ODE Time Series Transform: A Volume-Based Indicator Using SQL and the Cosine Function

Introduction

In high-frequency financial markets, understanding the dynamics of price formation and trade volume is a persistent challenge. Classical statistical indicators often fall short in capturing the temporal causality and geometric structures underlying tick-level price action. To address this, we explore a novel approach that bridges signal processing, dynamical systems, and market microstructure by embedding trade volume into the structure of a **harmonic oscillator**.

This article introduces a **volume-based indicator** grounded in the principles of ordinary differential equations (ODEs), where the relative change in trade volume is used to define an angular frequency. This angular representation allows for the projection of volume flow into a circular phase space via trigonometric functions. The resulting ODE-transformed path reveals oscillatory behavior akin to that of a damped harmonic oscillator.

This transform serves two purposes:

- It embeds trade activity in a physically interpretable phase space.
- It enables the **exploration** of whether price dynamics can be approximated by cumulative angular projections such as $\cos(\Theta)$ and $\sin(\Theta)$, revealing underlying oscillatory behaviors in market volume

By implementing the entire framework in SQL over real-time trade data, this work demonstrates that advanced analytical tools—traditionally reserved for continuous-time models—can be adapted for practical, real-time market monitoring

Most importantly, this study reveals for the first time a systematic, dynamic, and empirically verifiable correlation between trade volume and market price at the tick level. By projecting volume dynamics into a cumulative angular space and tracing their evolution as geometric paths, we uncover a latent structure where price emerges as a directional projection of volume flow. Under specific geometric regimes of volume flow, price dynamics exhibit conditional coupling to volume geometry, revealing a previously hidden microstructural dependency that vanishes at aggregated timescales. This transforms the classical view of market randomness into one of constrained, trajectory-governed motion. The result is not merely an indicator, but a discovery: **price is not independent of volume—it is embedded within its geometry**.

Variables Definition

Let us consider a high-frequency time series $v_i > 0$, defined as a sequence of discrete-time observations taken at successive (possibly irregular) time intervals $\Delta\tau_i$:

$$v = (v_0, v_1, \ldots, v_i, \ldots, v_N), \quad v_i = v(t_i), \quad \Delta v_i = v(t_{i+1}) - v(t_i), \quad \Delta\tau_i = t_{i+1} - t_i.$$

The series is fully determined by the initial value v_0 and its discrete derivative:

$$\dot{v}_i = \frac{\Delta v_i}{\Delta \tau_i}.$$

We introduce the instantaneous angular frequency w_i, a quantity of dimension T^{-1}. One natural candidate is the scaled relative variation:

$$w_i = 2\pi \cdot \frac{1}{v_i} \cdot \frac{\Delta v_i}{\Delta \tau_i}.$$

This frequency allows us to define a cumulative angular argument Θ_i:

$$\Theta_i = \sum_{j=0}^{i} w_j \Delta\tau_j.$$

Using this, we construct circular projections:

$$x_i = \cos(\Theta_i), \quad y_i = \sin(\Theta_i).$$

Embedding in a Harmonic Oscillator

If z_i satisfies a linear relation of the form:

$$z_i = \alpha x_i + \beta y_i$$

for constants $\alpha, \beta \in \mathbb{R}$, then z_i is a discrete solution of the damped harmonic oscillator equation:

$$\frac{\Delta^2 z_i}{\Delta \tau_i^2} + 2\zeta_i \Omega_i \frac{\Delta z_i}{\Delta \tau_i} + w_i^2 z_i = 0$$

where the damping frequency and ratio are defined by:

$$\Omega_i = -2\pi \cdot \frac{1}{w_i} \cdot \frac{\Delta w_i}{\Delta \tau_i}, \quad w_i = 2\pi \cdot \frac{1}{v_i} \cdot \frac{\Delta v_i}{\Delta \tau_i}, \quad \zeta_i = \frac{1}{4\pi}.$$

We now define the ODE-transformed path $\widetilde{Y}_i$ as:

$$\widetilde{Y}_i = \sum_{j=0}^{i} y_j$$

This summation encodes the cumulative harmonic response of the time series and defines a nonlinear transformation of the original signal v_i.

