Windowed Cross–Correlation and Peak Picking for the Analysis of Variability in the Association Between Behavioral Time Series

Steven M. Boker
Department of Psychology
The University of Notre Dame

Minquan Xu
Amdocs St. Louis Development Center

Jennifer L. Rotondo
Augustana College

Kadija King
University of Akron

January 16, 2002

Abstract

Cross–correlation and most other longitudinal analyses assume that the association between two variables is stationary. Thus, a sample of occasions of measurement is expected to be representative of the association between variables regardless of the time of onset or number of occasions in the sample. We propose a method to analyze the association between two variables when the assumption of stationarity may not be warranted. The method results in estimates of both the strength of peak association and the time lag when the peak association occurred for a range of starting values of elapsed time from the beginning of an experiment.

This work was supported in part by NIA grant R29 AG14983 and a grant from the University of Notre Dame. We also wish to thank David Parker, Melissa Sturge–Apple and Eric Covey whose assistance is greatly appreciated in gathering the example data presented here. We also wish to express our thanks to Scott Maxwell, Kathy Eberhard and four anonymous reviewers who provided many helpful editorial comments. Some of the example data has previously appeared in Boker and Rotondo (in press). Correspondence may be addressed to Steven M. Boker, Department of Psychology, The University of Notre Dame, Notre Dame Indiana 46556, USA; email sent to sboker@nd.edu; or browsers pointed to http://www.nd.edu/˜ sboker. C code to calculate the windowed cross–correlation matrices and Splus code for the peak picking algorithm are available for free on this web site.

Introduction

Many psychological experiments involve data that are comprised of multiple observations of the same variable over time. These data may be only a few waves of data in a panel study, from tens to hundreds of observations in a self–report journaling study or as many as tens of thousands of observations per variable in psychophysiological time series such as EEG, EKG or motion capture studies. In most cases, multiple variables are being observed (or as it is sometimes expressed in the time series literature, *sampled*) at each occasion of measurement. The data for which the methods in this article are appropriate include at least 100 occasions of measurement on at least two variables for one or more participants. While data with fewer occasions of measurement could be used, the method's usefulness would be reduced. The methods we propose use Pearson product moment correlations to estimate bivariate relationships between continuous variables. In addition, the algorithms presented here are specific to data that have equal intervals of time between observations, although in principle they could be adapted to unequal intervals.

The reason multiple occasions of measurement are included in a research design is generally because one wishes to understand not only the relationship between multiple variables at the same moment in time, but also the relationships between these variables as they change over time. Thus the relationship between variables observed at the same occasion of measurement may only be part of the story. Variables observed at different occasions of measurement may also be related to one another. These relationships between variables may have different strengths depending on the interval of time separating the measurements. The structure of the way that this relationship between variables changes as the interval between occasions of measurement changes can be an extremely informative diagnostic as to the nature of the underlying processes that gave rise to the data.

A number of methods are in common use to model these relationships between multiple variables at multiple occasions of measurement. These methods include, autocorrelation and cross-correlation, autoregressive and cross–lagged structural models (see e.g. Cook, Dintzer, & Mark, 1980; West & Hepworth, 1991), multivariate time series methods such as autoregressive moving average models (see e.g. Box, Jenkins, & Reinsel, 1994), cross–spectral or coherence analysis (Bloomfield, 1976; Warner, 1998), P–technique factor analysis (e.g. Nesselroade & Ford, 1985), dynamic factor analysis (Molenaar, 1985), various structural equation model variants proposed by McArdle and colleagues (e.g. Hamagami, McArdle, & Cohen, 2000), and nonlinear methods such as mutual information (Abarbanel, 1996; Boker, Schreiber, Pompe, & Bertenthal, 1998). Each of these methods uses related methods in that they all estimate some set of linear or nonlinear relations between observations separated by intervals of time while assuming that this structure remains constant over time. This assumption of constant relations over time boils down to an assumption known as *stationarity* (for discussions of stationarity and nonstationarity see Hendry & Juselius, 2000; Shao & Chen, 1987). In order for a process to be stationary (formally known as *weak stationarity*) for any starting time, t, the expected value of the means, standard deviations, autocorrelations, frequency spectra, and cross correlations of a sample from the process must be equal to the corresponding population values (Box et al., 1994; Itô, 1993). In other words, an assumption of stationarity implies that a set of statistical properties are assumed to hold across the entire length of time during which the psychological process was under observation.

With any simplifying assumption there are positive and negative consequences. There are two main benefits in assuming that a multivariate time series is stationary. The first is that one may conclude that any particular collection of occasions of measurement is representative of the whole behavior. Closely related, one may also conclude that all occasions of measurement separated by some time interval, s, have a relationship which has an expected value that does not change over the occasions of measurement in the experiment. Thus by assuming stationarity and making distribution assumptions, one may calculate means, covariances, regressions, and standard errors that may be concluded to be estimates of characteristics of the behavior as a whole.

Although these positive consequences might seem to provide arguments in favor of making an assumption of stationarity, it can be dangerous to make assumptions concerning data based on the convenience or power of a statistical method. Sometimes the assumption of stationarity is untenable. Traditional time series analyses use methods to remove nonstationarity so that the remaining stationary process can be analyzed, a process sometimes known as *prewhitening* the time series data (see e.g. West & Hepworth, 1991). For instance, bivariate time series analysis will typically remove co–occuring trends and cycles in order to examine the time–lagged relationship of the residuals of the two time series. However, many interesting behavioral phenomena are not only inherently nonstationary, but the very reason that they are interesting lies in the nature of that nonstationarity. One phenomenon of wide interest to behavioral researchers is human communication in its various forms, including face to face verbal conversation, nonverbal cues in conversation, written language, and music production and perception. It is our contention that the process of human communication is an excellent example of when understanding nonstationarity may be essential to understanding a behavioral phenomenon.

As an example, consider one form of human communication in which differing degrees of stationarity can be conveniently observed: the production and perception of music. If a musical composition were stationary, then there would be one function that, given an interval of time s milliseconds, would give the expected value of the correlation between the level of air pressure at any one moment in time and the level of the air pressure s milliseconds later. The logical consequence is that this musical composition would be composed of a single repeated waveform: essentially a single phrase played repeatedly without end by some combination of instruments.

While there exist musical compositions that display this characteristic of unending repetition, most musical compositions evolve over their span. And while there may be nearly repeating themes in a composition, new elements and relations between instruments tend to be added as the musical piece progresses. Humans do not tend to produce music that is stationary. Musical compositions tend to contain both elements of predictability and elements of surprise (Jones, 1993). Over short time spans one may observe repetition of a phrase, but over longer time spans the composition will tend to evolve in unexpected ways.

Similarly, an interesting conversation does not involve endlessly repeated verbiage, but contains both redundancy and surprise. It has been long known that over short time spans a listener may be able to predict the next word a speaker will say, but over longer time spans the words tend to be relatively unpredictable (Aborn, Rubenstein, & Sterling, 1959). Researchers in psycholinguistics, music perception, and auditory perception have begun to realize that many of the relationships of interest in their data are nonstationary, and that the nature of that nonstationarity is a crucial topic for analysis (see e.g. Bregman, 1990; Gregson & Harvey, 1992; Miller & Chomsky, 1963).

A related argument is advanced by Nesselroade and Featherman (1991) as applied to lifespan development. They suggest that the intraindividual variability of a construct over occasions of measurement may be an important indicator in and of itself. For instance variability in trial by trial performance has been used to help understand strategy learning in children (Siegler & Jenkens, 1989). Inherent in this argument is the notion that there may be critical changes in this variability, that is nonstationarity, which may be predictive of important outcomes. It is only a small step from Nesselroade and Featherman's position to suggest that nonstationarity in multivariate time series may itself prove to be a useful variable when addressing a variety of developmental questions.

