DeFi liquidity management via Optimal Control: Ohm as a case study

Tarun Chitra
tarun@gauntlet.network

Kshitij Kulkarni
ksk@eecs.berkeley.edu

Guillermo Angeris
angeris@stanford.edu

Alex Evans
aevans@baincapital.com

Victor Xu
victor@gauntlet.network

February 2022

Abstract

As decentralized finance grows to autonomously managing hundreds of billions of dollars of assets, capital efficiency has become an ever increasing component of protocol design. Recently, the Olympus protocol (also known as Ω) has utilized a novel liquidity provisioning mechanism that improves capital efficiency. This system introduces the concepts of a decentralized protocol renting, leasing, and buying liquidity when it is required for protocol functioning. In this note, we formalize the notions used by Ohm smart contracts in probabilistic and control theoretic terms. In particular, one can view the Ohm system as a stochastic non-linear control system. We show that the non-linear control mechanism is actually approximating the behavior of a simpler stochastic linear-quadratic regulator. We construct an associated Hamilton-Jacobi-Bellman equation for a mean-variance portfolio optimization problem, and show that the protocol can stabilize price by choosing appropriate portfolios. Our main result shows that the Ω protocol enjoys increasing ability to control price as the number of bond durations increases, but that this ability has diminishing marginal returns. Therefore, using this formalism, we show that with proper dynamic tuning and adjustments, the Ohm protocol can both improve capital efficiency and reduce risk to protocol users. We conclude by generalizing the Ohm controller model to a generic mechanism for optimizing risk and incentives in decentralized protocols, which includes other mechanisms like Tokemak and ve.

1 Introduction

Decentralized Finance (DeFi) has grown to manage well over \$150 billion of assets from 2019 to 2021. Much of this growth comes from protocols that allow two-sided (or more) markets to form around debt issuance, automated market making, and overcollateralized lending.

DeFi elides centralized third-parties who control liquidity flows (*e.g.* banks and brokers) and instead replaces them with a smart contract that manages rules for issuing debt and trading. Elision of third-parties, however, comes at a cost.

In order to execute functionality like liquidations of defaulted loans, DeFi protocols need to incentivize particular actors to interact and provide capital to a protocol. For instance, the two largest lending protcools, Aave and Compound, rely on actors known as liquidators to buy collateral backing a defaulted loan from the protocol [1, 2, 3]. The main reason a protocol does this is to enforce an invariant, *e.g.* the assets held in the protocol are worth more than the protocol's outstanding liabilities [4]. Other protocols, such as Liquity [5], instead incentivize pools of liquidity that are used for automatically buying bad loans, akin to an insurance capital buffer. For such protocols, incentivizing protocol liquidity is equivalent to reducing protocol risk (*e.g.* the quantity of defaulted loans that cannot be removed from the protocol's books). In order to entice users to lock their assets into such a capital buffer pool, protocols issue governance tokens, which act as pseudo-equity in the protocol. Users can use their governance tokens to propose and vote on changes to the protocol, including interest rates, margin requirements, and other parameters.

Liquidity Incentivization. When DeFi protocols initially grew rapidly in the summer of 2020, they incentivized liquidity by providing governance token-based liquidity incentives. Users who allocated capital to protocols in order to earn liquidity incentives were known as *yield farmers* [6]. Protocols effectively acquired users by paying in percentange ownership of the network, initially realizing the dream of decentralized protocols that are wholly owned by their users. However, as the number of protocols rose dramatically in 2020 and 2021 [7], yield farming became more efficient and allocators would rapidly migrate capital between protocols. This led to protocols that launched and were initially funded with billions of dollars of capital in liquidity, only to see that capital quantity dwindle dramatically once token incentives became scarcer.

One of the difficulties of paying for liquidity with equity-like instruments is that the quantity of liquidity provided to a protocol becomes highly correlated to the price of the equity instrument. When there is a large drawdown in the price of a governance token (which serves as the equity-like instrument), one can see liquidity leave from incentivized pools as the numéraire denominated yield also drops [8]. In a sense, a protocol is only really renting liquidity from a user, since they can remove their liquidity at any time. In traditional capital markets, companies usually pay for liquidity with bonds and swaps as opposed to direct sales of equity. However, the only asset that DeFi protocols truly own is the accrued fees from using the platform and the governance tokens that are in its treasury (*e.g.* on balance sheet). As such, they are forced to often pay for protocol liquidity using governance tokens when accrued fees aren't sufficient for network growth.

Automated market makers Curve and SushiSwap were the first liquidity provisioning protocols to try to slow down capital outflows from their protocols by incentivizing long-duration liquidity [9, 10]. Both protocols effectively provided higher interest rates to users who locked up their liquidity for a long time frame. In Curve's case, the main increase in incentivization came from 'boosts' which correspond to higher token incentives for longer lock-up commitments. By adding in duration constraints to ensure that liquidity didn't migrate, these protocols opened up the design space for liquidity incentivization. These vesting lock-ups and incentivization boosters effectively allowed for protocols to 'lease' liquidity for a given timeframe. However, these protocols provided such incentives in an ad hoc manner that did not systematically adjust incentives as a function of market activity and volume. As such, there was a tendency for control over liquidity parameters to be centralized which has led to some catastophic failures [11].

Another concept that became increasingly popular in 2021 is the concept of *protocol-owned liquidity*. This refers to idea that a protocol aggressively uses its earned fees and governance tokens to itself be a liquidity provider on a number of decentralized venues. One of the first protocols to explicitly do this was Fei, which used create-redeem fees to provide liquidity to a Uniswap pool. While the initial design had some setbacks [12], this opened up the design space to DeFi protocols that 'buy' liquidity via the usage of DAO treasury funds.

OlympusDAO. The first protocol to combine all three aspects of liquidity provisioning — buy, lease, and rent — was OlympusDAO or Ohm. OlympusDAO aims to create a low-volatilty reserve asset OHM, or Ω, that can be used as a reserve currency. There are two main actions that owners of Ω can take in the protocol, staking and liquidity provision (referred to as *bonding* in Ohm parlance). The goal of the protocol is to distribute token-based incentives to both stakers and liquidity providers in such a way that ensures reduced volatility while also allowing for new entrants (who want to hold and/or use Ω as a store of value) to join the protocol.

A user who stakes Ω earns a pro-rata portion of the inflation of the Ω token, much like a proof-of-stake cryptocurrency (*e.g.* Solana, Cosmos, or Polkadot). The protocol utilizes staking to control the liquid supply of the asset, much like a central bank that issues re-purchase agreements to reduce the liquid money supply in exchange for future interest rate payments. From the perspective of the Ω DAO, the main purpose of staking is to reduce the liquid supply of Ω, which acts (weakly) as a mechanism to mollify price volatility. On the other hand, a user who provides liquidity deposits Ω and another asset (such as ETH or a stablecoin) into a constant function market maker (CFMM) [13] and provides proof of liquidity provision to the Olympus protocol. These users (referred to as *bonders*) provide their assets to the protocol to ensure that new buyers who want to buy Ω have ample liquidity to enter into positions. Ohm offers different types of bonds — some that have shorter durations (weeks) and others than are longer (months). The longer your duration of liquidity provision, the higher your expected Ω reward. This represents the protocol *leasing* liquidity from liquidity providers rather than simply renting it. The protocol has to balance how much of its inflation and earned fees it spends on incentivizing these two actions.

Currently, the live Ohm protocol automates execution of some portions of the liquidity incentive allocation mechanism while implementing others components manually (*e.g.* via on-chain governance). Our description of the portfolio allocation problem in §3.1 is more general formulation than what is implemented in Olympus. In particular, if one can numerically solve for the parameters of the control models we propose in §3.2, one can optimize and update allocation parameters in a principled manner. Our goal is to demonstrate the versatility and robustness properties of the mechanism, when viewed from the lens of control theory. Once formulated in this manner, we construct an associated stochastic Hamilton-Jacobi-Bellman (HJB) equation for the control problem. This allows protocol developers and managers to utilize numerical methods for solving for the optimal allocation of incentives to different parts of the protocol.

Recently, OlympusDAO governance has proposed using internal bonds as a mechanism for liquidity management and has suggested that "Olympus position itself toward a bond-emphasized model of token emissions and distribution" [14]. We note that the results we derive below show that increasing the number of bond durations can allow the protocol to stabilize the price at a faster rate.

Our Contributions. In the sequel, we will focus on constructing a mathematical formalism for analyzing how Olympus balances incentives. We start in §2 by elucidating the mechanism that Olympus uses in production for determining the incentives paid to bonders as a non-linear feedback controller. The non-linear controller used in production is difficult to analyze from a stability perspective, *e.g.* it is difficult to discern the maximum magnitude of a price shock that the controller can mollify. However, we demonstrate in §3.1 that via a change of variables, the non-linear controller is effectively approximating a linear controller. This linear controller, which optimizes yields given to each pool instead of furnished price discounts, resembles the linear-quadratic regulator (LQR) [15, 16], and is related to a mean-variance portfolio optimization problem. The linear-quadratic regulator has many theoretical guarantees on the stability and long-term behavior of the system. In particular, one can formally answer questions about how much of the control variable (*e.g.* incentives in the form of Ω tokens) is needed to reduce the volatility of the asset by adding liquidity to CFMM pools.

Given this control mechanism, we subsequently formulate the Ohm protocol as a portfolio allocation problem. Within DeFi, equilibria between staking and liquidity provision for lending and staking derivatives have previously been analyzed as portfolio allocation problems [17, 18]. We extend these frameworks to include protocols that incentivize liquidity provision to CFMMs. The first portfolio allocation mechanism chosen is a mean-variance tracking model, described in §3.2. This model is a generalization of the model used in [17].