Nondimensionalized ODE Formulation

To express the system in terms of the angular variable Θ_i, we perform the following change of variable:

$$\frac{\Delta y_i(\Theta_i)}{\Delta \tau_i} = \frac{\Delta y_i(\Theta_i)}{\Delta \Theta_i} \cdot \frac{\Delta \Theta_i}{\Delta \tau_i}.$$

Substituting into the original differential equation yields the nondimensionalized form:

$$\frac{\Delta^2 y_i(\Theta_i)}{\Delta \Theta_i^2} + y_i = 0,$$

whose general solutions are again:

$$x_i = \cos(\Theta_i), \quad y_i = \sin(\Theta_i).$$

Summary

Time series analysis can therefore be conducted in either of two conjugate spaces:

- The temporal domain indexed by t_i,
- The angular domain indexed by Θ_i,

revealing geometric and causal structures not evident under conventional methods. The ODE transform path $\widetilde{Y}_i$ provides a novel embedding that bridges discrete stochastic signals and continuous dynamical systems.

A Volume-Based Indicator Using SQL and the Cosine Function

Building on the ODE Time Series Transform, we now demonstrate a practical application of this framework in the context of market data analysis. In particular, we introduce a novel **volume-based indicator** derived from high-frequency trade volume using the cosine of a time-evolving angular argument.

This work is based on a real-time Binance data stream for the XRPUSDT trading pair, using tick-level granularity.

This section shows how to:

- Define the angular frequency w_i where v_i represents the trading volume at time t_i
- Construct the angular argument Θ_i via SQL queries
- Compute $\cos(\Theta_i)$ directly in SQL
- Use the cumulative sum of $\cos(\Theta_i)$ as a signal for volume-driven market dynamics
- Compute the conjugate component $\sin(\Theta_i)$ for a phase-complete representation of market volume flow

Remark: This indicator can be applied in two distinct use cases:

1. *Tick Data Level:* Each v_i corresponds to the exact trading volume of an individual trade. If multiple trades share the same timestamp (i.e., a singularity in $\Delta\tau_i = 0$), the last non-zero time interval should be reused to ensure stability in frequency computation.
2. *Sampled Level:* The time series is resampled (e.g., per second or per minute), and v_i represents the aggregated volume over each sampling interval.

In this article, we will focus exclusively on the tick-level data use case, as it carries more granular and informative signals about market microstructure dynamics than its sampled counterpart.

The entire indicator is built using SQL functions supported by QuestDB, allowing real-time streaming analysis of market volume without external preprocessing or scripting. It can be directly visualized using time-series dashboards (e.g., Grafana) for interactive analysis.

SQL Code for Volume-Based Indicator(Sampled Level)

