ManiPerp: A Simple Pool-Based AMM for Perpetual Futures

Stephen Grugett
Manifold Markets, MNX

February 2026

Abstract

We present ManiPerp, a minimal automated market maker (AMM) for perpetual futures contracts. The mechanism uses dual liquidity pools—one for long positions and one for short—with a funding rate that rebalances the pools based on open interest imbalance. All positions are opened and closed at an externally provided oracle price; the system performs no internal price discovery. We define the core operations (opening, closing, funding, and liquidation) as state transitions over the pool and position state, and introduce a solvency factor that guarantees the system can always honor withdrawals without requiring auto-deleveraging (ADL). The design is intended for play-money prediction markets such as Manifold, where simplicity and support for thinly traded, user-created markets are more important than capital efficiency.

1 Introduction

Perpetual futures (perps) are derivative contracts with no expiry date. Unlike traditional futures, which settle on a fixed date, perps use a periodic *funding rate* to anchor the contract price to an external oracle. When a perp trades above the oracle price, long position holders pay short holders, creating downward pressure; the symmetric case applies when the perp trades below the oracle. This mechanism, introduced by BitMEX and now dominant in cryptocurrency markets, produces a simple instrument for gaining leveraged exposure to any continuously quoted index.

Orderbook-based perp exchanges (e.g., dYdX, Hyperliquid) require sufficient liquidity on both sides of the book to function. Pool-based designs such as GMX and Ostium replace the orderbook with shared liquidity pools that act as counterparty to all traders, making them better suited to long-tail markets with low trading volume. ManiPerp follows this approach but with a substantially simpler mechanism, trading capital efficiency for clarity and ease of implementation.

The primary motivation is to extend prediction market platforms like Manifold beyond binary and multi-choice questions to continuous time-series: asset prices, temperature indices, approval ratings, AI benchmark scores, and any other quantity with a regularly updated data feed.

2 Model

2.1 State

We define the system state as a tuple $(L, S, \mathcal{P})$, where L and S are the mana balances of the Long and Short liquidity pools respectively, and $\mathcal{P}$ is the set of open positions. Each position $p \in \mathcal{P}$ is a tuple (d, q, c, P_e) where:

- $d \in \{\texttt{long}, \texttt{short}\}$ is the direction,
- q is the position size (in notional mana),
- c is the cost basis (margin deposited),
- P_e is the entry price.

The liquidation price of a position with leverage $\ell = q/c$ is:

$$P_{\text{liq}} = \begin{cases} \left(1 - \dfrac{1}{\ell}\right) P_e & \text{if } d = \texttt{long} \\ \left(1 + \dfrac{1}{\ell}\right) P_e & \text{if } d = \texttt{short} \end{cases} \tag{1}$$

Let P denote the current oracle price, an externally provided value updated at regular intervals by a data source outside the system (e.g., an API feed for asset prices, a weather service for temperature indices). The system does not perform price discovery—all positions are opened and closed at the prevailing oracle price. The AMM's role is solely to manage liquidity, leverage, and solvency.

A market is initialized with $L_0 = S_0 > 0$ and $\mathcal{P} = \emptyset$.

2.2 Opening a Position

A trader opens a position by specifying a direction d, a mana amount m, and a leverage $\ell \leq \ell_{\text{max}}$. The position is entered at the current oracle price P.

Definition 2.1 (Open). $\texttt{open}(d, m, \ell)$ takes as input a direction, mana amount, and leverage, and updates the state:

$$(L, S, \mathcal{P}) \xrightarrow{\texttt{open}(d,m,\ell)} (L', S', \mathcal{P}') \tag{2}$$

where, if $d = \texttt{long}$:

$$L' = L + m, \qquad S' = S, \qquad \mathcal{P}' = \mathcal{P} \cup \left\{(\texttt{long},\ m \cdot \ell,\ m,\ P)\right\} \tag{3}$$

and symmetrically for $d = \texttt{short}$.

2.3 Funding

Every funding period, we compute the imbalance between the two pools and transfer mana from the dominant side to the minority side, while simultaneously haircutting the positions on the dominant side.

Definition 2.2 (Imbalance Function). Let $r = L/S$. The imbalance function I is defined as:

$$I(r) = \frac{r - 1}{r - 1 + k}, \qquad r \geq 1 \tag{4}$$

where $k > 0$ is a sensitivity parameter. I satisfies:

1. $I(1) = 0$,
2. I is monotonically increasing,
3. $\lim_{r \to \infty} I(r) = 1$,
4. $I(1 + k) = \frac{1}{2}$—i.e., the funding rate reaches half its maximum when the imbalance ratio exceeds equilibrium by k.

The funding rate for the period is $f = I(r) \cdot f_{\text{max}}$, where f_{max} is a market parameter.

Definition 2.3 (Fund). $\texttt{fund}()$ updates the state as follows. If $L > S$:

$$(L, S, \mathcal{P}) \xrightarrow{\texttt{fund}()} (L', S', \mathcal{P}') \tag{5}$$

where:

$$L' = (1 - f) \cdot L \tag{6}$$
$$S' = S + f \cdot L \tag{7}$$

and for each long position $p = (\texttt{long}, q, c, P_e) \in \mathcal{P}$:

$$p' = \big(\texttt{long},\ (1-f)\cdot q,\ (1-f)\cdot c,\ P_e\big) \tag{8}$$

Short positions are unchanged. The symmetric case applies when $S > L$.