In fact, the variability or nonstationarity of a variable may indicate only part of the story. We propose that there may be interesting patterns of variability in the association between variables. One consequence of this proposal is that variables may be nearly stationary in short durations, but much less so over longer time spans. We will use this idea of *local stationarity* in order to quantify variability in association.

Variability in a measure of association such as correlation may be important in order to understand, in particular, how adaptable creatures such as ourselves behave when the environment is also adapting to us. In any interpersonal exchange, two or more individuals may be adapting to one another and it therefore might be expected that the association between the individuals' behavior would show a pattern of variability that would be indicative of the underlying adaptive process.

The use of nonstationarity of association as a variable in psychological research requires a method that can quantify fluctuations in the relationships between variables over time. We propose a composite method, a few well–known steps and a simple algorithm, that can estimate time varying changes in patterns of bivariate predictability in a flexible manner. We then apply the method to a data set from an experiment in non–verbal communication in order to illustrate its use in a real–world psychological context.

In order to create a method that would be able to quantify variability of association, we identified three criteria that needed to be fulfilled. The first criterion is that the method must be able to track changes in the time lag and strength of association between the two time series over the course of the experiment. Suppose we measure two variables on multiple occasions separated by equal intervals and resulting in two vectors of observations $\mathbf{X}$ and $\mathbf{Y}$. If an event in $\mathbf{X}$ occurs before a similar event in $\mathbf{Y}$, we might reason that the event in $\mathbf{X}$ may have predicted the event $\mathbf{Y}$. But later we may see an event in $\mathbf{Y}$ that appears to predict a later event in $\mathbf{X}$. Such changes in lag and strength of maximum prediction can be indicative of an underlying dynamic relationship between the constructs that gave rise to the data in $\mathbf{X}$ and $\mathbf{Y}$. The variance of the time lag and variance of strength of association will give estimates of two types of nonstationarity in the bivariate time series.

Second, within some set bounds the method should estimate the interval of time between occasions of measurement at which a maximum association between two time series occurs and the strength of that association. The importance of finding a best lag of association has been long been recognized (e.g. Cattell, 1963). Often, one is only interested in the association between events that are separated by no more than some fixed bound (a few seconds, a few minutes, a day, etc.), while events that occur outside this bound are likely to be only spuriously related. Within that bounded interval of time, there may a lag between an event that occurs in time series **X** and a similar event that occurs in time series **Y**. The method should give an estimate of the time lag between the event in **X** and the event in **Y** as well as the strength of association between the two events.

Finally, the method should be flexible in the inherent tradeoff between the reliability of estimates of association between variables and the sensitivity to changes in the estimates of association. As more occasions of measurement are used to create an aggregated estimate of association, longer durations of time will be aggregated. This will tend to lead to a more accurate estimate of association from a statistical standpoint, but will simultaneously reduce the sensitivity needed to measure short duration fluctuations in a process. At the present time, we see no way around this reliability–sensitivity tradeoff other than both decreasing the interval of time between occasions of measurement and simultaneously increasing the number of occasions. The optimal tradeoff between sensitivity and reliability is likely to be different for each experiment, since how quickly the predictive relationship between psychological processes changes will be dependent on which processes are being studied, and each individual experiment may incorporate its own unique interval between occasions of measurement. Thus, the method must allow flexibility in assigning parameters that control the associated reliability and sensitivity.

As an example, we examined the coordination between movements made by pairs of individuals engaged in conversation. Facial expression, eye contact, pupil dilation, posture, gesture, and inter-personal distance may potentially be considered as elements of non-verbal communication. We tracked individual's hand and head motions during a 10 minute conversation and examined the association between the overall velocity of two conversants' head movements and between the left hand of one conversant and the right hand of the other. Previous work has indicated that the pattern of association between two conversants' movements can be nonstationary (Rotondo & Boker, in press). The example conversational experiment tests the hypothesis that auditory interference in the form of amplified traffic noise with a loudness equivalent to that of a subway train will change the structure of the coordination between two conversants' movements. The proposed method allows an estimate of the nonstationarity in the conversants' movements when noise is present as opposed to when noise is absent.

Cross–Correlation

A common method for estimating the association between events in two time series is cross–correlation, the correlation between two time varying stimuli or events over time intervals that may or may not be coincident. Essentially, a vector of sequential occasions of measurement is selected from each time series such that both vectors contain the same number of occasions and then the Pearson product moment correlation is calculated for these two vectors. The vectors may or may not begin at the same occasion of measurement. The interval of time separating the beginning occasion of measurement for the two vectors is the *lag* or *offset*. A vector of sequential measurements sampled from a time series is often called a *window*.

Suppose we wish to cross correlate two time series each containing N observations $\mathbf{X} = \{x_1, x_2, x_3, \ldots, x_N\}$ and $\mathbf{Y} = \{y_1, y_2, y_3, \ldots, y_N\}$ with equal intervals of time, s, between observations. If we assume stationarity and choose a positive lag of τ observations, the cross–correlation between $\mathbf{X}$ and $\mathbf{Y}$ at a lag τ is a function r of $\mathbf{X}$, $\mathbf{Y}$, and τ that can be defined as

$$r(\mathbf{X}, \mathbf{Y}, \tau) = \frac{1}{N-\tau} \sum_{i=1}^{N-\tau} \frac{(x_i - \overline{\mathbf{X}})(y_{i+\tau} - \overline{\mathbf{Y}})}{\mathrm{sd}(\mathbf{X})\mathrm{sd}(\mathbf{Y})} , \tag{1}$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ are the grand means and $\mathrm{sd}(\mathbf{X})$ and $\mathrm{sd}(\mathbf{Y})$ are the standard deviations of $\mathbf{X}$ and $\mathbf{Y}$ respectively over all occasions of measurement. This is merely an ordinary Pearson correlation between the two time series lagged by τ observations (a time interval corresponding to τ times s, the sampling interval).

Variations of cross–correlation similar to Equation 1 are commonly used in psychological research. For instance, cross–correlation has been used in theories in audition (e.g. Cherry, 1961; Licklider, 1959), of motion perception (e.g. Reichardt, 1961; Santen & Sperling, 1985), and of form detection and pattern recognition (e.g. Dodwell, 1971; Glass & Switkes, 1976; Dixon & Di Lolo, 1994). Most researchers assume the observed data come from stationary processes, where means and variances are constant over time. This yields mathematical tractability at the expense of possibly oversimplifying the model for the process of interest.

We will illustrate the use of cross–correlation with two example data sets, each containing data recorded using motion tracking equipment from the movements of pairs of participants sampled at 80 Hz (80 occasions of measurement per second). The first data set is from two participants who were asked to dance while mimicking each others' movements and the second data set is from two participants engaged in conversation. We use these two examples of coordinated behavior because the dancing behavior should be relatively stationary due to the repeating nature of the synchronizing auditory stimulus to which the participants were asked to dance. The coordination of head behavior from conversation has been found to exhibit a great deal of nonstationarity in previous work in our lab (Rotondo, 2000).

In the first data set, two individuals are dancing with one another while listening to a repeating rhythm. Dancer A is instructed to lead and dancer B is instructed to follow dancer A's movements as closely as possible. The movements that they make are likely to be highly synchronized with each other as well as being synchronized with the rhythm. Since the rhythm is repeating, the dancers are synchronizing with a stationary signal. Thus, we would expect that there would be a stable pattern of cross–correlation over the whole trial. A plot of the average cross–correlation between two dancers' movements as calculated using Equation 1 is shown in Figure 1–a. There is a high degree of association between the two dancers' movements at synchrony and at plus and minus 1600ms. The rhythm with which the dancers were synchronizing repeated every 1600ms, so this suggests that the dancers were making movements in time with this rhythm such that their movements also tended to repeat with the same cycle. These data are largely in conformance with an assumption of stationarity, so the overall cross–correlational pattern is strong.

