PC3 Spectral Analysis: A PCA and Fourier Modeling Approach for Yield Curve Curvature

Author: Danny Watkins
University of Arizona
MATH485: Mathematical Modeling
Professor Ildar Gabitov

May 11, 2025

Abstract

The yield curve embeds complex information about macroeconomic expectations, liquidity conditions, and market structure. Decomposing and trading its dynamic behavior requires precise modeling of both systematic factors and transient distortions. This paper presents a mathematical modeling framework that combines Principal Component Analysis (PCA) and Discrete Fourier Transform (DFT) techniques to isolate, smooth, and exploit curvature dislocations for systematic analysis. We apply rigorous statistical testing, mean-reversion analysis, and Fourier-based noise filtering to extract actionable signals from PC3, the curvature component. Our findings confirm the stationarity and short half-life of PC3 dislocations, making them well-suited for mean-reversion strategies. Unlike PC1 and PC2, which are highly sensitive to monetary policy shifts and volatility regime changes, PC3 remains relatively stable across macro environments. This makes it a cleaner and more persistent signal for identifying temporary yield curve distortions driven by technical or structural pressures.

We further show that traditional butterfly proxies fail to track PC3, exhibiting weak correlation with curvature dislocations even after smoothing. To address this, we construct a dynamic PCA-weighted curvature index that evolves with market structure and more accurately captures real-time changes in curvature. This index provides a scalable, interpretable foundation for curvature-based strategies. As one example, we demonstrate how it can be paired with a simple two-layer signal system to identify tradeable dislocations — but its broader value lies in offering a clean, statistically grounded representation of yield curve shape changes, suitable for deployment in analytics platforms, risk systems, or macro overlay models.

Contents

1 Introduction

Modeling financial time series requires understanding both global macro trends and local structural deviations. In the fixed income space, the U.S. Treasury yield curve offers a multidimensional signal reflecting market expectations for monetary policy, inflation, and economic growth across maturities. In addition to these macroeconomic drivers, the curve also reflects short-term technical distortions caused by liquidity imbalances, dealer positioning, and investor flows. Principal Component Analysis (PCA) consistently reveals that three orthogonal components — level, slope, and curvature — explain the majority of yield curve variation.

While PC1 and PC2 correspond to parallel shifts and steepening/flattening of the curve, PC3 captures systematic curvature — specifically, the relative movement of intermediate maturities (the "belly") versus the short and long ends. This structure is evident in PC3's eigenvector, which typically assigns opposite-signed loadings to the belly and the wings, isolating curvature shocks while remaining orthogonal to level and slope. Unlike PC1 and PC2, which are closely tied to volatility regimes and monetary policy changes, PC3 is relatively stable across macro environments. This makes it a consistent and interpretable signal for detecting yield curve dislocations that are technical rather than fundamental in nature.

This paper develops a modeling framework that extracts and smooths the third principal component (PC3) using rolling PCA and Fourier-based spectral filtering. We validate the curvature signal's statistical properties — including stationarity, low dispersion, and fast mean reversion — and demonstrate that traditional butterfly trades correlate poorly with these dynamics. In response, we construct a PCA-weighted curvature index that evolves with the yield curve's structure and directly reflects changes in curvature. While this index can be paired with simple signal logic to identify tradeable dislocations, its broader purpose is to provide a stable, interpretable foundation for curvature-aware analytics, overlays, and strategy design.

2 Mathematical Framework

2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a foundational tool in multivariate time series analysis, used to reduce dimensionality and identify dominant modes of variation. In the context of the yield curve, PCA helps extract uncorrelated factors that summarize the joint movement of rates across maturities.

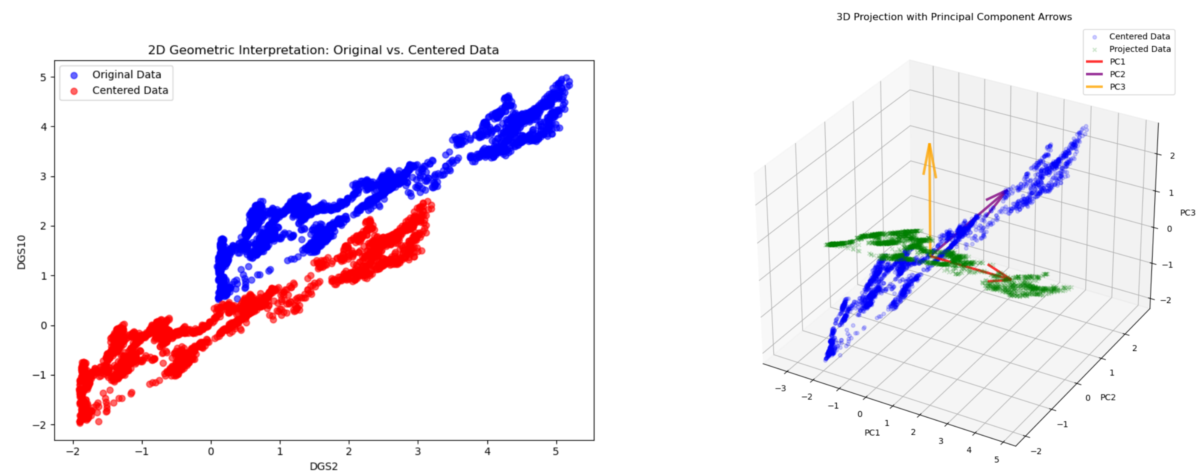
Given a data matrix $X \in \mathbb{R}^{T \times M}$ consisting of T observations across M maturities, PCA seeks an orthonormal basis that diagonalizes the covariance matrix:

$$\Sigma = \mathbb{E}\left[(X-\mu)(X-\mu)^T\right], \tag{1}$$

where μ is the mean vector across observations. The eigenvalue problem to solve is:

$$\Sigma v_i = \lambda_i v_i, \tag{2}$$

where v_i are eigenvectors (principal components) and λ_i are the associated eigenvalues, representing the variance explained by each component.



(a) 2D Geometric Interpretation **(b)** 3D PCA Projection with Yield Curve Axes

Figure 1: PCA Geometric Intuition: Yield curve as a point cloud and its principal directions of variation.

