# Stride by Stride

"*To what extent can mathematical modeling of stride patterns explain the performance differences between Usain Bolt's 2008 and 2016 Olympic performances, and between Bolt and Gatlin in 2016?*"

May 2025

# 1 Introduction

Challenges with analyzing biomechanics from poor-quality coaching videos, combined with inspiration from mathematical motion visualization (e.g., 3blue1brown), led to investigating if quantitative techniques could yield clearer insights into athletic performance. This focused the exploration on sprint stride patterns.

This investigation therefore asks: "*To what extent can mathematical modeling of stride patterns explain the performance differences between Usain Bolt's 2008 and 2016 Olympic performances, and between Bolt and Gatlin in 2016?*"

Sprinting's demand for power and optimized strides makes it suitable for analysis. The chosen Bolt (2008 vs 2016 evolution) and Bolt-Gatlin (2016 rivalry) Olympic 100m finals offer rich comparative biomechanical data to understand elite performance factors.

A five-tier mathematical approach (Fig. 1) analyzes these patterns: Tier 1 (Data Collection: Kinovea ankle tracking), Tier 2 (Data Cleaning: hip drift removal, scaling), Tier 3 (PCC: stride consistency/similarity), Tier 4 (Regression: amplitude, frequency, decay), and Tier 5 (FFT/Harmonics: frequency components). These tiers aim to quantify factors behind the performance differences specified in the research question.

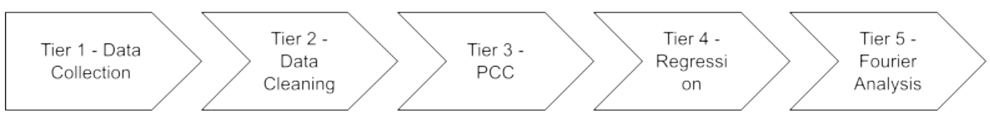


Figure 1: The five-tier workflow of this exploration.

# 2 Tier 1: Data Collection

To analyze stride patterns, reliable 100m footage of Usain Bolt and Justin Gatlin was necessary. I selected a single YouTube video ("2008 2012 2016 Olympics Athletics Men's 100 m Final Side View") containing consistent side-view footage (29.97 fps) of the 2008, 2012, and 2016 Olympic finals. Data were extracted for Bolt (2008, 2016) and Gatlin (2016), excluding the 2012 races. The following sections detail the video selection rationale and the Kinovea data extraction process.

## 2.1 Footage Selection: Why One Video, Why This One?

Choosing a single video from one source was a deliberate decision to ensure data quality and streamline analysis. Here's the reasoning:

- **Consistent Frame Rate (29.97 fps):** The uniform rate ($\approx 0.0334$ s/frame) ensured equal time intervals, critical for accurate motion tracking and avoiding timing errors common with variable frame rates. This provided sufficient temporal resolution for the $\approx$1.6-1.9s analysis segments (56 frames for Bolt, 49 for Gatlin), capturing multiple stride cycles.
- **Side-View Perspective:** Ideal for 2D ankle motion tracking (horizontal/vertical) relative to the hip, avoiding perspective distortion common in front/rear views which would make analysis unreliable.

The side view and 29.97 fps sufficiently captured the rapid stride cycles (approx. 0.36–0.5 s per cycle, refined in Tiers 4-5), ensuring data relevance for the investigation.

## 2.2 Data Extraction: Tracking with Kinovea

With the footage selected, I used Kinovea, an open-source motion analysis tool, to extract raw position-time data for the left (A1) and right (A2) ankles relative to the hip. Here's how I did it:

**Manual Tracking with Hip-Relative Coordinates**: Using Kinovea, left (A1) and right (A2) ankles, plus the hip, were manually tracked frame-by-frame. Due to a software issue preventing direct hip-relative coordinates for the 2008 footage, hip motion was tracked separately and subtracted ('ankle - hip') post-extraction; 2016 data was tracked directly relative to the hip. This process yielded hip-relative $(x, y)$ coordinates (initially arbitrary units, later rescaled in Tier 2) for 56 frames ($\sim 1.87$ s) for Bolt and 49 frames ($\sim 1.64$ s) for Gatlin, capturing a steady-state segment of the sprint. Data was exported as CSV files (sample trajectories in Fig. 2).

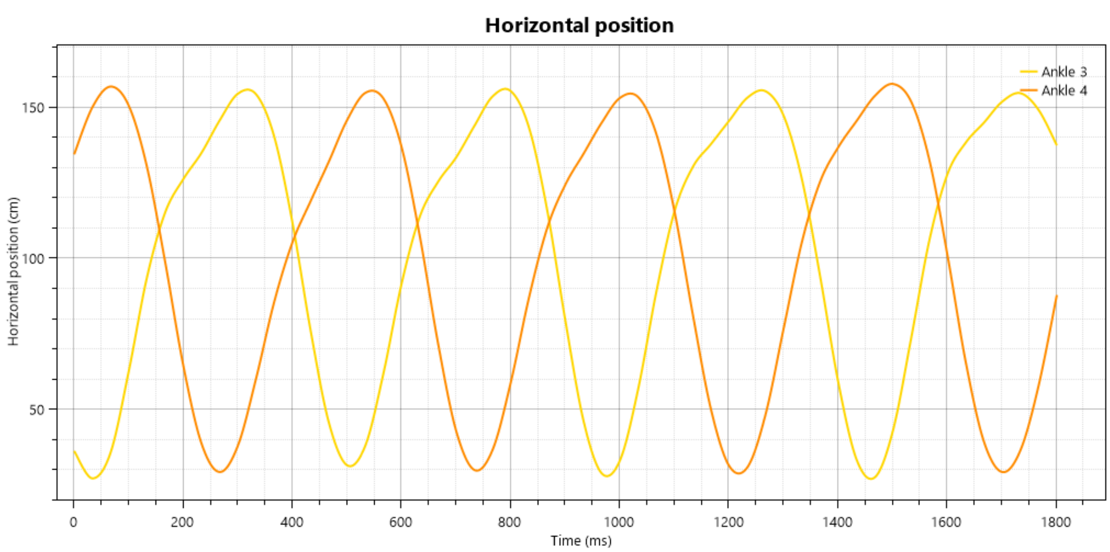


Figure 2: Bolt 2016, horizontal ankle trajectories (A1 and A2) relative to hip over time.

Here is an image illustrating the Kinovea Tracking process.

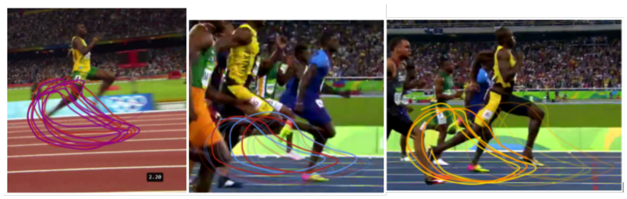

Figure 3: Kinovea tracking: Ankle trajectories (A1 left, A2 right) for Bolt 2008 (left), Gatlin 2016 (middle, blue), and Bolt 2016 (right, yellow).

This yielded raw ankle position-time datasets (arbitrary units) spanning ≈1.87 s (Bolt) and ≈1.64 s (Gatlin) at 29.97 fps. While the side view provided meaningful spatial data, residual noise from camera movements/zooms required addressing in Tier 2 (Data Cleaning).