The following SQL code implements the ODE-based volume indicator using QuestDB:

```
WITH ResampledTrades AS (
    SELECT
    DATE_TRUNC('second', timestamp) AS resampled_timestamp,
    symbol,
    AVG(price) AS average_price,
    SUM(quantity) AS total_quantity
    FROM
    binance_trades
    WHERE
```

```sql
symbol = 'XRPUSDT'
GROUP BY
resampled_timestamp,
symbol
),

PreviousValueCalc AS (
SELECT
resampled_timestamp AS timestamp,
symbol,
average_price AS price,
total_quantity AS quantity,
FIRST_VALUE(total_quantity) OVER (
PARTITION BY symbol
ORDER BY resampled_timestamp
ROWS BETWEEN 1 PRECEDING AND CURRENT ROW
) AS previous_value
FROM
ResampledTrades
),

FrequencyCalc AS (
SELECT
timestamp,
symbol,
price,
quantity,
previous_value,
(2 * PI() * (quantity - previous_value) / previous_value) AS frequency
FROM
PreviousValueCalc
WHERE
previous_value IS NOT NULL AND previous_value > 0
),

ThetaCalc AS (
SELECT
symbol,
timestamp,
price,
quantity,
frequency,
SUM(frequency) OVER (
PARTITION BY symbol
ORDER BY timestamp
ROWS UNBOUNDED PRECEDING
```

```sql
) AS theta
FROM
FrequencyCalc
)

SELECT
symbol,
timestamp,
price,
quantity,
frequency,
theta,
COS(theta) AS cos_theta,
SUM(COS(theta)) OVER (
PARTITION BY symbol
ORDER BY timestamp
ROWS UNBOUNDED PRECEDING
) AS cumulative_cos_theta,
SIN(theta) AS sin_theta,
SUM(SIN(theta)) OVER (
PARTITION BY symbol
ORDER BY timestamp
ROWS UNBOUNDED PRECEDING
) AS cumulative_sin_theta
FROM
ThetaCalc
ORDER BY
timestamp;
```

Volume-Based Computation Analysis The query uses trading volume (referred to as `quantity` in the query) as a basis for several key calculations:

1. **Total Volume per Second:**
   In the `ResampledTrades` CTE, the query calculates `SUM(quantity) AS total_quantity`. This gives us the total trading volume for each second.

2. **Volume Change:**
   In the `PreviousValueCalc` CTE, the query retrieves the previous second's volume using a window function. This sets up the ability to calculate the change in volume from one second to the next.

3. **Frequency Calculation:**
   In the `FrequencyCalc` CTE, the query calculates what it calls `frequency` using the formula:

   ```
   (2 * PI() * (quantity - previous_value) / previous_value) AS frequency
   ```

   This is essentially calculating the relative change in volume, scaled by

2π. In financial terms, this could be interpreted as a measure of volume volatility or the rate of change in trading activity.

4. **Trigonometric Transformations:**
   Finally, the query applies trigonometric functions to these cumulative measures. This transformation into circular functions could be used to identify cycles or patterns in trading volume behavior.

SQL Code for Volume-Based Indicator(Tick Data Level)