Sorting eigenvalues in decreasing order yields components ordered by the proportion of variance captured. In yield curve analysis, these typically align with intuitive curve movements:

- PC1 reflects level shifts (parallel movements across all maturities).
- PC2 reflects slope changes (short-term vs. long-term rates).
- PC3 captures curvature (relative movements of the belly versus the wings).

This curvature structure is visible in PC3's eigenvector, which tends to assign opposite-signed weights to intermediate maturities compared to short and long ends — isolating convexity changes while remaining orthogonal to level and slope.

Each principal component score at time t is computed by projecting the centered observation onto the eigenvector:

$$\mathrm{PC}_{i,t} = (X_t - \mu) \cdot v_i. \tag{3}$$

PCA decorrelates yield movements and reduces the problem to three orthogonal modes, enabling cleaner signal extraction and interpretation.

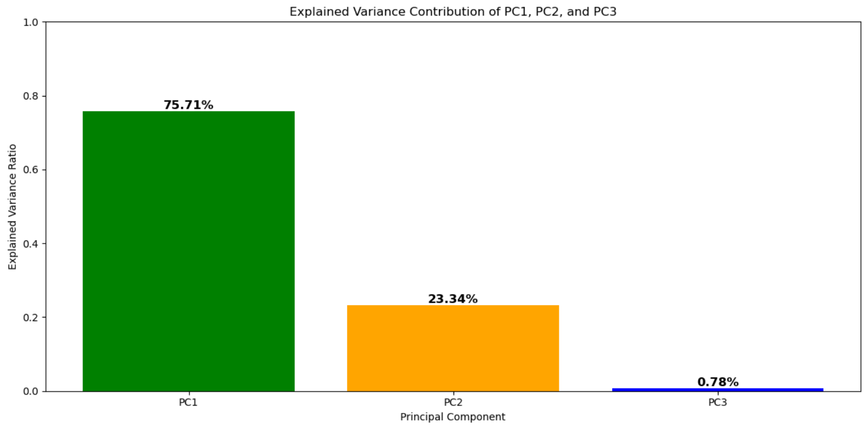


Figure 2: Explained variance contribution of each principal component. PC1 captures the majority of the yield curve's movement, while PC2 and PC3 capture slope and curvature patterns respectively. This summary helps validate the PCA decomposition used in constructing trading signals.

2.2 Illustrative PCA Example Using Yield Data

To make the PCA process more concrete, consider a simplified example using synthetic yield data for two maturities: 2-year and 10-year Treasuries, observed over five time points. The data matrix X is:

$$X = \begin{bmatrix} 3.2 & 4.5 \\ 3.5 & 4.6 \\ 3.3 & 4.7 \\ 3.1 & 4.8 \\ 3.4 & 4.9 \end{bmatrix}$$

Each row corresponds to a daily observation, and each column corresponds to a yield maturity.

We begin by subtracting the mean from each column to obtain the centered matrix X^*, then compute the sample covariance matrix:

$$\Sigma = \begin{bmatrix} 0.008 & 0.006 \\ 0.006 & 0.009 \end{bmatrix}$$

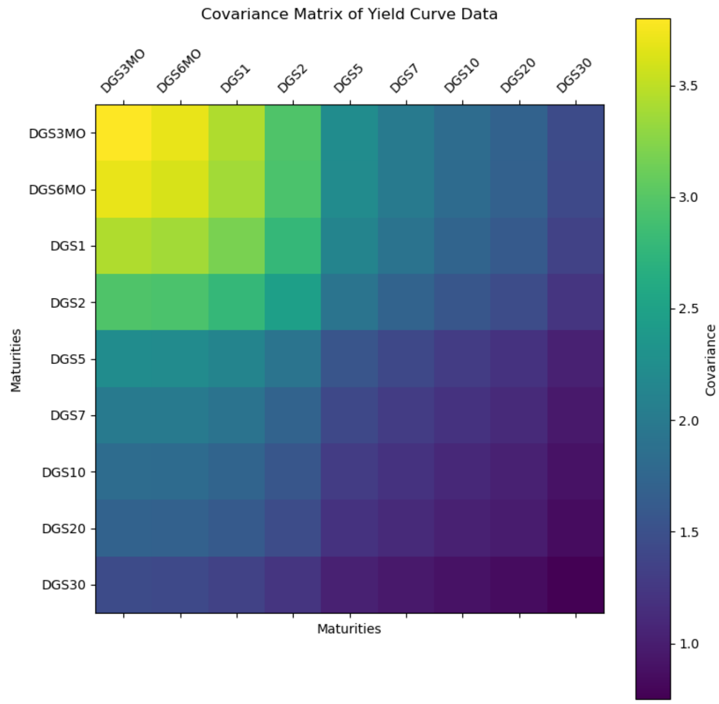


Figure 3: Covariance matrix of the yield curve time series. PCA acts on this matrix to identify orthogonal directions of maximal variance, enabling decomposition into interpretable components like level, slope, and curvature.

Solving the characteristic equation $\det(\Sigma - \lambda I) = 0$, we find the eigenvalues:

$$\lambda_1 = 0.014, \quad \lambda_2 = 0.003$$

These indicate that the first principal component explains approximately 82% of the total variance, while the second explains 18%. The associated eigenvectors are:

$$\text{PC1: } \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}, \quad \text{PC2: } \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}$$

This means PC1 reflects parallel movements in both maturities — a level shift. PC2 reflects tilting behavior: when the 2Y yield moves up, the 10Y moves down, and vice versa, consistent with a slope change.

This example illustrates how PCA transforms observed yield changes into interpretable curve shape shifts, and how the eigenstructure provides both the factor loadings and the shape intuition used throughout this paper.

2.3 Discrete Fourier Transform (DFT)

Financial time series often contain a mixture of persistent structural signals and short-term noise. In the context of PCA, the raw PC3 score series often reflects this: long-horizon curvature oscillations are overlaid with high-frequency market microstructure effects. To isolate the underlying curvature behavior, we apply the Discrete Fourier Transform (DFT) to the time series and retain only the dominant frequency components.