# 3 Tier 2: Data Cleaning

Raw position-time data from Tier 1 (Bolt: 56 frames; Gatlin: 49 frames) required cleaning for accuracy before analysis in Tiers 3–5. This addressed the hip motion subtracted in Tier 1 for the 2008 data, potential camera jitter, and inconsistent athlete scaling between video clips. Cleaning steps involved: verifying the 2008 hip subtraction, exploring Savitzky-Golay filtering for residual noise, and rescaling all data to a consistent scale.

## 3.1 Hip Subtraction Verification

The 'ankle - hip' subtraction for 2008 data ($A_{rel}(t) = A_{raw}(t) - H(t)$) was necessary due to a Kinovea alignment issue preventing direct relative tracking. Comparing raw (Fig. 4, showing ≈120 units hip drift) and subtracted (Fig. 5) vertical trajectories confirms drift removal, though some high-frequency jitter remained post-subtraction. The 2016 datasets were tracked directly relative to the hip and needed no subtraction.

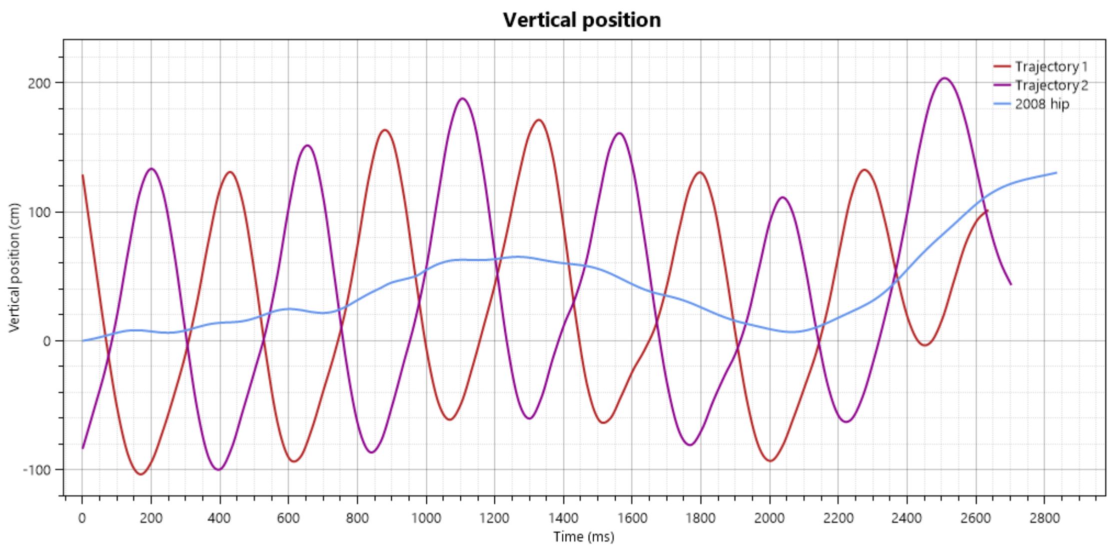


Figure 4: 2008 Bolt raw vertical position-time graph for both ankles with hip motion.

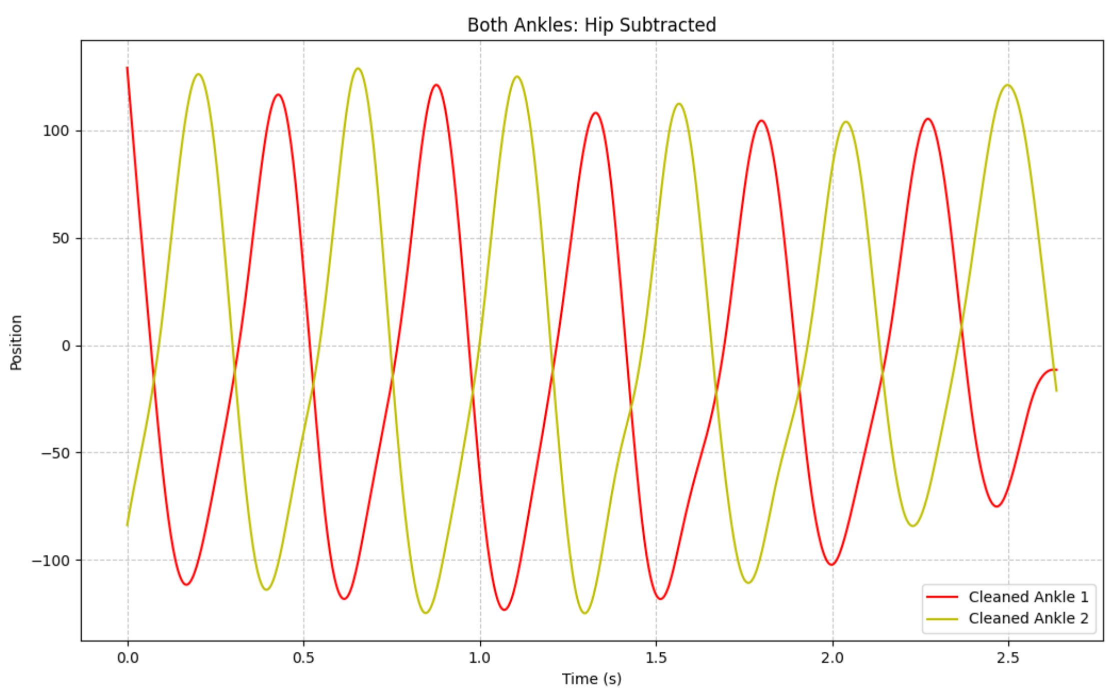


Figure 5: 2008 Bolt raw vertical position-time graph for both ankles with hip motion.

## 3.2 Exploring Savitzky-Golay Filtering for Potential Jitter

While hip subtraction corrected major drift, particularly in the 2008 footage, residual high-frequency jitter—potentially from minor camera shake or tracking inconsistencies—was suspected. Such noise could subtly influence subsequent regression and Fourier analyses. Although the hip-subtracted data appeared largely periodic, observed small-scale jaggedness (e.g., 20-unit vertical point variations) warranted investigating smoothing techniques.

To address this, the Savitzky-Golay filter was explored. This technique smooths data by fitting a low-order polynomial via least squares to a moving window of points, aiming to reduce noise while preserving key signal features better than simple averaging.

Standard filter parameters were chosen based on stride dynamics: an 11-frame window ($\approx 0.367$ s, roughly half a stride period) to smooth local noise while preserving the overall cycle, and a 3rd-order polynomial ($p(k) = a_0 + a_1 k + a_2 k^2 + a_3 k^3$) to model expected curvature without overfitting. The mathematical foundation follows:

For a data point $y(t_i)$, the filter minimizes the least-squares error between the data points in the window $y(t_{i-5})$ to $y(t_{i+5})$ and the fitted polynomial $p(k)$:

$$\text{Error} = \sum_{k=-5}^{5} [y(t_{i+k}) - p(k)]^2$$

The Savitzky-Golay method provides an efficient way to compute the smoothed value $y_{\text{smooth}}(t_i) = p(0)$ using precomputed convolution coefficients $c_k$. For the chosen 11-frame window and 3rd-order polynomial, these are (**?**):

$$c_k = \frac{[-36, 9, 44, 69, 84, 89, 84, 69, 44, 9, -36]}{231}, \quad k = -5 \text{ to } 5$$