The mean-variance model can be explicitly represented as a stochastic linear-quadratic problem, which is proved in §4.1. Given that the LQR has an explicit solution, we subsequently analyze how this model performs as the number of bonds increases. In §4.2, we show that the rate of convergence of the price process associated with the linearized Ohm controller increases as the number of bonds increases, although the marginal improvement from each new bond decays. This suggests that the protocol can achieve its stated goals of high liquidity and reduced variance with a small number of correctly parametrized bonds. The correct bond parameters can be discerned by solving the correct HJB equation (which is defined in §3.1).

Note that the mean-variance model doesn't explicitly optimize for liquidity efficiency. For instance, a desire of an Ohm-like protocol might be to optimize the amount of Ω spent on a particular pool relative to the realized liquidity within that pool. Moreover, the protocol may want to optimize for duration weighted liquidity within a pool. In §3.2, we describe a model for including such constraints. This can be viewed as an LQR with non-linear constraints whose associated HJB equation can be solved numerically.

Finally in §5, we extend this framework for Ohm to a more general online learning style problem and argue that DeFi protocols can solve allocation problems for both risk minimization and revenue optimization. This extension to other types of DeFi allocation problems suggests that future DeFi protocols can utilize stochastic optimal control and reinforcement learning to achieve high-level objectives such as risk minimization and incentive optimization. While impractical to solve on-chain, these calculations can be performed periodically off-chain and provided to the on-chain system with a proof of correctness.

2 Formalizing the Olympus Protocol

The Olympus protocol aims to create a lower volatility asset by dynamically adjusting the circulating supply of Ω and providing incentives to users to purchase Ω. The former is controlled by a staking mechanism which incentivizes users to lock their Ω in a smart contract in exchange for future Ω rewards. By staking their Ω in the contract, users reduce the circulating supply. On the other hand, in order to ensure the protocol has ample liquidity for users to trade Ω for numéraire assets such as USD and ETH, the protocol incentivizes users to lock liquidity in automated market makers. This process is known as bonding. The protocol has a treasury consisting of Ω, ETH, and other assets and it adjusts the composition of its treasury by creating bonds. The treasury is used to pay incentives and act indirectly as an insurance fund. Users create a bond by depositing either a non-Ω crypto asset or a share in an automated market maker into the Olympus contract. Upon creating a bond, users also pay a small fee to the protocol, which grows the protocol treasury. The interplay between incentives provided to users for staking and bonding impacts the drift and volatility of the price process for Ω (priced in numéraire terms). We will make the following assumptions throughout the subsequent sections:

- The protocol can adjust incentives according to a discrete clock $n \in \mathbf{N}$, which can be block height. For our results, we assume that these incentives are held piecewise constant in continuous time $t \in \mathbb{R}_+$, which is necessary for the solution to the mean-variance optimization problem we set up to be well-defined. That is, each iterate of the discrete clock n corresponds to Nt units of continuous time.
- The Ω price process is continuous and its drift and volatility are impacted by the protocol's incentive control variables.
- The Ω price process is denominated relative to a fixed numéraire, *e.g.* Ω/USD.

- We assume no-arbitrage so that Ω yields that are computed via prices are equivalent to yields computed via quantities (*e.g.* equations (2) and (6) are equivalent).

2.1 Staking

The Olympus protocol manages the issuance of the Ω token by adjusting an inflation schedule $i : \mathbf{R}_+ \to \mathbf{R}$. When $i(t) > 0$, the protocol mints new tokens and distributes them to users for providing their assets to be used for staking and bonding. On the other hand, when $i(t) < 0$, the protocol burns tokens and reduces the money supply. We view $i(t)$ as a control variable adjusted by the protocol in our formal model.

In practice, protocol burns are executed by either burning Ω held by the protocol's treasury or by performing an Ω buyback. The protocol executed buybacks using a so-called "inverse bond" mechanism, where it openly buys back Ω from holders at a discount to the spot price [19].

The protocol generates fees from users who create bonds and from fees it earns by providing liquidity in CFMMs. Collectively, these fees plus funds borrowed from users who created bonds make up the treasury of the protocol. Theoretically, were the protocol's tokenholders to decide it should shutdown, the protocol could distribute its treasury pro-rata to Ω token holders. One of the main goals of the protocol is to utilize its staking and bonding mechanisms to ensure that one can always redeem 1 Ω for 1 unit of numéraire. As such, a user is also incentivized to stake their Ω in order to preserve their pro-rata implied ownership of the protocol's treasury.

Users who hold Ω who wish to earn more Ω can stake Ω into the protocol. A fraction $\pi_s(t) \in [0, 1]$ of the inflation $i(t)$ is given to users who stake their assets in the protocol. Unlike bonds, users with staked assets can remove them at any time. Prior work has studied how rational users allocate their token wealth to staked and bonded assets albeit for layer 1 blockchains [17, 18]. The protocol's high level goal with how it chooses $\pi_s(t)$ is to keep the percentage of the total money supply that is staked near a target level. In Figure 1, one can see that the Olympus mechanism as deployed on Ethereum was able to track a target percentage with high accuracy. In §3.2, we will first analyze an analytically tractable model that does not include an explicit control for keeping the staked percentage of the monetary supply near a target. However, an analytically intractable model that can be simulated numerically is described in §3.2 that includes an explicit stake tracking term.

2.2 Bonding

Liquidity Provision. A second place that the protocol spends its inflation is on ensuring that there is ample liquidity for new users to purchase Ω. The most popular on-chain trading venues are constant function market makers (CFMMs) [21, 22, 23]. Briefly, these systems allow users, known as liquidity providers (LPs), to deposit two or more assets into an asset pool. Traders who wish to execute a trade against the pool of assets provide an input and output asset and the liquidity provided by LPs is used to execute the trade. The price that the trade is executed at is computed by ensuring that an invariant, known as the trading

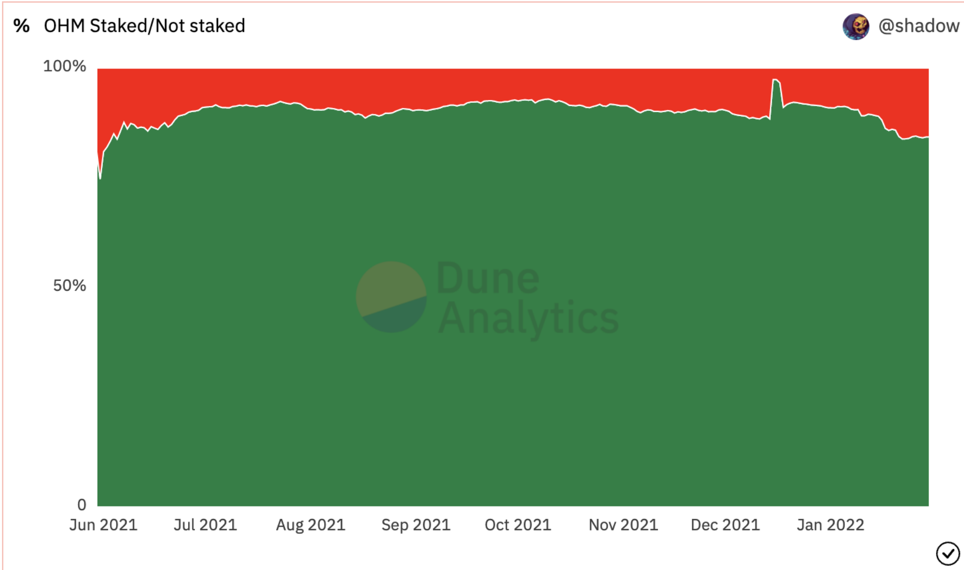

Figure 1: Percentage of Ω staked as a function of time (taken with permission from Dune Analytics [20])

function, is kept constant before and after the trade. The most common trading function used in decentralized exchanges is the constant product function which ensures that the product of the quantities of liquidity is kept constant before an after a trade.

When an LP deposits their assets into a CFMM pool, they receive a tokenized asset known as an *LP share*. The LP share serves as a receipt that the LP can use to redeem their pro-rata assets and fees earned. In order for LPs to receive Ω rewards from the Olympus protocol, they do the following steps:

- Create LP shares for CFMMs that have been whitelisted by protocol governance to receive bond rewards (*e.g.* OHM/ETH or OHM/USDC)
- Use Ohm's bonding contract to lock the LP shares inside the Olympus protocol, which returns the user with a bond share
- The bond share represents a claim on both the locked LP shares and the interest earned by the LP who locks their assets within the contract
- The LP can redeem their bond share for their linearly vested interest payments (which complete vesting over the duration of the bond)[1]

The Olympus contracts ensure that as the duration of a bond increases, so does the interest rate [24, 13, 20]. We note that all of our formalism in this paper is independent of the precise bond withdrawal mechanism. To compute non-asymptotic, quantitative results for particular

[1]We note that some forks of Ω allow for the user to withdraw their initial LP collateral, but Olympus itself does not do this

bond mechanisms (*e.g.* allowing the user to withdraw some of their LP share collateral in the future under some conditions), one needs to encode this in the loss function as we do in the incentive constrained mean-variance model of §3.2. In particular, the rates of convergence we derive for the stochastic linear quadratic regulator as a function of the number of bonds require the drift and variance terms in the dynamics in eq. (22) to be suitably bounded. We leave this for future work.

Bond control mechanism. In order to determine the amount of interest to pay to each bonder when they redeem their bond shares, the protocol uses a control mechanism. Mechanistically, the Olympus bond mechanism is similar to a linear actuator that aims to achieve a target that is set over a duration of time by voting. To describe the bond control mechanism, we assume that all quantities are indexed by bond durations, $j \in 1, \ldots, n_d$, and we drop dependence on j for readability. That is, the calculations below are for a **single** bond duration.