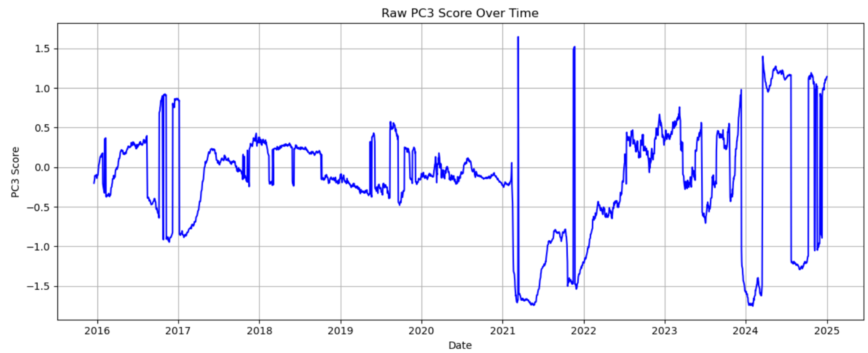


Figure 4: Raw time series of the third principal component (PC3), which captures curvature in the yield curve. This unfiltered signal contains meaningful dislocation patterns but also exhibits significant noise, motivating the use of smoothing techniques such as Fourier filtering.

For a discrete signal x_n of length N, the DFT transforms the data into the frequency domain:

$$X(f) = \sum_{n=0}^{N-1} x_n \, e^{-2\pi i f n/N}, \tag{4}$$

where $X(f)$ is the complex Fourier coefficient for frequency index f. The inverse DFT reconstructs the signal from its frequency components:

$$x_n = \frac{1}{N} \sum_{f=0}^{N-1} X(f) \, e^{2\pi i f n/N}. \tag{5}$$

The spectral energy at each frequency is given by $|X(f)|^2$. By sorting frequencies in descending order of energy and retaining only those that cumulatively explain a fixed threshold (typically 85–90%), we filter out noise while preserving the dominant structural cycles in the PC3 signal.

This filtering process smooths the raw PC3 series, enhancing its signal-to-noise ratio and making it more suitable for interpretation, statistical validation, and signal generation. The result is a clean representation of curvature oscillations that captures meaningful deviations in the yield curve shape without being distorted by short-term noise.

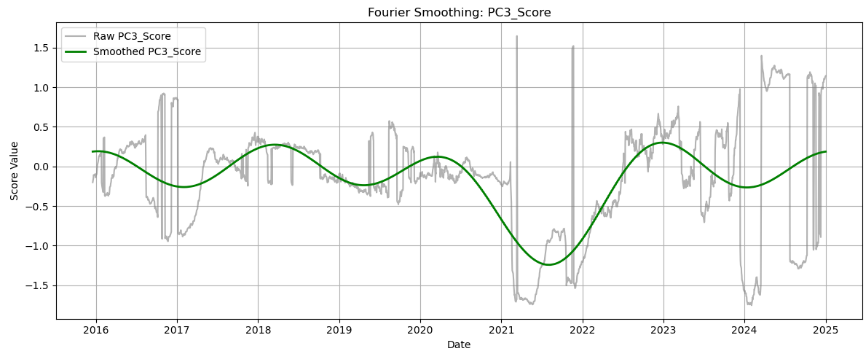


Figure 5: Smoothed PC3 time series using Fourier filtering. This cleaned signal isolates curvature dislocations in the yield curve by removing high-frequency noise, making it a suitable anchor for constructing a curvature trading index.

2.4 Fourier Filter Design: Hybrid Variance-Elbow Cutoff

To retain the structural curvature signal while removing short-term noise, we developed a custom Fourier filtering strategy that combines statistical rigor with shape-based interpretability. The key objective was to determine an optimal cutoff frequency — the point beyond which higher-frequency components would be discarded during inverse transformation.

We evaluated two filtering criteria:

- **85% Variance Retention:** Sort frequency components by energy contribution and retain the minimum set of components explaining at least 85% of the total spectral energy. This ensures that most of the structure-driving variance is preserved.
- **Elbow Point Heuristic:** Identify the point at which the cumulative spectral energy curve flattens — i.e., where the marginal energy gain of adding new frequencies drops below a small threshold (typically $\nabla E(f) < 0.01$). This provides a shape-based signal-noise boundary.

Rather than choosing one method arbitrarily, we implemented a weighted hybrid:

$$\text{Weighted Cutoff} = 0.8 \cdot f_{85\%} + 0.2 \cdot f_{\text{elbow}}$$

This hybrid approach balances statistical coverage with geometric intuition, and avoids overly aggressive filtering that might distort the signal's curvature dynamics.

Result: The smoothed PC3 signal produced by this weighting scheme offers a clean, interpretable trend while preserving the underlying curvature cycles visible in raw PC3. This filtered signal is used throughout the paper for stationarity testing, signal generation, and butterfly comparison.

2.5 Illustrative Fourier Example on PC3 Data

To illustrate the Fourier smoothing process, consider a sample calculation from the original PC3 signal. Suppose we select $x[0] = -0.505035$ as the first data point from a PC3 time series of length $N = 2500$.

We begin by computing the Discrete Fourier Transform (DFT) at frequency $k = 1$:

$$X[1] = \sum_{n=0}^{N-1} x[n] \cdot e^{-2\pi i \cdot 1 \cdot n/N}$$

Using Euler's identity, this becomes a combination of cosine and sine components:

$$X[1] = \mathrm{Re}(X[1]) + j \cdot \mathrm{Im}(X[1]) = -83.904 + j \cdot 104.444$$

The magnitude of this frequency component is:

$$|X[1]| = \sqrt{(-83.904)^2 + (104.444)^2} \approx 133.97$$

and the phase is:

$$\theta = \tan^{-1}\left(\frac{104.444}{-83.904}\right) \approx 2.25 \text{ radians}$$

The spectral power at this frequency is:

$$\text{Power}[1] = |X[1]|^2 \approx 17948.54$$

This accounts for approximately 2.26% of the signal's total energy. Since this frequency falls within the hybrid cutoff threshold established in the previous section, it would be retained during filtering.

To reconstruct the signal's contribution at time $t = 0$ from this frequency, we apply the inverse DFT:

$$x[0]^{(1)} = \frac{1}{N} \cdot |X[1]| \cdot \cos\left(\frac{2\pi \cdot 0 \cdot 1}{N} + \theta\right) \approx -0.0356$$

Interpretation: This means the $k = 1$ frequency component contributes roughly -0.0356 to the PC3 signal at $t = 0$. Repeating this process for all retained frequencies and applying the inverse DFT yields the smoothed PC3 signal used throughout the paper for curvature tracking and dislocation detection.

