Quantitative Trading Module

Relative Moving Averages and Distributional Intraday Trading

Daniel Bloch

VinUnivesity & Paris Sorbonne

18th January 2026

Outline

- **Motivation and Framework**
  - Limitations of classical technical indicators
  - From moving averages to local price distributions
  - Addressing non-stationarity in intraday markets
- **Relative Moving Average (RMA)**
  - Sliding-window construction and anchor-based normalisation
  - Mean-reverting distance and expected profit interpretation
  - Multi-anchor RMA profiles and local regime detection
- **Distributional Representations**
  - Quantile-based local distributions
  - Fractile mapping of RMA signals
  - Alpha curves and tail-risk identification
- **Intraday Trading Strategy Design**
  - Cross-reversal and cross-revert signal logic
  - Regime-dependent entry and exit mechanisms
  - Dynamic stop-loss and profit-target control

Outline

- **Empirical Illustration and Implementation**
  - Intraday S&P 500 case study
  - Signal behaviour, PnL profiles, and regime adaptation
  - Practical considerations for live trading systems

Course Reference:
A Course On Systematic Trading With RMA
`http://ssrn.com/abstract=5278107`

Relative Moving Average (RMA) Framework

Addressing Non-Stationarity

- **Adaptive structure:** Combines moving averages with local price action to characterise short-term market dynamics.
- **Sliding-window distribution:** Normalises price data around a moving average to capture local behaviour.
- **Expansion and contraction detection:** Identifies volatility regimes associated with trading opportunities.
- **No stationarity assumptions:** Operates without fixed distributional or time-series assumptions.
- **Signal robustness:** Supports reliable entry and exit signals under evolving market conditions.

RMA: Bridging Technical Indicators and Price Action

Motivation and Integration

- **Limits of classical tools:** Traditional indicators often lag and may conflict; price action trading adds context but lacks objectivity.
- **Integrated methodology:** RMA fuses moving averages with price action to construct adaptive local price profiles.
- **Local disequilibria focus:** Tracks normalised deviations to detect expansion, contraction, and regime shifts.
- **Improved signal quality:** Enhances intraday strategies through clearer, repeatable decision rules.
- **Versatile application:** Extends naturally to pairs and correlation trading via relative distribution analysis.

RMA Implementation: Adaptive Distribution-Based Trading

- **Dynamic price profiling:** Constructs a rolling, normalised distribution around a local moving average to capture evolving market structure.
- **Expansion and contraction phases:** Uses distributional spread to identify entry (expansion) and exit (contraction) opportunities.
- **Window-based control:** Sliding window size determines temporal resolution, balancing sensitivity and noise.
- **Embedded risk management:** Supports real-time position sizing, stop placement, and profit targets using local return-to-risk estimates.

Sliding Window Construction for Time-Series Segmentation

- **Input time series:** Let $X = \{x_1, x_2, \ldots, x_n\}$ with $x_t \in \mathbb{R}$.
- **Forward-looking segmentation:** Define overlapping windows

$$v_k = (x_k, x_{k+1}, \ldots, x_{k+W-1}), \quad k = 1, \ldots, n^*,$$

  where W is the window size and $n^* = n - W + 1$.
- **Backward-looking (causal) formulation:** For financial applications,

$$v_k = (x_{k-W+1}, \ldots, x_k), \quad k = W, \ldots, n,$$

  ensuring all information is available at time k.
- **Implementation note:** Forward indexing is used for exposition; backward indexing restores causality in live trading systems.

Anchor-Based Normalisation and RMA Construction

- **Anchor-based normalisation:** For each window $v_k = (x_k, \ldots, x_{k+W-1})$, select an anchor element x_{k+i-1} with $i \in \{1, \ldots, W\}$ and define the normalised window

$$v_k^{(i)} = \frac{1}{x_{k+i-1}}\, v_k = \left(\frac{x_k}{x_{k+i-1}}, \ldots, \frac{x_{k+W-1}}{x_{k+i-1}} \right).$$

- **Sample mean of the normalised window:**

$$\overline{v}_k^{(i)} = \frac{1}{W} \sum_{l=0}^{W-1} \frac{x_{k+l}}{x_{k+i-1}} = \frac{\overline{v}_k}{x_{k+i-1}},$$

where $\overline{v}_k$ is the arithmetic mean of the raw window.