The interest rate paid out to each bonder varies as a function of the *debt ratio* for that bond duration. The debt ratio, $d_r(n) \in \mathbb{R}$, is defined as the ratio of the total amount of Ω that is needed to pay existing debt holders, $D(n) \in \mathbb{R}$, to the total supply of Ω at block height n, $S(n) \in \mathbb{R}$. If the inflation rate is high and going to staking, then the debt ratio will shrink if no new bonders join the market. On the other hand, if staking inflation is low or zero, then the debt ratio will generally increase as new bonders join.

Instead of directly computing an interest coupon (*e.g.* amount of Ω to pay out a bondholder upon loan termination), Olympus uses a *pricing discount* model to imply the coupon value. If the implied Ohm price from the LP share at the time of bond j's share creation is $p_i(t)$, then the discounted price $p_d(t)$ used to compute the Ω coupon payment is:[2]

$$p_d(t) = \min\left(1 + c(n)d_r(n), p_i(t)\right) \tag{1}$$

where $c(n) \in \mathbb{R}$ is a dynamic variable known as the *control variable* that is adjusted by Olympus governance. Note that here, we have let $n = \lfloor t \rfloor$, by letting the variables depending discrete clock be piecewise constant while the continuous variables run. We note that by definition $p_d(t) \leq p_i(t)$ for all t and that the units of p_i and p_d are $\frac{\Omega}{\text{USD}}$. If the value of the collateral initially deposited by the bond is X_0 USD, then the bond coupon payout $C(t)$ is computed as,

$$C(t) = p_d(t) \times X_0$$

where C is in units of Ω. Since $p_d(t) \leq p_i(t)$, this implies a net bond yield $\gamma_b(t)$ of

$$\gamma_b(t) = \frac{\frac{1}{p_d(t)} - \frac{1}{p_i(t)}}{\frac{1}{p_i(t)}} = \frac{p_i(t)}{p_d(t)} - 1 \tag{2}$$

[2] The `bondPrice` function in the Ohm contract code [24] computes p_d as $p_d = \max\left(1 + cd_r, p_i\right)$ because the units of p_d are $\frac{\text{USD}}{\Omega}$. In this paper, we always use units $\frac{\Omega}{\text{USD}}$.

as $\frac{1}{p_i(t)}$ represents the per unit of USD value of Ω without a discount. If a bond is of duration $d \geq 0$, then at time $t \in [0, d]$, a bondholder's shares will have value $X(t)$ equal to

$$X(t) = X_0 \left(\frac{\gamma_b(t)t}{d} \right) \tag{3}$$

Once a bond is created, the total outstanding debt is updated as $D(n) = D(n-1) + X_0(1 + \gamma_b(t))$ and the debt ratio is updated to

$$d_r(n) = \frac{D(n)}{S(n)} = \frac{D(n-1) + X_0(\gamma_b(t))}{S(n)} \tag{4}$$

Upon expiry at time τ_b, the debt in the bond is decremented by:

$$D(n + \tau_b) = D(n-1) - X_0\gamma_b(t)$$

For Ω forks that allow the user to withdraw their LP collateral in the future, the debt is updated as

$$D(n + \tau_b) = D(n-1) - X_0(1 + \gamma_b(t))$$

Finally, the control variable $c(t)$ is updated over a duration τ according to the formula:

$$c(n) = \begin{cases} \min(c(n-1) + r, c') & \text{if } c(n-1) < c' \\ \max(c(n-1) - r, c') & \text{otherwise} \end{cases} \tag{5}$$

when $t \leq \tau$, where c', the reference control to track, is set using a multisig voting mechanism in which many users vote on what the appropriate discount rate should be. Then, there is a nonlinear update on the control value that tries to adjust $c(n)$ until it has converged to c' over τ duration of time, at which point a new c' is voted on by users. This approach is akin to solving a sequence of optimization problems, each of duration τ, where the optimization is done in a decentralized fashion by voting (philosophically, this is similar to model predictive control [25]).

Equations (3), (4), and (5) represent the state and control variables for an adjustment mechanism that controls the discount price. As we will show in the sequel, a simple linear feedback controller given by the solution to a tractable Hamilton-Jacobi-Bellman equation can find optimal solutions to a related control problem, when the goal is set to be price tracking and volatility minimization. We propose this problem as a proxy for what the Ohm system is approximating.

3 Portfolio Optimization

Our goal is to take describe the allocation problem for the staking and bonding mechanisms described in §2. At a high-level, the protocol's goal is to maximize the liquidity in the protocol while minimizing both spend per unit of liquidity and asset volatility. The protocol has to distribute its inflation at time t, $i(t)$, to stakers and holders of different bonds, to achieve these goals. To model this, we first assume that Ω has a price process that can be described as a controlled stochastic process. This modeling assumption holds if the majority of liquidity for Ω is in liquidity pools whose LP shares are staking in the Ohm bonding mechanism. That is, the Ω price can be modeled as a one-dimensional stochastic process (uncoupled from prices of other assets) driven by the inflation and bond controls, which we define below. In practice, this assumption has held up as $> 65\%$ of the Ω supply has been either staked or bonded over the last 8 months (see Figure 1).

3.1 Preliminaries

Notation. To formalize notation, we will assume that there are $n_d \geq 3$ bond durations, represented by a vector $d \in \mathbf{R}_+^{n_d}$. We will assume that there exists a finite maximum duration $d_{\max} \in \mathbf{R}$ and define an admissible duration vector d to be one such that $d_1 = 0, d_{n_d} = d_{\max}$. The zero duration bonds represent 'rented' liquidity (*e.g.* yield farming) where there is no lock-up and or vesting for earned interest. The bonds with duration d_i, $i \in \{2, \ldots, n_d - 1\}$ represent bonds whose lock-up/vesting period is d_i and represent *leased liquidity*. Associated to each duration is a control variable $c_i(t)$, a debt ratio $d_r^i(t)$, and a maximum debt ratio $d_r^{i,\max}$. Bonds of duration i cannot be issued if an update of the form (4) leads to $d_r^i(t) > d_r^{i,\max}$.

Moreover, one of the main improvements that the Olympus protocol made was the inclusion of *protocol-owned liquidity*. This corresponds to liquidity that the protocol smart contract supplies itself. As the protocol collects non-Ω denominated fees (*e.g.* for origination of bonds), it builds up a treasury consisting of non-Ω assets such as ETH or stablecoins. The protocol smart contract can use these non-Ω assets in combination with Ω-denominated assets to provide liquidity itself without needing to pay an incentive. One can view this as a very long duration bond that provides no discount (*e.g.* $c \to \infty$ in (1)), which is how we will model this. In particular, we will reserve d_{n_d} for protocol owned liquidity, which implies that $\gamma_{n_d} = 0$ and $c_{n_d} = \infty$. We will let $POL(t)$ be the random variable representing the maximum protocol-owned liquidity that can be minted. We define the vector $\Gamma = (\gamma_s, \gamma_1, \ldots, \gamma_{n_d}) \in \mathbf{R}^{n_d+1}$ to be the vector of yields for staking, γ_s, and for all of the bonds. We say that Γ_i represents the ith yield source (*e.g.* staking is the 0th yield source in this notation).

Finally, we define $l(t) \in \mathbf{R}^{n_d}$ to be the amount of liquidity bonded to each each duration. For $i \in [n_d]$, $l_i(t)$ is the amount of liquidity provided via bonds with duration d_i. The total liquidity in the system, $L(t)$, is defined as $L(t) = l(t) \cdot \mathbf{1} = \sum_{i=1}^{n_d} l_i(t)$.

Control variable: Incentive Portfolio. Given inflation $i(t) \geq 0$ at time t, the protocol's goal is to construct a probability distribution $\pi(t) \in \Delta^{n_d+1} = \{(p_0, \ldots, p_{n_d}) \in \mathbf{R}_+^{n_d+1} : \sum_{i=0} p_i = 1\}$ such that $i(t)\pi(t)_i$ represents the allocation to the ith yield source. The probability distribution $\pi(t)$ can be thought of as a portfolio that the protocol allocates to the different yield sources based on an optimization routine that will be described in §**??**. As such, we refer to it as the *incentive portfolio* of the Olympus protocol. It is the only control variable that we have in our system, as we can only divert inflation to staking or liquidity provision. Finally, note that we have an implicit constraint on protocol-owned liquidity for all times t:

$$i(t)\pi(t)_{n_d} \leq POL(t)$$

This constraint says that we cannot allocate more liquidity to the protocol-owned component than the protocol has locked within its smart contract.

Linearizing the Ohm Controller. We now provide a conversion from the variables introduced in the previous section to control variables that will be used in the Hamilton-Jacobi-Bellman equation introduced below. This conversion involves going switching the controller from modifying the price discount to one that directly modifies yields. Recall that $\gamma_i(t) \in \mathbb{R}$ governs the net bond yield on the ith bond. As in the previous section, we introduce variables $\pi_i(t), i(t) \in \mathbb{R}$ for $i = 0, \ldots, n_d$, where $\sum_{i=1}^{n_d} \pi_i(t) = 1$ for all $t \in \mathbb{R}$. We define a change of variable formula:

$$\gamma_j(t) = \frac{i(t)\pi_j(t)}{\int_0^t i(\tau)\pi_j(\tau)d\tau} \tag{6}$$

where the term in the denominator $\int_0^t i(\tau)\pi_j(\tau)d\tau = S_j(t)$, the cumulative supply of Ω in bond j. We correspondingly define the formula that defines $\pi_j(t), i(t)$ in terms of $\gamma_j(t)$:

$$\pi_j(t) = \frac{\gamma_j(t)S_j(t)}{i(t)}$$

with the constraint that $\sum_{j=1}^{n_d} \pi_j(t) = 1$. Summing over j, this allows us to set:

$$\sum_{j=1}^{n_d} \gamma_j(t)S_j(t) = i(t)$$

and thus this uniquely defines $\pi_j(t), i(t)$ for each $\gamma_j(t)$. Formula (6) defines the bond yield for the jth bond at time t as the quantity of Ω received at time t relative to the cumulative value of Ω in bond j up to time t. Going forward, we use the variables $(\pi_0(t), \ldots, \pi_{n_d}(t), i(t))$ as control variables for the Ohm system.

