

TECHNICAL SPECIFICATION: ANTICIPATORY-TRIANGLE-TREND (ATT) TRADING SYSTEM

This document serves as a complete, deterministic, and language-agnostic specification of the **Anticipatory-Triangle-Trend (ATT) Trading System** as developed in the Canvas database code. It defines the exact mathematical, logical, and chronological criteria required to build, compile, or translate this system into execution-ready code (e.g., Python, C#, JavaScript, Pine Script, or HTML).

1. SYSTEM OVERVIEW & DATA INPUTS

The system tracks structural trend reversals using a three-point ($T1, T2, T3$) geometric triangle formation across the baseline Volume Weighted Average Price (VWAP) of the first candle of the session. It aligns index-level trend projections with option-level momentum profiles to execute highly selective option buys.

Data Requirements

- **Symbol A (NIFTY Index):** 1-minute OHLC candles for `NSE:NIFTY50-INDEX`.
- **Symbol B (Options):** 1-minute OHLC candles for all CE and PE options belonging to the active underlying index.
- **Granularity:** 1-minute chronological interval bars.
- **Trading Window:** Reset daily. State does not carry over across trading days.

2. MATHEMATICAL CORE DEFINITIONS

Typical Price (hlc_3)

For any given candle, the typical price is the arithmetic mean of its high, low, and close:

$$hlc_3 = \frac{\text{High} + \text{Low} + \text{Close}}{3}$$

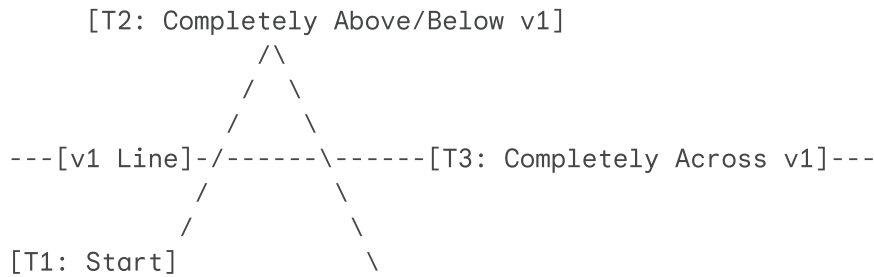
Baseline VWAP (v_1)

The baseline anchor line for a daily session is defined as the typical price of the very first candle (the 09:15 IST candle) of that day:

$$v_1 = hlc_{3, \text{first}}$$

3. TRIANGLE FORMATION ALGORITHM ($T1, T2, T3$)

The triangle formation logic is identical for both the underlying index and the resolved option contract.



Point T1 (The Anchor)

- **Definition:** The 1st chronological candle of the day.
- **Properties Pinned:** Timestamp (t_{T1}) and its typical price baseline (v_1).

Point T2 (The Deviation Point)

- **Definition:** The absolute first chronological candle *after* $T1$ where the entire price action (body and shadows) forms completely away from the baseline v_1 .
- **Mathematical Filter:**

$$\text{Candle } i \text{ is a valid } T2 \iff (\text{Low}_i > v_1) \vee (\text{High}_i < v_1), \quad \text{where } t_i > t_{T1}$$

- **Directional Trend Assignment:**

If $\text{Low}_{T2} > v_1 \implies \text{Trend} = \text{'PE'}$ (Overextended Upward, expecting downward)

If $\text{High}_{T2} < v_1 \implies \text{Trend} = \text{'CE'}$ (Overextended Downward, expecting upward)

- **Selection:** Pin the first chronological candle that satisfies this condition.

Point T3 (The Counter-Crossing Point)

- **Definition:** The absolute first chronological candle *after* $T2$ that forms completely **across** (on the opposite side of) the baseline v_1 relative to $T2$.
- **Mathematical Filter:**
 - If $T2$ Trend is 'PE' ($T2$ was above v_1):

$$\text{Candle } j \text{ is } T3 \iff \text{High}_j < v_1, \quad \text{where } t_j > t_{T2}$$

- If $T2$ Trend is 'CE' ($T2$ was below v_1):

$$\text{Candle } j \text{ is } T3 \iff \text{Low}_j > v_1, \quad \text{where } t_j > t_{T2}$$

- **Selection:** Pin the first chronological candle that satisfies this condition. Note down its timestamp (t_{T3}) and closing price.

4. MULTI-ASSET CORRELATION & MATCHING ENGINE

To execute a trade, the system correlates the Nifty Index's triangle structure with the candidate options' triangle structures.

Step 1: Nifty Triangle Resolution

1. Calculate Nifty $T1_N$, Nifty $T2_N$, and Nifty $T3_N$.
2. Resolve Nifty's Global Trend ($Trend_N$) and the closure time (t_{T3_N}).
3. Round Nifty's closing price at t_{T3_N} to the nearest multiple of 50 to define the **At-The-Money (ATM) Strike** (N_{50}):

$$N_{50} = \text{Round} \left(\frac{\text{Close}_{N,T3}}{50} \right) \times 50$$

Step 2: Option Candidate Filtering

1. Query the database for the two ATM option contracts matching strike N_{50} :
 - **CE Contract:** Ends with [Strike]CE (e.g., NSE:NIFTY2661623450CE)
 - **PE Contract:** Ends with [Strike]PE (e.g., NSE:NIFTY2661623450PE)

Step 3: Option Triangle Resolution

For both the CE and PE ATM option contracts, calculate their independent triangle points ($T1_O, T2_O, T3_O$) using their respective baseline typical prices (v_{1_O}).

