

CI2 for creating and comparing confidence-intervals for time-series bivariate plots

David R. Mullineaux

School of Sport and Exercise Science, College of Social Science, University of Lincoln, Brayford

Pool, Lincoln, LN6 7TS, UK; dmullineaux@lincoln.ac.uk

Corresponding author: David R. Mullineaux

Conflict of interest

No conflict of interest.

Contributions

DRM oversaw the collection of the test data, and was solely responsible for all other aspects of the study including inception, data analysis, preparing the manuscript, and reviewing and approving the final manuscript.

Acknowledgements

Thank you to Rory Heaney for collecting the test data.

CI2 for creating and comparing confidence-intervals for time-series bivariate plots

Highlights

- CI2 quantifies the confidence-intervals (CI) of bivariate time-series plots.
- Determines if the CI between two bivariate time-series overlap.
- Any time-lag between two time-series can be incorporated into the analysis.
- Knee-ankle angles plots on standard v curved treadmills differ at heel-strike.
- Matlab code is provided to facilitate standardisation of use of CI2.

Abstract

Currently no method exists for calculating and comparing the confidence-intervals (CI) for the time-series of a bivariate plot. The study's aim was to develop 'CI2' as a method to calculate the CI on time-series bivariate plots, and to identify if the CI between two bivariate time-series overlap. The test data were the knee and ankle angles from 10 healthy participants running on a motorised standard-treadmill and non-motorised curved-treadmill. For a recommended 10+ trials, CI2 involved calculating 95% confidence-ellipses at each time-point, then taking as the CI the points on the ellipses that were perpendicular to the direction vector between the means of two adjacent time-points. Consecutive pairs of CI created convex quadrilaterals, and any overlap of these quadrilaterals at the same time or ± 1 frame as a time-lag calculated using cross-correlations, indicated where the two time-series differed. CI2 showed no group differences between left and right legs on both treadmills, but the same legs between treadmills for all participants showed differences of less knee extension on the curved-treadmill before heel-strike. To improve and standardise the use of CI2 it is recommended to remove outlier time-series, use 95% confidence-ellipses, and scale the ellipse by the fixed Chi-square value as opposed to the sample-size dependent F-value. For practical use, and to aid in standardisation or future development of CI2, Matlab code is provided. CI2 provides an effective method to quantify the CI of bivariate plots, and to explore the differences in CI between two bivariate time-series.

Keywords

Angle-angle; Biomechanics; Coordination; Phase-plot; Spatial; Variability

INTRODUCTION

Time-series data from biomechanics provides the opportunity to explore the relationship or interaction between multiple variables over time. In the case of two variables, each can be a function of each other (e.g. angular velocity and angle of one joint), or independent from each other (e.g. two joint angles). Variables can be explored as the ratio of each other [1], and although this facilitates comparison of the interaction between the two variables information is lost through changing the dimensions of the data. More commonly, data have been presented as phase or bivariate plots, which are often explored and summarised as a single value representing the entire data using the perimeter, area and perimeter:area ratio [2]. It is useful to calculate standard measures of variability, particularly on a point-by-point basis including using various forms of confidence-intervals (CI)[3]. These applications of CI have treated the abscissa (x-axis) as an “error free” variable. However, for bivariate plots both the abscissa and ordinate (y-axis) axes need to be treated as both containing error.

Treating both axes of data as containing error and providing a measure of variability at each time-point is achieved using vector coding (VC). There are two main approaches that incorporate both the vector angles and magnitudes [4] or include just the vector angles [5]. VC has been used to explore coordination and functional-variability [6], but VC has not been used as a CI as it would not be suitable as the variability is of the point-to-point vectors rather than the space encompassed by the points.

Several approaches to analysing bivariate plots exist in the area of spatial statistics, where error for both variables is incorporated, and requires both measures of central tendency (e.g. minimum-distance-centre, centroid, linear-directional-mean, weighted-mean-centre, mean-centre) and variability (e.g. area, standard-distance, standard-deviational-ellipse, confidence-ellipse). There are various strengths and weakness of each method, but the mean centre (i.e. average of points for each dimension) and confidence-ellipse (i.e. semi-major and semi-minor axes of the ellipse) are the most familiar to biomechanists often used to analyse centre of pressure data. In addition to the spatial aspect, time-series data includes a temporal aspect that needs to be incorporated into the calculation of CI for comparing any overlap between two time-series.

The temporal aspect ‘within’ a times-series can be accommodated by exploring the area between the CI bands, which can be achieved by creating convex shapes between adjacent CI along the time-series. Two times-

series can then be compared by analysing any overlap of the convex shapes at the same time-points. Further though there can be a temporal difference or time-lag 'between' time-series. The time-lag for univariate analyses can be corrected for through a time transformation [7] but these have not been validated for their impact on bivariate comparisons. Currently, only a simple quantification of the time-lag, such as through cross-correlations, is advisable to determine the time-lag to incorporate into the analysis. Hence, the aim was to develop a method named CI2 to calculate the CI on time-series bivariate plots, and to determine any overlap of the CI between two bivariate time-series at the same time or incorporating any time-lag between the series.

METHODS: The test data were the knee and ankle angles for running on a motorized standard-treadmill (Powerjog-J100, Power Sport International) and non-motorized curved-treadmill (Curve, Woodway). Healthy males ($n=10$; 21.4 ± 4.1 years; 75.6 ± 7.0 kg; 1.81 ± 0.08 m) ran for 30s at 2.11 ± 0.05 m/s (standard-treadmill) and 2.15 ± 0.11 m/s (curved-treadmill), calculated from the left-leg posterior toe displacement and stance duration. Five retroreflective markers placed on each leg (lateral: greater-trochanter, femoral-epicondyle and malleolus, and; dorsal first: metatarsal-phalange and inter-phalange joints) were captured at 200Hz (9 Raptor cameras; Cortex5.2 software; Motion Analysis Corporation). Data were analysed in Matlab (2015a; Mathworks) and smoothed at 5Hz using a fourth-order Butterworth low-pass filter. Strides were extracted from heel-strike to heel-strike based on the maximum and minimum anterior-posterior velocity of the toe markers, respectively [8]. Strides were 148 ± 8 frames long, hence all strides were time-normalized to a similar size of 151 data points using a cubic spline.

A 2-stage outlier detection method with settings of 0.0001 and 0.005 was applied [9] to improve the normal distribution resulting in 2.8 ± 1.4 strides as times-series outliers and a further 1.5 ± 1.4 strides as mismatching between joints were removed leaving 33.0 ± 5.5 strides. In the sagittal plane angles between vector 1 (greater-trochanter to femoral-epicondyle) and vector 2 (femoral-epicondyle to malleolus) defined the knee angle, and between vector 2 and vector 3 (malleolus to first metatarsal-phalange joint) defined the ankle angle. Each angle was offset-normalised by subtracting the mean stride, and the minimum values defined the maximum knee extension and maximum ankle dorsiflexion.

For each bivariate plot, CI2 was calculated in 4-stages (Matlab code; Supplementary material). The 95% confidence-ellipses were calculated (stage-1) with the CI taken as the points on the ellipse that were perpendicular to the direction vector, which was calculated as the ellipse axis angle minus the direction angle between the current and next mean points of the time-series (stage-2). Convex quadrilaterals were then created between each adjacent pairs of CI (stage-3). Any overlap of the quadrilaterals between the two time-series were calculated, including a time-lag element to compare the CI at the same and adjacent time-points (stage-4). Using the means of both CI for both angles, the peak cross-correlation between series 1 and 2 occurred at -0.5 ± 1.0 frames, hence a time-lag of ± 1 frame was used. The overlaps, as binary scores (yes or no), were calculated for four comparisons: left v right legs on standard-treadmill and curved-treadmill, and between legs for each treadmill. The range between CI, percentage agreement for intra-participant overlap of left v right legs and percentage agreement for group full-overlap and zero-overlap periods, indicating group similarities and differences, were calculated and presented as means \pm standard-deviations. The notation indicates the time-series and position of the element within the matrix (e.g. matrix P, series 1, positions 2 to 3 was P(1,2:3)).