Controlled Stochastic Process. As the main goal of the Olympus protocol is to construct a low price volatility asset, we need to specify a price process that is controlled by our incentive variables. In stochastic optimal control, this process is usually a controlled Îto diffusion of the form (recall that $i(t)$ is the block reward or inflation at time t) [26] :

$$dP_t = \mu(P_t, \pi(t), i(t))dt + \sigma(P_t, \pi(t), i(t))dW_t \tag{7}$$

where P_t is numéraire valued price of Ω, $\mu(P_t, \pi(t), i(t))$ is the controlled drift, $\sigma(P_t, \pi(t), i(t))$ is the controlled volatility, and dW_t is the standard Wiener process. Intuitively, the idea is that our choice of $\pi(t)$ and $i(t)$ should be constructed so as to minimize $\sigma(P_t, \pi(t), i(t))$ while also providing some bounds on $\mu(P_t, \pi(t), i(t))$ over time. Since we are assuming that the dominant liquidity sources are on-chain venues (*e.g.* CFMMs), it is clear that when we reduce incentives in $\pi(t)$ and $i(t)$, we will also affect the drift and volatility. For instance, if we move all liquidity to be rented, $\pi(t) = (0, 1, 0, \ldots, 0)$, then we should expect liquidity to be more volatile (*e.g.* opportunity cost of capital to keeping liquidity in Ω if there is a protocol with higher yield) and increase the implied volatility, σ. We will encode constraints and expectations of this form in the desiderata below.

Loss Functional. Within optimal control, the quality of a control mechanism $(\pi(t), i(t))$ is measured by a loss functional,

$$J_T(P_0, \pi(t), i(t)) = \int_0^T \ell(P_t, \pi(t), i(t))dt$$

for a loss function $\ell : \mathbf{R}_+ \times \mathbf{Range}(\pi) \times \mathbf{Range}(i) \to \mathbf{R}$ where $\ell \geq 0$. Since we have a stochastic process (7), we can evaluate the expected loss functional,

$$C_T(x, \pi(t), i(t)) = \mathop{\mathbf{E}}_{P_t}[J_T(P_t, \pi(t), i(t))|P_0 = x] \tag{8}$$

Our goal is to construct $\pi^*(x,t), i^*(x,t$ such that $(\pi^*(x,t), i^*(x,t)) = \mathrm{argmin}_\alpha C_T(x, \alpha)$.

Stochastic Optimal Control. Stochastic optimal control studies the behavior of the expected loss function (8). The main object of study in this field is the value function $V(x,t)$, which is defined as

$$V(x,t) = \inf_{\pi(t), i(t)} C_T(x, \pi(t), i(t))$$

Given a value function, one can construct the associated stochastic Hamilton-Jacobi-Bellman equation,

$$\frac{\partial V(x,t)}{\partial t} + \inf_{\pi(t), i(t)} \left[\ell(P_t, \xi(t), l(t), i(t)) + \mu \cdot \nabla_x V(x,t) + \frac{\sigma^2}{2} \Delta V(x,t) \right] = 0 \tag{9}$$

where $\mu = \mu(P_t, \xi(t), l(t), i(t))$, $\sigma = \sigma(P_t, \xi(t), l(t), i(t))$, ∇_x is the gradient with respect to the x coordinate and Δ is the Laplacian, $\Delta = \sum_i \frac{\partial^2}{\partial x_i^2}$. In the simplified linear quadratic problem we derive for the Ω system, this HJB equation is a Riccati equation that can be explicitly analyzed.

3.2 Mean-Variance Models and LQR

In this section, we specialize the preliminaries of the prior section to computationally tractable models. To utilize these results, we have to specify a controlled price process (7) and a loss function ℓ. For all models in this section, we will use a common price process, the linearly controlled price process. The two models we propose will differ in their choice and construction of ℓ. Throughout the following, we fix $(\Omega, \mathcal{F}, P, \{\mathcal{F}_t\}_{t\geq 0})$, a filtered probability space, with a Brownian motion W_t.

Linear Controlled Price Process. We propose an optimal control problem that the Ohm protocol can approximate whose solution attempts asymptotic price and percentage staked tracking and terminal variance minimization over an infinite time horizon. We first assume that the price process is a linear controlled diffusion. Consider the following linear controlled stochastic process:

$$dP(t) = [AP(t) + B[\pi(t), i(t)]^\top]dt + [CP_t + D[\pi(t), i(t)]^\top]dW_t \tag{10}$$

where $P(t) \in \mathbb{R}$, $A, C \in \mathbb{R}$, $B^\top, D^\top \in \mathbb{R}^{n_d+2}$, $\pi(t) \in \Delta^{n_d+1} \subset \mathbb{R}^{n_d+1}$, $i(t) \in \mathbb{R}$. Going forward, we denote $[\pi(t), i(t)]^\top = u(t)$.

We also have the following dynamics for the percentage staked, which is $\xi(t) = S(t) - D(t)$. We can write this using the identities $S(t+1) = S(t) + i(t)$ and $D(t) = d_r(t)S(t) = D(t-1) + X_0(1 + \gamma_b(t))$, and therefore:

$$\begin{aligned}
\xi(n+1) &= S(n+1) - D(n+1) \\
&= S(n+1) - D(n) + X_0(\lfloor t \rfloor + \gamma_b(\lfloor t \rfloor)) \\
&= S(n) + i(n) - D(n) + X_0(\lfloor t \rfloor + \gamma_b(\lfloor t \rfloor)) \\
&= \xi(n) + i(\lfloor t \rfloor) + X_0(\lfloor t \rfloor + \gamma_b(\lfloor t \rfloor))
\end{aligned}$$

Unconstrained Mean-Variance Tracking Model. We assume that the protocol aims to find functions $\pi(\cdot)$ and $i(\cdot)$ (corresponding to the portfolio and inflation schedule, respectively) to minimize the following quadratic cost functional:

$$J_T(P_0, u(\cdot)) = \mathbf{E}\left[\int_0^T \|P(\tau) - P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \mathrm{Var}(P(T)) \tag{11}$$

with respect to the dynamics (10), where $P^\star \in \mathbb{R}$ is a reference price that the protocol attempts to track and $r \in \mathbb{R}$ is the cost on the control. We note that this functional approximates the two joint objectives of the protocol: price tracking and (terminal) volatility minimization. The expectation in (13) is taken over the randomness in the Brownian motion and in the initial price P_0. Note that this problem as written is not immediately a stochastic linear quadratic regulator, due to the presence of the terminal variance term, $\mathrm{Var}(P(T))$. In section 4.1, we show formally that solutions of the above problem are contained in solutions of a stochastic LQR, which is analytically tractable to solve via an associated Riccati equation.

Incentive Constrained Mean-Variance Tracking Model. There are a number of constraints that we can place on μ, σ and ℓ in the general nonlinear model based on existing market behaviors and properties of CFMMs. Some of these properties stem directly from mathematical properties of CFMMs whereas others represent observed user behavior. Technically, μ and σ also depend on $L(t) \in \mathbf{R}_+$, which is the total liquidity in the pools. We can view an adjustment of $\pi(t)$ as stochastically affecting $L(t')$ for $t' > t$. Moreover, these also depend on the total staked supply $\xi(t) = S(t) - D(t)$. If there is a high $S(t)$, there is less liquidity in the CFMM pools and one expects price impact and volatility when there is lower liquidity [6]. Similarly, we can only indirectly control $\xi(t)$ via an adjustment of $\Gamma_0(t') = \gamma_s(t')$ and $i(t')$ for $t' < t$. For simplicity, we will first assume that we can compute $\xi(t)$ and $L(t)$ as a deterministic function of $P_t, \pi(t)$ and $i(t)$ explicitly. Later, we will relax this constraint and allow them to be stochastic processes themselves. We detail the constraints on $\mu(P_t, \pi(t), i(t), L(t), \xi(t))$ below:

1. $\left|\frac{\partial \mu}{\partial L}\right| > 0$: If liquidity decreases, then the magnitude of the drift has to strictly increase (this is naturally true for CFMMs [27])

2. $\left|\frac{\partial \mu}{\partial \xi}\right| > 0$: If the staked supply increases, there is less liquidity and the process should have higher price impact per unit time

3. $\left|\frac{\partial \mu}{\partial \pi_i}\right| \leq \left|\frac{\partial \mu}{\partial \pi_j}\right|$ if $i < j$: Loss of longer locked up liquidity due to reduced incentives for duration d_j impacts the price drift at least as much as any lower duration $d_i < d_j$

4. $\frac{\partial \mu}{\partial i} \leq 0$: If issuance increases, then the (long-term) drift cannot increase (*e.g.* there is more supply on the market relative to numéraire)

Next, let's consider constraints on $\sigma(P_t, \pi(t), i(t), L(t), \xi(t))$:

1. $\exists L > 0$ such that $\frac{\partial \sigma}{\partial L} \in [-L, 0]$: Volatility is non-increasing as liquidity increases, but has a Lipschitz derivative in liquidity

2. $\exists L > 0$ such that $\left|\frac{\partial \sigma}{\partial \xi}\right| \leq L$: Volatility has a bounded change in the staked supply

3. $\left|\frac{\partial \sigma}{\partial \pi_i}\right| \leq \left|\frac{\partial \sigma}{\partial \pi_j}\right|$ if $i < j$: Loss of longer locked up liquidity due to reduced incentives for duration d_j impacts the volatility at least as much as any lower duration $d_i < d_j$

4. $\frac{\partial \sigma}{\partial i} \leq 0$: If issuance increases, then we expect more volatility as the number of ways that the increased supply can be redistributed goes up

Finally, we will consider what requirements we have for the loss function ℓ. As the loss function will be an important part of the Hamiltonian-Jacobi-Bellman equation of the next section, it is important to design both the constraints on it and its explicit functional form in a manner that is computationally efficient.