- **Relative Moving Average (RMA):**

$$\overline{v}_k^{(i,-1)} = \overline{v}_k^{(i)} - 1 = \frac{\overline{v}_k}{x_{k+i-1}} - 1, \quad i = 1, \ldots, W,$$

capturing the relative deviation from the anchor price.

Anchor-Based Normalisation and RMA Construction

- **Anchor selection:** Anchors may be chosen as the first, last ($i = W$), middle element, or via statistics (minimum, maximum, or quantiles), enabling flexible local price profiling.

Mean-Reverting Distance and Expected Profit in RMA

- **RMA as a scaled distance to the local mean:**

$$\overline{v}_k^{(i,-1)} = \frac{d_k^{(i)}}{x_{k+i-1}}, \quad i = 1, \dots, W,$$

  where

$$d_k^{(i)} = \overline{v}_k - x_{k+i-1}$$

  measures the deviation of the anchor price from the local window mean.

- **Interpretation:** The distance $d_k^{(i)}$ represents a local mean-reverting force, indicating how far the price is from its short-term equilibrium.

Mean-Reverting Distance and Expected Profit in RMA

- **Current-price signal:** For the most recent observation $(i = W)$,
$$d_k^{(W)} = \overline{v}_k - x_{k+W-1}$$
quantifies the deviation of the current price from the moving average.
- **Trading insight:** Larger absolute deviations suggest higher potential for mean reversion and, therefore, greater expected trading opportunity.

Mean-Reverting Distance and Expected Profit in RMA (Ctd.)

- **Link to a mean-reversion model:** In continuous time, price dynamics can be represented as

$$dS_t = \kappa\big(Z_t - S_t\big)\,dt + \sigma\,dW_t,$$

  where $\kappa > 0$ is the speed of mean reversion and $Z_t = \mathbb{E}_t[S_T]$ denotes the conditional expected terminal price.
- **RMA interpretation:** The RMA deviation acts as a local, data-driven proxy for $\big(Z_t - S_t\big)$, inferred directly from recent price history.
- **Expected profit:** For a position of size ω, the expected profit over the holding period satisfies

$$\mathbb{E}_t[\pi_T] = \omega\big(\mathbb{E}_t[S_T] - S_t\big).$$

- **Key insight:** Stronger RMA deviations correspond to larger expected mean-reversion gains, subject to risk and execution constraints.

Computing the RMA Indicator from Intraday Price Windows

- **Initial calibration:** The first 100 minutes of intraday data are used to select an appropriate window length W for each asset, balancing responsiveness and noise.
- **Sliding-window construction:** The price series is segmented into overlapping windows using NumPy:

```
windows = np.lib.stride_tricks.sliding_window_view(
    spot_time_series, window_shape=W
)
```

- **Window statistics:** For each window, compute the local mean to serve as the reference level for normalisation.
- **RMA normalisation:** Normalised deviations (RMA values) are obtained as

```
means = windows.mean(axis=1)
rma = windows / means[:, None] - 1.0
```

RMA Anchoring Strategies and Feature Extraction

- **Anchor-based normalisation:** RMA profiles can be computed by anchoring each window to specific reference points, highlighting different aspects of local price structure.
- **Common anchor choices:**

```
first_anchor = windows[:, 0]          # first element
mid_anchor   = windows[:, W // 2]     # middle element
last_anchor  = windows[:, -1]         # last element
min_anchor   = windows.min(axis=1)    # minimum
max_anchor   = windows.max(axis=1)    # maximum
```

- **Interpretation:**
  - First / last anchors emphasise trend-following or reversal behaviour.
  - Min / max anchors highlight local extremes and potential mean reversion.
  - Middle anchors provide symmetry around the local centre.
- **Output:** The resulting anchored RMA curves summarise local price distributions and form inputs for intraday signal generation.

Visualising RMA Normalisations Across Anchors

Anchor-dependent RMA profiles

Each dotted curve represents an RMA normalisation computed with respect to a different anchor within the same sliding window.

- **Last element (W):** lime green — highlights current price deviation.
- **Middle element ($\lfloor W/2 \rfloor$):** light pink — symmetric reference around the local centre.
- **First element (1):** gold — emphasises path-dependence from window entry.
- **Minimum ($\min$):** dodger blue — captures upside mean-reversion from local lows.
- **Maximum ($\max$):** aqua blue — captures downside mean-reversion from local highs.

Visualising RMA Normalisations Across Anchors

Interpretation: Comparing anchor-based profiles reveals regime changes, asymmetries, and expansion/contraction dynamics in local price behaviour.