1) Ellipse

Based on aspects of Principal Components Analysis [10] and modified Matlab code [11] an ellipse was fitted to the total number (n) of abscissa (x) and ordinate (y) data from a bivariate plot using a covariance matrix and matrix differential equations (where, c was the index of summation, \bar{x} was mean of x , and \bar{y} was mean of y).

a) Covariance matrix.

$$\begin{bmatrix} A & C \\ B & D \end{bmatrix} = \begin{bmatrix} 1/(n-1) \sum_{c=1}^n (x_c - \bar{x})^2 & 1/(n-1) \sum_{c=1}^n (x_c - \bar{x})(y_c - \bar{y}) \\ 1/(n-1) \sum_{c=1}^n (x_c - \bar{x})(y_c - \bar{y}) & 1/(n-1) \sum_{c=1}^n (y_c - \bar{y})^2 \end{bmatrix}$$

b) Eigen-values (where, $b=A+D$ and $d=AD-BC$), i.e. lengths of the semi axes of the ellipse.

$$\lambda_1 = (b - \sqrt{b^2 - 4d}) / 2$$

$$\lambda_2 = (b + \sqrt{b^2 - 4d})/2$$

c) Ellipse orientation adjustment, where if the covariance matrix elements were the same sign then $e=-1$, else $e=1$.

$$e = \begin{cases} -1 & \text{for } \text{sgn}(A) = \text{sgn}(B) = \text{sgn}(C) = \text{sgn}(D) \\ 1 & \text{else} \end{cases}$$

d) Eigen-vectors, i.e. direction of the ellipse axes.

$$[\vec{v}_1 \quad \vec{v}_2] = \begin{bmatrix} E & G \\ F & H \end{bmatrix} = \begin{bmatrix} B & B \\ \lambda_1 - A & \lambda_2 - A \end{bmatrix}$$

e) Scaled eigen-vectors (sum of squared versions scaled to unity) incorporating the ellipse orientation adjustment (stage-1c).

$$[\hat{v}_1 \quad \hat{v}_2] = \begin{bmatrix} I & K \\ J & L \end{bmatrix} = \begin{bmatrix} eE/\sqrt{E^2 + F^2} & G/\sqrt{G^2 + H^2} \\ eF/\sqrt{E^2 + F^2} & H/\sqrt{G^2 + H^2} \end{bmatrix}$$

f) Scaling factor of the inverse of the Chi-squared cumulative distribution function for probability p for 2 degrees of freedom (e.g. for $p=0.95$, $k=2.4478$). This fixed value scaling aids standardization as it is independent of sample size [12].

$$k = (-2\log(1 - p))^{0.5}$$

g) Ellipse data, where the perimeter was divided into n points from $t_{c=1}=0$ to $t_{c=n}=2\pi/n(n-1)$, hence $t_c=2\pi/n(c-1)$.

$$\begin{bmatrix} X_{c=1} & Y_{c=1} \\ \vdots & \vdots \\ X_n & Y_n \end{bmatrix} = k \cdot \begin{bmatrix} \cos(t_{c=1}) & \sin(t_{c=1}) \\ \vdots & \vdots \\ \cos(t_n) & \sin(t_n) \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix} \cdot [\hat{v}_1 \quad \hat{v}_2] + \begin{bmatrix} \bar{x} & \bar{y} \\ \vdots & \vdots \\ \times n & \times n \end{bmatrix}$$

2) Confidence-intervals (CI)

Where the data sets of x and y repeated r times to form a time-series (Fig. 1a), first the change between the centres of the current set (t) to the next set of time-points ($t=t+1$) was calculated to determine the direction

angles. Second, the difference between the ellipse angle and the direction angle were used to calculate the two points on the ellipse as the lower and upper CI.

a) Change in x and y were calculated using the single-sided difference for t=1 to t=r, and to maintain the same size matrix the last point equaled the change of the penultimate point.

$$\dot{x}_t = \begin{cases} \bar{x}_{t+1} - \bar{x}_t & \text{if } t = 1 \text{ to } t = r - 1 \\ \bar{x}_r - \bar{x}_{r-1} & \text{if } t = r \end{cases}$$

$$\dot{y}_t = \begin{cases} \bar{y}_{t+1} - \bar{y}_t & \text{if } t = 1 \text{ to } t = r - 1 \\ \bar{y}_r - \bar{y}_{r-1} & \text{if } t = r \end{cases}$$

b) Direction angle from the current to next set of time-point data.

$$\varphi_t = \text{atan2}(\dot{x}_t, \dot{y}_t)$$

c) Ellipse angle (using stage-1e).

$$\theta_t = \text{atan2}(K_t, L_t)$$

d) CI coordinates (for t=1 to t=r).

$$\begin{bmatrix} CI1x_t & CI1y_t \\ CI2x_t & CI2y_t \end{bmatrix} = k \cdot \begin{bmatrix} \cos(\varphi_t - \theta_t) & \sin(\varphi_t - \theta_t) \\ \cos(\varphi_t - \theta_t + \pi) & \sin(\varphi_t - \theta_t + \pi) \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_{1t}} & 0 \\ 0 & \sqrt{\lambda_{2t}} \end{bmatrix} \cdot [\widehat{v}_{1t} \quad \widehat{v}_{2t}] + \begin{bmatrix} \bar{x}_t & \bar{y}_t \\ \bar{x}_t & \bar{y}_t \end{bmatrix}$$

3) Convex hull of each pair of adjacent CI

Where the order of the CI from time t to t+1 created a simple or concave quadrilateral (Fig. 2a), the order of the CI was sorted to create a convex quadrilateral (Fig. 1b). Given that there were 4 points, the principles of Graham's scan [13] (stages 3a-c) and Cramer's rule (stage-3d) provided a simple solution.

a) Four CI coordinates (M) were offset by subtracting the CI coordinates with the minimum-y value (or if a tie the coordinates with the minimum-y and minimum-x). Note, the matrix was transposed from stage-2d.

$$N = M - [\min\{M_y\} \cdots \times 4] = \begin{bmatrix} CI1x_t & CI2x_t & CI1x_{t+1} & CI2x_{t+1} \\ CI1y_t & CI2y_t & CI1y_{t+1} & CI2y_{t+1} \end{bmatrix} - [\min\{M_y\} \cdots \times 4]$$

- b) Two argument inverse tangent (atan2) of each of the coordinates in N.

$$O = [\text{atan2}(N_1) \quad \text{atan2}(N_2) \quad \text{atan2}(N_3) \quad \text{atan2}(N_4)]$$

- c) M sorted from their corresponding minimum to maximum O.

$$P = [M_{O(1)} \quad M_{O(2)} \quad M_{O(3)} \quad M_{O(4)}]$$

- d) Third point (P(3)) tested to see if it was part of the convex hull (Fig. 2; point R) by assessing if the segments P(1) to P(3) and P(2) to P(4) intersected, and if they intersected then P(3) was part of the hull (Fig. 2a right), else the quadrilateral was concave (Fig. 2a middle) and P(3) was replaced with P(4).