2.6 Half-Life of Mean Reversion

To evaluate how quickly yield curve components revert after a distortion, we model each principal component score using a first-order autoregressive process (AR(1)):

$$x_t = \phi x_{t-1} + \epsilon_t, \tag{6}$$

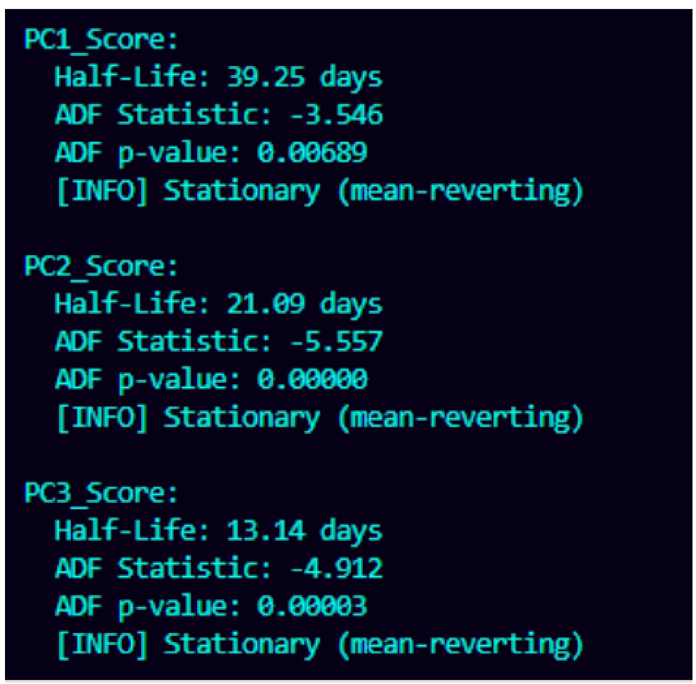
where ϵ_t is white noise. The half-life measures how long it takes a deviation to decay by 50%, and is given by:

$$t_{1/2} = \frac{\ln(0.5)}{\ln(\phi)}. \tag{7}$$

Shorter half-lives imply stronger mean reversion, which is favorable for short- to medium-term trading signals.

Applying this method to the PCA outputs, we find that PC3 has the fastest mean reversion:

- PC1: Half-Life = 39.25 days, ADF Statistic = -3.546, p = 0.00869
- PC2: Half-Life = 21.09 days, ADF Statistic = -5.557, p = 0.00000
- PC3: Half-Life = 13.14 days, ADF Statistic = -4.912, p = 0.00003


```
PC1_Score:
  Half-Life: 39.25 days
  ADF Statistic: -3.546
  ADF p-value: 0.00689
  [INFO] Stationary (mean-reverting)

PC2_Score:
  Half-Life: 21.09 days
  ADF Statistic: -5.557
  ADF p-value: 0.00000
  [INFO] Stationary (mean-reverting)

PC3_Score:
  Half-Life: 13.14 days
  ADF Statistic: -4.912
  ADF p-value: 0.00003
  [INFO] Stationary (mean-reverting)
```

Figure 6: Comparison of half-lives across PC1, PC2, and PC3. PC3 reverts fastest.

All three components pass the Augmented Dickey-Fuller (ADF) test for stationarity at high confidence, but PC3 clearly stands out: it not only reverts the fastest, but also maintains its curvature signal with relatively low exposure to volatility shocks or macroeconomic regime shifts.

This is further confirmed by correlation tests showing that PC3 movements are largely uncorrelated with VIX or policy regimes, unlike PC1 and PC2. Together, these properties make PC3 the most tradeable and interpretable signal for capturing temporary curve distortions.

3 Modeling Framework and System Implementation

This project is built as a modular Python system that implements the full curvature signal extraction pipeline from start to finish. It includes raw data acquisition, rolling PCA decomposition, Fourier-based noise filtering, signal generation, and performance visualization. The system is designed to support experimentation, extensibility, and reproducibility.

3.1 Project Structure

The project is organized with clear separation between source code, data, configuration, and output. Key components include:

- `main.py` — Orchestrates the end-to-end pipeline by executing all major components via subprocess. It runs the full modeling pipeline in order using a predefined list of scripts.
- `config.json` — Defines user-controlled parameters such as PCA window size, smoothing thresholds, and filepaths.
- `data/` — Contains both raw inputs (yield curves, VIX, macro data) and processed outputs (Fourier scores, PCA results, trading signals).
- `visuals/` — Stores outputs from each module, grouped into subdirectories:
  - `comparisons/` — butterfly vs. PC3
  - `fourier/` — spectral plots and smoothing
  - `macro_sensitivity/` — VIX and regime analysis
  - `pca_analysis/` — explained variance
  - `pca_geometry/` — geometric and projection visuals
  - `signals/` — trading signal overlays
  - `smoothing/` — PC3 filtering results
- `src/` — Core scripts that define each module of the modeling system, called sequentially via `main.py`.

Figure 7: Modular code structure of the curvature modeling system implemented in Python.

3.2 Pipeline Execution (main.py)

The file `main.py` runs the entire curvature modeling workflow using a series of operating system calls. It executes the following scripts in order:

```
python src/data\_loader.py
python src/rolling\_pca\_analysis.py
python src/fourier\_transform.py
python src/smooth\_all\_pc\_scores.py
python src/pc\_macro\_sensitivity\_analysis.py
python src/pc3\_mean\_reversion\_analysis.py
python src/pc3\_signal\_trading.py
python src/verify\_pc3\_vs\_butterfly.py
python src/pca\_geometry\_visuals.py
```

This design enables quick, reproducible execution and supports batch runs for experimentation.

3.3 Yield Curve and Macro Data Collection (data_loader.py)

This script uses the `fredapi` package to pull historical U.S. Treasury yields across multiple maturities, the VIX index, and Federal Funds Rate changes. It processes the yield curve into a forward-filled DataFrame indexed by date and saves it to `data/yield_curve_data.csv`. It also generates a covariance matrix plot for visual inspection.