Local Distribution Modelling via Quantiles

- **Quantile-based representation:** Each sliding window is transformed into a small set of quantiles, providing a compact summary of the local price distribution.
- **Quantile definition:** For a level $q \in \mathscr{Q}$ (e.g. $\mathscr{Q} = \{0.1, 0.25, 0.5, 0.75, 0.9\}$), the q-quantile is the value below which a fraction q of observations in the window fall.
- **Numerical implementation:**

```
quantiles = np.quantile(windows, q, axis=1)
```

- **Motivation:** Quantiles are robust to outliers and naturally capture skewness, spread, and tail behaviour of intraday price dynamics.

Quantile Dictionaries and Local ECDF Approximation

- **Structured storage:** Quantile estimates are stored in a dictionary for efficient access and downstream analysis.
- **Dictionary layout:**
  - **Keys:** quantile levels $q \in \mathscr{Q}$
  - **Values:** 1D arrays of length n_{windows}, where each entry corresponds to the q-quantile of a window
- **Local ECDF approximation:** Collectively, the quantiles provide a time-evolving approximation of the empirical cumulative distribution function (ECDF) of prices.
- **Use in trading:** This representation supports regime detection, tail-risk assessment, and distribution-aware intraday signal construction.

Normalising Quantiles for Scale-Invariant Comparison

- **Issue with raw quantiles:** Unadjusted quantiles depend on the absolute price level and are therefore not comparable across windows or assets.
- **Scale normalisation:** Dividing each quantile by the corresponding window mean removes level effects and ensures cross-window comparability.
- **Zero-centred representation:** Subtracting 1 recentres the normalised quantiles such that:
  - 0 indicates equality with the window mean,
  - values > 0 indicate positions above the mean,
  - values < 0 indicate positions below the mean.
- **Shape over scale:** The transformation emphasises distributional characteristics (spread, skewness) independently of price level.
- **Feature engineering benefit:** Produces stable, scale-free features suitable for statistical analysis and machine learning models.

Normalising Quantiles for Feature Robustness

- **Relative-to-mean transformation:** Each quantile is expressed as a deviation from the local window mean, yielding a zero-centred representation.
- **RMA-consistent normalisation:** The transformation mirrors the RMA definition (Equation (**??**)), ensuring methodological coherence across indicators.
- **Purpose:** Removes scale effects and improves comparability across time windows and assets.
- **Modelling advantage:** Highlights distributional asymmetry and tail behaviour, improving feature quality for statistical and machine learning models.
- **Implementation:**

```
for q, quantile_values in quantiles.items():
    normalized_by_quantiles[q] = quantile_values / means - 1.0
```

Fractile Mapping of RMA Curves

- **Fractile concept:** Similar to quantiles, fractiles map a value to its relative position within a distribution, expressed on the $[0, 1]$ scale.
- **Why fractiles?** Fractile values are scale-free and invariant to monotonic transformations, enabling robust comparison across time and assets.
- **Application to RMA:** Each RMA curve $(w, 1, mid, \min, \max)$ is transformed into a corresponding fractile series

$$(f_w, f_1, f_{mid}, f_{\min}, f_{\max}),$$

representing its percentile rank within the local window distribution.

Fractile Mapping of RMA Curves

- **Interpretation:**
  - Values near 0: extreme lower tail (oversold regime)
  - Values near 1: extreme upper tail (overbought regime)
  - Values near 0.5: neutral positioning
- **Outcome:** Fractile-mapped RMA curves provide a unified probabilistic representation of local price dynamics.

Fractile Mapping of RMA Curves (Continued)

- **Objective:** Convert each RMA curve value into a fractile representing its relative position within the local, window-specific distribution.
- **Implementation overview:**
  - Select the RMA-derived value (e.g. *w*, *mid*, 1, min, max)
  - Compare it against the window's normalised return distribution
  - Compute its empirical fractile
  - Store the result as a new feature
- **Python implementation:**

```
new_columns = {}
columns_to_fractile = ['1', 'mid', 'w', 'max', 'min']

for column_name in columns_to_fractile:
    fractile_column_name = f'f_{column_name}'
    column_values = normalized_df[column_name].values
    new_columns[fractile_column_name] = compute_fractile(
        column_values, normalized_returns
```

Fractile Mapping of RMA Curves (Continued)

- **Visual reference:** Fractile curves are illustrated in Figure (3), highlighting regime shifts and extremal behaviour.