1. Maximizes the time-weighted liquidity, $T_t = \frac{d \cdot l(t)}{d \cdot \mathbf{1}} P_t$, where $l(t) \in \mathbf{R}^{n_d}$ is the amount of liquidity provided to each bond
2. Bounds the spend per unit of liquidity, $\frac{i(t)\pi(t)_i}{l(t)_i} < k_i$
3. Ensures that $\ell(t)_i > 0$ for all $i \in [n_d]$
4. Keeps $\xi(t)$ near a target stake percentage, ξ^*
5. Penalizes volatility σ_t

A simple example of a function that meets these constraints is

$$\ell(P(t), \xi(t), l(t), i(t)) = \left\| \frac{d \cdot l(t)}{d \cdot \mathbf{1}} P(t) - P^* \right\|_2^2 + \|\xi(t) - \xi^*\|_2^2 - \sigma_t^2 - \sum_{i=1}^{n_d} \eta_i \left(\frac{i(t)\pi_i(t)}{l_i(t)} - k_i \right) + \lambda_i \ell(t)_i$$

where $\lambda \in \mathbf{R}^{n_d}$ and $\eta \in \mathbf{R}^{n_d}$ are Lagrange multipliers. The non-linear term $-i(t) \sum_{i=1}^{n_d} \frac{\pi_i(t)}{l_i(t)}$ is what prevents this problem from being an LQR. However, we can replace this function by instead enforcing n inequality constraints of the form $\frac{i(t)\pi_i(t)}{l_i(t)} < k_i$. Using the Pontryagin Maximum Principle [28, Ch. 4], there exists a co-state equation for λ and η that can be used to solve this. However, one can only solve the HJB equation directly for this loss function as opposed to using a Riccati equation [29]. This means that this realistic model can only be solved numerically, which we leave for future work.

4 Analysis of the Unconstrained Optimal Control Problem

The unconstrained mean-variance tracking model of the previous section is amenable to formal analysis. In this section we prove two main results:

1. *Unconstrained Mean-Variable Optimization is a stochastic LQR*: There exists a parametrization of the mean-variance portfolio optimization problem that embeds it into a stochastic linear quadratic regulator. This allows us to inherit the well-known stability properties of the LQR, and allows us to write the optimal solutions to our problem of interest in terms of a tractable Riccati equation.
2. *Controller stability increases in the number of bonds*: As the number of bonds represents the dimensionality of the control variable, a natural question is ask is: how many bonds does one need to achieve a certain control performance, as measured by the quadratic cost (13). We provide asymptotic bounds that demonstrate that as the number of bonds is increased, the control mechanism is more effective. However, the marginal increase in performance of each new incremental bond decreases, suggesting that there is a 'sweet spot' of the number of bonds to add to achieve a certain level of mechanism performance.

4.1 Unconstrained Mean-Variance Optimization is a Stochastic Linear Quadratic Regulator

Consider once again the following linear controlled stochastic process:

$$dP(t) = [AP(t) + Bu(t)]dt + [CP(t) + Du(t)]dW_t \tag{12}$$

where $P(t) \in \mathbb{R}$, $A, C \in \mathbb{R}$, $B, D \in \mathbb{R}^{1\times(n+1)}$, $u(t) \in \mathbb{R}^{n_d+2}$. We assume that the initial state $P_0 \in \mathbb{R}$ is chosen according to a distribution $P_0 \sim D$, and below, we take expectations over this distribution and the randomness in the Brownian motion.

Considering the optimal control problem associated with this system:

$$\min_{u(\cdot)} J_T(P_0, u(\cdot)) = \min_{u(\cdot)} \mathbf{E}\left[\int_0^T \underbrace{\|P(\tau) - P^\star\|_2^2}_{\text{tracking term}} + r\|u(\tau)\|_2^2 d\tau\right] + \lambda \underbrace{\mathrm{Var}(P(T))}_{\text{variance term}} \tag{13}$$

subject to the dynamics (12), where $P^\star$ is a reference price we would like to track. Denote the set of optimal solutions to this problem $U(\lambda)$. Recall that $U(\lambda) \subset C([0,T], \mathbb{R}^{n_d+2})$

The goal is to relate this problem to a stochastic linear quadratic regulator, i.e. as a problem with a cost functional of the form:

$$\min_{u(\cdot)} \mathbf{E}\left[\int_0^T P(\tau)^\top Q P(\tau) + u(\tau)^\top R u(\tau) d\tau + P(T)^\top Q_T P(T)\right]$$

for $Q, R \succ 0$.

First, we note $\mathrm{Var}(P(t)) = \mathbf{E}[P(t)^2] - (\mathbf{E}[P(t)])^2$. The second term is the nonstandard term because the expectation cannot be pulled out of the quadratic function. Therefore, following the treatment in [30], we consider the associated functional (we will further show that the solutions to our original problem are contained in the set of solutions of this problem):

$$\tilde{J}(P(0), u(\cdot)) = \mathbf{E}\left[\int_0^T \|P(\tau) - P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \mathbf{E}[\lambda P(T)^2 - \gamma P(T)] \tag{14}$$

where $-\infty < \gamma < \infty$. Denote the set of optimal solutions to this problem $\tilde{U}(\lambda, \gamma)$.

Proposition 1. *If $\bar{u} \in U(\lambda)$, then $\bar{u} \in \tilde{U}(\lambda, \bar{\gamma})$, where $\bar{\gamma} = 2\lambda\,\mathbf{E}[\bar{P}(t)]$, and $\bar{P}(t)$ is the trajectory of the control system when $\bar{u}$ are injected as inputs.*

Proof. Suppose that $\bar{u} \in U(\lambda)$ but $\bar{u} \notin \bar{U}(\lambda, \bar{\gamma})$. Then, there exists u, and a corresponding trajectory $P(t)$ such that:

$$\begin{aligned}
&\mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \mathbf{E}[\lambda P(T)^2 - \gamma P(T)] \\
&- \mathbf{E}\left[\int_0^T \|\bar{P}(\tau)-P^\star\|_2^2 + r\|\bar{u}(\tau)\|_2^2 d\tau\right] + \mathbf{E}[\lambda \bar{P}(T)^2 - \gamma \bar{P}(T)] \\
&= \mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 - \|\bar{P}(\tau)-P^\star\|_2^2 - r\|\bar{u}(\tau)\|_2^2 d\tau\right] \\
&\qquad + \lambda\,\mathbf{E}[P(T)^2 - \bar{P}(T)^2] - \bar{\gamma}\,\mathbf{E}[P(T) - \bar{P}(T)] < 0
\end{aligned}$$

This implies:

$$\begin{aligned}
&\lambda\,\mathbf{E}[P(T)^2 - \bar{P}(T)^2] - \bar{\gamma}\,\mathbf{E}[P(T) - \bar{P}(T)] \\
&< -\mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 - \|\bar{P}(\tau)-P^\star\|_2^2 - r\|\bar{u}(\tau)\|_2^2 d\tau\right]
\end{aligned}$$

Consider the function $f(x, y) = \lambda x - \lambda y^2$ which is concave in both its arguments. Then, note that:

$$f(\mathbf{E}[P(T)^2], \mathbf{E}[P(T)]) = \lambda\,\mathbf{E}[P(T)^2] - \lambda(\mathbf{E}[P(T)])^2 = \lambda \mathrm{Var}(P(T))$$

with $\frac{\partial f}{\partial x} = \lambda$ and $\frac{\partial f}{\partial y} = -2\lambda y$. Now, using the first order conditions of optimality of $f(\cdot, \cdot)$:

$$f(\mathbf{E}[P(T)^2], \mathbf{E}[P(T)]) \le f(\mathbf{E}[\bar{P}(T)^2], \mathbf{E}[P(T)]) + \lambda(\mathbf{E}[P(T)^2 - \bar{P}(T)]) - 2\lambda\,\mathbf{E}[\bar{P}(T)](\mathbf{E}[P(T) - \bar{P}(T)])$$

Adding $\mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right]$ to both sides we have:

$$\begin{aligned}
&\mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + f(\mathbf{E}[P(T)^2], \mathbf{E}[P(T)]) \\
&= \mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \lambda \mathrm{Var}(P(T)) \\
&\le \mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \lambda \mathrm{Var}(\bar{P}(T)) \\
&\quad + \lambda(\mathbf{E}[P(T)^2 + \bar{P}(T)] - 2\lambda\,\mathbf{E}[\bar{P}(T)](\mathbf{E}[P(T) - \bar{P}(T)]) \\
&< \mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] + \lambda \mathrm{Var}(\bar{P}(T)) \\
&\quad - \mathbf{E}\left[\int_0^T \|P(\tau)-P^\star\|_2^2 + r\|u(\tau)\|_2^2 - \|\bar{P}(\tau)-P^\star\|_2^2 - r\|\bar{u}(\tau)\|_2^2 d\tau\right] \\
&= \mathbf{E}\left[\int_0^T \|\bar{P}(\tau)-P^\star\|_2^2 + r\|\bar{u}(\tau)\|_2^2 d\tau\right] + \lambda \mathrm{Var}(\bar{P}(T))
\end{aligned}$$

This is a contradiction to $\bar{u}$ being an optimal solution to $U(\lambda)$. Therefore, we are done and $\bar{u} \in \tilde{U}(\lambda, \bar{\gamma})$. □

We will now show that solving (14) is equivalent to solving the stochastic LQ problem:

$$\min_{u(\cdot)} \mathbf{E}\left[\int_0^T \|\tilde{P}(\tau) - \tilde{P}^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau + \lambda \tilde{P}_T^2\right]$$

where $d\tilde{P}_t = [A\tilde{P}(t) + Bu(t)]dt + [C\tilde{P}(t) + Du(t)]dW_t$, $\tilde{P}_t = P_t - \frac{\gamma}{2\lambda}$, and $\tilde{P}^\star = P^\star - \frac{\gamma}{2\lambda}$.