Now consider the coordination between head movements produced as two individuals converse. It seems likely that these movements might not occur at exactly the same time. Sometimes conversant A would initiate a head movement that conversant B would respond to a short time later. Similarly, conversant B may sometimes make a head movement that

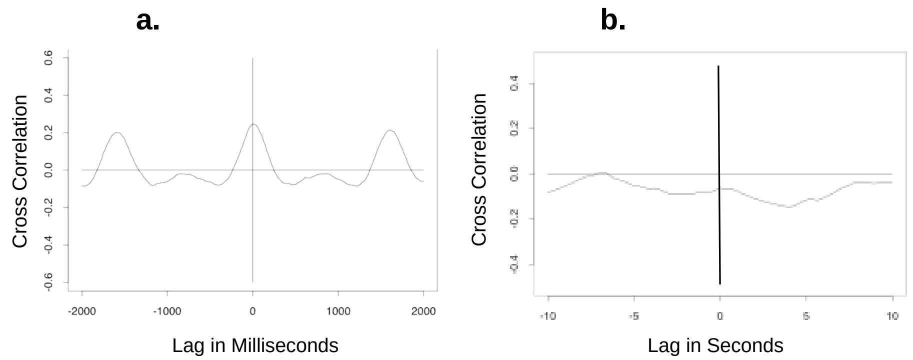

Figure 1. Overall lagged cross–correlations for (a) whole body movements of two individuals dancing during a 40 second trial and (b) head movements of two individuals during a 5 minute dyadic conversation.

conversant A would later respond to. Thus the correlation between lagged observations of position or orientation might turn out to be greater than that between synchronous observations. But sometimes the response may be in the same direction as the stimulus, and sometimes it may be in the opposite direction. Thus it might be that over short time scales there could be a high degree of association, but due to nonstationarity, overall there might be only low values of cross–correlation. Figure 1–b plots the overall cross–correlation between two conversants' head movements. Only weak relationships are evident. This could mean that there really is not much coordination in the conversation, or it could be that there is short–term coordination along with nonstationarity. Standard methods which assume stationarity cannot distinguish between these two possibilities.

Thus while a single measure of cross–correlation may give a good estimate of association between two behavioral time series, it may not give an estimate of the expected value of that association in the same way as calculating a mean of numbers drawn from a normal distribution. Instead, aggregating across the whole time series when global stationarity does not hold may be more akin to calculating the mean of a multimodal distribution. In fact, there may not be a stable expected value of association between two behavioral time series. The examination of patterns of change in the association between two time series may be a legitimate area of inquiry in and of itself. For this reason, we now consider a more temporally detailed analysis that can be calculated using many short cross–correlational windows in which the starting time of windows of observations is incremented or *swept* over the whole data set.

Windowed Cross–Correlation

One way to examine how the strengths and lags of association between two time series are changing over time is to use only short intervals of data from each time series to estimate the association and then select these windows so that their starting points represent increasing elapsed time from the beginning of the experiment. This has the advantage of only making an assumption of local stationarity rather than assuming stationarity over the whole time series.

Using short, overlapping windows that cover the time series results in a moving estimate of association and lag that needs to be calculated in a way that does not favor one variable over another, since global stationarity is not assumed. Thus we must split the lags and calculate the cross–correlation in the following manner. Suppose we have two data vectors each with N observations, $\mathbf{X} = \{x_1, x_2, x_3 \ldots, x_N\}$ and $\mathbf{Y} = \{y_1, y_2, y_3 \ldots, y_N\}$, with equal intervals between observations of length s. Further suppose a window size w_{max}, a time lag τ on the integer interval $-\tau_{max} \leq \tau \leq \tau_{max}$ and an elapsed time index i from the beginning of the data vector. Note that for every τ_{max} there will always be an odd number of integers in the interval $-\tau_{max} \leq \tau \leq \tau_{max}$. For each $i = \{\tau_{max}+1, \tau_{max}+2, \ldots, N-\tau_{max}-w_{max}\}$, a pair of windows $\mathbf{Wx}$ and $\mathbf{Wy}$ can be selected from the two data vectors $\mathbf{X}$ and $\mathbf{Y}$ respectively as follows

$$\mathbf{Wx} = \left\{ \begin{array}{ll} \{x_i, x_{i+1}, x_{i+2}, \ldots x_{i+w_{max}}\} & \text{if } \tau \leq 0 \\ \{x_{i-\tau}, x_{i+1-\tau}, x_{i+2-\tau}, \ldots x_{i+w_{max}-\tau}\} & \text{if } \tau > 0 \end{array} \right\} \text{ and} \tag{2}$$

$$\mathbf{Wy} = \left\{ \begin{array}{ll} \{y_{i+\tau}, y_{i+1+\tau}, y_{i+2+\tau}, \ldots y_{i+w_{max}+\tau}\} & \text{if } \tau \leq 0 \\ \{y_i, y_{i+1}, y_{i+2}, \ldots y_{i+w_{max}}\} & \text{if } \tau > 0 \end{array} \right\} . \tag{3}$$

Now the cross–correlation between the windows $\mathbf{Wx}$ and $\mathbf{Wy}$ can be defined as

$$r(\mathbf{Wx}, \mathbf{Wy}) = \frac{1}{w_{max}} \sum_{i=1}^{w_{max}} \frac{(\mathbf{Wx}_i - \mu(\mathbf{Wx}))(\mathbf{Wy}_i - \mu(\mathbf{Wy}))}{\text{sd}(\mathbf{Wx})\text{sd}(\mathbf{Wy})} , \tag{4}$$

where $\mu(\mathbf{Wx})$ and $\mu(\mathbf{Wy})$ are the means and $\text{sd}(\mathbf{Wx})$ and $\text{sd}(\mathbf{Wy})$ are the standard deviations of the windows $\mathbf{Wx}$ and $\mathbf{Wy}$ respectively.

Note that the choice of which window is to be lagged back in time from i, the original elapsed time index into the data vectors $\mathbf{X}$ and $\mathbf{Y}$ depends on whether τ is greater than or less than zero. When a time series is nonstationary, not making this distinction can bias the overall value and lag of the correlation. Thus if the calculation were made in the same way as in Equation 1 it could matter which variable was assigned to $\mathbf{X}$ and which to $\mathbf{Y}$. By selecting the windows in the manner of Equations 2 and 3 we guarantee a mirror symmetry such that the resulting set of cross–correlations as τ ranges from $-\tau_{max}$ to $+\tau_{max}$ will contain the same values in reverse order when the variables in $\mathbf{X}$ and $\mathbf{Y}$ are swapped.

To illustrate how the windows are chosen and results are stored, a simplified example is shown in Figure 2. In this example, a maximum lag of $\tau_{max} = 1$ is chosen and the steps are displayed in which the first four pairs of windows are selected from two data vectors $\mathbf{X}$ and $\mathbf{Y}$. At the bottom of the diagram is displayed one method for storing the resulting correlations between selected windows into a results matrix. Windowed and lagged cross–correlation requires four parameters to be selected by the researcher: window size (w_{max}), window increment (w_{inc}), maximum lag (τ_{max}), and lag increment (τ_{inc}). In the simplified illustration in Figure 2, window size is 6, window increment is 2, maximum lag is 1 and lag increment is 1. The results matrix will have a number of columns equal to $(\tau_{max} * 2) + 1$ and number of rows equal to the largest integer less than $(N - w_{max} - \tau_{max})/w_{inc}$.

