High-Frequency Binary Inventory Management Strategy

Legging into a Risk-Free Box Spread

Algorithmic Specification

December 12, 2025

1 Definitions and State Space

Let the state of the trading portfolio at time t be defined by the vector $\mathcal{S}_t$:

$$\mathcal{S}_t = \{Q_{yes}, C_{yes}, Q_{no}, C_{no}\} \tag{1}$$

Where:

- Q_i: Total quantity of shares held for outcome $i \in \{\text{YES}, \text{NO}\}$.
- C_i: Total cost basis (USD spent) for outcome i.

We define the **Volume Weighted Average Price (VWAP)** for the current inventory:

$$\mu_{yes} = \frac{C_{yes}}{Q_{yes}}, \quad \mu_{no} = \frac{C_{no}}{Q_{no}} \tag{2}$$

(Note: If $Q_i = 0$, then $\mu_i = 0$).

2 The Profit Invariant

The strategy's objective is to satisfy the **Box Spread Inequality**. A risk-free profit is mathematically guaranteed if the sum of the average costs is less than the payout.

We define a global **Target Cost Basis** (C_{target}):

$$C_{target} = 1.00 - \xi \tag{3}$$

Where ξ is the minimum required profit margin (e.g., 0.02).

The Invariant condition to maintain is:

$$\mu_{yes} + \mu_{no} \leq C_{target} \tag{4}$$

The realized, locked profit (P_{lock}) is:

$$P_{lock} = \min(Q_{yes}, Q_{no}) \times \xi \tag{5}$$

3 Execution Logic

The algorithm switches between two modes based on the **Inventory Imbalance** ΔQ:

$$\Delta Q = Q_{yes} - Q_{no} \tag{6}$$

3.1 Mode A: Open (The Trap Entry)

Condition: $\Delta Q \approx 0$ (Portfolio is balanced).

In this phase, the algorithm acts as a Market Maker, placing "Trap" orders to capture volatility. The limit price π_{limit} for side i is derived from the **Inverse Price** of the opposing side j:

$$\pi_{limit}^{i} = C_{target} - P_{ask}^{j} - \mathcal{R}(P_{ask}^{i}) \tag{7}$$

Where $\mathcal{R}(p)$ is a dynamic **Risk Premium** determined by the probability of the outcome (see Section 6).

3.2 Mode B: Hedge (Closing the Leg)

Condition: $|\Delta Q| > 0$ (Portfolio is exposed).

The goal is to neutralize ΔQ while maintaining Invariant (4). Assume we are Long YES ($\Delta Q > 0$). We must buy NO.

We solve for the **Maximum Limit Price** (π_{limit}^{no}) such that the combined VWAP remains valid:

$$\pi_{limit}^{no} = C_{target} - \mu_{yes} \tag{8}$$

If $P_{ask}^{no} \leq \pi_{limit}^{no}$, execute the hedge immediately.

4 Safety Constraints

To prevent ruin from trending markets, the following constraints are mandatory:

1. **Maximum Exposure Cap (E_{max}):**

$$|\Delta Q| \leq E_{max} \tag{9}$$

   If breached, enter **Emergency Liquidation** (Market Orders).

2. **Time Decay Stop:** If $t > T_{expiry} - 2\text{min}$ and $|\Delta Q| > 0$, close positions immediately.

5 Algorithmic Loop

Algorithm 1 Box Spread Legging Strategy

Input: E_{max}, C_{target}
while $t < T_{expiry}$ **do**
  Update Portfolio: $Q_{yes}, \mu_{yes}, Q_{no}, \mu_{no}$
  Update Market: $P_{ask}^{yes}, P_{ask}^{no}$
  Calculate Imbalance: $\Delta Q \leftarrow Q_{yes} - Q_{no}$
  if $|\Delta Q| \geq E_{max}$ **then**
    EMERGENCY_CLOSE(ΔQ)
  else if $\Delta Q > 0$ **then** ▷ **Mode B:** Long YES, Buy NO
    $\pi_{limit} \leftarrow C_{target} - \mu_{yes}$
    if $P_{ask}^{no} \leq \pi_{limit}$ **then**
      PLACE_LIMIT_BUY(Side=NO, Price=π_{limit}, Qty=ΔQ)
    end if
  else if $\Delta Q < 0$ **then** ▷ **Mode B:** Long NO, Buy YES
    $\pi_{limit} \leftarrow C_{target} - \mu_{no}$
    if $P_{ask}^{yes} \leq \pi_{limit}$ **then**
      PLACE_LIMIT_BUY(Side=YES, Price=π_{limit}, Qty=$|\Delta Q|$)
    end if
  else ▷ **Mode A:** Balanced, Set Traps
    $\pi_{yes} \leftarrow C_{target} - P_{ask}^{no} - \mathcal{R}(P_{ask}^{yes})$
    $\pi_{no} \leftarrow C_{target} - P_{ask}^{yes} - \mathcal{R}(P_{ask}^{no})$
    UPDATE_ORDER(Side=YES, Price=π_{yes})
    UPDATE_ORDER(Side=NO, Price=π_{no})
  end if
  Sleep(Latency_Tick)
end while

6 Decision Matrix and Risk Thresholds

The Risk Premium function $\mathcal{R}(p)$ protects against adverse selection in low-probability outcomes.