- i) Four points created from P in the order 1, 3, 2 and 4.

$$Q = \begin{bmatrix} P(1)_x \\ P(1)_y \end{bmatrix}$$

$$R = \begin{bmatrix} P(3)_x \\ P(3)_y \end{bmatrix}$$

$$S = \begin{bmatrix} P(2)_x \\ P(2)_y \end{bmatrix}$$

$$T = \begin{bmatrix} P(4)_x \\ P(4)_y \end{bmatrix}$$

- ii) Three vectors created for: segment QR; segment ST, and; between points RT.

$$\overrightarrow{QR} = R - Q$$

$$\overrightarrow{ST} = T - S$$

$$\overrightarrow{RT} = T - R$$

- iii) Applying Cramer's rule using determinants, the x and y scaling parameters U and V were calculated. Note, \circ and \oslash indicate Hadamard product and division on an element-by-element basis, respectively, relevant when the matrices contain more than one pair of elements as in stage-4b.

$$U = (ST_x \circ RT_y - ST_y \circ RT_x) \oslash (QR_x \circ ST_y - QR_y \circ ST_x)$$

$$V = (QR_x \circ RT_y - QR_y \circ RT_x) \oslash (QR_x \circ ST_y - QR_y \circ ST_x)$$

iv) Where segments intersected (i.e. both U and V were between 0 and 1), P(3) was retained as it was part of the hull else P(3) was replaced with P(4).

$$P(3) = \begin{cases} P(3) & \text{if } 0 \leq U \& V \leq 1 \\ P(4) & \text{else} \end{cases}$$

4) Overlap of CI quadrilaterals for two series of data

The overlap of the quadrilaterals from two time-series 1 and 2 was determined as a binary score of yes or no overlap. To incorporate any time lag between the two series, quadrilaterals of series 1 at each time-point t (t=1 to m, where m=r-1) were compared to the quadrilaterals of series 2 at corresponding time-points (h), where h incorporated the time-lag g (e.g. g=±1). Hence h=t-g to t+g (if h<0 then h=1, or if h>m then h=m).

a) Combinations of the four points between the two series were created so that all four segments of the quadrilaterals were analysed. Hence, Q, R, S and T each contain 16 columns, where Q and R were for series 1, and S and T were for series 2.

$$Q_t = \begin{bmatrix} P(1,1)_{x_t} & \dots & \dots & \dots & P(1,2)_{x_t} & \dots & \dots & \dots & P(1,3)_{x_t} & \dots & \dots & \dots & P(1,4)_{x_t} & \dots & \dots & \dots \\ P(1,1)_{y_t} & \dots & \dots & \dots & P(1,2)_{y_t} & \dots & \dots & \dots & P(1,3)_{y_t} & \dots & \dots & \dots & P(1,4)_{y_t} & \dots & \dots & \dots \end{bmatrix}$$

$$R_t = \begin{bmatrix} P(1,2)_{x_t} & \dots & \dots & \dots & P(1,3)_{x_t} & \dots & \dots & \dots & P(1,4)_{x_t} & \dots & \dots & \dots & P(1,1)_{x_t} & \dots & \dots & \dots \\ P(1,2)_{y_t} & \dots & \dots & \dots & P(1,3)_{y_t} & \dots & \dots & \dots & P(1,4)_{y_t} & \dots & \dots & \dots & P(1,1)_{y_t} & \dots & \dots & \dots \end{bmatrix}$$

$$S_h = \left[\begin{pmatrix} P(2,1)_{x_h} & P(2,2)_{x_h} & P(2,3)_{x_h} & P(2,4)_{x_h} \\ P(2,1)_{y_h} & P(2,2)_{y_h} & P(2,3)_{y_h} & P(2,4)_{y_h} \end{pmatrix} \dots \times 4 \right]$$

$$T_h = \left[\begin{pmatrix} P(2,2)_{x_h} & P(2,3)_{x_h} & P(2,4)_{x_h} & P(2,1)_{x_h} \\ P(2,2)_{y_h} & P(2,3)_{y_h} & P(2,4)_{y_h} & P(2,1)_{y_h} \end{pmatrix} \dots \times 4 \right]$$

b) Whether quadrilaterals overlap was calculated using Q_t, R_t, S_h and T_h (stage-3d ii-iv). Where any of the 16 element-by-element U and V pairs were both between 0 and 1, then there was an overlap and so went on to stage-4c, else went to next stage.

$$W_{t,h} = \begin{cases} 1 & \text{if any } 0 \leq U \& V \leq 1, \text{ go to stage 4c} \\ \text{undefined} & \text{else, go to next stage} \end{cases}$$

i) Checked if one quadrilateral was inside the other (Fig. 2b-c).

1st) Quadrilaterals' coordinates were combined so the first point contained the minimum-y (or if a tie minimum-y and minimum-x), and this pair of coordinates were subtracted from all 8 points. This order meant that one only needed to check whether the second quadrilateral was inside the first, and so there was no need to check the reverse of whether the first quadrilateral was inside the second.

$$Z = \begin{cases} [P(1,1:4) \quad P(2,1:4)] - [P(1,1) \cdots \times 8] & \text{if } P(1,1)_y < P(2,1)_y \\ [P(2,1:4) \quad P(1,1:4)] - [P(2,1) \cdots \times 8] & \text{if } P(1,1)_y > P(2,1)_y \end{cases}$$

2nd) Two argument inverse tangent angles were calculated for Z.

$$\alpha = [\text{atan2}(Z_1) \quad \cdots \quad \text{atan2}(Z_8)]$$

3rd) Whether quadrilaterals overlap (Fig. 2b) was calculated where if all angles for quadrilateral 2 ($\alpha(5:8)$) were between angles $\alpha(2)$ and $\alpha(4)$ then went on to next stage, else the quadrilaterals did not overlap and went on to stage-4c.

$$W_{t,h} = \begin{cases} \text{undefined} & \text{if } \alpha(2) < \alpha(5:8) < \alpha(4), \text{ go to next stage} \\ 0 & \text{else, go to stage 4c} \end{cases}$$

ii) Whether quadrilaterals overlap (Fig. 2c) was determined using U and V where if QR segment (P(1,1) to P(2,1)) intersected either the ST segments (P(1,2) to P(1,3); P(1,3) to P(1,4)) then the quadrilaterals did not overlap.

$$W_{t,h} = \begin{cases} 0 & \text{if } 0 \leq U \&V \leq 1, \text{ no overlap} \\ 1 & \text{else, overlap} \end{cases}$$

c) The overlap for series 1 with 2 ($\beta(1)$) and series 2 with 1 ($\beta(2)$) were created (where indices <1 were made 1, and indices >m were made m). $\beta(1)$ and $\beta(2)$ were the same for no time lag ($g=0$), but otherwise differed.

$$\beta(1)_t = \sum_{t=1, h=t-g}^{t=m, h=t+g} W_{t,h} = \begin{bmatrix} W_{1,t-g} + \cdots W_{1,t+g} \\ \vdots \\ W_{m,t-g} + \cdots W_{m,t+g} \end{bmatrix} \{ \text{if } \beta(1)_t \geq 1, \text{ overlap} \}$$

$$\beta(2)_h = \sum_{t=1, h=t-g}^{t=m, h=t+g} W_{t,h} = \begin{bmatrix} W_{1,t-g} + \cdots W_{1,t+g} \\ \vdots \\ W_{m,t-g} + \cdots W_{m,t+g} \end{bmatrix} \{ \text{if } \beta(2)_h \geq 1, \text{ overlap} \}$$

d) Quadrilaterals overlap where $\gamma \geq 1$, determined by combining $\beta(1)$ and $\beta(2)$.