Alpha Curves from RMA Quantiles

- **Objective:** Quantify the distributional position of each RMA curve over time by comparing it to local quantile bounds.
- **Alpha interpretation:** Lower alpha values represent the probability level in $[0, 1]$ below which the RMA curve lies within the windowed distribution.
- **Why alpha curves?** They provide a probabilistic measure of extremeness, supporting regime detection, signal filtering, and risk-aware trade decisions.
- **Computation (per RMA curve):**

Alpha Curves from RMA Quantiles

```
normalized_df = compute_lower_higher_quantiles(
    normalized_df, quantiles, "w"
)
normalized_df = compute_lower_higher_quantiles(
    normalized_df, quantiles, "1"
)
normalized_df = compute_lower_higher_quantiles(
    normalized_df, quantiles, "mid"
)
```

- **Result:** Alpha curves track how often RMA signals reside in the tails of their local distributions.

Alpha Curves from RMA Quantiles (Continued)

- **Generated output:**
  - *w_lower_alphas*: Raw lower-alpha values (quantile levels) associated with the *w* RMA curve.
  - *w_lower_alphas_means*: Smoothed alpha series obtained via a rolling average over *n* minutes, reducing microstructure noise.
- **Purpose of smoothing:**
  - Enhances regime persistence
  - Improves interpretability of signal strength
  - Stabilises downstream trading rules
- **Visual interpretation (Figure 2):**
  - Green curve: *w_lower_alphas_means*
  - Magenta curve: *mid_lower_alphas_means*
  - Orange curve: 1*_lower_alphas_means*

Dynamic Stop-Loss and Profit-Target Mechanism

- **Objective:** Control intraday risk by dynamically adjusting trading halts based on both realised profits and accumulated losses.
- **Core principle:** Risk tolerance tightens as the trading session progresses, protecting gains while limiting late-session drawdowns.
- **Profit-based protection:**
  - Track the maximum realised net PnL during the session.
  - Allow normal fluctuations around profitable trends.
  - Stop trading only if the drawdown from the peak exceeds a time-dependent threshold.

Dynamic Stop-Loss and Profit-Target Mechanism

- **Loss-based protection:**
  - Begin with a relatively loose absolute loss limit (e.g. 0.7%).
  - Gradually tighten this limit toward zero as end-of-day approaches.
  - Enforce immediate stop once the loss threshold is breached.
- **Outcome:** A smooth, adaptive alternative to static stop-loss rules that balances profit preservation and controlled risk exposure.

Dynamic Stop-Loss and Profit-Target Mechanism (Continued)

- **Design advantages:**
  - Avoids premature exits during sustained profitable trends.
  - Implements a trailing-stop logic without relying on rigid price levels.
  - Gradually increases risk aversion as the trading session matures.
  - Naturally adapts to both volatile and low-volatility regimes.
- **Decision logic:** Trading is halted whenever either a drawdown-based condition or a loss-based condition is triggered.
- **Python implementation:**

Dynamic Stop-Loss and Profit-Target Mechanism (Continued)

```
def check_stop_trading(
    net_pnl,
    max_net_pnl,
    drawdown_limit,
    loss_limit
):
    drawdown = max(max_net_pnl - net_pnl, 0.0)
    condition_drawdown = drawdown >= drawdown_limit
    condition_loss = net_pnl <= -loss_limit
    return condition_drawdown or condition_loss
```

- **Practical implication:** This mechanism enforces discipline at the portfolio level while allowing individual trades to evolve naturally.

Visual Summary of RMA and Fractile Indicators

- **Context:** This slide summarises the RMA-based indicators and derived trading signals for the **S&P 500 Index (GSPC)** on **February 3, 2025**.
- **Model configuration:**
  - Sliding window length: $\textit{window\_size} = 106$
  - Visualisation starts at $t = 120$ to account for initialisation and smoothing
- **Displayed figures:**
  - **Spot price and moving average:** Figure (1)
  - **RMA curves (anchor-based normalisations):** Figure (2)
  - **RMA fractile curves:** Figure (3)
  - **PnL with extremum-revert trigger:** Figure (4)
  - **PnL with full-distribution crossover (no reversion):** Figure (5)
- **Key takeaway:** The combined RMA, fractile, and alpha representations provide a coherent, distribution-aware framework for intraday regime detection and trade management.