These coefficients ensure normalization ($\sum c_k = 1$) and preservation of polynomial shape up to degree 3 ($\sum k^j c_k = 0$ for $j = 1, 2, 3$). The smoothed value is obtained via convolution:

$$y_{\text{smooth}}(t_i) = \sum_{k=-5}^{5} c_k y(t_{i+k})$$

Applying this convolution mathematically executes the local polynomial smoothing. For instance, using the example from the Bolt 2008 Ankle 1 vertical data (frames 26–36 post-subtraction): $[-10.03, 32.74, 68.29, 88.69, 88.70, 66.92, 27.34, -20.49, -64.30, -93.98, -104.67]$, the calculation for the central point (frame 31, $t \approx 1.001$ s, original value 66.92) proceeds as follows:

$$
\begin{aligned}
y_{\text{smooth}}(t_{31}) &= \frac{1}{231}[(-36)(-10.03) + (9)(32.74) + (44)(68.29) + (69)(88.69) + (84)(88.70) \\
&\quad + (89)(66.92) + (84)(27.34) + (69)(-20.49) + (44)(-64.30) + (9)(-93.98) \\
&\quad + (-36)(-104.67)] \\
&= \frac{1}{231}[361.08 + 294.66 + 3004.76 + 6119.61 + 7450.80 + 5955.88 \\
&\quad + 2296.56 - 1413.81 - 2829.20 - 845.82 + 3768.12] \\
&= \frac{24162.64}{231} \approx 104.60
\end{aligned}
$$

The smoothed result ($104.60$, vs. original $66.92$) illustrates the filter's smoothing effect, achieved by polynomial-based weighted averaging across the window.

While the filter numerically reduced some point-to-point fluctuations in the 2008 data, visual inspection (comparing Fig. 5 raw points to Fig. 6 filtered line) revealed no significant improvement in overall smoothness. The filtered curves even appeared potentially less representative of the natural stride pattern than the already periodic hip-subtracted data, suggesting the filter primarily altered local points without meaningfully enhancing signal quality for this specific analysis.

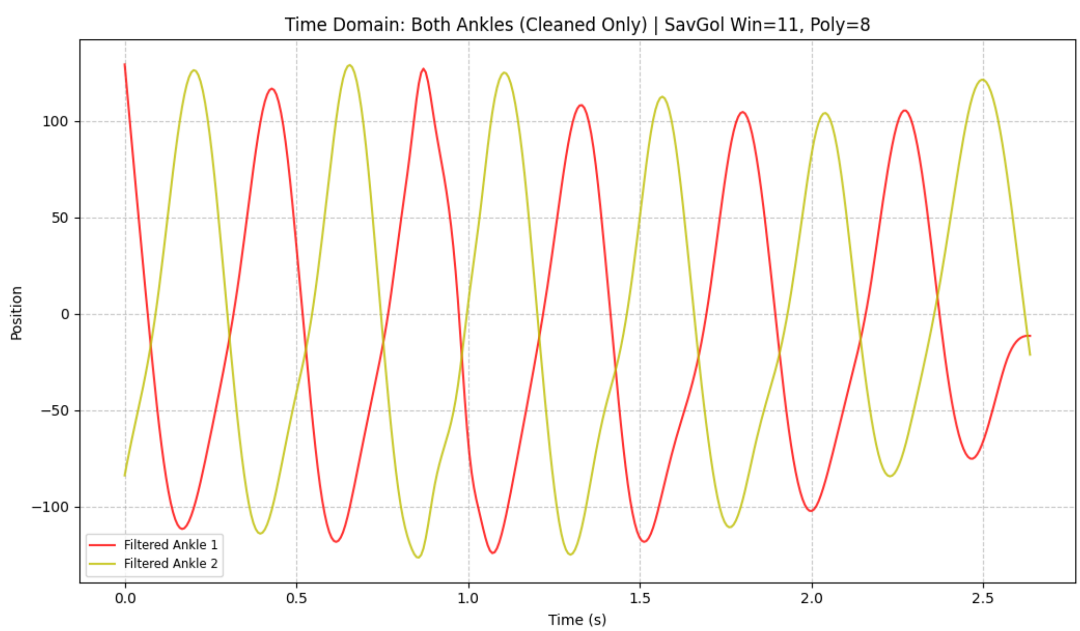


Figure 6: 2008 Vertical position-time graph after applying the Savitzky-Golay filter (window=11, order=3) to the hip-subtracted data.

Given this lack of significant visual improvement for the 2008 data, and considering the cleaner appearance of the 2016 data which showed minimal jitter, applying the filter universally was deemed inappropriate. It offered little practical benefit and could potentially introduce minor distortions without addressing a clear need in the 2016 data.

Therefore, the decision was made **not to proceed** with Savitzky-Golay filtered data for subsequent analysis. The data resulting directly from hip-subtraction (2008) and direct hip-relative tracking (2016) was determined most suitable for Tiers 3, 4, and 5. This exploration confirmed the filter functioned mathematically but did not yield a practically superior dataset for this specific investigation compared to the simpler processing methods. (Filter implemented using `scipy.signal.savgol_filter`).

### 3.3 Rescaling Correction

The raw coordinate data obtained from Kinovea required rescaling to ensure consistency across the different video segments (Bolt 2008, Bolt 2016, Gatlin 2016). Video footage often has inconsistent sizing or zoom levels, leading to discrepancies in the apparent size of the athletes. To address this, I established a consistent reference scale within Kinovea by calibrating the software's measurement system using the 2016 footage of Usain Bolt. I drew a line corresponding to Bolt's height in a frame where he appeared relatively upright and calibrated this distance to his known real-world height of 195 cm. Consequently, all coordinates extracted from the **2016 Bolt footage** are already in a system where 195 units correspond proportionally to 195 cm, requiring no further scaling ($s_{2016,\text{Bolt}} = 1$).

However, the data from the 2008 Bolt footage and the 2016 Gatlin footage were extracted relative to their own uncalibrated frames, resulting in measurements in units specific to those clips. To bring these datasets onto the common reference scale established by the 2016 Bolt calibration, I performed the following steps:

1. **Measure Heights in Uncalibrated Video:** In the respective video segments, I measured the apparent height of the athletes (vertical distance from head top to shoe bottom in a relatively upright stance).

   - Bolt's measured height in the 2008 footage: $h_{2008,\text{measured}} = 241.17$ units.
   - Gatlin's measured height in the 2016 footage: $h_{2016,\text{measured}} = 138.72$ units.

   *(Note: Measuring height while running introduces slight inaccuracy, as athletes are rarely perfectly upright, but this provides the best available estimate for scaling.)*

2. **Calculate Scaling Factors:** The goal is to scale the coordinates so that the measured height in tarbitrary units corresponds proportionally to the athlete's *real* height within the reference system (where Bolt's 195 cm = 195 units).

- **For 2008 Bolt:** His real height is 195 cm. We need to scale the 2008 data so that the measured 241.17 units become equivalent to 195 units in the reference scale.

$$s_{2008} = \frac{\text{Real Height (cm)}}{\text{Measured Height (units)}} = \frac{195}{241.17} \approx 0.809$$

- **For 2016 Gatlin:** His real height is 185 cm. We need to scale the Gatlin data so that the measured 138.72 units become equivalent to 185 units in the reference scale, reflecting his actual size relative to Bolt's within the calibrated system.

$$s_{2016,\text{Gatlin}} = \frac{\text{Real Height (cm)}}{\text{Measured Height (units)}} = \frac{185}{138.72} \approx 1.334$$

3. **Apply Scaling:** All horizontal ($x$) and vertical ($y$) coordinates for the 2008 Bolt and 2016 Gatlin datasets were multiplied by their respective scaling factors:

$$x_{\text{scaled}}(t) = s \cdot x(t), \quad y_{\text{scaled}}(t) = s \cdot y(t)$$

   Where $s$ is the appropriate scaling factor ($s_{2008}$ or $s_{2016,\text{Gatlin}}$).