$$\gamma = \beta(1) + \beta(2) \{if \gamma_t \geq 1, overlap\}$$

RESULTS:

The output from CI2 produces CI bands for each time-series, which were divided up by the quadrilaterals formed between adjacent pairs of CI. The CI range had low variability of $\leq 2^\circ$ (CI range: standard-treadmill left-leg $8 \pm 2^\circ$, right-leg $6 \pm 2^\circ$; curved-treadmill left-leg $7 \pm 1^\circ$, right-leg $6 \pm 1^\circ$) and was similar across the stride-time showing no clear periods of greater variability (Fig. 3). A sample bivariate plot for one participant shows good similarity, or overlap, in the knee-ankle angles coupling relationship for the left v right leg on the standard-treadmill (Fig. 3a) and on the curved-treadmill (Fig. 3b). The most notable difference between treadmills was before heel-strike with less knee extension on the curved-treadmill. In comparing the same leg between treadmills there was poorer similarity, which arose from the different patterns before heel-strike and some temporal differences throughout the stride (Fig. 3c-d).

Within participants there was good agreement between conditions (Fig. 3), which was best between legs on the standard-treadmill ($89 \pm 9\%$ -agreement; Table 1) and moderate within legs between treadmills (34-36%-agreement). In contrast, there was wide inter-participant variability, which makes comparisons to the 'mythical average' trace unrepresentative of any participant [14] hence periods where all participants had the same overlap were determined. Between legs there was greater group similarity on the standard-treadmill at 37%-full-overlap versus 9%-full-overlap on the curved-treadmill, which was further supported by 0%-zero-overlap for both treadmills (Fig. 4a-b). In contrast, between treadmills there was poorer group agreement with 0%-full-overlap and 13-15%-zero-overlap. This period of zero-overlap where neither leg overlapped for all participants occurred before heel-strike (left-leg 85-100%-stride-time; right-leg 87-100%-stride-time; Fig. 4c-d).

DISCUSSION:

This paper demonstrates CI2 as an effective method for quantifying the bivariate CI and determining any overlap between two time-series. The CI2 method identified clear differences in the knee-ankle angles relationship between the two treadmills, particularly at the end of the swing phase in all participants. This was probably owing to the raised and inclined position of the curved-treadmill versus the lower and flat position of

the standard-treadmill, or contribution of the energy imparted from motorised treadmills [15]. There was large inter-participant variability in the coordination patterns between participants. Individuality often limits group comparisons, hence periods of when all participants demonstrated the same response of zero-overlap or full-overlap were highlighted as the 'group' response (Fig. 4). Future research should be designed to explore these test data in more depth.

There are a few assumptions of CI2. First, it was assumed the confidence-ellipse best encapsulates the variability of the data. The distribution of the data should primarily possess multivariate normality. Owing to the large number of ellipses it was unlikely that this assumption would be met for all ellipses. To improve this assumption two recommendations are proposed. First, outlier trials should be removed, for example using the two-stage method for time-series data that leads to the improvement in the number of time-points being normally distributed [9]. Second, the trial size should be at least 10, as through simulations this was found to provide a 'sufficiently' low bias in the ellipse area versus convex area [16]. Related to trial size, as recommended for standardisation [12], instead of using the bivariate F-distribution that varies with sample size [17] the ellipse was scaled using the Chi-square distribution that leads to a fixed scaling factor for all sample sizes. Others propose calculating CI using different F-values [e.g. 17 v 12], bootstrapping [18], randomness-based distributions [19] and minimal detectable difference bands [2]. These would all generally lead to larger CI, but in this application of CI2 given the large number of ellipses and individuality across groups the fixed Chi-squared method with smaller CI was deemed preferable to protect against type-II errors.

The presence of a time-lag between time-series was assessed through cross-correlation, which revealed a small time-lag of ≤ 1 frame that was incorporated into the analysis. Cross-correlation provides a simple solution, which assumes a uniform time-shift throughout the entire time-series, but this was unlikely, as evidenced from the matched quadrilaterals where there are varying amounts of time-lags at different time-points (Fig. 3). Potentially more advanced time-aligning methods could be used, although none have yet been developed for bivariate data. In stage-4c, any overlap of series 1 on 2 and 2 on 1 was calculated. These are the same when there is no time-lag, but with a time-lag introduced a series can overlap another series more often on one series than the other. The overlaps from both ways were therefore combined to create a single indication of the overlap between the two time-series.

There are other methodological considerations. Data are often offset-normalised, such as through subtracting the standing anatomical position, or as in this study, subtracting the mean trial. Some form of offset-normalisation is advised, which, reduces spatial variability between time-series. Scale-normalisation would also be required where the data is of a different range, which would be more apparent in a bivariate plot of data with different units. The CI for each plot showed consistent variability across the stride. Larger variability has been found at turning points, and although potentially meaningful the possibility of a statistical artefact has been identified [5]. This artefact appears to be reduced in CI2, and where rapid changes of direction could lead to distortion of the quadrilateral in the presence of high accelerations the sampling frequency should be increased to reduce this impact. In comparing future studies with similar frequencies of movement, there is an interaction of a higher sampling frequency decreasing the time-span or width of quadrilaterals, which could be compensated for by increasing the time-lag.

The mathematical process of CI2 can be achieved using Excel, which may be beneficial for an educational task using a few time-points. On a practical level, code is required to loop through the large number of calculations, hence Matlab code is provided (Supplementary material). This will facilitate standardisation in the use of CI2, where it is recommended that there are 10+ trials, data are offset-normalised, outlier time-series are removed, 95% confidence-ellipses are calculated, and the fixed Chi-square value is used for the scaling. CI2 provides an effective method to quantify the CI of bivariate plots and to explore differences in the CI between two time-series.

REFERENCES:

1. Evangelidis PE, Pain MT, Folland J. Angle-specific hamstring-to-quadriceps ratio: a comparison of football players and recreationally active males. *J Sport Sci* 2015;33(3):309-19.
2. Garofalo P, Cutti AG, Filippi MV, Cavazza S, Ferrari A, Cappello A, Davalli A. Inter-operator reliability and prediction bands of a novel protocol to measure the coordinated movements of shoulder-girdle and humerus in clinical settings. *Med Biol Eng Comput* 2009;47(5):475-86.
3. Hershler C, Milner M. Angle-angle diagrams in the assessment of locomotion. *Am J Phy Med* 1980;59(3):109-25.