Spot Prices and Moving Average

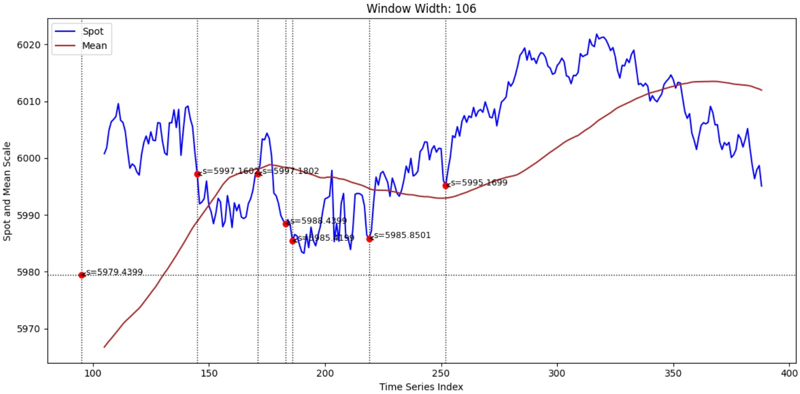

Figure 1: Prices and rolling MA for GSPC on the 03/02/25.

RMA Curves

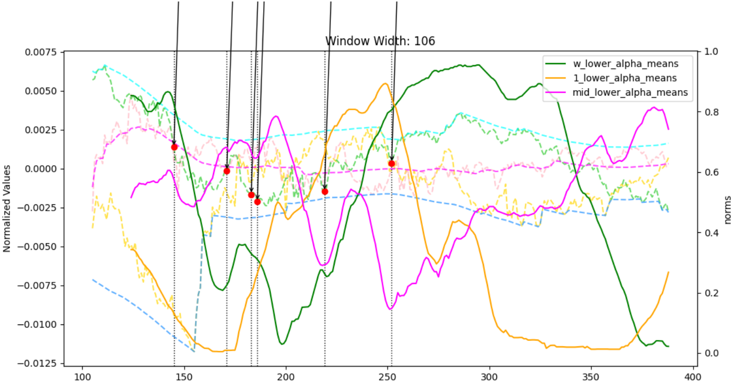

Figure 2: Rolling RMA for GSPC on the 03/02/25.

RMA Fractile Curves

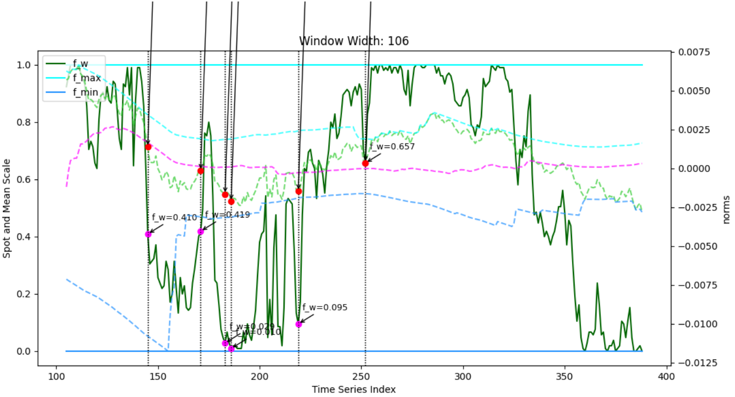

Figure 3: Rolling RMA Fractiles for GSPC on the 03/02/25.

PnL with Extremum-Revert-Trigger

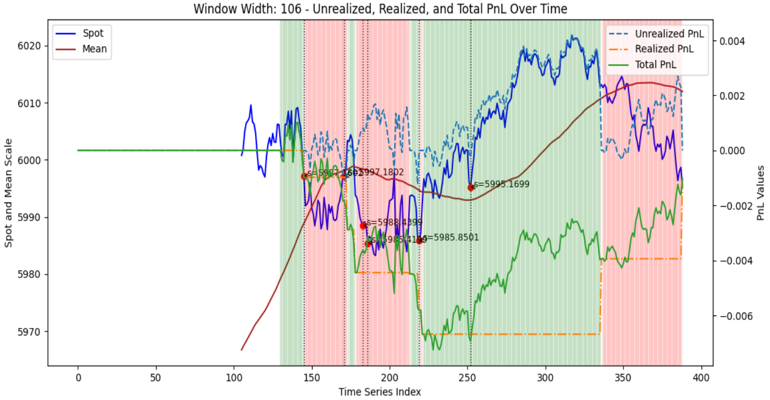

Figure 4: Prices and rolling PnL with Extremum-Revert-Trigger for GSPC on the 03/02/25.

