Market-Neutral Volatility Harvesting (MNVH) and Ledger-SPT

— A deployable framework to monetize dispersion under a self-financing constraint in discrete time

Daiki Sasaki, MD, PhD

September 23, 2025

Abstract

This paper formalizes **Market-Neutral Volatility Harvesting (MNVH)** in discrete time together with an implementation layer, **Ledger-SPT**, that satisfies **self-financing** and **accounting exactness**. MNVH adjusts quantities using the deviation between each asset's step return $r_{i,t}$ and the portfolio return R_t, i.e., $R_t - r_{i,t}$, harvesting **cross-sectional dispersion** under short-horizon **mean reversion** while remaining near **market-neutral**. Trading starts from **zero initial positions** ($q_{i,0} = 0$), and no external cash is injected thereafter.

1 Introduction

Shannon's Demon shows the growth effect of rebalancing, while SPT (Stochastic Portfolio Theory) provides a dispersion-driven drift in relative wealth for functionally generated portfolios. Many theoretical treatments assume continuous time and frictionless markets, omitting ledgers, fees, and lots. Our contribution is a **minimal, deployable** MNVH controller in discrete time and an execution loop, **Ledger-SPT**, that makes self-financing, fees, and lot adjustments exact. We also make explicit the initial condition $q_{i,0} = 0$, reconciling zero-start operation with neutrality.

2 Notation and Assumptions (Discrete Time; Zero Start)

- Assets $i = 1, \ldots, n$; prices $p_{i,t} > 0$; simple returns $r_{i,t} = (p_{i,t+1} - p_{i,t})/p_{i,t}$.
- Held quantities $q_{i,t}$; portfolio market value $V_t = \sum_i p_{i,t} q_{i,t}$.
- Target notional weights w_i with $\sum_i w_i = 1$ (signs allowed).

- **Initial condition**: $q_{i,0} = 0,\ V_0 = C_0$ (cash only). **Self-financing**: per-step trading cashflow + fees = 0; external cashflow $\mathrm{CF}_t = 0$.

3 MNVH Update Rule (Quantity Control)

Update (proposed quantities): $\boldsymbol{q_{i,t+1} = \lambda q_{i,t} + \beta(w_i/p_{i,t})(R_t - r_{i,t})}$, with $R_t = \sum_j w_j r_{j,t}$, inventory leash $\lambda \in [0,1]$, and aggressiveness $\beta > 0$. Buy recent underperformers and sell outperformers to harvest next-step mean reversion. Quantity control reduces sensitivity to price-level effects.

4 Ledger-SPT: Accounting-Exact, Self-Financing Execution

Maintain per-asset lots (quantity, price, fees) and cash C_t. Before orders, aggregate $\Delta q_{i,t} = q_{i,t+1} - q_{i,t}$ and fees $\phi_{i,t}$. Enforce

$$\sum_i p_{i,t} \Delta q_{i,t} + \sum_i \phi_{i,t} = 0. \tag{1}$$

If violated, find $\kappa_t \in (0,1]$ and set $\Delta \tilde{q}_{i,t} = \kappa_t \Delta q_{i,t}$. PnL (mid mark) decomposes as

$$\Delta V_t = \underbrace{\sum_i q_{i,t} p_{i,t} r_{i,t}}_{\text{MTM}} + \underbrace{\sum_i p_{i,t} \Delta \tilde{q}_{i,t}}_{\text{trading cashflow}} - \sum_i \phi_{i,t}. \tag{2}$$

Log each component for auditability.

5 Shape of Expected Return (Mean-Reversion Assumption)

Under $(r_{i,t+1} - R_{t+1}) \approx -\rho(r_{i,t} - R_t) + \varepsilon$, a first-order approximation yields

$$\mathbb{E}[\Delta V^{\text{trade}}_{t \to t+1}] \approx \beta V_t \rho\, \mathbb{E}\Big[\sum_i w_i (r_{i,t} - R_t)^2\Big] - \mathbb{E}[\text{costs}]. \tag{3}$$

A sufficient condition is $\beta V_t \rho\, \mathbb{E}[\mathrm{XSVar}_t] >$ fees+slippage, where $\mathrm{XSVar}_t = \sum_i w_i (r_{i,t} - R_t)^2$.