4. Tepavac D, Field-Fote EC. Vector coding: a technique for quantification of intersegmental coupling behaviours. *J Appl Biomech* 2001;17:259-70.
5. Heiderscheit BC, Hamill J, van Emmerik REA. Variability of stride characteristics and joint coordination among individuals with unilateral patellofemoral pain. *J Appl Biomech* 2002;18(2):110-21.
6. Cunningham TJ, Mullineaux DR, Noehren B, Shapiro R, Uhl TL. Coupling angle variability in healthy and patellofemoral pain runners. *Clin Biomech* 2014;29(3):317-22.
7. Helwig NE, Hong S, Hsiao-Weckslar ET, Polk JD. Methods to temporally align gait cycle data. *J Biomech* 2011;44(3):561-6.
8. Zeni JA Jr, Richards JG, Higginson JS. Two simple methods for determining gait events during treadmill and overground walking using kinematic data. *Gait Posture* 2008;27(4):710-4.
9. Mullineaux DR, Irwin G. A simple outlier detection method for intra-subject time-series data. In: Sato K, Sands WA, Mizuguchi S, editors. *Proceedings of the 32nd Conference of the International Society of Biomechanics in Sports*. Johnson City, TN: Eastern Tennessee University; 2014;220-3.
10. Oliveira LF, Simpson DM, Nadal J. Calculation of area of stabilometric signals using principal component analysis. *Physiol Meas* 1996;17(4):305-12.
11. Duarte M, Zatsiorsky VM. Effects of body lean and visual information on the equilibrium maintenance during stance. *Exp Brain Res* 2002;146(1):60-9.
12. Schubert P, Kirchner M. Ellipse area calculations and their applicability in posturography. *Gait Posture* 2014;39(1):518-22.
13. Graham RL. An efficient algorithm for determining the convex hull of a finite planar set. *Inform Process Lett* 1972;1:132-3.
14. Dufek JS, Bates BT, James CJ, Stergiou N. Interactive effects between group and single-subject response patterns. *Hum Movement Sci* 1995;14:301-23.
15. Savelberg HH, Vorstenbosch MA, Kamman EH, van de Weijer JG, Schambardt HC. Intra-stride belt-speed variation affects treadmill locomotion. *Gait Posture* 1998;7(1):26-34.
16. Jackson AL, Inger R, Parnell AC, Bearhop S. Comparing isotopic niche widths among and within communities: SIBER - Stable Isotope Bayesian Ellipses in R. *J Anim Ecol* 2011;80(3):595-602.
17. Fraser DAS, Guttman I. Tolerance regions. *Ann Math Stat* 1956;27:162-79.

18. Lenhoff MW, Santner TJ, Otis JC, Peterson MG, Williams BJ, Backus SI. Bootstrap prediction and confidence bands: a superior statistical method for analysis of gait data. *Gait Posture* 1999;9(1):10-7.
19. Pataky TC, Vanrenterghem J, Robinson MA. Zero- vs. one-dimensional, parametric vs. non-parametric, and confidence interval vs. hypothesis testing procedures in one-dimensional biomechanical trajectory analysis. *J Biomech* 2015;48(7):1277-85.

Fig. 1. Overview of representing two-dimensional time-series confidence-intervals (CI) as quadrilaterals and determining where there is any overlap of the quadrilaterals between two sets of time series. A: data (o) at three sets of time-points (t_1 to t_3), each encompassed by 95% confidence-ellipses (solid elliptical lines), with ellipse axes (dotted lines) and time-series 95%CI points (*) are used to create 95%CI bands (dashed lines). Arrows indicate direction vectors between consecutive ellipses. B: 95%CI points (*) from adjacent time-points are joined to create convex quadrilaterals along the time-series. C: where corresponding quadrilaterals from two sets of time-series partially overlap, then the overlapping quadrilaterals from each time-series are considered to overlap and are shaded.

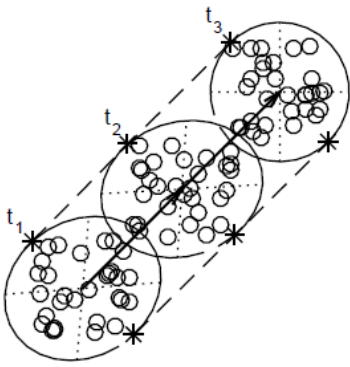
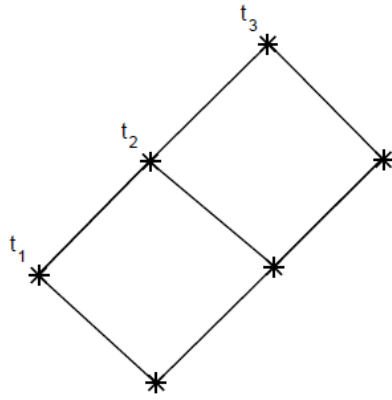
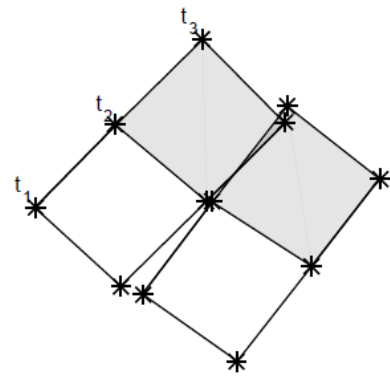
Fig. 2. Checks in determining the overlap of quadrilaterals from two sets of bivariate time-series confidence-intervals (CI). A: the extracted four 95%CI points from adjacent time-points when joined can create shapes classified as simple (left), concave (middle) or convex (right). Creating vectors QR and ST between various points are tested to see if they are simple, concave and convex, and points reordered or replaced to create a convex quadrilateral (see stage-3d). B: to assess if two quadrilaterals overlap where no sides of the two quadrilaterals intersect, first assess if all the points of the second quadrilateral (dashed lines) fall outside of the shaded area (none do; see stage-4bi). C: where all points fall outside of shaded areas, check if vector (dot-dashed lines) from lower point of quadrilateral 1 (solid line) to any point on quadrilateral 2 (dashed lines) crosses the upper sides of the quadrilateral 1 ('a' does not cross hence quadrilaterals overlap; 'b' does cross hence quadrilaterals do not overlap; see stage-4bii).

Fig. 3. Knee-ankle angles bivariate plots for one participant for four comparisons of two conditions (series 1 and 2). The 95%CI for series 1 are solid lines (left-leg or standard-treadmill) and for series 2 are dashed lines (right-leg or curved-treadmill). Time periods of overlap are shaded (light or medium-light grey) and periods of no-overlap or differences between series are white, ignoring the highlighted quadrilateral every 10% of stride-time shaded for series 1 (black) and series 2 (medium-dark grey) that illustrate the temporal alignment between series. Data is from heel-strike to heel-strike, moving in the direction of the arrow (indicated in D), with the approximate location of heel-strike (0%-stride-time) and toe-off ($56.4 \pm 0.8\%$ -stride-time; indicated in A). Minimum values are maximum knee extension and maximum ankle dorsiflexion.

Fig. 4. Periods of overlap for participant by stride-time for four comparisons of two series of knee-ankle angles bivariate data. Series 1 was left-leg or standard-treadmill, and series 2 was right-leg or curved-treadmill. Shaded regions indicate where all 10 participants for series 1 and 2 overlapped (i.e. group full-overlap; grey) or none overlapped (i.e. group zero-overlap; black), which indicates group similarities or differences within the comparison. Data were from heel-strike to heel-strike (0%-stride-time to 100%-stride-time), and toe-off occurs at $56.4 \pm 0.8\%$ -stride-time.

Table 1. Intra-participant and groups similarities presented as percentage agreement for intra-participant overlap, group full-overlap and group zero-overlap, for knee-ankle angles bivariate plots on a motorised standard-treadmill and non-motorised curved-treadmill for both legs (n=10 healthy males).