Remark 2.1. The funding mechanism haircuts each position's size and cost basis by a factor of f. Since the pool and all positions on the dominant side are scaled by the same factor, the pool is always able to return the cost bases of its participants. Short position holders on the minority side pay nothing and passively receive the transferred mana through their increased share of the Short pool.

2.4 Liquidation

When the oracle price updates, we check all open positions for liquidation.

Definition 2.4 (Liquidate). `liquidate`(P) takes a new oracle price and updates the state. For each long position $p = (\texttt{long}, q, c, P_e) \in \mathcal{P}$ with $P_{\text{liq}} \geq P$:

$$q' = 0, \qquad c' = 0 \tag{9}$$

The position's margin remains in the Long pool (L is unchanged). The symmetric case applies for short positions where $P_{\text{liq}} \leq P$.

2.5 Closing a Position

When a trader closes a position, the payout consists of two components: (1) their cost basis, returned from their own pool, and (2) any profit, paid from the opposing pool and scaled by a solvency factor.

Definition 2.5 (Close). `close`(p) for a long position $p = (\texttt{long}, q, c, P_e)$ at the current oracle price P updates the state as follows. The position is exited at P—no slippage or price impact is applied.

The position equity is:

$$\pi = \frac{P - P_e}{P_e} \cdot q \tag{10}$$

If $\pi \leq 0$ (the position is at a loss), the payout is $\max(c + \pi,\ 0)$, drawn entirely from the Long pool.

If $\pi > 0$ (the position is profitable), the payout is:

$$\text{payout} = c + s \cdot \pi \tag{11}$$

where s is the *solvency factor*:

$$s = \frac{S - C}{E} \tag{12}$$

and:

- S is the current balance of the Short pool,
- $C = \sum_{p_j \in \mathcal{P}_S} \min(c_j,\ v_j)$ is the reserved cost bases for open short positions, where v_j is the current value of position p_j,
- $E = \sum_{\substack{p_i \in \mathcal{P}_L \\ \pi_i > 0}} \pi_i$ is the total unrealized equity of profitable long positions.

The solvency factor satisfies $0 \leq s \leq 1$ and ensures the system can always honor all withdrawals. It serves the same role as auto-deleveraging (ADL) in centralized perp exchanges, but applies proportionally to all profitable positions rather than forcibly closing selected ones.

The pool is updated:

$$L' = L - c, \qquad S' = S - s \cdot \pi, \qquad \mathcal{P}' = \mathcal{P} \setminus \{p\} \tag{13}$$

2.6 Liquidity Provision

Multiple liquidity providers (LPs) can co-fund a market. Each LP's ownership is tracked as a fractional share of each pool.

Definition 2.6 (Add Liquidity). `addLiquidity`(Δ_L, Δ_S) for an LP depositing Δ_L and Δ_S into the Long and Short pools respectively updates the state:

$$(L, S) \xrightarrow{\texttt{addLiquidity}(\Delta_L, \Delta_S)} (L + \Delta_L,\ S + \Delta_S) \tag{14}$$

The LP's ownership shares are:

$$\omega_L = \frac{\Delta_L}{L + \Delta_L}, \qquad \omega_S = \frac{\Delta_S}{S + \Delta_S} \tag{15}$$

An LP may withdraw their proportional share of each pool at any time. In practice, the protocol may wish to restrict withdrawals to amounts in excess of current obligations—i.e., the cost bases and unrealized profits claimable by open positions. A simple implementation is to permit withdrawals only when the solvency factor $s = 1$ for both sides, ensuring that no withdrawal can push the system into a state where profitable positions cannot be fully paid out.

Remark 2.2. Subsidizing a perp market is a potentially better proposition for LPs than subsidizing a prediction market. In a binary market, the price converges to 0% or 100%, leaving LPs with nothing. In a perp market, as long as the oracle price does not move dramatically in one direction, LPs can recover a substantial portion of their funds. If trading fees are introduced on opens and closes, LPs could reasonably expect to earn a net return.

3 Market Parameters

To instantiate an ManiPerp market, the creator specifies the parameters listed in Table 1.

Parameter	Description
Oracle	An external index, updated at regular intervals
$\ell_{\max}$	Maximum leverage
$f_{\max}$	Maximum funding rate per period
k	Funding sensitivity
Funding period	Frequency of funding events (e.g., hourly, daily)
(L_0, S_0)	Initial liquidity pool subsidy

Table 1: Market parameters required to instantiate an ManiPerp market.

4 Discussion

ManiPerp is deliberately minimal. All trades execute at the oracle price with no slippage or price impact—the funding rate is the sole mechanism that discourages imbalanced positioning. The design omits several features present in production perp protocols—including price-impact curves, dynamic fees, and internal price discovery—in favor of a system that is easy to reason about, implement, and explain to non-expert users. The solvency factor is the key mechanism that guarantees safety without complexity: by proportionally scaling payouts when the opposing pool is under-capitalized, the system avoids both insolvency and the need for forced position closures.

The design opens a large space of continuous-valued markets that are difficult to express as binary or multi-choice prediction questions: weather indices, economic indicators, approval ratings, benchmark scores, and more. Any quantity with a reliable, regularly updated data feed is a candidate for an ManiPerp market.