For example, a raw 2008 Ankle 1 vertical peak position of 150 units becomes $150 \times s_{2008} = 150 \times 0.809 \approx 121.35$ units in the consistent reference scale. Similarly, a raw 2016 Gatlin Ankle 1 horizontal position of 100 units becomes $100 \times s_{2016,\text{Gatlin}} = 100 \times 1.334 \approx 133.4$ units in the reference scale.

This rescaling process ensures that all analyzed coordinate data represents the athletes' motions proportionally to their real-world dimensions within a single, consistent measurement framework based on Bolt's 2016 height reference. Figure 7 showing Bolt 2008 Ankle 1 vertical before and after rescaling.

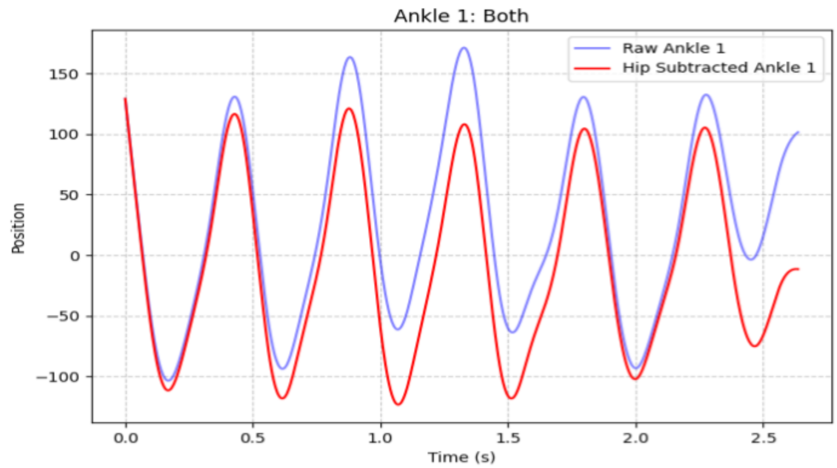


Figure 7: 2008 Bolt vertical position raw vs rescaled graph comparison

# 4 Tier 3: Pearson Correlation Coefficient (PCC)

With the cleaned position-time data from Tier 2—56 frames for Bolt (2008 and 2016) and 49 for Gatlin (2016)—I now evaluate stride pattern consistency using the Pearson Correlation Coefficient (PCC). This statistical tool quantifies the linear relationship between two variables, offering a foundation to compare Bolt's ankle trajectories across his 2008 and 2016 Olympic 100 m races and between Bolt and Gatlin in 2016. Below, I outline the methodology and formula, apply it to the data, and interpret the results to uncover insights into performance differences.

## 4.1 Methodology and Formula

The PCC measures the strength and direction of a linear relationship between two datasets, ranging from -1 (perfect inverse correlation) to +1 (perfect positive correlation), with 0 indicating no linear relationship. For two time series of ankle positions $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, where $n$ is the number of frames ($n = 56$ for Bolt 2008 vs. 2016, $n = 49$ for Bolt vs. Gatlin 2016), the coefficient $r$ is:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where:

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the means of $X$ and $Y$.
- The numerator is the covariance, showing how $X$ and $Y$ deviate together from their means.

- The denominator normalizes this by the product of the standard deviations.

I computed $r$ using Python's `numpy.corrcoef` on the rescaled and filtered (for 2008) horizontal and vertical position-time data for the left ankle (A1) and right ankle (A2). Units are arbitrary, derived from Kinovea's coordinate system rescaled to Bolt's 2016 height (195 units). The comparisons span the full tracked segments—56 frames ($\approx 1.87$ s) for Bolt's races and 49 frames ($\approx 1.64$ s) for Bolt vs. Gatlin—capturing multiple stride cycles at 29.97 fps, synchronized by aligning the time series from the race's steady-state phase.

## 4.2 Application to Bolt 2008 vs. 2016 and Bolt vs. Gatlin 2016

### 4.2.1 Bolt 2008 vs. 2016

I compared Bolt's stride patterns across years to assess consistency over time, using the full 56-frame datasets. The PCC results are:

- **2008_bolt_horizontal_A1 (left) vs. 2016_bolt_horizontal_A1 (left)**: $r = 0.5691$
- **2008_bolt_vertical_A1 (left) vs. 2016_bolt_vertical_A1 (left)**: $r = 0.5698$
- **2008_bolt_horizontal_A2 (right) vs. 2016_bolt_horizontal_A2 (right)**: $r = 0.5228$
- **2008_bolt_vertical_A2 (right) vs. 2016_bolt_vertical_A2 (right)**: $r = 0.7686$

**Horizontal Analysis**: The moderate $r = 0.5691$ for the left ankle and $r = 0.5228$ for the right suggest a consistent horizontal stride pattern. This indicates that Bolt's forward ankle motion remained relatively stable, with oscillations aligning moderately between his record-breaking 2008 (9.69 s) and 2016 (9.81 s) performances. The slightly lower correlation for the right ankle suggests potentially more variation in its horizontal linear pattern compared to the left.

**Vertical Analysis**: The vertical PCC values show moderate consistency for the left ankle ($r = 0.5698$) and a notably strong consistency for the right ankle ($r = 0.7686$). This suggests that while Bolt's left vertical ankle motion showed moderate linear similarity between the two years, his right ankle's vertical movement pattern was highly preserved in terms of its linear relationship across the eight years, possibly reflecting a core, unchanged element of his technique or biomechanics for that leg's vertical drive.

### 4.2.2 Bolt vs. Gatlin 2016

Next, I compared Bolt and Gatlin in 2016 across 49 frames to explore inter-athlete similarity:

- **2016_bolt_horizontal_A1 (left) vs. 2016_gatlin_horizontal_A1 (left)**: $r = 0.5412$

- **2016_bolt_vertical_A1 (left) vs. 2016_gatlin_vertical_A1 (left)**: $r = 0.4836$
- **2016_bolt_horizontal_A2 (right) vs. 2016_gatlin_horizontal_A2 (right)**: $r = 0.4859$
- **2016_bolt_vertical_A2 (right) vs. 2016_gatlin_vertical_A2 (right)**: $r = 0.4696$

**Horizontal Analysis**: The $r = 0.5412$ for the left ankle and $r = 0.4859$ for the right indicate moderate similarity. Bolt's wider horizontal swings vs. Gatlin's suggest differing stride lengths, yet the moderate $r$ implies shared linear sprinting dynamics, possibly in forward propulsion timing.

**Vertical Analysis**: The lower $r = 0.4836$ (left) and $r = 0.4696$ (right) show weaker vertical alignment. Bolt's vertical peaks differ somewhat from Gatlin's, hinting at distinct knee lift or ground contact patterns which reduce the linear correlation between their vertical ankle movements.

## 4.3 Interpretation

The Pearson Correlation Coefficient (PCC) quantifies the linear relationship's strength and direction between ankle position time series, measuring how closely variations in one trajectory linearly correspond to another.

