing a discontinuity in the output.

Limits The proportional, integral, and derivative terms, as well as the total output γ , are each clamped between configurable upper and lower bounds. Clamping the integral term in particular prevents integral windup during extended periods where the target utilization cannot be reached.

Sample Period Each market defines a minimum and maximum sample period. Updates attempted before the minimum period has elapsed are rejected. If an update arrives after the maximum period has elapsed, the derivative term is ignored for that step, preventing an irregular update period from producing an erratic output.

9.5 Interest Rate Spread

Neptune protocol generates revenue through a spread in the borrowing and lending interest rates. Each market defines a protocol interest fee r_f , which is charged to borrowers in addition to the rate produced by the PID transformed curve. Fees generated through the interest rate spread are allocated to the war chest.

$$I_{Borrow} = \gamma(t) \sum_{k=0}^n a_k u(t)^k + r_f \quad (18)$$

$$I_{Lend} = (I_{Borrow} - r_f) \frac{B(t)}{D(t)} \quad (19)$$

where:

I_{Borrow} is the interest rate provided to borrowers.

I_{Lend} is the interest rate given to lenders.

B is the total value of borrowed assets.

D is the total value of deposits.

r_f is the protocol interest fee.

The output of the PID transformed curve is never allowed to fall below zero, so the protocol interest fee also serves as the minimum borrow rate. This guarantees that the lending rate can never become negative.

9.6 Interest Accrual

Interest is accrued continuously rather than in discrete payouts. Whenever any action is taken on a market, the market first calculates the time Δt elapsed since interest was last distributed, and accrues interest for that period. The debt pool balance B is increased by $I_{Borrow}B\Delta t$, the lending pool balance D is increased by $(I_{Borrow} - r_f)B\Delta t$, and the remaining $r_fB\Delta t$ is set aside as protocol revenue for the war chest. Because interest is always distributed before any deposit or withdrawal is processed, a lender can only ever earn interest for the period of time over which their liquidity was actually lent. There is no distribution event that can be front run, and no need for an external party to trigger distributions.

9.7 Modeling

Modeling the time dependent market response to changing interest rates is an inherently difficult task to perform accurately. The markets were modeled using a Monte Carlo simulation made up of a large quantity of randomized users, each consisting of various properties that were thought to be important for the purposes of this study. In no particular order, these properties are:

Threshold Interest Rate The interest rate at which a user (lender or borrower) will invest or divest their capital in the market.

Capital The quantity of capital that a user has to place in the market.

Response Time The time between when the Threshold Interest Rate is crossed, and when a user invests or divests their capital.

Investment Status Holds state for a user, keeping track of whether they are currently invested or divested.

Trigger Time Keeps track of the time at which an investment or divestment transition has begun.

Random values are assigned to each of these properties according to a Gaussian distribution and a large number of both borrowers and lenders are generated. Various perturbations were applied to the system to simulate the market responding to a variety of external forces. Examples of these perturbations include:

- A sudden large influx/efflux of borrowers.

PID Interest Rate Controller

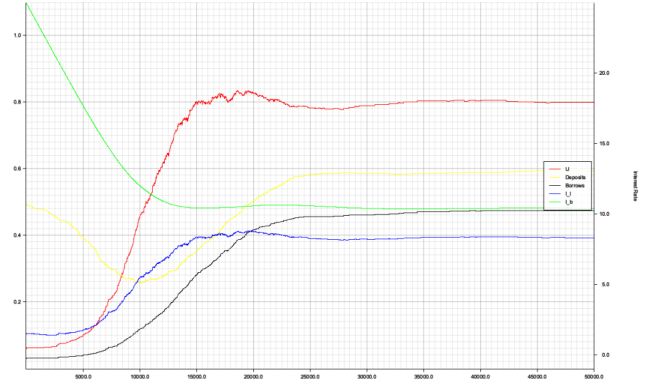


Figure 2: The market response during a rapid initialization. The market starts with a surplus of lenders and a very low utilization ratio. The control system rapidly drops the interest rates, causing an influx of borrowers, and the utilization ratio stabilizing at its set point, 0.8 in this case.

- A sudden large influx/efflux of lenders.
- A sudden large change in overall market interest rates.

In our testing we found that our control system was able to stabilize interest rate volatility, maintain market liquidity, and accurately track the target utilization ratio set point throughout each tested perturbation.