We see:

$$\begin{aligned}
\mathbf{E}\left[\int_0^T \|\tilde{P}(\tau) - \tilde{P}^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau + \lambda \tilde{P}_T^2\right] &= \mathbf{E}\left[\int_0^T \|P(\tau) - \frac{\gamma}{2\lambda} - P^\star + \frac{\gamma}{2\lambda}\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right. \\
&\quad + \frac{1}{2}\lambda\left(P(T) - \frac{\gamma}{2\lambda}\right)^2 \\
&= \mathbf{E}\left[\int_0^T \|P(\tau) - P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right. \\
&\quad + \lambda(P(T)^2 - P(T)\frac{\gamma}{\lambda} - \frac{\gamma^2}{4\lambda^2}) \\
&= \mathbf{E}\left[\int_0^T \|P(\tau) - P^\star\|_2^2 + r\|u(\tau)\|_2^2 d\tau\right] \\
&\quad + \mathbf{E}[\lambda P(T)^2 - \gamma P(T)]
\end{aligned}$$

Therefore, by shifting the dynamics, we can recover back solutions to our original problem $\tilde{U}(\lambda, \gamma)$.

4.2 Controller stability increases in the number of bonds

One natural question to ask is regarding how well the control mechanism behaves as we increase the number of bonds n_d. Ω adaptively adjusts the number of bonds in an attempt to better control the price process and achieve tracking. We now consider the infinite horizon version of the optimal control problem:

$$dP(t) = [AP(t) + Bu(t)]dt + [CP(t) + Du(t)]dW_t$$

$$J(P(0), u(\cdot)) = \mathbf{E}\left[\int_0^\infty q(P(t) - P^\star)^2 + r\|u(\tau)\|_2^2 d\tau\right] \tag{15}$$

We formally analyze this dimension dependence by using the Riccati equation associated with the linear quadratic control problem for the Ohm protocol to bound the rate of decay of the price process to the reference price when the drift and control matrices are chosen at random. Informally, we prove:

A 1-dimensional controlled price process where the controls are m-dimensional and dynamics are chosen at random, with controls minimizing the costs given in (15), converges to $P^\star$ at least at the rate $c\sqrt{m}$, for some $c > 0$.

Interpreting this in the context of Ω, we see that as the number of bonds increases ($m = n_d + 2$), the linear controller associated with the protocol's price process is able to control the system and achieve tracking at a faster rate, but experiences diminishing marginal utility as m becomes large.

We first prove the above result for scalar systems with zero Brownian motion (deterministic systems), of the form:

$$\dot{P}(t) = AP(t) + Bu(t)$$

with costs:

$$J(P_0, u(\cdot)) = \int_0^\infty qP(\tau)^2 + r\|u(\tau)\|_2^2 d\tau \tag{16}$$

($P(t), A \in \mathbb{R},\ B \in \mathbb{R}^{1\times m}, u(t) \in \mathbb{R}^{m\times 1},\ q, r \in \mathbb{R}$), to provide intuition for how the rate of convergence changes as the dimension of the control increases. We prove a more general stochastic version of the following theorem in Appendix A.

Theorem 1. *If $A \sim \mathcal{N}(0,1)$ and $B \sim \mathcal{N}(0, I_m)$, there exists a feedback controller $u(t) = KP(t)$ minimizing (16) such that $\dot{P}(t) = AP(t) + Bu(t)$ satisfies*

$$\mathbf{E}[P^2(t)] \leq \mathbf{E}[e^{-c\sqrt{m}t}P^2(0)] \tag{17}$$

for some $c > 0$.

Proof. We recall from linear system theory [31] that if there exists $\rho \in \mathbb{R}$ such that for any $q > 0$, ρ satisfies:

$$A^\top \rho + \rho A - \rho B r^{-1} B^\top \rho = -q \tag{18}$$

then the static feedback controller $K = -r^{-1}B^\top \rho$ minimizes (16). Given that A is also a scalar, the Riccati equation (18) is:

$$2\rho A - \rho^2 B r^{-1} B^\top = 2\rho A - \rho^2 r^{-1}\|B\|^2 = -q$$

Therefore, letting $r^{-1}\|B\|^2 = \tilde{B}$, we have the following quadratic equation that ρ must satisfy:

$$\tilde{B}\rho^2 - 2\rho A - q = 0$$

and solving for ρ, we have:

$$\rho = \frac{2A \pm \sqrt{(2A)^2 + 4\tilde{B}q}}{2\tilde{B}} = \frac{A \pm \sqrt{A^2 + \tilde{B}q}}{\tilde{B}} \tag{19}$$

We now use

$$V(P) = \rho P^2$$

as a Lyapunov function to bound the state [31]. Taking the time derivative of V along trajectories of $P(t)$:

$$\begin{aligned}
\frac{d}{dt}V(P) &= (AP + Bu)^\top \rho P + P^\top \rho (AP + Bu) \\
&= ((A + BK)x)^\top \rho P + P^\top \rho (A + BK)P \\
&= P^\top (A + BK)^\top \rho P + P^\top \rho (A + BK)P \\
&= P^\top ((A + BK)^\top \rho + \rho(A + BK))P \\
&= P^\top (A^\top \rho + K^\top B^\top \rho + \rho A + \rho BK)P \\
&= P^\top (A^\top \rho + \rho A + (-r^{-1}B^\top \rho)^\top B^\top \rho + \rho B(-r^{-1}B^\top \rho))P \\
&= P^\top (A^\top \rho + -\rho^\top B(r^{-1})^\top B^\top \rho - \rho B r^{-1} B^\top \rho)P \\
&= P^\top (A^\top \rho + \rho A - r^{-1}\rho B B^\top \rho - r^{-1}\rho B B^\top \rho)P \\
&= P^\top (\underbrace{2\rho A - r^{-1}\rho^2 \|B\|^2}_{=-q} - r^{-1}\rho^2 \|B\|^2)P \\
&= P^\top (-q - r^{-1}\rho^2 \|B\|^2)P
\end{aligned}$$

Now, because $V(P) = \rho P^2$, and ρ is a scalar, we can bound:

$$\dot{V}(P) \leq -\frac{q}{\rho}V(P) - \frac{\rho^2}{r}P^\top \|B\|^2 P$$

where $V(P) = \rho P^2$. And so:

$$\rho \frac{d}{dt}P^2 \leq -qP^2 - \frac{(\rho)^2 \|B\|^2}{r}P^2 = -\left(q + \frac{\rho^2 \|B\|^2}{r}\right)P^2$$

Integrating both sides, we have:

$$P^2(t) \leq e^{-Rt}P^2(0)$$

where we define the rate R as

$$R = \frac{q}{\rho} + \frac{\rho \|B\|^2}{r}$$

Note that the rate R is a random variable since B is a Gaussian random variable and p is a function of Gaussian random variables. Therefore to bound $\mathbf{E}[R]$, we need to compute bounds on $\mathbf{E}[\rho]$. Using (19), we will first compute the conditional expectation of ρ so that we can write everything in terms of $\|B\|_2$

$$
\begin{aligned}
\mathbf{E}[\rho|\tilde{B}] &= \mathbf{E}\left[\frac{a \pm \sqrt{a^2+\tilde{B}q}}{\tilde{B}}\Big|\tilde{B}\right] \\
&= \frac{1}{\tilde{B}}\left(\mathbf{E}[a|\tilde{B}] + \mathbf{E}[\sqrt{a^2+\tilde{B}q}|\tilde{B}]\right) \\
&= \frac{1}{\tilde{B}}\mathbf{E}[\sqrt{a^2+\tilde{B}q}|\tilde{B}] && (a \text{ independent of } \tilde{B}, a \sim \mathcal{N}(0,1)) \\
&\leq \frac{1}{\tilde{B}}\sqrt{\mathbf{E}[a^2+\tilde{B}q|\tilde{B}]} && \text{(Jensen's inequality)} \\
&= \frac{\sqrt{1+\tilde{B}q}}{\tilde{B}} && (20)
\end{aligned}
$$

Using equation (20) and $\tilde{B} = \mathcal{O}(\|B\|^2)$ this means that

$$\mathbf{E}[\rho|\tilde{B}] = \frac{\mathcal{O}(\sqrt{\|B\|^2})}{\mathcal{O}(\|B\|^2)} = \mathcal{O}(\|B\|^{-1})$$

so that by iterated expectation $\mathbf{E}[\rho] = \mathbf{E}[\mathbf{E}[\rho|\tilde{B}]] = \mathcal{O}(m^{-1/2})$ since $\mathbf{E}[\|B\|^2] = m = n_d + 2$. This allows us to reason about the expected rate:

$$
\begin{aligned}
\mathbf{E}[R] &= q\,\mathbf{E}\left[\frac{1}{\rho}\right] + \frac{1}{r}\mathbf{E}[\rho\|B\|^2] \\
&\geq \frac{q}{\mathbf{E}[\rho]} + \frac{1}{r}\mathbf{E}[\rho\|B\|^2] && \text{(Jensen's inequality applied to } \tfrac{1}{x}) \\
&\geq c\sqrt{m}
\end{aligned}
$$

Finally we can use this bound to get

$$\mathbf{E}[P^2(t)] \leq \mathbf{E}[e^{-Rt}P^2(0)] \leq \mathbf{E}[e^{-c\sqrt{m}t}P^2(0)]$$

This demonstrates that as $m = n_d + 2$ increases, the controlled process converges faster to the reference price. $\square$

5 Beyond Olympus: Generalizations

Multidimensional Extensions. One natural extension is to try to control a multidimensional price process. Suppose that we instead had $\mathsf{P}(t) \in \mathbf{R}^k$ and had a controlled stochastic process of the form

$$d\mathsf{P}(t) = \boldsymbol{\mu}(\mathsf{P}(t), \boldsymbol{\pi}(t), \mathsf{i}(t))dt + \boldsymbol{\sigma}(\mathsf{P}(t), \boldsymbol{\pi}(t), \mathsf{i}(t))d\mathsf{W}_t$$

where $\boldsymbol{\mu}, \mathsf{i} \in \mathbf{R}^k$, $\boldsymbol{\sigma} \in \mathbf{R}^{k\times k}$, and $\boldsymbol{\pi} \in \mathbf{R}^{k\times m}$ where m is an upper bound on the number of bond durations. Protocols such as Tokemak [32, 33] claim to generalize the original Ohm model by solving this multidimensional problem. The goal of such protocols is to manage liquidity for multiple protocols simultaneously, generalizing Ohm to liquidity management for many protocols. This is purported to be more efficient than Ohm because liquidity can be optimized such that trading routes (*e.g.* trading from asset A to B to C) can be optimized [33, 34]. However, there exist no formal or numerical proofs of such claims.