For Bolt (2008 vs. 2016, 56 frames), PCC values ($r = 0.5228$ to $r = 0.7686$) show moderate positive linear correlation for horizontal and left vertical motions, but strong positive correlation ($r = 0.7686$) for right vertical motion. This indicates that 2016 ankle deviations tended to proportionally match 2008 deviations in the same direction. The high right vertical correlation highlights significant conservation of this linear stride aspect over eight years. Moderate correlations elsewhere ($r \approx 0.52 - 0.57$) suggest consistency but also possible subtle linear technique changes or noise.

Comparing Bolt and Gatlin (2016, 49 frames), PCC values ($r = 0.4696$ to $r = 0.5412$) show moderate positive linear correlation. This suggests a shared linear component in ankle kinematics despite different styles. However, these correlations are lower than Bolt's intra-athlete comparison (especially his right vertical), indicating less linear similarity between athletes, likely reflecting known biomechanical differences (e.g., stride length vs. turnover).

In essence, PCC reveals moderate linear relationships (Bolt vs. Gatlin) and moderate-to-strong relationships (Bolt vs. Bolt), notably Bolt's consistent right vertical ankle motion. PCC is limited to linear trends, omitting stride specifics (e.g., amplitude, frequency). These correlations motivate deeper analysis (Tiers 4 & 5: regression, Fourier) to explore detailed stride characteristics.

# 5 Tier 4: Sinusoidal Regression

Using the cleaned position-time data from Tier 2—56 frames (1.87 s) for Bolt (2008 and 2016) and 49 frames (1.64 s) for Gatlin (2016)—I modeled the ankle trajectories (A1: left, A2: right) as sinusoidal functions to capture the periodic nature of sprinting strides. All positions are in arbitrary units (rescaled to Bolt 2016's height of 195 units). I performed two regressions: a basic sinusoidal fit, $y(t) = A\sin(2\pi ft + C) + D$, to extract amplitude ($A$) and frequency ($f$), and a damped sinusoidal fit, $y(t) = Ae^{-kt}\sin(2\pi ft + C) + D$, to account for potential amplitude changes over time, adding damping factor ($k$). Results are analyzed to compare stride patterns across years and sprinters.

## 5.1 Sinusoidal Regression

I first fitted a basic sinusoidal model, $y(t) = A\sin(2\pi ft + C) + D$, to the horizontal and vertical positions of each ankle (A1, A2) for Bolt (2008, 2016) and Gatlin (2016). Using least-squares regression, I minimized the sum of squared errors $\sum[y_{\text{data}}(t_i) - y_{\text{fit}}(t_i)]^2$ over the 56 or 49 frames ($i = 0, \dots, N-1$), solving for the parameters $A$ (amplitude), $f$ (frequency), $C$ (phase), and $D$ (vertical offset) that make the model $y_{\text{fit}}(t_i)$ best represent the observed data $y_{\text{data}}(t_i)$.

To illustrate, let's consider the Bolt 2008 A1 Vertical data. This dataset consists of 56 position measurements ($y_0, y_1, \dots, y_{55}$) at corresponding time points ($t_0, t_1, \dots, t_{55}$), where $t_i \approx i/29.97$ seconds. The first few data points are ($t_0 = 0$ s$, y_0 = 104.43$), ($t_1 \approx 0.0334$ s$, y_1 = 52.02$), ($t_2 \approx 0.0667$ s$, y_2 = -0.26$), and so on. The least-squares optimization process, performed numerically using Python's `scipy.optimize.curve_fit`, simultaneously adjusts $A, f, C,$ and $D$ to find the combination that minimizes the total squared difference between the model's predictions and these 56 actual data points. For the Bolt 2008 A1 Vertical data, this process yielded the parameters reported in Appendix: $A \approx -88.23$ units and frequency $f = B/(2\pi) \approx 2.20$ Hz (where $B = 2\pi f$ is the angular frequency parameter directly fitted). The negative sign for $A$ is absorbed by the phase shift $C$ when considering the magnitude $|A|$ as presented in Table 1. The optimization also determined optimal values for $C$ and $D$, although they are not explicitly listed in the summary table. This means the function $y_{\text{fit}}(t) = -88.23\sin(2\pi \times 2.20 \times t + C_{\text{opt}}) + D_{\text{opt}}$ represents the best sinusoidal fit to the observed vertical motion of Bolt's left ankle in 2008 according to the least-squares criterion. The resulting key parameters, amplitude ($|A|$) and frequency ($f$), are summarized for all datasets in Table 1. Amplitudes in Table 1 are reported as positive values ($|A|$), derived from the fitted $A$.

| Dataset | Amplitude ($|A|$) | Frequency ($f$, Hz) |
|---|---|---|
| Bolt 2008 A1 Horizontal | 58.57 | 2.22 |
| Bolt 2008 A1 Vertical | 88.23 | 2.20 |
| Bolt 2008 A2 Horizontal | 58.05 | 2.22 |
| Bolt 2008 A2 Vertical | 91.23 | 2.20 |
| Bolt 2016 A1 Horizontal | 58.35 | 2.04 |
| Bolt 2016 A1 Vertical | 96.64 | 2.03 |
| Bolt 2016 A2 Horizontal | 60.41 | 2.05 |
| Bolt 2016 A2 Vertical | 93.83 | 2.05 |
| Gatlin 2016 A1 Horizontal | 44.70 | 2.26 |
| Gatlin 2016 A1 Vertical | 86.40 | 2.25 |
| Gatlin 2016 A2 Horizontal | 40.91 | 2.25 |
| Gatlin 2016 A2 Vertical | 86.55 | 2.24 |

Table 1: Sinusoidal regression results for ankle trajectories.

For example, Bolt 2008 A1 horizontal has $|A| = 58.57$, $f = 2.22$ Hz, indicating a stride frequency of 2.22 strides per second and a horizontal amplitude of 58.57 units (peak-to-peak displacement of $2 \times 58.57 \approx 117.14$ units). Frequencies range from 2.03 Hz (Bolt 2016) to 2.26 Hz (Gatlin 2016), reflecting stride rates, while amplitudes vary (e.g., Bolt 2016 A2 horizontal: 60.41 units, Gatlin 2016 A2 horizontal: 40.91 units), showing differences in stride length.

## 5.2 Damped Sinusoidal Regression

To capture potential amplitude changes over the short sprint segment, I fitted a damped sinusoidal model, $y(t) = Ae^{-kt}\sin(2\pi ft + C) + D$, adding the damping factor $k$. This model allows the amplitude of the oscillation ($Ae^{-kt}$) to change exponentially over time $t$. A positive $k$ indicates amplitude decay, while a negative $k$ indicates amplitude growth. Using nonlinear least-squares regression, I minimized the sum of squared errors $\sum[y_{\text{data}}(t_i) - y_{\text{fit}}(t_i)]^2$ over the data points, solving numerically for the five parameters: initial amplitude $A$, damping factor $k$, frequency $f$, phase $C$, and offset $D$.