PnL with Full-Distribution-Crossover

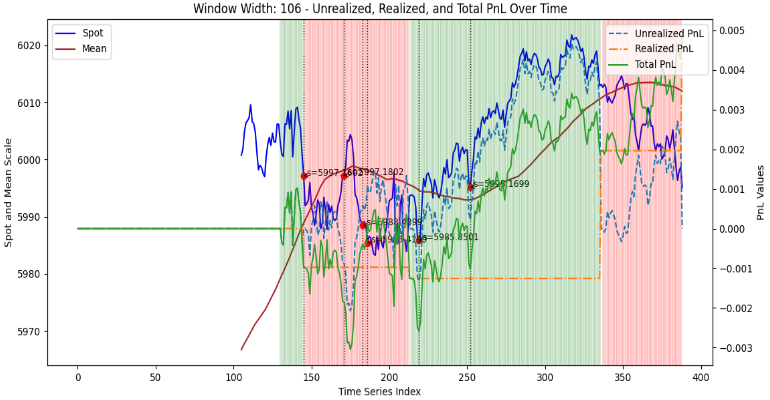

Figure 5: Prices and rolling PnL with Full-Distribution-Crossover for GSPC on the 03/02/25.

Riding Distributional Oscillations with the RMA-Based Trading Strategy

- **Overview:**
  - The Relative Moving Average (RMA) measures price deviations from a dynamic equilibrium, approximated by a sliding moving average (SMA).
  - The SMA is computed over a calibrated rolling window designed to capture the asset's intrinsic cycles of expansion and contraction.
- **Strategy principle:**
  - Intraday prices oscillate around the SMA, forming a time-varying distribution of normalised deviations.
  - The RMA represents these deviations through three reference levels: $(w, mid, 1)$.
  - The normalised price w fluctuates between the evolving minimum and maximum of the distribution, generating adaptive entry and exit opportunities.

Riding Distributional Oscillations with the RMA-Based Trading Strategy

- **Trading logic:**
  - **Go long:** when price transitions upward from the lower extremum toward the upper bound of the RMA distribution.
  - **Go short:** when price reverts downward from the upper extremum toward the lower bound.
- **Objective:** Exploit the cyclical and distributional structure of normalised deviations to adaptively trade both mean-reverting and momentum phases within intraday windows.

Detecting Cross-Reversal Signals within the RMA Distribution

- **Purpose:** Identify potential trading opportunities by monitoring both the position and directional movement of the normalised price w within its dynamic RMA distribution.
- **Key definitions:**
  - **Long-Cross-Reverse:** A downward movement of w, originating near the upper extremum of the distribution and progressing toward the lower extremum.
  - **Short-Cross-Reverse:** An upward movement of w, originating near the lower extremum of the distribution and progressing toward the upper extremum.

Detecting Cross-Reversal Signals within the RMA Distribution (Cont.)

- **Signal conditions:**
  - **Long-Cross-Reverse** triggers when:
    1. w lies above the median (upper region), or
    2. w lies below the median but remains above the minimum (lower half).
  - **Short-Cross-Reverse** triggers when:
    1. w lies below the median (lower region), or
    2. w lies above the median but remains below the maximum (upper half).
- **Interpretation:** These cross-reversal patterns exploit the symmetry of the RMA distribution to detect potential trend inflections and mean-reversion phases.

Exit Conditions in the RMA Trading Strategy

1. **Full Distribution Crossover Exit**
   - Assumes a complete traversal of the RMA distribution (maximum $\leftrightarrow$ minimum).
   - Exit for a short-cross-reverse coincides with the entry condition for a long-cross-reverse, and vice versa.
   - Enables symmetric, rule-based switching between long and short positions.
   - Best suited to clean, full-range distributional moves.
2. **Extremum Revert Trigger**
   - Designed for partial traversals of the distribution.
   - If the normalised price w starts a crossover but reverses before completion:
     - Exit the current position,
     - Immediately reverse exposure:

$$\text{long-cross-reverse} \to \text{long-cross-revert},$$

$$\text{short-cross-reverse} \to \text{short-cross-revert}.$$

   - Ensures continuous exposure while adapting to directional reversals.