Each of the parameter selections used in calculating the matrix of windowed and lagged cross–correlations has consequences with respect to the measures of relation that are calculated and the degree to which the results matrix is a summary rather than a

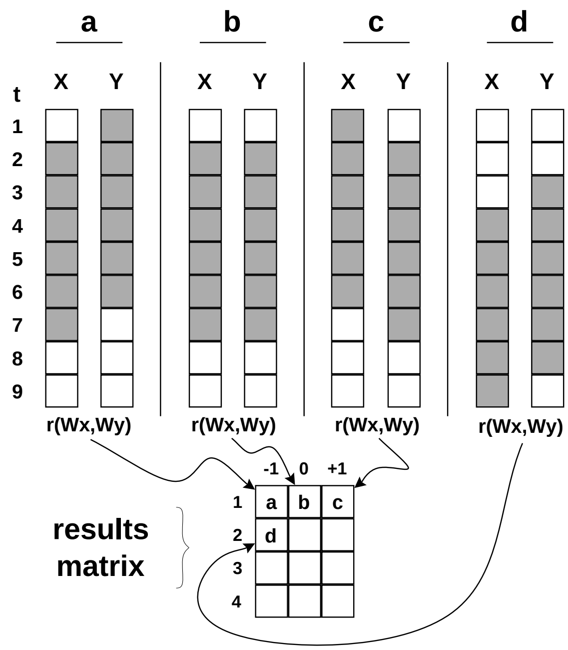

Figure 2. Four pairs of windows selected from two data vectors, $\mathbf{X}$ and $\mathbf{Y}$. Results of correlating each pair of windows is stored into the results matrix whose columns represent the relative lag of the two windows and whose rows represent the starting time of the window selected from $\mathbf{X}$.

recapitulation of the data vectors. It is important to be guided in the selection of these parameters by substantive theory as well as an exploration of a set of pilot data. In this way these parameters can then be fixed and used to estimate windowed and lagged associations in a second data set in a statistically testable manner.

The window size, w_{max}, (the number of observations in the data vectors W_x and W_y) should be chosen to be small enough so that the assumption can be made of little change in lead–lag relationship within the number samples in the window. However, if the window size is too small, the reliability for the correlation estimate for each sample will be reduced. Choosing the window size involves confronting the reliability/sensitivity tradeoff discussed earlier. The analyst should make this choice based on substantive and theoretic considerations in order to optimize the method for her particular data.

The window increment, w_{inc}, is the number of samples between successive changes in the window for the $\mathbf{X}$ vector as illustrated in Figures 2–c and d. The window increment is therefore also the amount of time that elapses between the correlations stored in successive rows in the results matrix. If the window increment is too short, there may be little change between successive rows in the results matrix, but if the window increment is too long, there may be so much change that successive rows in the results matrix will appear to be unrelated. Short window increments also lead to large numbers of rows in the results matrix. Thus one wishes to choose a window increment as long as possible, but not so long that the relation between successive rows in the results matrix is lost.

The maximum lag, τ_{max} is the maximum interval of time that separates the beginning of the two windows selected from their respective data vectors. The greater the maximum lag, the greater the interval of time that can separate behaviors for which estimates of coordination can be obtained. However, large maximum lags will tend to lead to large numbers of columns in the results matrix. It is thus up to the researcher to select the greatest interval of time separating a behavior from participant **X** and behavior from participant **Y** that would be considered to be of interest.

The lag increment, τ_{inc}, is the number of samples between successive changes in the window for the **Y** vector as shown by the difference between columns a and b in Figure 2. The lag increment is thus also the interval of time separating successive columns in the results matrix. Short lag increments lead to little change between successive columns in the results matrix, while long lag increments lead to apparently unrelated successive columns. The shorter the lag increments the more columns there will be in the results matrix. Thus, a good choice of lag increments will be the longest lag increment that still results in related change between successive columns. In this way the size of the results matrix can be minimized while patterns in change in the lead–lag structure of the coordination of the conversants can still be examined.

An implementation written in portable C code of the windowed lagged cross–correlation algorithm is available on the web page http://www.nd.edu/˜ sboker. The code can be compiled for MSDOS, Apple OS X, or most flavors of Unix. It inputs an ASCII textfile with two columns of data outputs a results matrix as shown above and includes command line options to control all the parameters discussed above.

Examples of the results of a windowed cross–correlation analysis applied to two example behavioral data sets are shown in Figure 3. In each of these graphs, the abscissa plots the lag of the two windows, the ordinate plots the elapsed time during the trial and the color represents the value of the cross–correlation at each combination of lag and elapsed time. Thus, the rows and columns in these two graphs correspond to the rows and columns in the results matrix from Figure 2 and the colors correspond to the values in the results matrix. Note that in these graphs, elapsed time increases as the value on the ordinate increases. Thus, the first row from the results matrix is the bottom row of the graph and the last row of the results matrix is at the top of the graph. In this way the cross–correlations at any particular value of elapsed time are shown as the colors from a horizontal slice through the graph.

The cross–correlations between two dancers' movements are plotted in Figure 3–a. Note that after about 8 seconds into the trial, the pattern of cross–correlations becomes stable; that is each horizontal slice through the graph is much like the next horizontal slice. Thus, vertical stripes are formed when the associations between variables are stable over time and therefore the association between the variables is stationary. When we observe the vertical stripes in Figure 3–a, it is evident that the pattern of association between the dancers' movements becomes stationary shortly about 8 seconds after the beginning of the trial.

On the other hand, the cross–correlations between the two conversants' head motions plotted in Figure 3–b does not show a pattern of vertical stripes of color. In fact, there are relatively large changes between subsequent horizontal slices through the graph. Thus, the lags of the cross–correlational association between the two conversants' head movements are changing rapidly as elapsed time in the trial increases. This pattern of association is

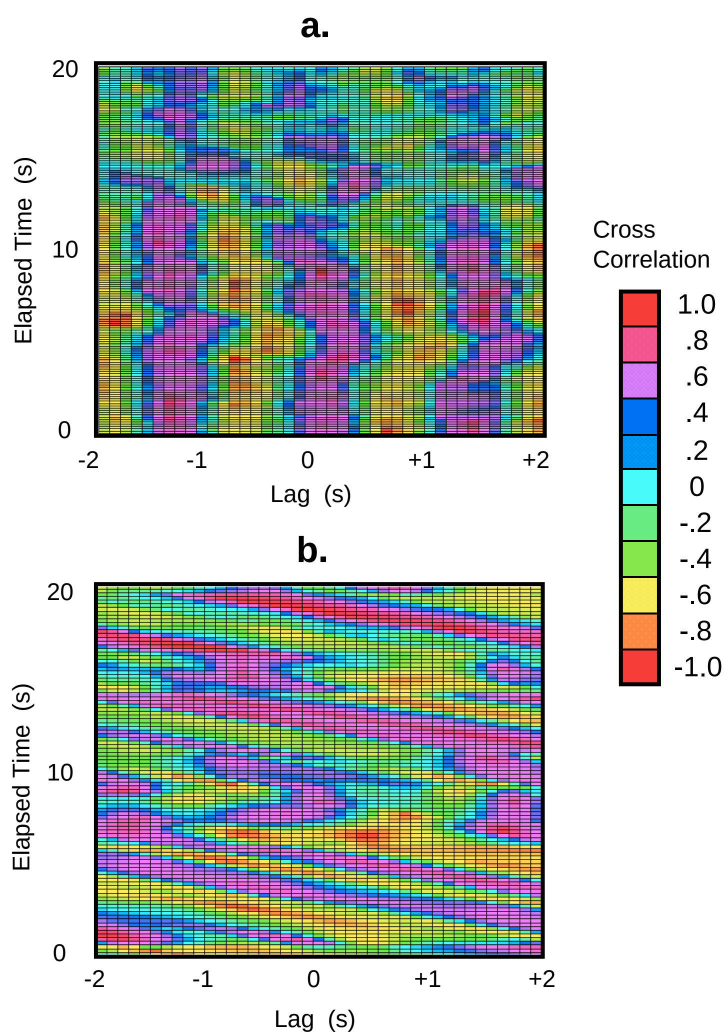

Figure 3. Density plots of windowed and lagged correlation result matrices from (a) 20 seconds of body velocities during dyadic dance, and (b) from 25 seconds of head velocity during dyadic conversation. High positive values of correlation are red, zero values of correlation are green and high negative values of correlation are orange. For these plots, window size is 120 samples, window increment is 20 samples, maximum lag is 400 samples, lag increment is 10 samples, and sampling rate is 80 Hz.

nonstationary.