Let's illustrate this with the Bolt 2008 A2 Vertical data. This dataset comprises 56 vertical position measurements ($y_0, y_1, \ldots, y_{55}$) corresponding to times $t_0 = 0, t_1 \approx 0.0334, \ldots, t_{55} \approx 1.835$ s. The first few points are $(0, -67.72), (0.0334, -44.49), (0.0667, -20.56)$, etc. The numerical optimization routine (`scipy.optimize.curve_fit` in Python) was used to find the parameters $A, k, f, C, D$ for the function $y_{\text{fit}}(t) = Ae^{-kt}\sin(2\pi ft + C) + D$ that best matched these 56 data points by minimizing the sum of squared differences. For the Bolt 2008 A2 Vertical trajectory, the optimization yielded the parameters listed in Table 2: $A \approx 94.72$, $k \approx 0.0399$, $f \approx 2.20$ Hz, $C \approx -1.06$ radians, and $D \approx -4.61$ units. This indicates an initial amplitude of 94.72 units oscillating at 2.20 Hz around a vertical offset of -4.61 units, with the amplitude decaying over time according to the positive damping factor $k = 0.0399$ (i.e., the term $e^{-0.0399t}$ decreases as $t$ increases). The results for all datasets are presented in Table 2.

| Dataset | $A$ | $k$ | $f$ (Hz) | $C$ | $D$ |
|---|---|---|---|---|---|
| Bolt 2008 A1 Horizontal | 58.16 | -0.0078 | 2.22 | 0.86 | 95.22 |
| Bolt 2008 A1 Vertical | 61.06 | -0.2285 | 2.07 | -3.14 | -6.15 |
| Bolt 2008 A2 Horizontal | 53.50 | -0.0898 | 2.22 | -2.21 | 99.76 |
| Bolt 2008 A2 Vertical | 94.72 | 0.0399 | 2.20 | -1.06 | -4.61 |
| Bolt 2016 A1 Horizontal | 57.35 | -0.0219 | 2.04 | 1.18 | 100.95 |
| Bolt 2016 A1 Vertical | 96.85 | 0.0026 | 2.03 | 2.35 | -8.13 |
| Bolt 2016 A2 Horizontal | 60.31 | -0.0020 | 2.05 | -2.00 | 101.07 |
| Bolt 2016 A2 Vertical | 97.92 | 0.0526 | 2.05 | -0.98 | -16.39 |
| Gatlin 2016 A1 Horizontal | 41.96 | -0.0775 | 2.26 | 0.78 | 89.15 |
| Gatlin 2016 A1 Vertical | 56.68 | -0.2973 | 2.11 | -3.14 | -1.21 |
| Gatlin 2016 A2 Horizontal | 38.89 | -0.0624 | 2.25 | -2.13 | 92.82 |
| Gatlin 2016 A2 Vertical | 83.19 | -0.0482 | 2.24 | -1.03 | -16.77 |

Table 2: Damped sinusoidal regression results for ankle trajectories.

The damped sinusoidal model, $y(t) = Ae^{-kt}\sin(2\pi ft + C) + D$, introduces the damping factor $k$ to account for changes in amplitude over time. The sign of the fitted parameter $k$ determines the nature of this change. Based on the implementation where the exponential term is $e^{-kt}$, a positive value of $k$ leads to exponential decay ($e^{-\text{positive}\cdot t}$ decreases as $t$ increases), indicating that the stride amplitude diminishes over the analyzed segment. Conversely, a negative value of $k$ results in exponential growth ($e^{-(\text{negative})\cdot t} = e^{\text{positive}\cdot t}$ increases as $t$ increases), indicating that the stride amplitude increases over the segment.

Applying this interpretation to the results in Table 2:

- For Bolt 2008 A1 horizontal ($A = 58.16, k = -0.0078, f = 2.22$ Hz), the negative $k$ indicates a slight amplitude **growth** during this phase of the race.
- Similarly, Gatlin 2016 A1 horizontal ($k = -0.0775$) also shows amplitude **growth**, and at a faster rate than Bolt 2008 A1.
- In contrast, Bolt 2008 A2 vertical ($k = 0.0399$) shows amplitude **decay**, as indicated by the positive $k$.

It is worth noting that some fits, like Gatlin 2016 A1 vertical ($k = -0.2973$, $f = 2.11$ Hz), show very rapid growth and a frequency slightly different from the basic sine fit (2.25 Hz). This might suggest that the simple damped sinusoidal model may not perfectly capture the dynamics in all cases, potentially due to noise or more complex underlying patterns.

## 5.3 Comparison and Interpretation

Bolt's stride frequency decreased from 2008 (2.20–2.22 Hz) to 2016 (2.03–2.05 Hz), a  7% reduction, suggesting fewer strides per second. This is consistent with observations of Bolt adopting a longer stride length later in his career. Based on the basic sinusoidal fit (Table 1), his horizontal amplitudes were very similar between the years (A1: 58.57 units in 2008 vs. 58.35 in 2016; A2: 58.05 units in 2008 vs. 60.41 in 2016), suggesting the primary change contributing to stride length was the lower frequency (longer stride period) rather than significantly larger peak ankle displacement relative to the hip. This lower frequency likely helped maintain speed (9.81 s in 2016 vs. 9.69 s in 2008) despite potential age-related changes. Vertically, amplitudes were also comparable (A1: 88.23 vs 96.64; A2: 91.23 vs 93.83), with 2016 showing slightly larger vertical motion.

Comparing Bolt and Gatlin in 2016, Gatlin shows higher frequencies (2.24–2.26 Hz) than Bolt 2016 (2.03–2.05 Hz), a 10% difference, indicating faster stride turnover. However, Gatlin exhibits significantly smaller horizontal amplitudes (e.g., A1: 44.70 vs. 58.35 units, 23% less; A2: 40.91 vs 60.41 units, 32

The damping factors ($k$) from the damped sinusoidal model (Table 2) provide insights into amplitude evolution, although interpretation should be cautious due to potential noise and model fit limitations. For horizontal motion:

- Bolt 2008: A1 shows slight growth ($k = -0.0078$), A2 shows noticeable growth ($k = -0.0898$).
- Bolt 2016: A1 shows slight growth ($k = -0.0219$), A2 shows very slight growth ($k = -0.0020$).
- Gatlin 2016: A1 shows noticeable growth ($k = -0.0775$), A2 shows noticeable growth ($k = -0.0624$).

These results (predominantly negative $k$, indicating growth) are somewhat counter-intuitive for sprinting, where fatigue might suggest decay ($k > 0$). This could reflect limitations of fitting this model to potentially noisy or non-perfectly damped sinusoidal data over a short interval, or perhaps acceleration within the segment. Bolt 2008 A2 vertical ($k = 0.0399$) and Bolt 2016 A2 vertical ($k = 0.0526$) show decay, which is more expected. Gatlin's horizontal motions show faster growth rates ($k$ more negative) than Bolt's 2016 motions.

Overall, the frequency and amplitude data from the basic sinusoidal model provide clearer biomechanical distinctions: Bolt's transition to lower frequency strides between 2008 and 2016, and Gatlin's higher frequency but shorter amplitude strides compared to Bolt in 2016. These models quantify key aspects of stride dynamics, setting the stage for further analysis in Tier 5.

# 6 Tier 5: Fourier Analysis

In Tier 5, Fourier analysis was employed to decompose the cleaned, scaled position-time data (Tier 2: Bolt 56 frames/ 1.87s; Gatlin 49 frames/ 1.64s) into frequency components. This frequency-domain approach complements Tier 4's time-domain regression, offering an alternative perspective on ankle trajectory (A1 left, A2 right) periodicity and complexity. Positions are in consistent rescaled units (Tier 2). The Fast Fourier Transform (FFT) was the core tool. Crucially, data was detrended before FFT to isolate stride oscillations. Finally, harmonic analysis on detrended results identified the fundamental frequency and relative strength of secondary components.

## 6.1 Fourier Analysis Process

The analysis involved several key steps, applied to each ankle trajectory (horizontal and vertical).