Adaptive crossover exit based on market regimes

- **Empirical Observation:**
  - As illustrated in Figure (**??**), contraction regimes exhibit tightly bounded oscillations of the normalised price w.
  - In this phase, w evolves within a quasi-cylindrical region of nearly constant diameter.
- **Distributional Dynamics:**
  - During contraction, w is forced to traverse the full RMA distribution:
    $$\text{minimum} \rightarrow \text{maximum} \quad \text{or} \quad \text{maximum} \rightarrow \text{minimum}.$$
  - Breakouts typically occur only during transitions into expansion regimes.
- **Regime Identification:**
  - Relative Extremum Ratios are used in real time to classify the market state as:
    - Contraction,
    - Expansion, or
    - Transition.

Adaptive crossover exit

Objective: Dynamically select the most appropriate exit mechanism based on the prevailing market regime.

1. **Expansion or Transition Regimes**
   - Price movements exhibit partial traversals and frequent reversals.
   - Apply the *Extremum Revert Trigger*.
   - Enables early exits and rapid position reversals when momentum fades.
2. **Contraction Regimes**
   - Normalised price w oscillates tightly between distributional bounds.
   - Apply the *Full Distribution Crossover Exit*.
   - Assumes clean, complete traversals from minimum to maximum (and vice versa).

Adaptive crossover exit

Key Benefit

This adaptive exit framework aligns the strategy with regime-dependent volatility, improving robustness, timing precision, and responsiveness across market conditions.

Implementation of the strategy: Key conditions

- **Long-Cross-Revert Entry**
  - Triggered when a long-cross reversal fails to reach the minimum.
  - The normalised price w reverses upward and moves back toward the maximum.
  - Implemented via the flag:

```
selected_cross_up_for_revert_event
```

- **Short-Cross-Revert Entry**
  - Triggered when a short-cross reversal fails to reach the maximum.
  - The normalised price w reverses downward and moves back toward the minimum.
  - Implemented via the flag:

```
selected_cross_down_for_revert_event
```

Implementation of the strategy: Key conditions

- **Crossing Quantile Configuration**
  - Quantile thresholds are updated dynamically to detect downward crossings.
  - Controlled through:

```
set_crossing_quantiles_down
```

Basic algorithms: Cross-reverse entry conditions

Algorithm 1 Entry conditions for long-cross-reverse and short-cross-reverse

Require: Current time t_k, normalised price levels $(w, mid, 1)$

Initialise entry flags:

  $set\_long\_cross\_reverse \leftarrow$ False

  $set\_short\_cross\_reverse \leftarrow$ False

if $w\_q\_lower\_long\_cross\_reverse$ **then**

  Enter **short** position

  Set $t_{entry} \leftarrow t_k$

else if $w\_q\_lower\_short\_cross\_reverse$ **then**

  Enter **long** position

  Set $t_{entry} \leftarrow t_k$

else

  No action

end if

Basic algorithms: Cross-reverse exit conditions

Algorithm 2 Exit conditions for long-cross-reverse and short-cross-reverse

```
Require: Current time t_k, normalised price levels (w, mid, 1)
Initialise exit flags:
    long_cross_exit ← False
    short_cross_exit ← False
if long_cross_exit then
    Exit short position
    Set t_exit ← t_k
else if short_cross_exit then
    Exit long position
    Set t_exit ← t_k
else
    Maintain current position
end if
```

Basic algorithms: Cross-revert entry conditions

Algorithm 3 Entry conditions for long-cross-revert and short-cross-revert

Require: Current time t_k, normalised price levels $(w, mid, 1)$

Initialise entry flags:
    $set\_long\_cross\_revert \leftarrow$ False
    $set\_short\_cross\_revert \leftarrow$ False
if $w\_q\_lower\_short\_cross\_revert$ **then**
    Enter **short** position
    Set $t_{entry} \leftarrow t_k$
else if $w\_q\_lower\_long\_cross\_revert$ **then**
    Enter **long** position
    Set $t_{entry} \leftarrow t_k$
else
    No trade signal
end if

Basic algorithms: Cross-revert exit conditions

Algorithm 4 Exit conditions for long-cross-revert and short-cross-revert

Require: Current time t_k, normalised price levels $(w, mid, 1)$

```
if short_cross_revert_exit then
    Exit short position
    Set t_exit ← t_k
else if long_cross_revert_exit then
    Exit long position
    Set t_exit ← t_k
else
    Maintain current position
end if
```

The end

Thank You !