It is well known that the multidimensional version of this problem has a 'curse of dimensionality' in that it takes exponentially longer in k to solve the Hamilton-Jacobi-Bellman equation associated with the mean-variance tracking problem [35]. Such models are often only tractable using reinforcement learning and provide significantly worse guarantees than results like Theorem 2. Moreover, such models are less robust as dimension increases. One way to see this is that if an adversary can control some small subset of the price processes (*e.g.* $o(k)$), the correlation structure between the processes can allow such an adversary to take over the controller (*e.g.* $\pi_i(t)$ for all $i \in [k]$).

Adversarial Control Mechanisms. Another mechanism for liquidity management that has managed billions of dollars of crypto assets is the so-called ve model [36, 37]. The ve model was invented by the market making protocol Curve [38, 6] which pays liquidity providers in the protocol with the native governance token CRV. When the protocol first launches the CRV token, users were 'yield farming' it and immediately selling it on the market. This leads to a negative feedback loop, as CRV going down in value leads to less liquidity being placed in Curve CFMM pools due to lower incentives. To stem immediate selling, the Curve protocol added a feature very similar to Ohm bonds: if a user locks up their earned CRV token for a time period, they would receive a 'boost' or extra CRV. For instance, if a user agreed to lock their CRV in the protocol for a year, they could receive $2.5\times$ the rewards they would had they not committed to be locked up for a year. The eponym of the name ve model comes from the fact that locked up CRV is represented by a token named veCRV.

One difference between Ohm bonds and veCRV is that veCRV holders were allowed to vote within Curve's governance smart contracts on changes to Curve incentives. This meant that veCRV holders, who had more governance power due to their boosts, could vote on adjusting boosts to particular pools. For instance, veCRV holders in one market making pool could create a governance proposal to give all of the CRV rewards (analogous to $i(t)$ in the Ohm scenario) to their pool and none to other pools. This dynamic of having increased governance influence on changing parameters such as the inflation rate $i(t)$ or the proportion of CRV rewards given to a certain pool (which is analogous to $\pi(t)$) created what became known as 'bribing' games. A third-party protocol would incentivize users to deposit assets into a smart contract. It then would reallocate those assets into different Curve market making pools and automatically generate proposals to maximize the return to the pools that it deposited in. The protocol would then take a fee from the excess earnings that it earned by manipulating the CRV governance mechanism This reallocation mechanism was first pioneered by Convex Finance [39] and led to an enormous increase in the total liquidity held by Curve.

We can model the dynamics of the CRV bribing game by augmenting (10). Suppose, for simplicity, that we have a two-player game where each player tries to provide a control $u^1(t)$ and $u^2(t)$ that maximizes their expected return, $J^1(u^1(t), u^2(t)), J^2(u^1(t), u^2(t))$. This is equivalent to having the two players play a zero-sum game to control the parameters A, B, C, D in (10). Such models have been studied for two-player, zero-sum games and instead of Hamilton-Jacobi-Bellman equations, one instead finds Hamilton-Jacobi-Isaacs equations [40]. While beyond the scope of this paper, Hamilton-Jacobi-Isaacs equations can be thought of an equation that replaces the supremum of (9) with a minimax formulation that minimizes over one players control parameters u_t^1 and maximizes over the others. We also note that the lack of unique equilibria that we find here is similar to those found in adversarial neural networks such as GANs [41].

We will instead provide a simplified description of what the ve model looks like for the linear-quadratic regulator of §3.2. Consider the following price process, which now has the inputs of both players $u^1(t)$ and $u^2(t)$:

$$dP(t) = \mu(P(t), u^1(t), u^2(t))dt + \sigma(P(t), u(t)^1, u(t)^2)dW_t \tag{21}$$

for $P_t^1 \in \mathbb{R}$. That is, there are two competing controllers that try to modify a single price process.

Thus, we set up two costs:

$$J^1(P(0), u^1(\cdot), u^2(\cdot)) = \mathbf{E}\left[\int_0^T \|P(t) - P^\star\|_2^2 + [u(\tau)^1, u(\tau)^2]^\top R^1 [u(\tau)^1, u(\tau)^2] d\tau\right]$$
$$J^2(P(0), u^1(\cdot), u^2(\cdot)) = \mathbf{E}\left[\int_0^T \|P(t) - P^\star\|_2^2 + [u(\tau)^1, u(\tau)^2]^\top R^2 [u(\tau)^1, u(\tau)^2] d\tau\right]$$

Therefore, these costs along with dynamics (21) give rise to a dynamic noncooperative game [42]. A solution concept that one can propose is finding $u^{1,\star}, u^{2,\star}$:

$$J^1(P(0), u^{1,\star}, u^{2,\star}) \leq J^1(P(0), u^1, u^{2,\star})$$
$$J^2(P(0), u^{1,\star}, u^{2,\star}) \leq J^1(P(0), u^{1,\star}, u^2)$$

for all $u^1 \neq u^{1,\star}$ and $u^2 \neq u^{2,\star}$. We note that generally, finding such solutions requires solving Hamilton-Jacobi-Isaacs equations (or in the linear-quadratic case, coupled Riccati equations) that may be difficult to tractably analyze.

6 Conclusion

As DeFi comes to manage larger pools of assets, it is important to understand how incentives in liquidity management work at scale. Our work is the first to formally define the controller used in the Ω liquidity management protocol. Using this definition, we were able to show that the non-linear controller that is used in production can actually be viewed as a linear-quadratic regulator using a change of coordinates. We then provide an analysis of the Ω liquidity management protocol by casting it as a control problem; we provide dynamics and costs that approximate the behavior of the protocol, and in particular, we analyze a simple stochastic linear quadratic model that resembles a tractable mean-variance portfolio optimization problem (price tracking combined with volatility minimization) that Ω can solve. We then analyze solutions to this problem by writing the associated Riccati equation, and show formally that the efficacy of the liquidity management protocol scales as $\mathcal{O}(\sqrt{m})$, where m is the number of bond durations that are allowed to be optimized over. This in turn means that adaptively adding durations can help the protocol achieve its tracking goals, but that this process has diminishing marginal utility.

The usage of feedback control in DeFi was first espoused in [43]; however, the appearance of optimal control and model predictive control in DeFi appears to have originated in the Ω protocol. Our formulation can help simplify existing protocols like Ω and help with the design of improvement to liquidity management protocols. Future work involves numerical simulations of the non-analytic loss functions of §3.2 and comparing simulation performance to real data. As mentioned, improving parameter selection in liquidity management protocols can lead to reduced risk and improved incentive spend. We believe that applications such as those in §5 will be enabled by further development of applications of stochastic control techniques, including reinforcement learning, to problems in DeFi.

References

[1] H.-T. Kao, T. Chitra, R. Chiang, and J. Morrow, "An analysis of the market risk to participants in the compound protocol," in *Third International Symposium on Foundations and Applications of Blockchains*, 2020.

[2] H.-T. Kao and N. Lord, "Makerdao auction assessment," 2021, https://maker-report.gauntlet.network.

[3] W. Fu, T. Chitra, R. Chiang, and J. Morrow, "Aave market risk assessment," 2021, https://gauntlet.network/reports/aave.

[4] T. Chitra, G. Angeris, A. Evans, and H.-T. Kao, "A note on borrowing constant function market maker shares," 2021.

[5] H.-T. Kao, T. Chitra, R. Chiang, and J. Morrow, "Liquity market risk assessment," 2021, https://liquity-report.gauntlet.network.

[6] G. Angeris, A. Evans, and T. Chitra, "When does the tail wag the dog? curvature and market making," *arXiv preprint arXiv:2012.08040*, 2020.

[7] 0xngmi, "Defi dashboard." [Online]. Available: https://defillama.com

[8] G. Pohl-Zaretsky, "Sushiswap increased liquidity mining efficiency 5x and added $300m daily volume with the gauntlet platform," Oct 2021. [Online]. Available: https://medium.com/gauntlet-networks/sushiswap-increased-liquidity-mining-efficiency-5x-and-added-300m-daily-volume-with-the-gauntlet-a

[9] C. Team, "Figuring out your required boost." [Online]. Available: https://resources.curve.fi/guides/boosting-your-crv-rewards

[10] S. Team. [Online]. Available: https://help.sushidocs.com/guides/how-to-stake-sushi-for-xsushi

[11] E. Nicolle, "Crypto secrecy makes defi a financial felon's wonderland," Jan 2022. [Online]. Available: https://www.bloomberg.com/news/articles/2022-01-27/crypto-s-cloak-of-anonymity-makes-defi-a-wonderland-for-felon

[12] R. Behnke, "Explained: The fei protocol bug (april 2021)," Oct 2021. [Online]. Available: https://halborn.com/explained-the-fei-protocol-bug/

[13] O. D. team, "What is bonding?" [Online]. Available: https://docs.olympusdao.finance/main/basics/bonding

[14] U. Zeus, "Exploring a bond-centric future," Feb 2022. [Online]. Available: https://forum.olympusdao.finance/d/1061-exploring-a-bond-centric-future

[15] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, vol. 20, no. 4, pp. 633–679, 2020.