Although the cross–correlations in Figure 3–b do not form vertical stripes, there does seem to be some sort of pattern to the variability of this association. As elapsed time increases, it seems that the stripes mostly are diagonal from lower right to upper left. That is, the peak cross–correlations appear to change from positive to negative lags. In order to be able to analyze patterns of change in the peak cross–correlation we have developed a *peak picking* algorithm that selects the peak correlation at each elapsed time according to some flexible criteria. The next section introduces this peak picking method and presents the parameters for controlling the selection of peak cross–correlations.

The Peak Picking Algorithm

One possible way to estimate the time lag of the predictive association between two time series is to find the peak cross–correlation that is closest to a lag of zero. For instance, suppose an event of duration one second occurs in data vector window $\mathbf{Wx}$ and a similar event occurs two seconds later in data vector window $\mathbf{Wy}$. In this case, we would expect a peak cross–correlation between the two windows at a lag of $\tau = 2/s$ where s is the sampling interval of the observations in the windows expressed in seconds. In order to find such a lag between two similar events a definition of what is meant by a "peak" will be required. As is so often the case in data analysis, the best definition may depend on the characteristics of the phenomena and the design of the experiment. We have defined a peak to be a maximum value of cross–correlation centered in a local region in which values are monotonically decreasing on each side of the peak. The analyst must define the size of this local region to be large enough so that spurious local noise is rejected, but small enough so that meaningful peaks are not rejected.

We now describe the peak picking algorithm in specific terms and then will describe it diagrammatically so as to provide an intuitive understanding of both how the algorithm works, and how changing its control parameter will affect the results obtained. A complete implementation of the algorithm is provided in S–plus language source code in Appendix A and is available for download from http://www.nd.edu/˜ sboker.

The peak picking algorithm operates on a vector, $\mathbf{V}$, of cross–correlations (one row from the results matrix shown in Figure 2) by starting at the element in this vector whose index, c, corresponds to a lag of zero. Since the lagged and windowed cross–correlation procedure described above always generates a results matrix with an odd number of columns ($[\tau_{max} * 2] + 1$), the initial value of c will be $c = \tau_{max} + 1$. The output from the peak picking algorithm is two numbers, the lag of the selected peak relative to c and value of the cross–correlation at that peak. The control parameter that needs to be defined prior to starting the algorithm is L_{size}, the size of the local region that defines a peak. The algorithm is described in general below and is enumerated in detail in Appendix B. A few additions to this algorithm have been made in order to account for exceptional cases and are discussed separately at the end of this section.

In Figure 4 we present the steps of the algorithm diagrammatically for an easy–to–find peak. In general, what happens is that the search region is incrementally increased until finally a local region is centered over a peak; then the lag index at the peak and the value of the cross–correlation at the peak are returned.

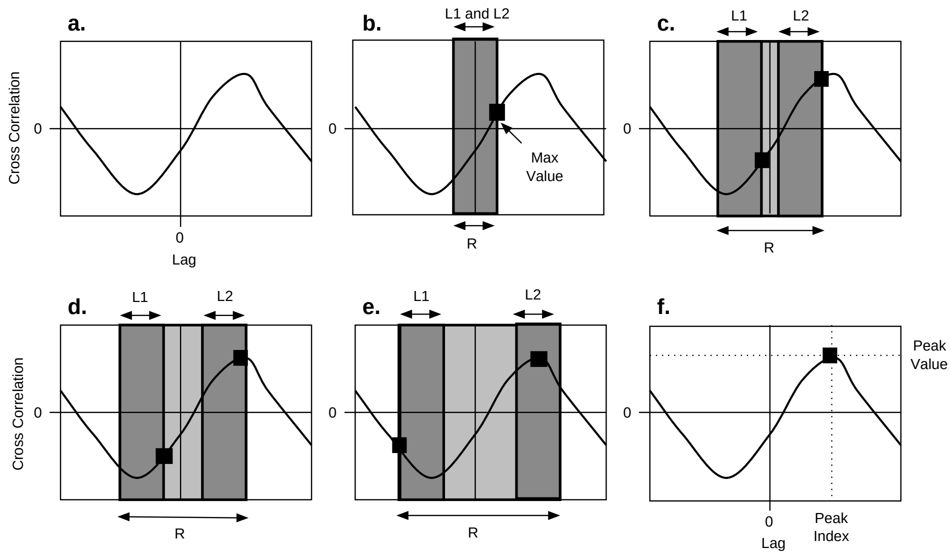

Figure 4. Diagrammatic representation of the peak picking algorithm.

The search starts at a lag of zero within the vector of cross–correlations plotted in the solid curved line as shown in Figure 4–a. In Figure 4–b, the search region begins with a size equal to the size of the local regions and is always centered over a lag of 0. The local regions $L1$ and $L2$ overlap each other at the start of the search. A maximum is found for each local region, but the maximum is not centered within the local region. In Figures 4–c and d the search region has increased enough so that the local regions no longer overlap. Still, the maximum in each local region $L1$ and $L2$ are not centered within its respective region. In Figure 4–e the search region is increased once more and the local region L2 now has a maximum at its center and the values of the cross–correlation are monotonically decreasing on either side of the maximum. A peak is identified as being found. (f) The value and index of the cross–correlation at the peak are returned and the algorithm stops.

We have isolated four problems that can lead to the spurious selection of a peak. We first summarize these problems and the solutions we implemented in the Splus code in Appendix A. We then present a more detailed description of the parametric choices involved in tailoring this peak picking algorithm to a particular variable of choice. Remember, one must tailor the algorithm in an exploratory manner on a set of pilot data prior to applying the algorithm in a confirmatory manner on an data set of interest.

The first problem is that the selection of a spurious peak could be due to high frequency noise in the cross–correlation; that is if the lines plotted in a graph like Figure 1 were "rough". In order to minimize this problem we use a *loess* smoothing function (Cleveland & Devlin, 1988) on the cross–correlation values prior to selecting a peak. We will further discuss the smoothing procedure below. The second problem is that a peak might be a minor fluctuation, a local maximum, rather than a larger maximum that is at a slightly larger lag with the same sign. In order to reduce this problem, we require a peak to be defined as a maximum with some chosen number of successively smaller values on each side of it. The third problem is that a spurious peak might have a smaller value than a larger peak with a somewhat larger lag of the opposite sign. To reduce this problem, we do not select a peak with a lag of one sign when there are successively increasing values with an equal lag of the opposite sign. The fourth problem is that there may be no credible peak within a range of lags that is appropriate to the phenomena under study, for instance all correlations within the range of lags might be equal to one another. To solve this problem, we allow the search to terminate as having failed after searching within some appropriate range and to return an indicator for missing values for the value and index of the peak. All of these solutions are implemented in the Splus code in Appendix A.