The following SQL code implements the ODE-based volume indicator for tick-level data using QuestDB:

```
WITH base_data AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    LAG(quantity) OVER (
      ORDER BY
        timestamp,
        trade_id
    ) AS prev_quantity,
    LAG(timestamp) OVER (
      ORDER BY
        timestamp,
        trade_id
    ) AS prev_timestamp
  FROM
    binance_trades
  WHERE
    symbol = 'XRPUSDT'
  ORDER BY
    timestamp,
    trade_id
),
with_delta AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    prev_quantity,
```

```sql
    timestamp - prev_timestamp AS delta_tau_raw
  FROM
    base_data
),
with_valid_deltas AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    prev_quantity,
    delta_tau_raw,
    CASE WHEN delta_tau_raw > 0 THEN delta_tau_raw ELSE NULL END AS valid_delta
  FROM
    with_delta
),
with_safe_delta AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    prev_quantity,
    delta_tau_raw,
    LAST_VALUE(valid_delta) IGNORE NULLS OVER (
      ORDER BY
        timestamp,
        trade_id ROWS BETWEEN UNBOUNDED PRECEDING
        AND CURRENT ROW
    ) AS associated_delta
  FROM
    with_valid_deltas
),
with_frequency AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    prev_quantity,
    delta_tau_raw,
    associated_delta,
    CASE WHEN delta_tau_raw = 0 THEN associated_delta ELSE delta_tau_raw END AS delta_tau_sa
```

```sql
    CASE WHEN delta_tau_raw = 0 THEN 'SINGULARITY_FIXED' WHEN delta_tau_raw IS NULL THEN 'FI
    CASE WHEN prev_quantity > 0
    AND (
      CASE WHEN delta_tau_raw = 0 THEN associated_delta ELSE delta_tau_raw END
    ) > 0 THEN 2 * PI() * (quantity - prev_quantity) / prev_quantity / (
      CASE WHEN delta_tau_raw = 0 THEN associated_delta ELSE delta_tau_raw END
    ) ELSE NULL END AS frequency
  FROM
    with_safe_delta
),
with_theta AS (
  SELECT
    timestamp,
    symbol,
    trade_id,
    price,
    quantity,
    prev_quantity,
    delta_tau_raw,
    associated_delta,
    delta_tau_safe,
    status,
    frequency,
    SUM(frequency * delta_tau_safe) OVER (
      ORDER BY
        timestamp,
        trade_id ROWS UNBOUNDED PRECEDING
    ) AS theta
  FROM
    with_frequency
  WHERE
    frequency IS NOT NULL
)
SELECT
  timestamp,
  symbol,
  trade_id,
  price,
  quantity,
  prev_quantity,
  delta_tau_raw,
  associated_delta,
  delta_tau_safe,
  status,
  frequency,
  theta,
```

```
  COS(theta) AS cos_theta,
  SIN(theta) AS sin_theta,
  SUM(
    COS(theta)
  ) OVER (
    ORDER BY
      timestamp,
      trade_id ROWS UNBOUNDED PRECEDING
  ) AS cumulative_cos_theta,
  SUM(
    SIN(theta)
  ) OVER (
    ORDER BY
      timestamp,
      trade_id ROWS UNBOUNDED PRECEDING
  ) AS cumulative_sin_theta
FROM
  with_theta
ORDER BY
  timestamp,
  trade_id;
```