[16] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 5582–5588.

[17] T. Chitra, "Competitive equilibria between staking and on-chain lending," *Cryptoeconomic Systems*, Nov. 2020. [Online]. Available: https://doi.org/10.21428/58320208.9ce1cd26

[18] T. Chitra and A. Evans, "Why stake when you can borrow?" *Available at SSRN 3629988*, 2020.

[19] abipup, "Oip-76: Create inverse bond policy lever." [Online]. Available: https://forum.olympusdao.finance/d/1020-oip-76-create-inverse-bond-policy-lever

[20] U. @shadow, "Olympus dao dune analytics dashboard," Jan 2022. [Online]. Available: https://dune.xyz/shadow/Olympus-(OHM)

[21] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, "An analysis of uniswap markets," *Cryptoeconomic Systems*, Nov. 2020. [Online]. Available: https://doi.org/10.21428/58320208.c9738e64

[22] G. Angeris, A. Agrawal, A. Evans, T. Chitra, and S. Boyd, "Constant function market makers: Multi-asset trades via convex optimization," 2021.

[23] ——, "Constant function market makers: Multi-asset trades via convex optimization," *arXiv preprint arXiv:2107.12484*, 2021.

[24] O. D. core team, "Olympus dao verified smart contract code," Jan 2022. [Online]. Available: https://etherscan.io/address/0x575409F8d77c12B05feD8B455815f0e54797381c#code

[25] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[26] B. Øksendal, "Stochastic differential equations," in *Stochastic differential equations*. Springer, 2003, pp. 65–84.

[27] A. Evans, G. Angeris, and T. Chitra, "Optimal fees for geometric mean market makers," *arXiv preprint arXiv:2104.00446*, 2021.

[28] L. C. Evans, "An introduction to mathematical optimal control theory version 0.2," *Lecture notes available at http://math. berkeley. edu/˜ evans/control. course. pdf*, 1983.

[29] P. O. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Transactions on automatic control*, vol. 43, no. 8, pp. 1163–1169, 1998.

[30] J. Yong and X. Y. Zhou, *Stochastic controls: Hamiltonian systems and HJB equations*. Springer Science & Business Media, 1999, vol. 43.

[31] F. M. Callier and C. A. Desoer, *Linear system theory*. Springer Science & Business Media, 2012.

[32] A. Wan, Jan 2022. [Online]. Available: https://mirror.xyz/andywan.eth/KylisgFyv9KA-Z21A6PcmbVF-3K8Jia-VS38wT2s8GQ

[33] T. Team, "Introduction to tokemak: Tool for sustainable liquidity." [Online]. Available: https://docs.tokemak.xyz/

[34] G. Angeris, T. Chitra, A. Evans, and S. Boyd, "Optimal routing for constant function market makers," 2021. [Online]. Available: https://web.stanford.edu/~boyd/papers/cfmm_routing.html

[35] D. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[36] K. Clarke, "Curve wars: The future of defi dependent on a windows 98 application," Nov 2021. [Online]. Available: https://medium.com/momentum6/the-great-curve-war-is-heating-up-dont-miss-the-alpha-9b70d99cf694#:~:text=What%20is%20the%20Curve%20War,increased%20buying%20pressure%20for%20%24CRV.

[37] A. Cronje, “Ve(3,3),” Jan 2022. [Online]. Available: https://andrecronje.medium.com/ve-3-3-44466eaa088b

[38] M. Egorov, “StableSwap - efficient mechanism for Stablecoin liquidity,” p. 6.

[39] E. Au, “Game theory behind curve and convex finance,” Jan 2022. [Online]. Available: https://www.theblockresearch.com/game-theory-behind-curve-and-convex-finance-129432

[40] L. C. Evans and P. E. Souganidis, “Differential games and representation formulas for solutions of hamilton-jacobi-isaacs equations,” *Indiana University mathematics journal*, vol. 33, no. 5, pp. 773–797, 1984.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[42] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory.* SIAM, 1998.

[43] H.-T. Kao and T. Chitra, “Feedback control as a new primitive for defi,” Nov 2020. [Online]. Available: https://medium.com/gauntlet-networks/feedback-control-as-a-new-primitive-for-defi-27b493f25b1

[44] E. A. Buehler, J. A. Paulson, and A. Mesbah, “Lyapunov-based stochastic nonlinear model predictive control: Shaping the state probability distribution functions,” in *2016 American Control Conference (ACC).* IEEE, 2016, pp. 5389–5394.

[45] J. Yong, “Linear-quadratic optimal control problems for mean-field stochastic differential equations—time-consistent solutions,” *Transactions of the American Mathematical Society*, vol. 369, no. 8, pp. 5467–5523, 2017.

[46] M. Scheutzow, “A stochastic gronwall lemma,” *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, vol. 16, no. 02, p. 1350019, 2013.

[47] A. Hudde, M. Hutzenthaler, and S. Mazzonetto, “A stochastic gronwall inequality and applications to moments, strong completeness, strong local lipschitz continuity, and perturbations,” in *Annales de l’Institut Henri Poincaré, Probabilités et Statistiques*, vol. 57, no. 2. Institut Henri Poincaré, 2021, pp. 603–626.

A Stochastic Quadratic Control

We generalize the formulation of §4.2 to the stochastic setting. Formalizing consider the stochastic linear quadratic control problem:

$$dP(t) = [AP(t) + Bu(t)]dt + [CP(t) + Du(t)]dW_t \tag{22}$$

$$J(P_0, u(\cdot)) = \mathbf{E}\left[\int_0^\infty qP(\tau)^2 + r\|u(\tau)\|_2^2 d\tau\right] \tag{23}$$

where $P(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}^m$, $A, C \in \mathbb{R}$, $B^\top, D^\top \in \mathbb{R}^m$, and $q, r \in \mathbb{R}$. Here, for simplicity of notation and analysis we have assumed that $P^\star = 0$. We show the following:

Theorem 2. *If $A, C \sim \mathcal{N}(0,1)$ and $B, D \sim \mathcal{N}(0, I_m)$, there exists a feedback controller $u(t) = KP(t)$ minimizing* (23) *such that $dP(t) = [(A+BK)P(t)]dt + [(C+DK)P(t)]dW_t$ satisfies*

$$\mathbf{E}[P^2(t)] \leq \mathbf{E}[e^{-c\sqrt{m}t}P^2(0)]$$

for some $c > 0$.

Proof. Defining the value function $V(P_0) = \inf_u J(P_0, u(\cdot))$, and assuming that it takes on a quadratic form similar to the deterministic case, $V(P) = \rho\|P\|^2$ where ρ is as in (19). Computing $\dot{V}(P)$ [44] gives

$$\dot{V}(P) = P^T \left(A^T\rho + \rho A + K^T B\rho + \rho K B + C^T \rho C + C^T \rho D K + K^T D \rho C + K^T D \rho D K\right) P$$

From [45], the control K has the form

$$K = (r + \rho D^T D)^{-1}(B^T P + \rho D^T D)$$

and satisfies a stochastic Riccati equation of the form

$$A^T\rho + \rho A + C^T\rho C - (B^T\rho D^T\rho C)^T(r + \rho D^T D)^{-1}(B^T\rho + D^T\rho C) = -1$$

Note that $\alpha = (r + \rho D^T D)$ is a positive scalar. Recall that B, D are drawn independently from $N(0, I_m)$. Using the Jensen relation $\mathbf{E}[1/x] \geq 1/\mathbf{E}[x]$, we can write the following bound for the expectation of the second term in the above equation

$$\begin{aligned}
\mathbf{E}[(B^T\rho + D^T\rho C)^T(r + \rho D^T D)^{-1}(B^T\rho + D^T\rho C)] &\geq \frac{\mathbf{E}[(B^T\rho D^T\rho C)^T(B^T\rho + D^T\rho C)]}{\mathbf{E}[(r + \rho D^T D)]} \\
&= \frac{\rho^2(1 + 2c + c^2)\Omega(m)}{r + \rho O(m)} \\
&= \frac{\rho^2(c+1)^2\Omega(m)}{r + \rho O(m)} \\
&\geq C''\rho
\end{aligned}$$

where the second inequality used the fact that $\mathbf{E}[B^T B] = \mathbf{E}[D^T D] = m$ and expanded the quadratic terms. Using the stochastic Riccati equation and this bound, once can simplify $\dot{V}(P)$ to

$$\mathbf{E}[\dot{V}(P)] \leq P^T \left(-O(\rho) - \mathbf{E}\left[\frac{1}{\alpha^2}(\rho B + \rho C D)(\rho D^T D)(B^T\rho + D^T\rho C)\right]\right) P$$

We claim the term in the expectation is also $O(\rho)$; one can see this by noting that the numerator is $O(\rho^3 m^2)$ whereas the denominator is $O(r + \rho^2 m^2) = O(\rho^2 m^2)$. This implies that we have:

$$\mathbf{E}[\dot{V}(P)] \leq -O(\rho)P^T P = -O(\rho)V(P)$$

Under mild measure theoretic assumptions on $\dot{V}(P)$, we can apply the Stochastic Grönwall inequality [46, 47] to achieve the result.

$\square$