Figure 5 presents four example cross–correlation graphs that illustrate problems that arise in selecting peak values. All of the plots in Figure 5 have been smoothed using the loess smoother (Cleveland & Devlin, 1988). This smoother is a semi–parametric smoother in that it uses locally weighted regression to provide an estimate of the changes in cross–correlation with respect to the values of the lag. Loess requires two parameters: a fitting function and a span. The fitting function is generally either linear or quadratic. The span refers to the proportion of the total data vector that is used in smoothing. For instance, a span of .25 uses the 25% of the elements in the data vector with indices nearest to a target index to produce a smoothed value at the target index. In the peak picking algorithm we have used a quadratic regression function and a span of .25 in calling the loess function. This effectively fits a quadratic function to each lag in the cross–correlation using a weighted selection of those 25% of the cross–correlations that are nearest in time to the target lag. Our peak picking function then fits an interpolation spline through the results of the loess function so that there is an interpolated cross–correlation value between each lag. In this way we have found that we can improve the temporal resolution of the estimate of the lag of maximum association. That is to say, by using interpolation, we can estimate the time lag of peaks that occur between occasions of measurement. This relies on an assumption of continuity in the change in time lag of maximum association as elapsed time increases. All of these smoothing steps are included in the Splus source code in Appendix A and their parameters should not need to be changed in most cases.

The smoothed and interpolated cross–correlation graphs shown in Figure 5 come from pilot data in a nonverbal conversation experiment in which head nodding was tracked for two participants over a 5 minute conversation. These cross–correlations are calculated from 4 seconds of head nodding tracked from the two conversants. Figure 5–a and –c are cases in which the location of the peak is obvious. Still, even in this case, the peak is only obvious within the confines of the chosen maximum lag interval τ_{max} that was used to calculate the cross–correlation function. Note that in Figure 5–c the overall maximum within the displayed window occurs at a lag of -2 seconds. However, this maximum does not fit our definition of a peak since it does not have a region of smaller correlations on each side of it within the chosen maximum lag. Thus, in this case we have chosen to ignore maxima such as these because they fall outside the bounds of lags that are of interest for this particular substantive phenomenon. One must choose a maximum lag so as to capture the possible lags of interest and exclude lags that are not of interest. We chose a maximum lag $\tau_{max} = 2$ seconds $\times\ s$ where s is the sampling interval in order to capture events that were within 2 seconds of each other, but also to exclude events separated by greater intervals of time. This interval was chosen by comparing results of a range choices of τ_{max} and then

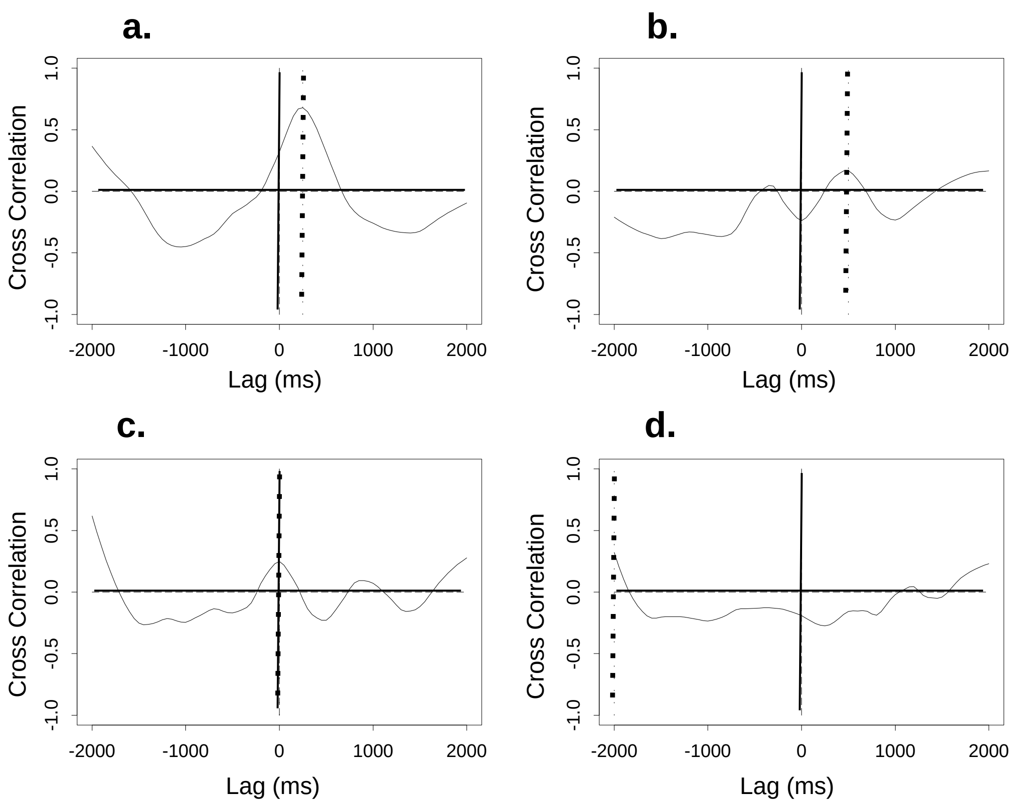

Figure 5. Four types of peaks found in cross–correlation graphs. (a) The peak is chosen at the dotted vertical line in an easy to find case. (b) An ambiguous case in which two small peaks are near lag zero in which the algorithm chooses the larger peak even though it is farther from zero. (c) Sometimes the peak is exactly at zero. (d) In some cases there may be no peak that fits the chosen parameters and so the algorithm terminates and returns missing value indicators.

running the lagged windowed cross–correlation method on pilot data from a separate head motion experiment. Greater values of τ_{max} tended to obscure the correlational structure and smaller values of τ_{max} tended to produce results that were not smooth and thus contained many spurious peaks. We tempered this decision with qualitative observations of human head nodding behavior which indicated that a 2 second range was not an unreasonable time frame in which agreement nods could occur.

The pattern of cross–correlations in Figure 5–b contains two peaks, a small peak with a negative lag near a lag of zero and a larger peak with a positive lag somewhat farther from lag zero. Such patterns can often occur when cross–correlations are calculated on cyclic phenomena. Since the increasing correlations for the positive lag within the search region become greater than the smaller peak before the smaller peak is identified as being centered in its local region, the local region switches to the positive lag and thus the larger peak becomes identified. Therefore, the size of the local region is critical in identifying which peak will be selected in cases like this. One must decide what temporal interval is of interest, since the size of the local region acts as a low pass filter, effectively removing cyclic phenomena with frequencies greater than $1/L_{size}$ where L_{size} is the time interval of the local region.

The cross–correlations in Figure 5–d do not have a peak that fits our definition. Thus, the algorithm stops when it reaches τ_{max} the maximum size that its search region can assume. The values that are returned for the peak index and peak value in this case are codes for missing values, since no peak was identified. If the local region had been set to be smaller, there would have been several possible candidate peaks. However, we have chosen in this case to reject such small peaks by the choice of L_{size}, the size of the local region.

The power of this approach to finding the lag of the maximum correlation closest to a lag of zero becomes evident when we use the algorithm on a matrix of windowed cross–correlations. Now we can automatically obtain, for each elapsed time, an estimate of the maximum association between two variables with the minimum time lag. Thus, the peak picking algorithm can be used for each row in a results matrix of windowed cross–correlations, resulting in a vector of lags and a vector of strengths of peak association.