Finding accurate distributions to apply to individual users is difficult, as actual experimental data is very hard to come by or unavailable altogether. This makes performing a rigorous study and precisely tuning PID parameters purely from simulation very impractical. For these reasons, the distributions chosen for the Monte Carlo were selected based off of qualitative examination of real world market trends, and idealized incentives for borrowers and lenders. These simulations serve purely as a proof of concept, rather than a rigorous tool for tuning control system parameters. In practice, the control system is initialized with very conservative gains. It has been demonstrated by other DeFi protocols that a simple static curve is sufficient for producing market stability. This means that Neptune can operate safely with low PID gains, and slowly increase these parameters over time as more real world market data is collected, thus improving Neptune's ability to quickly respond to market changes.

10 Off Chain Software

The off chain software (OCS) is responsible for executing various heart beat functions that take place on chain. These include triggering control system updates, executing liquidations, and updating oracles.

Neptune hosts an incredibly robust network of public nodes which are utilized extensively to access the blockchain as fast as possible. Crypto markets can be incredibly volatile and transaction speed execution is critical in securing the safety of the protocol. Neptune has also deployed a globally decentralized and automated network of bots that function to trigger mission critical events on chain, and to update price oracles quickly and reliably. The OCS can be generally divided into 4 separate types of software, trigger bots, executer bots, indexers, and web apps.

10.1 Trigger Bots

Trigger bots have the simple responsibility of triggering on chain events at some predetermined interval. The execution's payload contains no important external information, it simply serves as a starting point for logic to execute on chain. Neptune employs the use of a single trigger bot, the PID Updater.

10.1.1 PID Updater

The PID Updater has the responsibility of triggering the PID to update and obtain the next control output. Each update first distributes any outstanding interest on the affected markets, bringing their utilization accumulators up to date, and then computes the next control output for each market. The PID updater is called at a fixed interval, and necessarily so. For PID control systems to respond correctly it is imperative that there is minimal variation in the update period. The Interest Model Contract has logic built into it that prevents irregular update periods from leading to erratic behavior, but it is still best to avoid this situation altogether.

10.2 Executer Bots

Executer bots are responsible for executing on chain events that require external information. The Neptune Money Market uses 2 such bots, the Price Oracle Feeder and the Liquidation Bot.

10.2.1 Price Oracle Feeder

The Price Oracle Feeder is Neptune's most critical piece of off chain software. It is responsible for updating the Price Oracle Contract. The Oracle Feeder is a globally distributed network of software which fetches prices of all supported assets from various web2 sources, aggregates them, and uses a robust algorithm to discard outliers and determine the most accurate market price. A new price is pushed on chain whenever it deviates from the last published price by more than a set threshold, or when a maximum amount of time has elapsed since the last update. This keeps gas costs low during periods of calm while ensuring that prices

are always updated well within the validity window enforced by the market.

10.2.2 Liquidation Bot

Liquidations are integral to the health of the protocol, and for this reason Neptune encourages and incentivizes other engineers and teams to take part in liquidations. Liquidation is fully permissionless, and anyone may liquidate an under-collateralized account. Alongside these community liquidators, the Neptune team operates its own liquidation bot, ensuring that under-collateralized accounts are always liquidated promptly.

Neptune's liquidation bot checks the state of all margin accounts of the money market and liquidates under-collateralized accounts back to the liquidation health target. The bot attempts to execute profitable liquidations via flash loans while considering the cost of swapping via liquidity pools. It keeps track of all margin accounts via event-listening of on-chain state-changing transactions and has periodic self-healing mechanics to ensure precise and fast processing of liquidations. It keeps track of all the available liquidity pools' statuses and keeps an up-to-date swapping router graph for profit optimization. This bot has no direct access to funds as its transaction fees are paid by another account via the Fee Grant module. All required assets to liquidate come from the flash loan and all profits are sent to another cold wallet.

10.3 Indexers

Indexers are responsible for indexing the state of the blockchain and to make this information quick and easy to access for other parts of the OCS to consume. In particular, the liquidation bot makes extensive use of the indexer to quickly determine the state of all active margin accounts within the protocol. The indexer is also used by the web app to provide users with a rich dashboard of their account information.