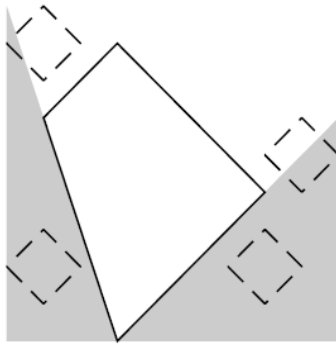
Agreement			
	Intra-participant	Group full-overlap	Group zero-overlap
Left v Right legs			
Standard	89±9%	37%	0%
Curved	79±21%	9%	0%
Standard v Curved			
Left-leg	36±14%	0%	15%
Right-leg	34±16%	0%	13%

A**B****C**

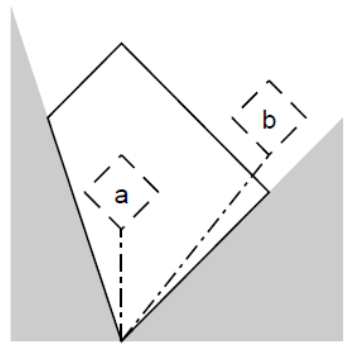
A



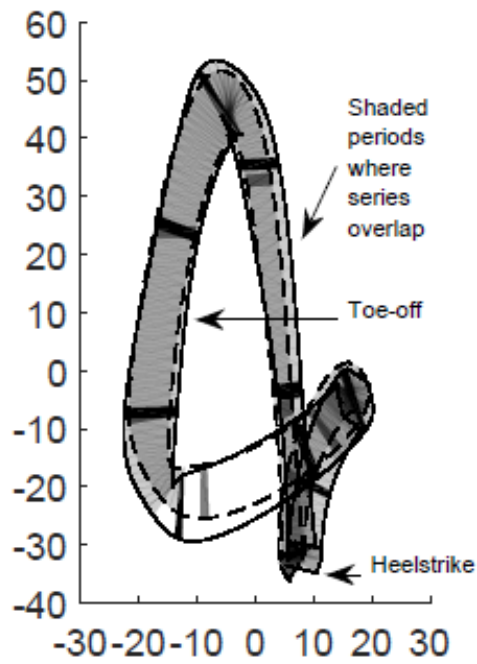
B



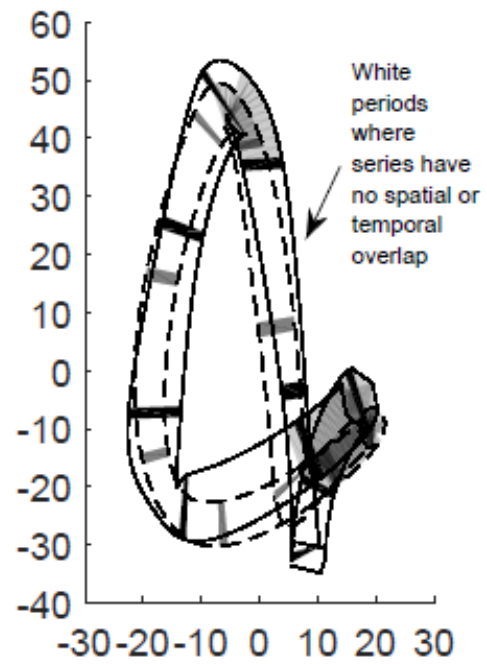
C



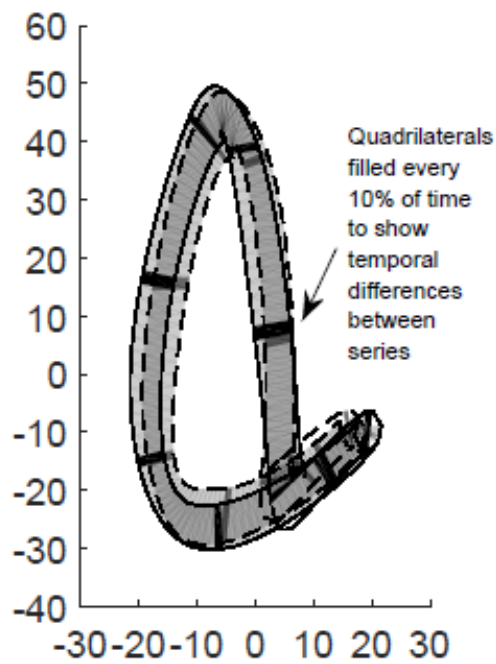
A: Left v Right legs (standard)



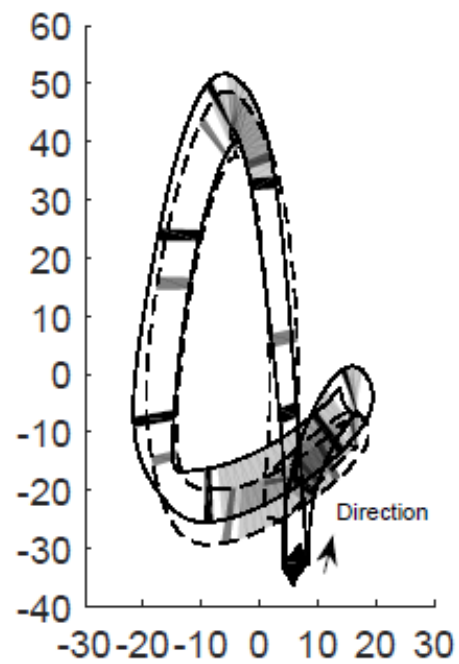
C: Standard v Curved (left leg)



B: Left v Right legs (curved)

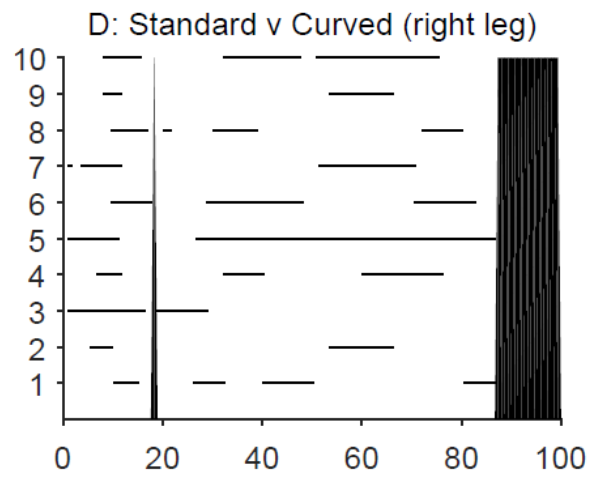
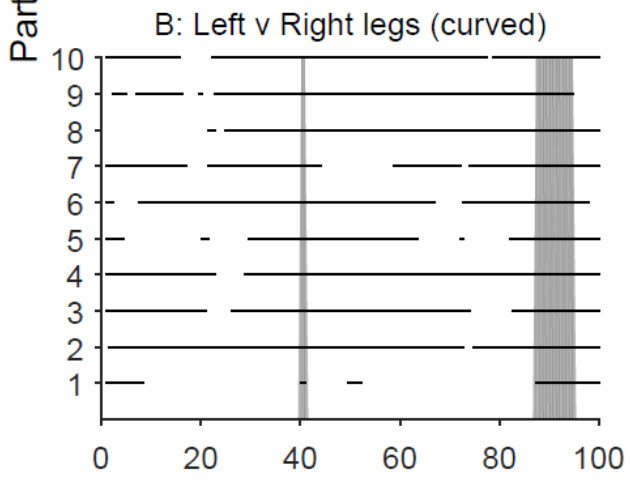
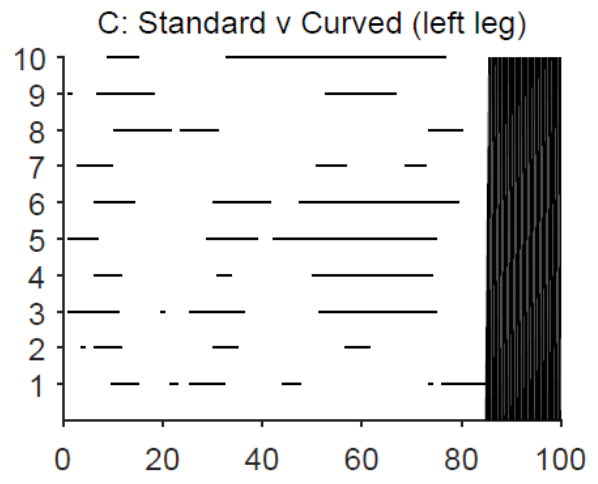
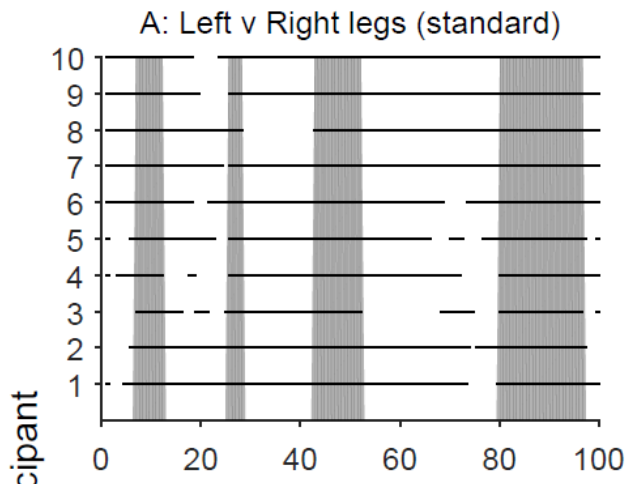