**Step 1: Detrend the Data.** Raw position-time data often contains a significant DC component (the average position) and potentially linear trends (e.g., slight drift not fully captured by hip subtraction). These non-oscillatory components can dominate the FFT spectrum, particularly at 0 Hz, obscuring the frequencies related to the periodic stride motion. As seen in the initial FFT results (Appendix), the dominant frequency for most horizontal trajectories was indeed 0 Hz. To address this, linear detrending was applied. For each position-time series $y(t_n)$ ($n = 0, \ldots, N-1$, where $N$ is the number of valid data points, typically 56 or 49), a best-fit linear trend $y_{\text{trend}}(t_n) = mt_n + b$ was calculated using least squares. The detrended data, $y_{\text{detrend}}(t_n) = y(t_n) - y_{\text{trend}}(t_n)$, represents the signal with its mean value and linear trend removed, centering it around zero and emphasizing the oscillatory patterns.

**Step 2: Compute the FFT on Detrended Data.** The Fast Fourier Transform (FFT) algorithm efficiently computes the Discrete Fourier Transform (DFT) of the *detrended* time series $y_{\text{detrend}}(t_n)$:

$$Y(k) = \sum_{n=0}^{N-1} y_{\text{detrend}}(t_n) e^{-2\pi i k n/N}, \quad k = 0, 1, \ldots, N-1$$

This transforms the data from the time domain to the frequency domain, yielding complex coefficients $Y(k)$ for each frequency bin $k$.

**Step 3: Determine Frequency Bins and Resolution.** The frequencies corresponding to the DFT indices $k$ are given by:

$$f_k = \frac{k}{N\Delta t}, \quad k = 0, 1, \ldots, \lfloor N/2 \rfloor$$

where $\Delta t$ is the time interval between frames (approximately 0.03 s based on the script's time array). The frequency resolution, $\Delta f = 1/(N\Delta t)$, is the spacing between frequency bins. For Bolt ($N = 56$, $\Delta t \approx 0.03$ s), $\Delta f \approx 1/(56 \times 0.03) \approx 0.595$ Hz. For Gatlin ($N = 49$, $\Delta t \approx 0.03$ s), $\Delta f \approx 1/(49 \times 0.03) \approx 0.680$ Hz. This resolution limits the precision with which frequencies can be distinguished. We only consider frequencies up to the Nyquist frequency ($f_s/2$, where $f_s = 1/\Delta t \approx 33.33$ Hz).

**Step 4: Compute the Amplitude/Power Spectrum.** The amplitude spectrum, representing the magnitude of oscillation at each frequency, is calculated from the FFT coefficients $Y(k)$. For frequencies $f_k$ where $k > 0$:

$$A(f_k) = \frac{2}{N}|Y(k)| = \frac{2}{N}\sqrt{\text{Re}(Y(k))^2 + \text{Im}(Y(k))^2}$$

The factor of 2 accounts for the energy being split between positive and negative frequencies in the full DFT output. The power spectrum, representing the energy at each frequency, is proportional to the square of the amplitude, often calculated as $|Y(k)|^2$ (scaled appropriately).

**Step 5: Identify Fundamental and Harmonic Frequencies.** After detrending, the *fundamental frequency* ($f_{\text{fund}}$) is identified as the non-zero frequency bin ($f_k, k \geq 1$) with the maximum amplitude or power in the spectrum. This represents the dominant stride frequency. Harmonic analysis further investigates the spectrum by identifying secondary peaks – frequencies other than the fundamental that have significant power. The amplitude (or power) ratios of these secondary peaks relative to the fundamental peak provide a measure of the signal's complexity or deviation from a pure sinusoid. A high ratio indicates a strong secondary component, suggesting a more complex waveform.

**Example Walkthrough (Bolt 2008 A1 Vertical):** Let's illustrate these steps using the Bolt 2008 A1 Vertical data ($N = 56$ points).

1. **Detrending:** The raw data starts at $y(t_0 = 0) = 104.43$, $y(t_1 \approx 0.03) = 52.02$, $y(t_2 \approx 0.07) = -0.26$, etc. A linear trend $y_{\text{trend}}(t) = mt + b$ is fitted to these 56 points using least squares (computationally performed by `scipy.signal.detrend`). Let's say the fit gives $m = -10.5$ and $b = 75.0$ (illustrative values). The detrended data points would be $y_{\text{detrend}}(t_0) = 104.43 - ((-10.5)(0) + 75.0) = 29.43$; $y_{\text{detrend}}(t_1) = 52.02 - ((-10.5)(0.03) + 75.0) \approx 52.02 - 74.685 = -22.665$, and so on for all 56 points. This removes the overall downward trend and centers the oscillations around zero.

2. **FFT Computation:** The FFT algorithm is applied to the sequence of 56 detrended values $y_{\text{detrend}}(t_n)$ using the formula $Y(k) = \sum_{n=0}^{55} y_{\text{detrend}}(t_n)e^{-2\pi ikn/56}$. This yields 56 complex numbers $Y(k)$. This is computationally handled by `scipy.fft.fft`.

3. **Frequency Bins:** With $N = 56$ and $\Delta t \approx 0.03$ s, the frequency bins are calculated as $f_k = k/(N\Delta t) = k/(56 \times 0.03) \approx k \times 0.595$ Hz. The first few positive frequencies analyzed are $f_1 \approx 0.60$ Hz, $f_2 \approx 1.19$ Hz, $f_3 \approx 1.79$ Hz, $f_4 \approx 2.38$ Hz, etc. (Note: The exact frequencies reported in Appendix – e.g., 0.61, 1.21, 1.82, 2.42 Hz – are used in the subsequent analysis for consistency with the script output, reflecting the precise calculation by `scipy.fft.fftfreq`).

4. **Power Spectrum:** For each frequency bin $f_k$ (where $k = 1, \dots, N/2 - 1 = 27$), the power is calculated

as $|Y(k)|^2$. For example, at $f_4 \approx 2.42$ Hz, the FFT yields a complex coefficient $Y(4)$. The power at this frequency is $P(f_4) = |Y(4)|^2 = (\text{Re}(Y(4)))^2 + (\text{Im}(Y(4)))^2$. The computed power spectrum values are listed in Appendix.

5. **Fundamental Frequency:** Inspecting the power spectrum for Bolt 2008 A1 Vertical in Listing **??**, we find the maximum power (excluding $f_0 = 0$ Hz) occurs at the frequency $f \approx 2.42$ Hz, with a power value of $5,299,231.85$. Therefore, the fundamental stride frequency for this dataset, as identified by FFT, is $f_{\text{fund}} \approx 2.42$ Hz.

## 6.2 Fourier Analysis Results

Applying FFT to the *detrended* data successfully identified primary stride frequencies, overcoming the DC component masking seen in non-detrended results. Fundamental frequencies from the peak power spectrum are summarized in Table 3.