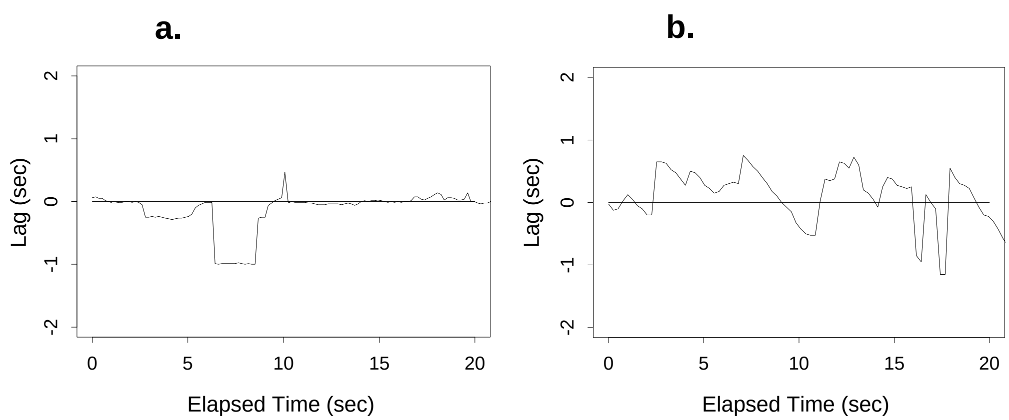

Figure 6. Peak picking algorithm results for time lag of peak correlation for (a) 20 seconds of body velocities during dyadic dance, and (b) from 25 seconds of head velocity during dyadic conversation.

In Figures 6–a and 6–b are plotted the lags of the peak cross–correlation for the same example data that was previously plotted in Figures 3–a and 3–b respectively. As was noted, it is apparent that the lag of the peak correlation is relatively stable in the dance data after an initial adaptation phase lasting about 8 or 9 seconds. However, it is apparent by inspection that there is much more variability in the association during conversation. In addition, there is an intriguing pattern of relatively shallow negative slopes and much greater positive slopes — like a "sawtooth" pattern — that may be indicative of a behaviorally relevant pattern in conversation. One substantive interpretation of these shallow negative slopes is that during conversation, individual A begins to nod his head leading to a similar head nod of individual B, but then after a short period of time individual A slows his head nod so that individual B begins to lead. This exchange of predictive leads in head motions may be an important and as yet unreported communicative aspect of nonverbal communication.

We next present an example of how this method can be put to use in an experimental setting. The example application uses motion capture data from our lab where we have been studying adaptive associations between individuals' movements during conversation.

Example Application

Many researchers suggest there is a close relationship between posture, gesture and speech (e.g. Condon & Ogston, 1966; Dittman & Llewellyn, 1969; Scheflen, 1964). According to the functions posture and gesture serve, speech–related non–verbal cues have been divided by Ekman and Friesen (1969) into three main types: emblems, illustrators and regulators. Emblems refer to those non-verbal acts that have a direct translation, such as nodding the head when meaning "yes". Their function is explicitly communicative and is recognized as such. Illustrators are movements which are tied directly to speech and facilitate communication by amplifying and elaborating the verbal content of message; for instance, swinging ones arms when speaking of playing golf. Regulators are movements which guide and control the flow of conversation, influencing who is speaking and how much is said. Instances of regulators include postural shifts and changes in gaze direction. Kendon (1981) argued that people may choose to use emblems in preference to speech, because in certain communicative contexts there may be distinct advantages in using gesture. For instance, since communication through posture and gesture is silent, environmental noise would be unlikely to interfere with nonverbal communication whereas spoken language could become difficult to understand in the presence of a loud sustained noise.

We hypothesize that when people are engaged in conversation in a noisy environment, they may use posture and gesture in a way that carries more informational content so to disambiguate their verbal communications. Higher informational content means that movements are less likely to be redundant and are more likely to be surprising to an interlocutor. Given this hypothesis, the association between two individuals' movements in a noisy environment would be expected to have greater variability in both time lag and maximum correlation than would the association between movements of the same individuals conversing in a quiet environment. We performed an experiment in order to test this hypothesis and used windowed cross–correlation and the peak picking algorithm to test whether there was greater variability in interpersonal coordination when loud environmental noise was present.

Methods

Participants.

Participants were 8 (4 dyads) female undergraduate senior students from Clark Atlanta University who volunteered for the study and who were not compensated. Participants were previously acquainted, having participated in a summer internship program with one another.

Apparatus.

An Ascension Technologies MotionStar 16 sensor magnetic motion tracking system was used to track the motions of participants. Each sensor is a cube approximately 3cm on a side and provides 3 dimensions of position and 3 dimensions of orientation information sampled at 80 Hz with a resolution of approximately 1.5mm in position and 2 degrees of arc in orientation. Eight sensors were placed on each individual: one on a the back of a baseball cap worn tightly on the head, one strapped just below each elbow using a neoprene and velcro around–the-limb strap, one held to the back of the back of each hand with an elastic weightlifting glove, one held to the sternum with a neoprene and velcro vest, and one strapped just below each knee with a neoprene and velcro around–the–limb strap. Each sensor was connected to the MotionStar system computer with a long cable. Thus each individual had a bundle of 8 cables that were gathered and positioned behind them in order to provide the minimum of interference with movement.

Participants were seated approximately 2 meters from each other with the magnetic transmitter sitting to one side of them. Due to the relatively small size of the room (3.5m × 3.5m) the transmitter was so close to the individuals that the motion of the hands nearest the transmitter was inaccurately recorded. Thus, we have only used head motion of each participant and one hand from each participant (the left hand of one participant and right hand of the other) in our analysis.

Procedure.

Participants were informed that they were in an experiment measuring “magnetic fields during conversation” in order to minimize self–consciousness about their posture and gestures. Participants were not informed that the experiment would include the noise manipulation In debriefing, no participant guessed that we were actually recording their movements. Participants were strapped into the sensors, and were asked to engage in a 5 minute free conversation while seated about 2 meters apart from each other. The experimenter then left the room. During one half of the conversation the room was quiet and during the other half of the conversation a loud (90db A-weighted) traffic noise was played over loudspeakers located in two corners of the room. The order of the noise was counterbalanced so that for half the dyads the noise occurred in the first half of the conversation and for the other half of the dyads the noise occurred in the latter half of the conversation. The traffic noise had been recorded in stereo at a local street corner and included sounds of cars, trucks and motorcycles that were multitracked, time delayed and re-recorded with reverb so as to approximate the sound at a busy intersection in the downtown of a large city. To give an intuitive understanding of the apparent loudness of the sound, the sound pressure in the room was equivalent to that present at a subway platform when a subway is arriving.

Results

Since the stated hypothesis concerns the association between movements, we first calculated the velocity of the head and hands for each person for each sampled time during the experiment. Velocity was calculated for the head, V_{head}, and one hand, V_{hand}, of each participant for each time t as

$$V(t) = \left((\frac{x_{t+1} - x_{t-1}}{2S})^2 + (\frac{y_{t+1} - y_{t-1}}{2S})^2 + (\frac{z_{t+1} - z_{t-1}}{2S})^2 \right)^{1/2} , \quad (5)$$

where x_{t-1}, y_{t-1}, and z_{t-1} are the positions of the sensor in centimeters along the three spatial axes at at one sample prior to time t, and S is the interval of time between samples, $S = 1/80$ sec. The resulting velocity vectors (in cm/sec) for the head motions for each participant from the whole session were then used as input to the windowed cross–correlation analysis. The velocity vectors from the left hand of one participant and right hand of the other participant were cross–correlated in the same way.