3.4 Rolling PCA Decomposition (rolling_pca_analysis.py)

Using `sklearn`'s PCA and a 250-day window, this module computes a rolling decomposition of the yield curve into PC1, PC2, and PC3. The eigenvector loadings and time series of component scores are saved to `rolling_pca_results.csv`, and explained variance is visualized over time.

3.5 Fourier Smoothing of Curvature (fourier_transform.py)

This script applies a Discrete Fourier Transform (DFT) to the PC3 score series, calculates the cumulative power spectrum, and identifies a frequency cutoff using a hybrid of 85% variance and an elbow-point heuristic. It then applies the inverse transform to generate a smoothed signal, saved to `fourier_filtered_signals.csv`.

3.6 Full PCA Signal Smoothing (smooth_all_pc_scores.py)

This module generalizes the Fourier smoothing process to PC1 and PC2. It saves `fourier_smoothed_pc_sco` for downstream correlation and volatility analysis.

3.7 Statistical Validation and Reversion Behavior (pc3_mean_reversion_analy

This script tests the raw PC scores for stationarity and mean-reversion. It reports the AR(1) half-life, ADF test statistic and p-value, zero-crossing counts, and mean absolute deviation. It overlays normalized VIX to highlight macro regime sensitivity and generates a PC–VIX correlation heatmap.

3.8 Macro Sensitivity Regime Correlation (pc_macro_sensitivity_analysis.py)

To study macro linkage, this script segments data by VIX regime (high vs. low) and computes Pearson correlations between VIX and each PC. Results are visualized with seaborn bar plots.

3.9 Butterfly Proxy Comparison (verify_pc3_vs_butterfly.py)

This module compares the smoothed PC3 score with both a traditional 2s5s10s butterfly and a PCA-weighted synthetic trade using contemporaneous eigenvector weights. Both correlation coefficients are printed and plotted for validation.

3.10 Signal Logic and Trade Simulation (pc3_signal_trading.py)

This script implements a two-layer trade entry signal: a 3-month Z-score threshold combined with a slope reversal in the smoothed PC3. It generates labeled trades (BUY/SELL/EXIT) and stores the output in `pc3_signal_trades.csv`. It also plots the signal overlay on the PC3 series.

3.11 PCA Visualization Utilities (pca_geometry_visuals.py)

For geometric intuition, this module creates both 2D and 3D projections of the yield curve before and after centering, and overlays principal component directions. These plots clarify how PCA interprets curvature structure spatially.

3.12 Python Libraries Used

The project relies on several key libraries:

- `numpy`, `pandas`, `scipy`, `matplotlib`, `seaborn` for data analysis and visualization
- `scikit-learn` for PCA and preprocessing
- `statsmodels` for ADF testing
- `fredapi` for macroeconomic data

All dependencies are listed in `requirements.txt` for easy installation.

This modular system design allows for scalable experimentation and targeted debugging at each stage of the yield curve curvature signal pipeline.

4 Empirical Results and Signal Behavior

This section presents statistical findings from the PC3 signal construction process, including stationarity testing, mean-reversion diagnostics, Fourier filtering effects, and an evaluation of traditional butterfly spreads as curvature proxies.

4.1 Stationarity and Half-Life Estimation

The Augmented Dickey-Fuller (ADF) test applied to the raw PC3 score series $\{x_t\}$ rejects the null hypothesis of a unit root (p-value $= 4.0 \times 10^{-5}$), confirming that the curvature signal is statistically stationary.

Fitting an AR(1) process,

$$x_t = \phi x_{t-1} + \epsilon_t,$$

yields an estimated coefficient $\hat{\phi} \approx 0.946$. This corresponds to a half-life:

$$t_{1/2} = \frac{\ln(0.5)}{\ln(\hat{\phi})} \approx 12.8 \text{ days}.$$

For comparison, PC1 and PC2 show half-lives of approximately 38.4 and 20.6 days, respectively, suggesting that curvature dislocations revert much more quickly than level or slope movements.

4.2 Tightness Around the Mean and Oscillation Frequency

The mean absolute deviation (MAD) and annual zero-crossing frequency were calculated for each principal component:

- PC1: MAD = 2.25, Zero Crossings 5/year
- PC2: MAD = 0.69, Zero Crossings 11/year
- PC3: MAD = 0.51, Zero Crossings 9/year

PC3's tighter dispersion and regular reversion support its suitability for short-term strategies seeking mean-reverting behavior.

4.3 Impact of Fourier Smoothing

To improve signal quality, the raw PC3 series $\{x_t\}$ was smoothed using a hybrid Fourier filter retaining 85% cumulative spectral energy and guided by an elbow-point heuristic.

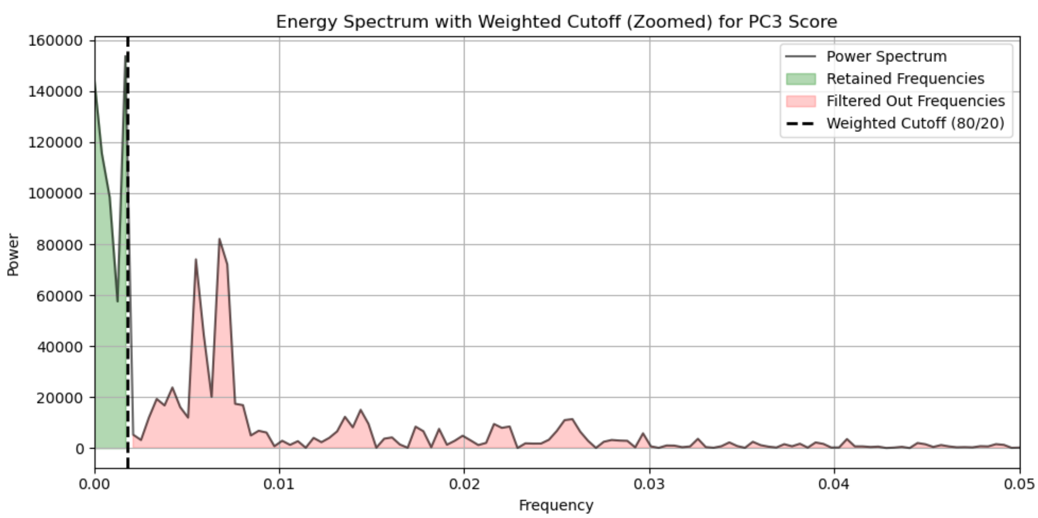


Figure 8: Energy spectrum of the PC3 score. The green region shows retained low-frequency components that passed the weighted cutoff threshold (80% variance + 20% elbow). The red region highlights filtered high-frequency components considered noise. The dashed line marks the final cutoff used in signal reconstruction.