6 Relation to SPT

MNVH is a discrete-time, control-theoretic analogue of SPT's excess-growth effect: an empirical gradient step proportional to $R_t - r_{i,t}$ to deliberately monetize dispersion. Ledger-SPT ensures full accounting capture.

7 Deriving $(R - r)$ from SPT Definitions and Embedding into a Market-Neutral System

7.1 SPT basics and a discrete-time proxy

In continuous-time SPT, for a portfolio with prices S_i and weights π_i,

$$d \log V_t^\pi = \sum_i \pi_{i,t} \, d \log S_{i,t} + \frac{1}{2} \Gamma_t(\pi) \, dt \quad \text{(schematic)}. \tag{4}$$

The second term is the *excess growth rate*, conceptually "sum of constituent variances minus portfolio variance," contributing positively. As a one-step discrete proxy with simple returns $r_{i,t}$ and $R_t = \sum_i w_i r_{i,t}$,

$$\mathcal{G}_t(w) := -\frac{1}{2} \sum_i w_i \, (r_{i,t} - R_t)^2 \tag{5}$$

acts as a generating function whose gradient magnitude grows with cross-sectional variance.

7.2 Gradient $\Rightarrow$ quantity step (natural emergence of $(R - r)$)

Let $V_t = \sum_j p_{j,t} q_{j,t}$ and $w_i = (p_{i,t} q_{i,t})/V_t$. The first variation (under a one-step constant-V_t approximation) gives

$$\Delta q_{i,t} \propto + \frac{w_i}{p_{i,t}} (R_t - r_{i,t}). \tag{6}$$

Thus the canonical $R_t - r_{i,t}$ signal appears mechanically from the cross-sectional-variance-based generator.

7.3 Embedding into a self-financing, neutral system

(i) Enforce self-financing by scaling with κ_t so that $\sum_i p_{i,t} \Delta q_{i,t} + \sum_i \phi_{i,t} = 0$. (ii) Keep regression beta near zero via a hedge or a neutrality term. (iii) Respect the zero-start condition $q_{i,0} = 0$.

8 Estimation and Parameterization

Weights: equal / inverse-vol / shrinkage Markowitz. Estimate ρ via rolling regression; clip to $[0, 1]$. Choose β from turnover, margin and fee budget (stabilize fee/gross at 20–40%); $\lambda \in [0.8, 0.98]$ per step. Costs: explicit fees + participation$\times$spread$\times$volatility.

9 Experimental Design (Zero Start Enforced)

- **Initial condition**: $q_{i,0} = 0,\ C_0 = V_0$. Thereafter self-financing only.
- **Universe**: 30–200 liquid perps/futures or cash equities.
- **Time scale**: 1–60 minutes or daily; synchronized steps.
- **Backtest**: OOS rolling; realistic fees; queue slippage; funding/borrow.
- **Metrics**: CAGR, vol, Sharpe, turnover, fee/gross, hit rate, skew/kurtosis, drawdown, crowding/stress robustness.
- **Ablations**: λ, β, weight design, with/without hedge, driver (here $R - r$ only).

10 Risk Management and Market Neutrality

Monitor beta neutrality; pre-compute worst-case margin usage; throttle β when dispersion is trend-dominated.

11 Typical Pitfalls

Timestamp misalignment; underestimating fees; inventory drift if λ too high; breakdown of reversion in stress.

Appendix A: Pseudocode Satisfying Zero-Start and Self-Financing

```
INPUT: prices p[i], weights w[i], lambda, beta
STATE: cash C, positions q[i]=0 (for all i at t=0), ledger lots = {}

At each step t:
1) compute simple returns r[i] from p[i] -> p_next[i]
2) R = sum_i w[i]*r[i]
3) propose q_star[i] = lambda*q[i] + beta*(w[i]/p[i])*(R - r[i])
   dq_star[i] = q_star[i] - q[i]
4) simulate fees phi[i](dq_star[i]); S = sum_i p[i]*dq_star[i] + sum_i phi[i]
5) if S != 0:
      find kappa in (0,1] s.t. sum_i p[i]*(kappa*dq_star[i]) + sum_i phi_i(
    kappa) = 0
      dq[i] = kappa*dq_star[i]
   else:
      dq[i] = dq_star[i]
```

```
6) place orders dq[i]; update ledger and cash; set q[i] <- q[i] + filled(dq[i])
7) risk checks (beta, leverage, drawdown); logs (MTM, trade cashflow, fees)
```