- Identify their internal Option Trends ($Trend_O$).

Step 4: Directional Trend Matching

Options must always uptrend to be bought. For an option to uptrend, its $T2_O$ must form *below* its VWAP, which mathematically resolves its internal trend to 'CE'.

Apply the following deterministic mapping to lock the trading asset:

- **Case 1: Global Nifty Trend ($Trend_N$) is 'CE'**
 - Target Contract: **CE Option**
 - Execution Rule: Verify that the CE Option's internal trend ($Trend_O$) matches 'CE'. If validated, lock this symbol.

- **Case 2: Global Nifty Trend ($Trend_N$) is 'PE'**
 - Target Contract: **PE Option**
 - Execution Rule: Verify that the PE Option's internal trend ($Trend_O$) matches 'CE' (Opposite/Inverse Match). If validated, lock this symbol.

The signal timestamp is locked to the selected option's $T3_O$ time (t_{T3_O}).

5. EXECUTION MECHANICS

Entry Execution (|BUY|)

- **Timing:** Position entry occurs at the closing price of the **immediate next chronological 1-minute candle** after the option's $T3$ trigger.

$$t_{\text{Entry}} = t_{T3_O} + 1 \text{ minute}$$

- **Entry Price (P_{Entry}):** The closing price of the option contract at t_{Entry} .

Exit Targets & Boundaries

Once a position is entered, monitor all subsequent 1-minute candles (k) of the locked option contract where $t_k \geq t_{\text{Entry}}$:

1. Take-Profit (TP) Target (Configurable):

$$P_{\text{Target}} = P_{\text{Entry}} \times \left(1.0 + \frac{\text{Target Percent}}{100.0} \right)$$

- *Exit Trigger:* If the `High` of candle k is $\geq P_{\text{Target}}$, exit immediately at P_{Target} .

2. Absolute Time Cut-off:

- *Exit Trigger:* If the current candle timestamp $t_k \geq 13:13$ IST and the TP target has not been hit, force-liquidate the trade at the `Close` price of candle k .

6. TRANSLATION PSEUDOCODE FOR AI GENERATION

```
# DAILY INITIALIZATION LOOP
for day in trading_days:
    # 1. Resolve Nifty T1, T2, T3
    nifty_candles = get_candles("NSE:NIFTY50-INDEX", day)
    t1_n = nifty_candles[0]
    v1_n = (t1_n.high + t1_n.low + t1_n.close) / 3.0

    t2_n = None
    for c in nifty_candles[1:]:
        if c.low > v1_n:
            t2_n = c
            n_trend = "PE"
```

```

        break
    elif c.high < v1_n:
        t2_n = c
        n_trend = "CE"
        break

if not t2_n: continue # Skip day if no structural deviation

t3_n = None
for c in nifty_candles[nifty_candles.index(t2_n)+1:]:
    if n_trend == "PE" and c.high < v1_n:
        t3_n = c
        break
    elif n_trend == "CE" and c.low > v1_n:
        t3_n = c
        break

if not t3_n: continue # Skip day if triangle doesn't close

# 2. Select ATM Option contracts
atm_strike = round(t3_n.close / 50.0) * 50
target_option_type = n_trend # CE for CE, PE for PE
target_symbol = find_option_contract_name(atm_strike, target_option_type)

# 3. Resolve Option T1, T2, T3
opt_candles = get_candles(target_symbol, day)
t1_o = opt_candles[0]
v1_o = (t1_o.high + t1_o.low + t1_o.close) / 3.0

t2_o = None
for c in opt_candles[1:]:
    if c.low > v1_o:
        t2_o = c
        o_trend = "PE"
        break
    elif c.high < v1_o:
        t2_o = c
        o_trend = "CE"
        break

if not t2_o: continue

t3_o = None
for c in opt_candles[opt_candles.index(t2_o)+1:]:
    if o_trend == "PE" and c.high < v1_o:
        t3_o = c
        break
    elif o_trend == "CE" and c.low > v1_o:
        t3_o = c
        break

if not t3_o: continue

# Validate that option is in an uptrend (T2 below VWAP, meaning trend is CE)
if o_trend != "CE":
    continue # Skip if option's geometry is not configured for an uptrend

```

```
# 4. Entry & Exit Tracking
entry_index = opt_candles.index(t3_o) + 1
if entry_index >= len(opt_candles): continue

buy_candle = opt_candles[entry_index]
entry_price = buy_candle.close
target_price = round(entry_price * (1.0 + target_percent / 100.0), 2)

for c in opt_candles[entry_index:]:
    # Target hit check
    if c.high >= target_price:
        exit_price = target_price
        exit_time = c.time
        break
    # Time-stop check (13:13 IST)
    if c.time.strftime("%H:%M") >= "13:13":
        exit_price = c.close
        exit_time = c.time
        break
```