Table 1: Dynamic Entry Strategy Matrix ($C_{target} = 0.98$)

Market State	Strategy Mode	YES Bid (π_{limit}^{yes})	NO Bid (π_{limit}^{no})	Risk Profile
50 / 50	Double Trap	$C_{target} - P_{ask}^{no}$	$C_{target} - P_{ask}^{yes}$	**Low**
40 / 60	Double Trap	$C_{target} - P_{ask}^{no}$	$C_{target} - P_{ask}^{yes}$	**Low**
30 / 70	Skewed Trap	$(C_{target} - P_{ask}^{no}) - 0.02$	$C_{target} - P_{ask}^{yes}$	**Medium**
20 / 80	Heavy Skew	$(C_{target} - P_{ask}^{no}) - 0.05$	$C_{target} - P_{ask}^{yes}$	**High**
10 / 90	Vulture Only	*No Bid*	*No Bid*	**Extreme**

6.1 Risk Premium Function Definition

The formalized Risk Premium $\mathcal{R}(p)$ used in the algorithm is:

$$\mathcal{R}(p) = \begin{cases} 0 & \text{if } p \geq 0.40 \quad \text{(Fair Value)} \\ 0.02 & \text{if } 0.25 \leq p < 0.40 \quad \text{(Skew)} \\ 0.05 & \text{if } 0.15 \leq p < 0.25 \quad \text{(Deep Skew)} \\ \infty & \text{if } p < 0.15 \quad \text{(Vulture/No Quote)} \end{cases} \tag{10}$$

6.2 Execution Example: The Double Trap

Scenario: Balanced Market (Zone 1).

- **Market State:** $P_{ask}^{yes} = 0.51, \quad P_{ask}^{no} = 0.51$
- **Objective:** $C_{target} \leq 0.98$

Step 1: Trap Placement

$$\pi_{limit}^{yes} = C_{target} - P_{ask}^{no} = 0.98 - 0.51 = \mathbf{0.47}$$
$$\pi_{limit}^{no} = C_{target} - P_{ask}^{yes} = 0.98 - 0.51 = \mathbf{0.47}$$

Action: Submit Limit Buys on both sides at 0.47.

Step 2: The Trigger (First Leg)

Panic selling drives YES down. The YES trap is filled at 0.47.

- Inventory: $\{Q_{yes} = 1, C_{yes} = 0.47\}$.
- Action: Cancel NO trap.

Step 3: The Hedge (Second Leg)

Recalculate max price for NO:

$$\pi_{limit}^{no} = C_{target} - \mu_{yes} = 0.98 - 0.47 = \mathbf{0.51}$$

Action: Submit Limit Buy (NO) @ 0.51. When filled, profit is guaranteed.

7 Strategy Mechanics and Parameter Sensitivity

This strategy functions as a **High-Frequency Volatility Harvester**. Unlike traditional arbitrage, which executes simultaneous risk-free trades, this strategy "legs into" a risk-free position over a short temporal window (typically seconds or milliseconds).

The core thesis is that in binary markets, liquidity is fragmented. During volatility spikes, the sum of the bid-ask spread momentarily decouples from the probability unity:

$$P_{ask}^{yes}(t) + P_{ask}^{no}(t + \Delta t) < 1.00 \tag{11}$$

The strategy acts as a liquidity provider during these decoupling events.

7.1 Parameter Roles and Tuning

The performance of the algorithm is strictly governed by three primary control parameters. Tuning these values involves a trade-off between *Fill Rate* (Volume) and *Safety* (Risk of Ruin).

C_{target} **(Target Cost Basis):**

Definition: The maximum combined price we are willing to pay for one share of YES and one share of NO. Defined as $1.00 - \xi$.
Typical Value: 0.98 (implies 2% profit margin).
Role:

- *Aggressive (> 0.99):* High fill rate, but highly sensitive to trading fees. Risk of net loss after commission.
- *Conservative (< 0.96):* Guarantees high profit per trade, but rarely executes. Only catches extreme crashes.

E_{max} **(Maximum Exposure Cap):**

Definition: The hard limit on the inventory imbalance $|\Delta Q|$.
Typical Value: 50 – 500 shares (dependent on bankroll).
Role: This is the primary defense against **Trend Risk**.

- If E_{max} is too high, the bot may accumulate a massive losing position during a market crash (e.g., Bitcoin dumps $1,000 in 1 minute) before it can hedge.
- If E_{max} is too low, the bot stops buying too early during a temporary dip, missing the opportunity to lower its cost basis.

$\mathcal{R}(p)$ **Thresholds (Risk Premium Zones):**

Definition: The price points at which the strategy demands a discount (0.40, 0.25, 0.15).
Role: These thresholds model the **convexity of risk** in binary options.

- As a probability approaches 0, the likelihood of a total loss increases exponentially, not linearly.
- The parameters enforce "Stink Bids" to ensure that we only hold low-probability outcomes if the potential payout justifies the risk of the asset going to zero.

7.2 The Phases of Execution

The lifecycle of a single trade unit follows a strict state transition to ensure the invariant is never violated.

1. **Phase 1: The Trap (Maker):** The bot provides liquidity on both sides (or the skewed side) at calculated prices. It does not know which way the market will move. It is passive.
2. **Phase 2: The Catch (Execution):** A market participant hits one of the traps. The bot creates an open inventory position ($\Delta Q \neq 0$). The bot is now *at risk*.
3. **Phase 3: The Pivot (Maker/Taker):** The bot immediately cancels the opposing trap. It calculates the specific price needed for the second leg to lock the profit.
4. **Phase 4: The Lock (Hedge):** The bot waits for mean reversion. Once the second leg fills, the risk is neutralized. The profit is now mathematical and independent of the market outcome.