Visualization

To evaluate the effectiveness of the proposed volume-based indicator, we visualize its behavior using real-time Binance trade data for the **XRPUSDT** pair. **Figure 1** displays the price evolution, serving as a reference for market movements. **Figure 2** shows the cumulative cosine projection $\sum \cos(\Theta)$, and **Figure 3** presents the cumulative sine projection $\sum \sin(\Theta)$, both computed directly from high-frequency trade volumes. These angular components are then plotted in a phase portrait in **Figure 4**, where cumulative sine is plotted against cumulative cosine. This portrait reveals hidden geometric structures and possible oscillatory regimes in the flow of market volume, enabling the exploration of whether price dynamics can be approximated by these cumulative angular projections.

Globally, the structure of the phase portrait emerges from the accumulation of directional steps governed by angular frequency, where each direction is shaped by local volume change and time between trades. When volume variation vanishes ($\Delta v_i = 0$), the angular frequency w_i drops to zero, freezing the angular coordinate θ_i, but not resetting it. This results in directional persistence — segments of the trajectory where motion continues along a fixed direction, forming straight radial paths in the phase space.

Locally, at the tick level, we see bursts of coordinated movement followed by periods of geometric stillness.

Global View

Fig.1

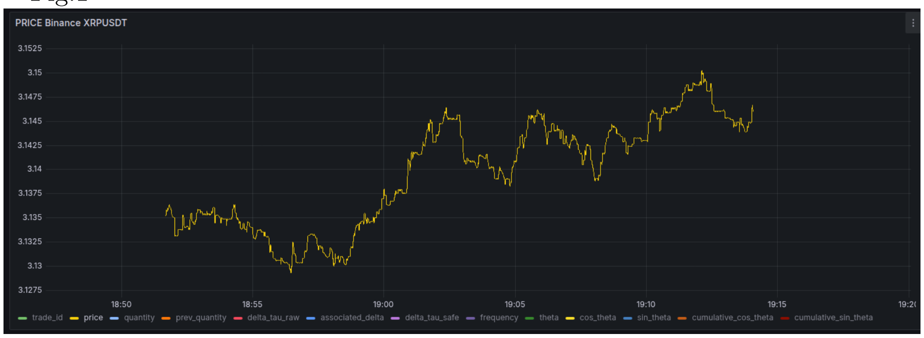
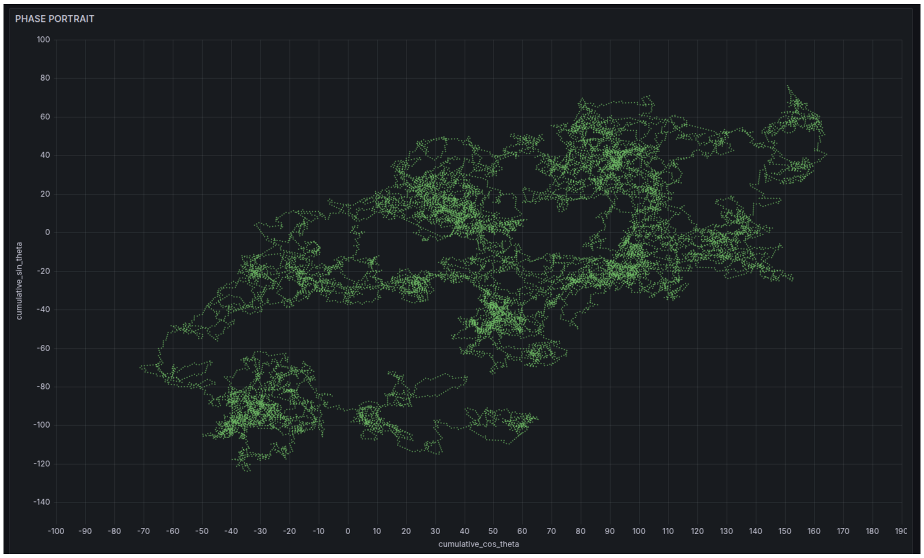


Fig.2

Fig.3

Fig.4



Local View

Fig.1

Fig.2

Fig.3

Interpretation

Global View: ODE-Based Cumulative Embedding The evolution of the cumulative cosine and sine projections shown in Figures 2 and 3 is entirely governed by the choice of the angular frequency function:

$$w_i = 2\pi \cdot \frac{1}{v_i} \cdot \frac{\Delta v_i}{\Delta \tau_i}.$$

This formulation treats relative volume change as the driver of angular velocity in a rotating phase space. It is one possible mapping among infinitely many, and each choice of w_i defines a distinct geometric embedding of the market's microstructure.

By systematically exploring alternatives—such as including volume acceleration, normalized ratios, or non-linear combinations—the ODE framework becomes a **general engine** for discovering which aspects of volume flow contain latent predictive structure.

The phase portrait in Figure 4 reveals that price evolution traces structured trajectories within this volume-driven space, suggesting that the embedding does more than reflect noise—it encodes interpretable geometric regimes.

Local View: Tick-Level Coupling and Microstructural Synchronization When zooming in at the tick scale (see Figures 1–3), we uncover a striking phenomenon: changes in price align closely with changes in the cumulative projections. This alignment is **not visible** at aggregated timescales, highlighting the temporal fragility of this dependency.

This leads to a major discovery:

> Under specific geometric regimes of volume flow, price dynamics exhibit conditional coupling to volume geometry, revealing a previously hidden microstructural dependency that vanishes at aggregated timescales.

This coupling implies that price does not evolve independently but **synchronizes with the phase flow induced by volume rotation**. In this sense, price becomes interpretable as a projection **onto the directional manifold defined by cumulative phase rotation**, opening the door to a fundamentally new class of volume-based models.

Related Applications

The same singularity-handling approach used in the ODE-based volume indicator—where zero inter-trade intervals are managed by propagating the last known valid time delta—is also integrated into the SKA real-time learning system. This ensures robust and artifact-free entropy accumulation, particularly during high-frequency trade bursts. As a result, the identification of market regime transitions in XRPUSDT using SKA is not only structurally aligned with the volume geometry framework, but also inherits the same microstructural integrity at the tick level.