The resulting signal $\{\tilde{x}_t\}$ retained its stationarity (ADF p-value ¡ 0.01), half-life (15 days), low MAD, and stable zero-crossing rate. Short-term noise was removed, revealing a smoother curvature trend driven by structural market dynamics.

4.4 Failure of Standard Butterfly Proxies

To evaluate whether traditional instruments still track curvature, we compared the 2s5s10s butterfly spread with PC3 scores. Pearson correlation between the daily butterfly spread and PC3 was found to be:

$$\text{Correlation} = -0.2186,$$

suggesting an inverse and weak relationship.

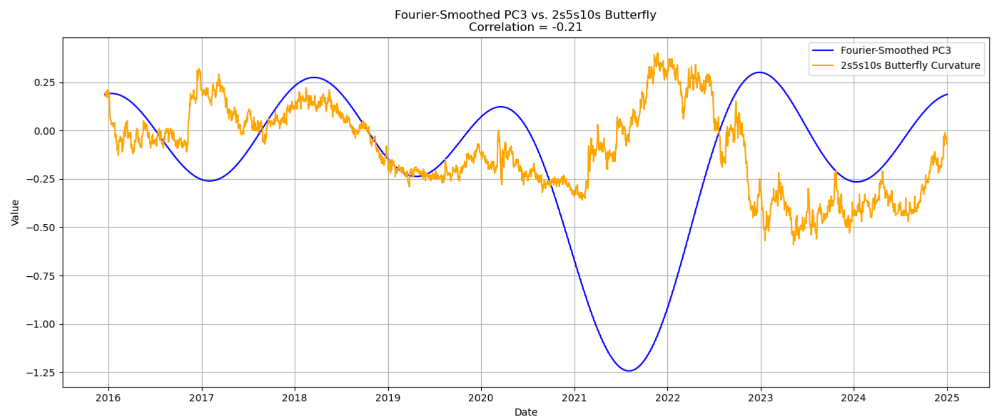


Figure 9: Smoothed PC3 score (curvature component) compared against the 2s5s10s butterfly spread over time. Despite common use in trading, the butterfly spread fails to capture the dynamic curvature tracked by PC3 — evidenced by the weak correlation.

This breakdown underscores the structural limitations of static butterfly constructions. While often used as heuristic proxies for curvature, their fixed weights do not align with the actual curvature dynamics extracted from PCA. The weak correlation with PC3 indicates that traditional butterflies may no longer provide reliable or consistent exposure to yield curve curvature, especially as market behavior evolves.

5 From Butterfly Spreads to a Curvature Index

5.1 How Traders Traditionally Express Curvature

In fixed income markets, curvature trades are often implemented using bond combinations known as butterfly spreads. A common structure is the 2s5s10s butterfly, which combines:

- +1 unit of 2-year Treasury
- -2 units of 5-year Treasury
- +1 unit of 10-year Treasury

This setup is designed to isolate changes in the ”belly” of the yield curve (the 5-year point), and is intended to be neutral to parallel rate shifts (level) and to linear steepening (slope). The payoff relies on curvature changes — that is, bumps or dips in the middle of the curve.

The spread is calculated as:

$$\text{Butterfly Spread} = y\_5Y - \frac{1}{2}(y\_2Y + y\_10Y) \tag{8}$$

This measures how far the 5Y yield deviates from the average of the 2Y and 10Y — a simple estimate of curvature at the 5-year point.

5.2 The Problem: Butterfly Proxies Are Noisy and Inaccurate

While simple and intuitive, butterfly trades are noisy approximations. Their static weights do not reflect the actual statistical structure of curvature dynamics. To test their validity, we computed the Pearson correlation between the daily change in the 2s5s10s butterfly and the smoothed PC3 score — our mathematically defined curvature signal.

Result: Correlation = -0.21 — weak, and even slightly inverse.

This suggests butterfly spreads no longer align well with PC3. Their performance as curvature proxies is degraded by changing curve dynamics and the rigid `+1 / -2 / +1` weight structure.

5.3 A Better Approach: PCA-Aligned Curvature Exposure

To directly capture true curvature, we construct a synthetic index using PCA loadings themselves. Each day, we apply PCA to a rolling 250-day window of yield changes and extract the third eigenvector (PC3), which defines curvature.

We then assign weights to maturities according to their PC3 loading values. These weights dynamically reflect the shape of curvature dislocations in that moment. The synthetic curvature signal is computed as a linear combination:

$$\text{PC3 Index} * t = \sum *i = 1^n w\_i^{(t)} \cdot y\_i, t \tag{9}$$

where $w_i^{(t)}$ is the PC3 loading for maturity i at time t, and $y_{i,t}$ is the corresponding yield.