D: Standard v Curved (right leg)



Knee angle (°)

Ankle angle (°)



Stride time (%)

```

function [gamma,CIseries1,CIseries2]=myCI2(xAin,yAin,xBin,yBin,p,g,plotsyes,tshade)
% output: overlap of two bivariate times-series (0 no overlap; 1 overlap); option: plots
% input: xAin,yAin,xBin,yBin: two xy series 1 and 2; n>=3 trials in columns (same n
within series, but n can be different between series 1 and 2)
% input: time-points in rows same length: n>=2; good if before have removed outliers &
offset normalised
% input: p is probability for confidence-interval of ellipse (e.g. 0.95)
% input: g is timelag from 0 (exact match) to g forward & backwards (so if any of series
A overlaps B-g to B+g then identified as overlap)
% input: option: plotsyes as 'yes' for plots; tshade=1 to r (e.g. 10 shade every 10th
quads on plots to show any time diff between series)
% See accompanying paper for explanations: Mullineaux DR. CI2 for creating and comparing
confidence-intervals for time-series bivariate plots Gait Posture. 2017;xx
% -----
[r,~]=size(xAin);
m=r-1; %connecting current to next CI to create quadrilaterals reduces size by 1
%-----STAGE1&2 ELLIPSE AND ELLIPSE CONFIDENCE INTERVALS-----
% calculate 2D CI for each bivariate series A and B
[x1A,y1A,x2A,y2A]=ellipseCIIts(xAin,yAin,p);
[x1B,y1B,x2B,y2B]=ellipseCIIts(xBin,yBin,p);
CIseries1=[x1A,y1A,x2A,y2A];
CIseries2=[x1B,y1B,x2B,y2B];
% -----STAGE3 CONVEX HULL, AND PLOT OPTION-----
% create quadrilaterals for series 1 and 2 (and option to plot data)
switch plotsyes % creates figure where option selected
    case 'yes'; figure; hold on;
end
% create matrices before loop (5 points as 4 corners +1 to close quadrilateral)
Px1=NaN(5,m); Py1=NaN(5,m); Px2=NaN(5,m); Py2=NaN(5,m);
for T=1:m; % T is t, but use capital version for plots
    [Px1(:,T),Py1(:,T)]=convexquad([x1A(T:T+1,1);x2A(T:T+1,1)], [y1A(T:T+1,1);y2A(T:T+1,1)]);
    [Px2(:,T),Py2(:,T)]=convexquad([x1B(T:T+1,1);x2B(T:T+1,1)], [y1B(T:T+1,1);y2B(T:T+1,1)]);
    switch plotsyes
        case 'yes'
            fill(Px1(:,T),Py1(:,T),'w','FaceAlpha',0.0,'EdgeColor','none')
            fill(Px2(:,T),Py2(:,T),'w','FaceAlpha',0.0,'EdgeColor','none')
            TT=mod(T,tshade); % shade/colour every tshade quad to see any time-lag between series
            if TT<=0
                fill(Px1(:,T),Py1(:,T),'k','FaceAlpha',1,'EdgeColor','none')
                fill(Px2(:,T),Py2(:,T),'k','FaceAlpha',.5,'EdgeColor','none')
            end
    end
    axis equal; % NOTE: if series different units, then normalise scales for this to look ok
    plot([x1A,x2A],[y1A,y2A],'k');
    plot([x1B,x2B],[y1B,y2B],'k','LineStyle','--');
end
end

% -----STAGE4 OVERLAP OF QUADRILATERALS-----
% finds overlap of all quadrilaterals from series1 with all quadrilaterals from series2
beta1=zeros(m,1); %Series 1; default zeros is no overlap
beta2=zeros(m,1); %Series 2; default zeros is no overlap (beta1 and beta2 will be same
if no timelag)
Q=NaN(2,16); R=NaN(2,16); S=NaN(2,16); T=NaN(2,16); %16 to compare 4 vectors of quad1 to 2
for t=1:m
    % STAGE4a: create combinations of the 16 vectors; series1 vectors
    Q(1,:)=[repmat(Px1(1,t),1,4),repmat(Px1(2,t),1,4),repmat(Px1(3,t),1,4),repmat(Px1(4,t),1,4)];
    Q(2,:)=[repmat(Py1(1,t),1,4),repmat(Py1(2,t),1,4),repmat(Py1(3,t),1,4),repmat(Py1(4,t),1,4)];
    R(1,:)=[repmat(Px1(2,t),1,4),repmat(Px1(3,t),1,4),repmat(Px1(4,t),1,4),repmat(Px1(1,t),1,4)];
    R(2,:)=[repmat(Py1(2,t),1,4),repmat(Py1(3,t),1,4),repmat(Py1(4,t),1,4),repmat(Py1(1,t),1,4)];
    for hh=t-g:1:t+g
        h=hh; %these few lines to adjust indexes to stay within ranges