10.4 Web Apps

The web app is the interface between the end user and the protocol which exists on the blockchain. The UI is designed to be both sleek and intuitive, so that users can enjoy interacting with our systems. It provides the functionality required to connect a user's wallet and to interact with the smart contracts, along with an information rich dashboard that allows users to track their positions and to view the health of their personal account, and the health of the protocol as a whole as well.

11 The NEPT Token

NEPT is the protocol token of the Neptune Money Market. The purpose of this token is two fold. Firstly, it provides users of the system with governance. This puts the future of the protocol in the hands of its users. Secondly, holders of the token who stake it are able to improve the terms on which they interact with the protocol, through improved account health and access to flash loans.

11.1 Staking

NEPT is staked by bonding it to the Token contract for one of a set of predefined durations. Longer durations represent a greater commitment to the protocol, and as such each duration is assigned its own set of weights which determine its influence over staking rewards, governance voting, account health, and flash loan access. Once a new duration has been added it can never be removed, and its reward and governance weights can never be changed, so that stakers can be confident in the terms of their bond.

When a user chooses to unbond, their stake does not become liquid immediately. It must first serve out a cooldown equal to its bond duration. Stakes may also be configured to cascade, stepping down to the next shortest duration once a cooldown has elapsed, and eventually entering the unbonding queue once the shortest duration is reached. Tokens that are unbonding can be rebonded at any time.

Staking rewards are emitted continuously at a fixed annual emission rate, and are split between all stakers in proportion to their reward weighted stake. Emissions cease once the total supply of NEPT has been issued.

11.2 Token Utility

11.2.1 Account Health

Stake can be bonded against a specific margin account index, which improves the health of that account. The health weighted value of the stake S is compared against the total value of the account's collateral C to produce a health modifier

$$M = 1 + m \min \left(1, \frac{S}{\rho C} \right) \quad (20)$$

where:

m is the maximum health improvement provided by staking.

ρ is the ratio of stake to collateral value required to obtain the maximum improvement.

The account health given by equation 5 is then multiplied by M , and the same modifier is applied to the allowable LTV check and to the target health used

in partial liquidations. This allows dedicated stakers to borrow a little more against the same collateral, and gives them a little more room before their position becomes eligible for liquidation.

11.2.2 Flash Loans

In order to take out a flash loan, a user must hold a flash loan weighted stake whose value is at least a fixed fraction of the value of the requested loan. This ties the use of flash loans to a long term commitment to the protocol, and rewards the liquidators who keep Neptune safe.

11.2.3 Governance

The governance weighted stake of each user is used as their voting power in protocol governance, discussed in section 12.

11.3 Token Distribution

Tokens allocated outside of staking emissions are distributed through on chain token unlocks held by the Token contract. Each unlock specifies a start time and a schedule, either a lump sum or a linear release over a set duration, and may optionally be assigned an admin who is able to reclaim unvested tokens.

12 Governance

The Neptune Money Market is a decentralized protocol, and as such it is important that the future of the protocol is in the hands of its users. Holders of staked NEPT are able to vote on proposals governing every aspect of the protocol, from the use of war chest funds to the configuration of individual markets.

Every privileged action across the Neptune smart contracts, such as adding a new market or collateral, modifying risk parameters, or retuning the PID, is gated by a permission system. Governance is a distinct permission group within this system and is able to perform every action that the contract owner can. Voting power is determined by each user's governance weighted stake, which is automatically kept up to date with the governance contract whenever stake is bonded, unbonded, or cascaded.

12.1 The War Chest

As the protocol accumulates fees through the distribution of interest, through liquidations, and through flash loans, they are sent to a special contract named the war chest. The war chest is controlled by governance and serves multiple important purposes.

During volatile market movements and a massive decrease in demand for lending, if the market utilization were to saturate, the war chest can be utilized to

811 take over exiting lenders' positions. This would allow
812 lenders to recoup their capital quickly, rather than
813 waiting for changing interest rates to affect the market
814 state.

815 Another use for the war chest is to insure the proto-
816 col against a catastrophic failure. If the protocol were
817 to be exploited or a large margin position were some-
818 how able to become significantly under-collateralized
819 before being liquidated, the war chest could be used
820 to cover the losses.

821 The war chest is also used to fund community
822 projects and developer teams to encourage them to
823 build upon the protocol.