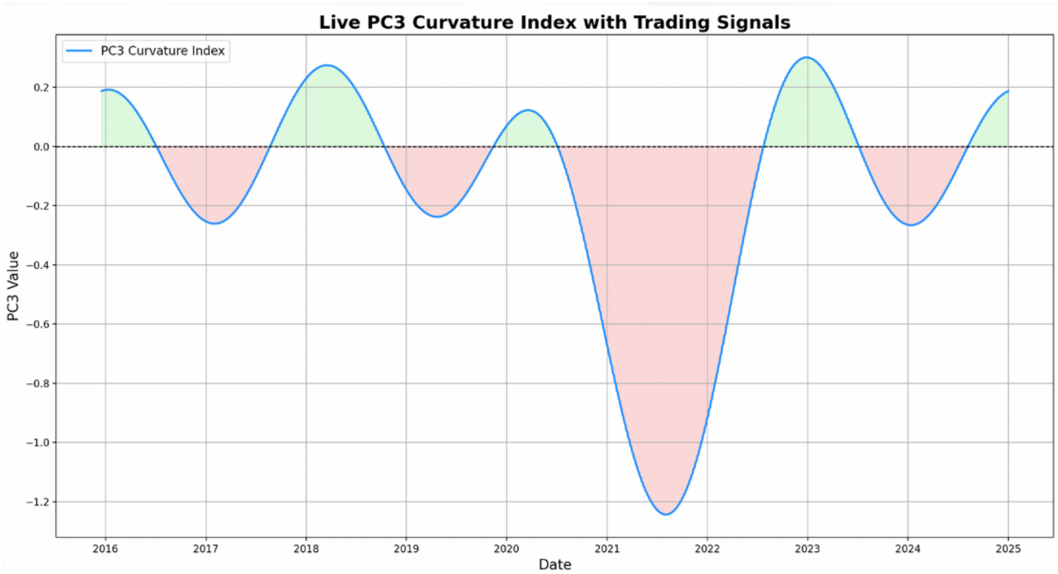


Figure 10: Live PC3 Curvature Index constructed using PCA-weighted yield components. Green areas represent upward-bending curve regimes (long curvature bias), and red areas represent downward curvature phases. This real-time index captures evolving yield curve shape more accurately than fixed-weight proxies.

5.4 Example: Constructing a PCA-Weighted Curvature Signal

Consider the full set of maturities used in our PCA analysis, with corresponding yields and PC3 loadings on a specific day:

Maturity	Yield (%)	PC3 Loading
3M	5.20	0.03
6M	5.14	0.05
1Y	4.98	0.12
2Y	4.65	−0.35
3Y	4.44	−0.42
5Y	4.17	0.71
7Y	4.05	−0.20
10Y	3.98	−0.14
20Y	4.12	−0.01
30Y	4.20	0.01

We compute the PC3 index by multiplying each yield by its corresponding loading and summing the results. Letting y_i be the yield and w_i the PC3 loading for each maturity i:

$$\text{PC3 Index}_t = \sum_i w_i \cdot y_i$$

Substituting in the values:

$$
\begin{aligned}
\text{PC3 Index}_t &= (0.03)(5.20) + (0.05)(5.14) + (0.12)(4.98) \\
&\quad + (-0.35)(4.65) + (-0.42)(4.44) + (0.71)(4.17) \\
&\quad + (-0.20)(4.05) + (-0.14)(3.98) + (-0.01)(4.12) + (0.01)(4.20) \\
&= 0.156 + 0.257 + 0.598 - 1.628 - 1.865 + 2.961 \\
&\quad - 0.810 - 0.557 - 0.041 + 0.042 \\
&= \boxed{-0.887}
\end{aligned}
$$

This value represents the curvature signal for that day — a data-driven index of how the yield curve bends, constructed from the empirical PC3 shape.

5.5 Why This Index Matters

The PCA-aligned index offers a cleaner, more accurate curvature measure. It:

- Reflects the actual statistical structure of curve dislocations.
- Dynamically adapts as curve behavior shifts.
- Is orthogonal to level and slope by construction.
- Can serve as a reference for tracking, alerts, or overlay strategies.

Rather than betting on the shape of the yield curve through fixed-weight trades, we can measure curvature directly. This index transforms PC3 from an abstract eigenvector into a real, observable signal of curve shape that can evolve with the market.

5.6 Using the Curvature Signal: Macro Insight and Trading Logic

The curvature of the yield curve — captured cleanly through PC3 — reflects the market's expectations about medium-term economic direction. Unlike short maturities (which respond primarily to Fed policy) or long maturities (which reflect structural growth and inflation expectations), the belly of the curve acts as a pivot between the two.

Monitoring the evolution of PC3 allows us to detect when investors are pricing in inflection points — shifts from tightening to easing cycles, or from slowdown to recovery. When PC3 is rising sharply from deeply negative levels, it may indicate growing confidence in a rebound. When PC3 is falling from elevated levels, it may suggest that optimism is fading or that tightening is expected to compress medium-term growth.

Trading Application: One way to systematically use this curvature signal is through a two-layer dislocation-and-confirmation rule:

1. **Dislocation Condition:** A trade is considered only when the raw PC3 score moves beyond a threshold $k \cdot \sigma$ from its mean — indicating a statistically significant deviation in curve shape.

2. **Slope Confirmation:** The signal must also exhibit a directional reversal. That is, the first derivative (slope) of the smoothed PC3 signal must be pointing back toward equilibrium. This confirms mean-reversion and avoids chasing trending noise.

Example Rules:

- If raw PC3 $< -1.5 \cdot \sigma$ and smoothed PC3 slope turns upward $\rightarrow$ long curvature trade.
- If raw PC3 $> +1.5 \cdot \sigma$ and smoothed PC3 slope turns downward $\rightarrow$ short curvature trade.

This structure ensures trades are only entered during extreme dislocations that are beginning to correct — exploiting the mean-reverting nature of PC3 while filtering out noise.