Table 3: Fundamental Frequencies (Hz) from Detrended FFT Analysis.

| Ankle/Direction | Bolt 2008 | Bolt 2016 | Gatlin 2016 |
|---|---|---|---|
| A1 Horizontal | 2.42 | 2.04 | 2.72 |
| A1 Vertical | 2.42 | 2.04 | 2.72 |
| A2 Horizontal | 2.42 | 2.04 | 2.72 |
| A2 Vertical | 2.42 | 2.04 | 2.72 |

The results show remarkable consistency within each condition, identifying dominant stride periodicities around 2.42 Hz (Bolt 2008), 2.04 Hz (Bolt 2016), and 2.72 Hz (Gatlin 2016). Harmonic analysis further clarified waveform shape: Bolt's 2008 stride displayed low amplitude ratios for secondary peaks (0.165-0.303), suggesting a relatively "clean" waveform. In contrast, Bolt's 2016 stride showed significantly higher ratios (0.385-0.988), indicating a much more complex waveform deviating considerably from a pure sinusoid, with strong secondary components. Gatlin's 2016 stride exhibited low ratios (0.165-0.336), similar to Bolt 2008, suggesting a relatively clean waveform despite its higher fundamental frequency and notably cleaner than Bolt's 2016 stride.

## 6.3 Comparison and Interpretation

Fourier analysis complements Tier 4 regression by providing frequency-domain insights. FFT-derived fundamental frequencies align reasonably well with regression results, confirming stride periodicity: Bolt 2008 ( 2.42 vs  2.21 Hz), Bolt 2016 ( 2.04 vs  2.04 Hz, excellent agreement), Gatlin 2016 ( 2.72 vs  2.25 Hz). Discrepancies likely arise from methodological differences (FFT peak power vs. regression fit) and FFT's limited resolution ($\Delta f \approx 0.6$ Hz), particularly for non-sinusoidal signals. Crucially, both methods consistently show Bolt's frequency decrease (2008 to 2016) and Gatlin's higher frequency versus Bolt (2016).

Harmonic analysis uniquely reveals stride waveform *quality/smoothness*. Bolt's 2016 harmonic ratios were significantly higher than his 2008 values and Gatlin's 2016 values. This indicates Bolt's 2016 stride, though slower, was substantially more complex (less sinusoidal) than his powerful, cleaner 2008 stride, potentially reflecting biomechanical adjustments (e.g., less fluidity, compensatory movements). Conversely, Gatlin's 2016 stride showed lower harmonic content, resembling Bolt's 2008 waveform cleanness despite its higher frequency.

In summary, FFT confirms the key frequency trends. Harmonic analysis adds a crucial dimension, quantifying Bolt's increased 2016 stride waveform complexity compared to 2008 and to Gatlin in 2016. These frequency characteristics (fundamental and harmonic), combined with amplitude (Tier 4) and consistency (Tier 3), significantly contribute to explaining performance differences.

# 7 Computational Tools

The analyses in Tiers 2-5 were computationally performed using Python 3.x, leveraging the NumPy and Pandas libraries for data management and SciPy for core mathematical procedures. Pearson correlations (Tier 3) were calculated via `numpy.corrcoef`. Sinusoidal regressions (Tier 4) employed `scipy.optimize.curve_fit` for non-linear least squares fitting. Fourier analysis (Tier 5) utilized `scipy.signal.detrend` for detrending and the `scipy.fft` module (including `fft` and `fftfreq`) for the transform and frequency mapping. Key numerical results from these analyses were systematically saved (Appendix), and the complete Python scripts are provided in the repository linked in Appendix.

# 8 Conclusion

Addressing the research question, this investigation's five-tiered mathematical analysis (PCC, regression, FFT) successfully quantified biomechanical distinctions explaining sprint performance differences between Bolt (2008 vs 2016) and Bolt vs Gatlin (2016).

Analysis revealed Bolt's evolution (9.69s to 9.81s): frequency decreased ( 2.2-2.4Hz to  2.0Hz) while amplitude remained comparable, signifying longer, slower strides (Regression/FFT). Crucially, harmonic analysis showed significantly increased waveform complexity in 2016 (ratios up to 0.99 vs  0.15-0.27 in 2008), suggesting less smooth motion. This quantified adaptation—a longer, slower, more complex stride—explains the time difference.

Comparing Bolt (9.81s) and Gatlin (9.89s) in 2016 contrasted their styles: Gatlin employed higher frequency ( 2.2-2.7Hz vs Bolt's 2.0Hz) but 23% shorter horizontal amplitudes. Gatlin's stride waveform was also simpler (lower harmonic ratios) than Bolt's complex 2016 stride. Modeling explains Bolt's victory by quantifying how his longer, albeit slower and more complex, stride proved more effective than Gatlin's rapid, shorter strategy.

In conclusion, the combined mathematical modeling (regression: freq/amp; FFT: freq/complexity) significantly explained performance differences. It quantified Bolt's evolution and the distinct Bolt-Gatlin strategies, demonstrating these techniques' analytical power on video data.

# 9 Evaluation

This investigation's tiered mathematical analysis provided quantitative insights into Bolt's and Gatlin's stride patterns and performance differences. Strengths lie in the systematic methodology, consistent data source and cleaning procedures, critical evaluation of processing steps, and the use of complementary regression and FFT analyses to reveal stride frequency, amplitude, and complexity. However, findings must be interpreted cautiously due to several limitations. These include constraints inherent in the data source (single video quality, manual tracking subjectivity, short analysis duration), assumptions made during processing (approximate scaling, hip subtraction simplification, potential effects of detrending), limitations of the models employed (PCC linearity, sinusoidal simplification, FFT resolution), and the restricted scope focusing solely on ankle kinematics while excluding other key biomechanical factors.

## Validity of Conclusions and Assumptions

Core assumptions (consistent video/tracking, model suitability) are reasonable, but limitations (tracking error, short segment, model simplification) necessitate cautious interpretation. Nevertheless, the models consistently quantified key distinctions: Bolt's decreased frequency and increased 2016 complexity, alongside Gatlin's higher frequency but lower amplitude. These findings directly support the conclusion that modeling significantly explains performance differences within the study's constraints, affirming the utility of these mathematical tools for video kinematic analysis.

## Appendix: Online Repository for Supporting Materials

All supporting materials for this investigation, including the Python scripts used for data processing and analysis (Tiers 2-5), generated graphs and visualizations, and detailed supplementary data files (e.g., processed coordinates, numerical outputs from correlations, regressions, and FFT), are available online in the project's GitHub repository:

`https://github.com/disCodeOri/Math-IA-programs-and-results-files`

Readers are encouraged to consult this repository for access to the code implementation, visual representations of the results, and the underlying data referenced throughout this paper.

# Bibliography


1. Savitzky, Abraham, and Marcel J. E. Golay. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry*, vol. 36, no. 8, 1964, pp. 1627-1639, `https://doi.org/10.1021/ac60214a047`

2. SciPy Developers. "scipy.signal.savgol_filter." *SciPy*, , Accessed 25 Mar. 2025, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html`

3. SciPy Developers. "scipy.optimize.curve_fit." *SciPy*, , Accessed 25 Mar. 2025, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html`

4. NumPy Developers. "numpy.corrcoef." *NumPy*, , Accessed 25 Mar. 2025, `https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html`

5. Huet, Serge, et al.. "Statistical Tools for Nonlinear Regression: A Practical Guide with S-PLUS and R Examples." *Springer*, , 2004,

6. Oppenheim, Alan V., and Ronald W. Schafer. "Discrete-Time Signal Processing." *Pearson*, 3rd ed., 2009,

7. Choi, K. S., et al.. "FFT-based analysis of human walking and running." *Computers in Biology and Medicine*, vol. 32, no. 6, 2002, pp. 477–491,

8. SciPy Developers. "scipy.fft." *SciPy*, , Accessed 25 Mar. 2025, `https://docs.scipy.org/doc/scipy/reference/fft.html`