```

```

    if hh<=0
        h=1;
    elseif hh>=m
        h=m;
    end
S(1,:)= repmat(Px2(1:4,h)',1,4); %series2 vectors
S(2,:)= repmat(Py2(1:4,h)',1,4);
T(1,:)= repmat([Px2(2:4,h)',Px2(1,h)],1,4);
T(2,:)= repmat([Py2(2:4,h)',Py2(1,h)],1,4);
% STAGE4b: test intersect of all 16 vectors
W=segmentcross(Q,R,S,T);
% STAGE4bi: if no intersect, now check if quadrilateral 2 not inside quadrilateral 1
    if W<=0 %skips this section if already know there is an overlap
% find which quadrilaterals has lowest y value, and subtract min y from all y's
% order so only need check 2 is inside 1 (and no need to check reverse that 1 inside 2)
Z=[Px1(1:4,t)',Px2(1:4,h)';Py1(1:4,t)',Py2(1:4,h)'];
[~,mincol]=min(Z(2,:));
Z=Z-repmat(Z(:,mincol),1,8);
    if mincol>=5 %switch series 1 and 2 for that with minimum
        Z=[Z(:,5:8),Z(:,1:4)];
    end
% check all points quad2ofZ are not between 2nd and 4th angles (QtoR and QtoT) of
quadlofZ (i.e. outside sides of quadlofZ)
alpha=atan2(Z(2,:),Z(1,:)); % QQ,QR,QS,QT are angles of quadlofZ
alphatest=sum(sign(alpha(5:8)-
repmat(alpha(2),1,4)))+sum(sign(repmat(alpha(4),1,4)-alpha(5:8)))); %will give 8 if all
points between angles QR and QT
    if alphatest<=7 %as not all 8 + then one quad can't be inside the other
        W=0;
    else
% STAGE4bii: if all 8 between, check if quad1Q to quad2Q crosses quad1RS or quad1ST
        Q2=[Z(:,1),Z(:,1)]; % [quadlofZ Q, quadlofZ Q]
        R2=[Z(:,5),Z(:,5)]; % [quad2 ofZ Q, quad2 ofZ Q]
        S2=[Z(:,2),Z(:,3)]; % [quadlofZ R, quadlofZ S]
        T2=[Z(:,3),Z(:,4)]; % [quadlofZ S, quadlofZ T]
        W=segmentcross(Q2,R2,S2,T2);
        if W>=1 %if crosses=1, then outside quadrilateral, i.e. no overlap
            W=0;
        else
            W=1; %i.e. no cross, so quad2ofZ within quadlofZ, so quads overlap
        end
    end
end
    beta1(t)=beta1(t)+W; % STAGE4c
    beta2(h)=beta2(h)+W;
end
end
gamma=isfinite(1./(beta1+beta2)); % STAGE4d
%-----
switch plotsyes
    case 'yes'% plots traces with shading/colour showing where no overlap
        fill(Px1(:,gamma),Py1(:,gamma),'k','FaceAlpha',.2,'EdgeColor','none')
        fill(Px2(:,gamma),Py2(:,gamma),'k','FaceAlpha',.2,'EdgeColor','none')
    end
%-----
end % end of main function
%=====
%-----STAGE1 ELLIPSE-----
function [XY,theta,IJKL,lambdas]=ellipse(x,y,p) %calculates ellipse
%Input: x and y are 1 column each; p probability to scale ellipse, e.g. 0.95
%Output: XY: ellipse data; theta: axes angles; IJKL/lambdas: eigen vectors/values
%modified from: Duarte M, Zatsiorsky VM. Exp Brain Res 2002;146(1):60-9.
k=(-2*log(1-p))^0.5; %chi scales ellipse (e.g. otherwise 95% would be 85.35%)
ABCD=cov(x,y); %2x2 covariance matrix of x and y
[IJKL,lambdas]=eig(ABCD); %scaled eigen vectors and values

```

```

n=length(x); %number of time-points used to create circle intervals
c=linspace(0, (2*pi)/n*(n-1), n)'; %circle intervals; bias if use (0,2*pi,n)
XYcentered=k*[cos(c), sin(c)]*sqrt(lambdas)*IJKL; %ellipse around [0,0]
XY=(XYcentered+repmat([mean(x), mean(y)], n, 1)); %ellipse around [meanx, meany]
theta=atan2(IJKL(2,:), IJKL(1,:)); % ellipse axes angles
end
%-----STAGE2 ELLIPSE CONFIDENCE INTERVALS-----
function [CI1x, CI1y, CI2x, CI2y]=ellipseCIts(x, y, p) %calculates ellipse CI
% input: x and y data in 1 column; p probability scaling for ellipse, e.g. 0.95
% output: two CI for bivariate time-series. CI relative to direction of time-series
[r, ~]=size(x);
k=(-2*log(1-p)).^0.5; %chi scales ellipse (e.g. otherwise 95% would be 85.35%)
% find angle of general direction of time-series movement
Xmean=mean(x, 2); Ymean=mean(y, 2);
Xchange=diff(Xmean); Xchange=[Xchange; Xchange(1, end)]; %repeat final point
Ychange=diff(Ymean); Ychange=[Ychange; Ychange(1, end)];
phi=atan2(Ychange, Xchange);
% preset variables before loop
CI1x=NaN(r, 1); CI1y=NaN(r, 1); CI2x=NaN(r, 1); CI2y=NaN(r, 1);
for t=1:r
[~, thetas, IJKL, lambdas]=ellipse(x(t,:), y(t,:), p); % retrieve ellipse analysis
omega2=thetas(2)-phi(t); %make ellipse angle relative to direction angle
omega1=omega2+pi(); %find second angle relative to direction
%calculate first and second CI (i.e. positions on ellipse relative to direction)
CI1=k*[cos(omega1), sin(omega1)]*sqrt(lambdas)*IJKL; %matrix order correct to work
CI1x(t)=CI1(1); CI1y(t)=CI1(2);
CI2=k*[cos(omega2), sin(omega2)]*sqrt(lambdas)*IJKL;
CI2x(t)=CI2(1); CI2y(t)=CI2(2);
end
CI1x=CI1x+Xmean; CI1y=CI1y+Ymean; CI2x=CI2x+Xmean; CI2y=CI2y+Ymean; % add mean to CI
end
%-----STAGE3 CONVEX HULL-----
function [Px, Py]=convexquad(Mx, My) %convex hull of 4 points (similar to Graham's scan)
% input: 4 x-points in 1 column; 4 y-points in 1 column;
% output: x y in convex order (creates quadrilateral, triangle if 1 point inside hull)
[ymint]=find(My<=min(My)); % find min y
if length(ymint)>=2 % if >1 min(My), find min(My) with min(Mx)
[~, ymint2]=min(Mx(ymint));
ymint=ymint(ymint2);
end
Nx=Mx-repmat(Mx(ymint, 1), 4, 1); %subtract x for min-y point from all x points
Ny=My-repmat(My(ymint, 1), 4, 1); %subtract min-y from all y points
O=atan2(Ny, Nx); %calculate angle (range will be 0 to 180)
[~, korder]=sort(O); %sort on angle (small to large; i.e. anti-clockwise)
Px=Mx(korder); Px=[Px; Px(1)]; %add first point at end to close shape
Py=My(korder); Py=[Py; Py(1)];
%check if convex (i.e. if third point within quadrilateral replace else keep)
Q=[Px(1, 1); Py(1, 1)]; R=[Px(3, 1); Py(3, 1)]; %convex, so QR is points 1 and 3
S=[Px(2, 1); Py(2, 1)]; T=[Px(4, 1); Py(4, 1)]; %convex, so ST is points 2 and 4
W=segmentcross(Q, R, S, T); %test if QR intercepts ST
if W<=0; %convex will intercept (>=1), but if 0 no intercept so replace p3 with p4
Px(3, 1)=Px(4, 1); Py(3, 1)=Py(4, 1);
end
end
%-----STAGE3d & 4b & 4bii SEGMENT INTERSECT-----
function W=segmentcross(Q, R, S, T) %calculates intersect of QR and ST using Cramer's rule
% input: four points (QRST, x row 1, y row 2); columns n pairs of coordinates
% output: 0 if no intersection between any n pairs; 1 if any pair of vectors intersect
QR=R-Q; ST=T-S; RT=T-R; % create vectors from points
U=(ST(1, :).*RT(2, :)-ST(2, :).*RT(1, :))./(QR(1, :).*ST(2, :)-QR(2, :).*ST(1, :));
V=(QR(1, :).*RT(2, :)-QR(2, :).*RT(1, :))./(QR(1, :).*ST(2, :)-QR(2, :).*ST(1, :));
U=(U<1) & (U>0); % reduces to 0 (not meet criteria) or 1 (meets criteria)
V=(V<1) & (V>0);
W=sum(sum(U+V, 1))>=2; %reduces to 1 (both criteria=1) else 0, then sums for single value
end%=====

```