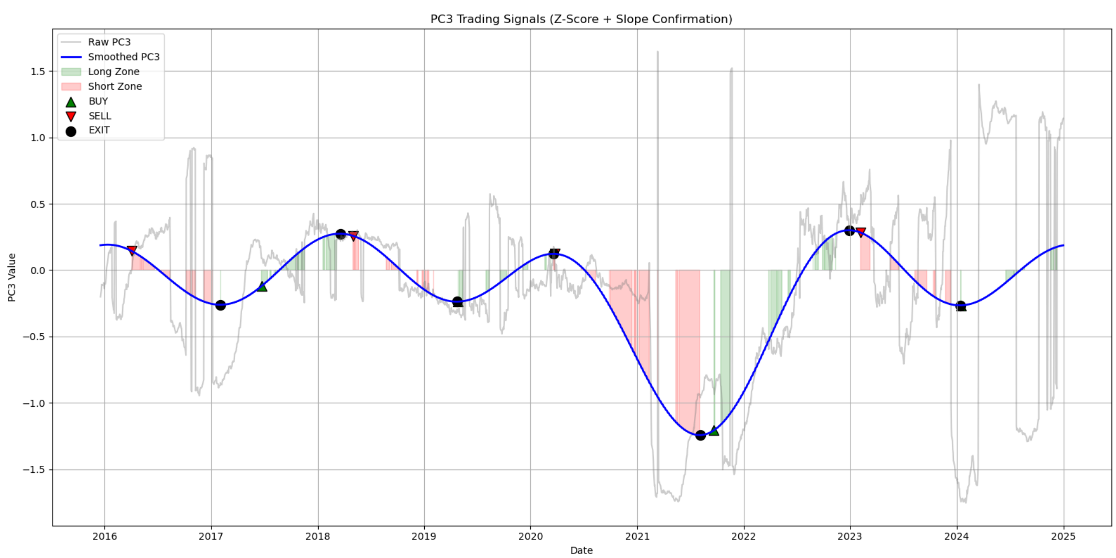


Figure 11: PC3 signal trading overlay. Green and red shaded areas highlight periods where the raw PC3 score diverges from the smoothed signal by more than one standard deviation and the slope turns toward equilibrium. Markers indicate entry (BUY, SELL) and exit points based on this two-layer logic.

Beyond trading, this curvature signal can also be used in macro monitoring systems, overlay dashboards, and as a signal in cross-asset allocation — especially during turning points in the economic cycle where the curve's shape often leads the broader data.

6 Conclusion and Next Steps

This project began as a search for a tradable, mean-reverting signal within the U.S. Treasury yield curve. What it uncovered was something more fundamental: a powerful, interpretable signal that isolates curvature itself. By extracting the third principal component (PC3) through rolling PCA and refining it with spectral smoothing, we arrived at a dynamic, real-time representation of curve shape that is statistically sound, macroeconomically meaningful, and practically usable.

The resulting PC3 curvature index offers a true measure of how the yield curve bends — not just in theory, but as expressed by actual market behavior. Unlike traditional methods such as butterfly spreads, which rely on fixed heuristics, our approach adapts to the market's evolving structure. This discovery enables new ways to track, interpret, and respond to shifts in investor sentiment, particularly around the critical belly of the curve where policy expectations and economic outlooks often collide.

Key Contributions:

- A dynamic PCA-weighted curvature index that evolves with market conditions and captures true yield curve shape changes.
- A statistically validated framework for smoothing and interpreting PC3 using Fourier techniques.
- A directional signal logic that links PC3 behavior to potential inflection points in macroeconomic sentiment.

6.1 Next Steps

Building on this foundation, several paths forward remain:

- **Live Feed Deployment:** Operationalize the curvature index as a real-time signal for macro dashboards, strategy overlays, and bond trading desks.
- **Empirical Strategy Testing:** Evaluate how well this signal performs in P&L terms across historical data using Treasury futures or swaps.
- **Adaptive Signal Refinement:** Apply machine learning tools — such as LSTM networks or reinforcement learning agents — to dynamically tune thresholds, confirm signal regime changes, and optimize trade execution.
- **Fourier Filtering Refinement:** Future work will also explore more rigorous approaches to filtering the PC3 signal in the frequency domain. Rather than relying on fixed variance retention thresholds, we aim to benchmark spectral cutoffs against the characteristics of red noise processes — a common model for financial time series that exhibit autocorrelation. By generating Monte Carlo simulations of red noise, we can identify the expected distribution of spectral power under the null hypothesis of randomness. Frequencies in the actual PC3 signal that exceed these thresholds can then be considered statistically significant and retained, improving signal quality while minimizing overfitting. This approach has the potential to make the smoothing process more adaptive, interpretable, and grounded in time series theory.

6.2 Final Takeaway

This work demonstrates that yield curve curvature can be more than just a conceptual shape — it can be measured, tracked, and understood through the lens of PCA. The PC3-weighted signal developed here isn't just a tool for trading. It's a framework for understanding how the bond market expresses belief in the economic cycle — and how that belief shifts in real time.

Acknowledgments

I would like to sincerely thank Professor Ildar Gabitov of the University of Arizona for his guidance and support throughout this project. As part of the MATH485 Mathematical Modeling course, Professor Gabitov provided critical feedback and helped sanity-check many of the modeling decisions made along the way. His encouragement through his class to explore the intersection of applied mathematics and financial systems helped shape both the direction and clarity of this research.

Project Repository

This research project, including all Python scripts, visualizations, and documentation, is available on GitHub for public reference and collaboration. The repository contains modular code for rolling PCA, Fourier smoothing, curvature signal construction, and trade signal generation.

Scan to view the full project repository:
`https://github.com/danny-watkins/pc3-arbitrage-analysis`

References

[1] Jolliffe, I. T. (2002). *Principal Component Analysis (2nd ed.)*. Springer Series in Statistics.

[2] Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.

[3] Bracewell, R. N. (2000). *The Fourier Transform and Its Applications (3rd ed.)*. McGraw-Hill.

[4] Shumway, R. H., Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples (4th ed.)*. Springer.

[5] Campbell, J. Y., Lo, A. W., MacKinlay, A. C. (1997). *The Econometrics of Financial Markets*. Princeton University Press.

[6] Vasicek, O. (1977). An Equilibrium Characterization of the Term Structure. *Journal of Financial Economics*, 5(2), 177–188.

[7] Rudebusch, G. D., Williams, J. C. (2009). Forecasting Recessions: The Puzzle of the Enduring Power of the Yield Curve. *Journal of Business & Economic Statistics*, 27(4), 492–503.

[8] Dickey, D. A., Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, 74(366), 427–431.

[9] Fama, E. F., Bliss, R. R. (1987). The Information in Long-Maturity Forward Rates. *American Economic Review*, 77(4), 680–692.

[10] Mann, M. E., Lees, J. M. (1996). Robust estimation of background noise and signal detection in climatic time series. *Climatic Change*, 33(3), 409–445.

[11] Makridakis, S., Wheelwright, S. C., Hyndman, R. J. (1998). *Forecasting: Methods and Applications (3rd ed.)*. Wiley.

[12] Harris, C. R., Millman, K. J., et al. (2020). Array programming with NumPy. *Nature*, 585, 357–362.

[13] Virtanen, P., Gommers, R., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272.

[14] Federal Reserve Economic Data (FRED). (2024). *St. Louis Federal Reserve Bank*. Retrieved from `https://fred.stlouisfed.org`

[15] Van Rossum, G., Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.