We chose to analyze windows of 4 seconds of data since based on previous experiments (Boker & Rotondo, in press; Rotondo & Boker, in press) it seemed reasonable that the production and perception of a gesture or head nod would occur within 2 seconds of each other. Thus the window size, w_{max}, was chosen to be $w_{max} = 4$ secs $\times$ 80 samples/sec = 320 samples. We chose to increment the window by 1/8 second in order to be able to capture rapid changes in lead and lag, so w_{inc} was set to be 1/10 second $\times$ 80 samples/sec = 8 samples. We set the maximum lag to be $\tau_{max} = 2$ secs $\times$ 80 samples/sec = 160 samples and the lag increment $\tau_{inc} = 1/10$ second $\times$ 80 samples/sec = 8 samples.

The windowed cross–correlations were run separately on the portions of the conversation with noise and without noise so that there would not be a set of windowed correlations that crossed the boundary between the noise and no noise conditions. The resulting matrices from the windowed cross–correlational analysis were submitted to the peak picking algorithm to calculate peak correlations nearest a lag of zero and their respective time lags. A loess smoothing span of .25 was used, producing a moderate amount of regression spline smoothing of the cross–correlation data. The peak–picking algorithm was called with a local region size $L_{size} = 4$, which corresponds to 4 cross–correlation lags each of which was $\tau_{inc} = 1/10$ second. Thus the effective local region used for the peak picking was 0.4 seconds so cyclic movements faster than 2.5 cycles per second were rejected.

The mean and variance of the vector of peak correlation values and the associated vector of lags were calculated for each dyad within each condition. Thus comparisons could be made between estimates of the overall amount of coordination between individuals in the two conditions. Similarly, the variability of the coordination magnitude and lag could also be compared across conditions. The resulting means and variances of the peak correlation values and their associated lags were used as dependent variables in univariate mixed model regression analyses grouped by dyad and fit using the Splus software function *nlme*. A test for order effects was made, but in no case did the order of the noise (first half versus latter half of the conversation) significantly predict any of the outcome variables so the order predictor variable was dropped from the analysis.

The mixed effects model was specified such that the intercept term was allowed to vary across dyads. The noise condition was dummy coded (0=silence and 1=noise) and used as a fixed predictor of the selected outcome variable as follows:

$$y_{ij} = b_{0j} + b_1 x_{ij} + e_{ij} \ , \qquad (6)$$

where y_{ij} is the selected dependent variable for the ith noise condition during the jth dyad session, x_{ij} is the ith noise condition during the jth dyad session, b_{0j} is the intercept term for the jth dyad session, b_1 is the regression coefficient of noise predicting the selected outcome variable, and e_{ij} is a independent identically distributed random variable. The results of these analyses are presented in Tables 2 and 1.

Both the mean and variance of the peak correlation increased significantly when the environmental noise was present (Table 1). Thus individuals movements were more highly correlated in the presence of noise and at the same time there was more variability in the value of the peak association. Only the variance of the lag between individuals increased

Table 1: Noise condition predicting the mean and variance of the value of the peak windowed cross–correlation for head and one hand using four independent univariate mixed model regressions grouped by dyad. Noise was coded silence=0, noise=1. Asterisk indicates $p < 0.05$

Dependent Variable	b	SE	Z
Head			
mean	.041	.015	2.745*
variance	.009	.002	3.523*
Hand			
mean	.043	.011	3.843*
variance	.012	.002	6.946*

Table 2: Noise condition predicting the mean and variance of the lag of the peak windowed cross–correlation for head and one hand using four independent univariate mixed model regressions grouped by dyad. Noise was coded silence=0, noise=1. Asterisk indicates $p < 0.05$

Dependent Variable	b	SE	Z
Head			
mean	.374	.331	1.131
variance	5.091	1.241	4.102*
Hand			
mean	-.221	.287	-.767
variance	13.818	4.505	3.067*

significantly when noise was present (Table 2). Thus the predictive association between individuals' movements tended to have lags that were farther from synchrony (a lag of zero) when exposed to the environmental noise.

Discussion

The results supported the hypothesis of greater variability in the coordination between conversing individuals when in the presence of noise than when there was no interfering noise. Thus, the variability of the time lag and value of peak association increased significantly for both the head and the hand in the presence of noise. No change was observed in the mean value of the time lag of peak association, but this was expected. Since individuals were randomly assigned to chairs, there was no reason why one would expect individuals in chair A to behave differently than those in chair B and therefore there should be no change in the mean value of the lag of the peak cross–correlation across noise conditions.

However, there was a significant increase in the coordination between conversants when the noise was present. While this finding was not anticipated in the hypothesis, it is intriguing. It appears that individuals more closely coupled their movements to each other when verbal communication became more difficult. These findings, while interesting, are from a small, and most likely nonrepresentative sample of individuals and thus may not generalize to the population of English language conversational dyads. We present these results solely in order to illustrate the use of the methods.

General Discussion

The combination of the use of windowed cross–correlation with our peak picking algorithm allows the empirical estimation of variability in a fundamentally different way than the usual examination of within–variable change over time. These methods allow the estimation of variability in the association between variables. The way in which the value of the peak association between variables changes and the way in which the temporal lag changes can be extremely informative as to the structure of adaptive relationships between individuals or the predictive capacity exhibited between variables within an individual over time.

We expect that methods such as those explicated here will become much more prevalent as psychologists begin increasingly to use experimental designs that measure a set of variables on many occasions from a sample of individuals. Prime examples for the use of the methods described here are physiological measurements (i.e. EEG, EKG, GSR, single cell recordings of neuron activation) taken over the course of an experiment, data from journaling studies in which individuals repeatedly self–report on several variables over an extended period of time, or motion capture experiments such as the example application reported above. Other psychological data such as longer daily diary studies with approximately 100 occasions of measurment or more may also be amenable to exploratory analysis with the methods presented here. In order for the algorithms to be useful, (a) each window must contain sufficient observations to reliably estimate the cross–correlations and (b) there must be a sufficient number of windows in order to reliably estimate the pattern of change in the time lags and strength of association between the variables. Estimating the actual number of observations that are sufficient for (a) and (b) above requires a power calculation dependent on the hypothesized effect size and required significance level for each of (a) and (b).

Other methods have been developed for the analysis of nonstationary and nonlinear time series. These methods fall into four basic groups: (a) those that emphasize forward prediction such as Kalman filtering, nonlinear prediction and mutual information; (b) those that emphasize system identification and description such as Lyapunov exponents, correlation dimension, and surrogate data; (c) those that emphasize graphical description and exploration such as state space plots and recurrence diagrams; and (d) those that emphasize noise reduction (for introductions see Abarbanel, 1996; Heath, 2000; Kantz & Schreiber, 1997). The windowed cross–correlation and peak picking method differs from these other techniques in that its primarily goal is the analysis of the change in the association between variables over time. The most closly related method known to the authors is an information transfer method proposed by Schreiber (2000) in which entropy is used as the measure of association between the time series. Schreiber's method differs substantially from ours in that his method doesn't attempt to estimate time–varying patterns of association between variables.

The primary disadvantage to the windowed cross–correlation and peak picking analysis is that it requires the analyst to choose several parameters in order to to appropriately estimate the changing pattern of association between variables. A preliminary sample of data must be used to tune these parameters prior to using the method for hypothesis testing. We have found this process of exploration of preliminary samples to be illuminating in its own right, and so now run pilots of our experiments so that we can adjust both the parameters and our hypotheses prior to gathering larger and inevitably more expensive samples to test these revised hypothesis without adjusting the parameters of the windowed cross–correlation and peak picking algorithms.

The particular choices of parameters for the windowed cross–correlation and peak picking algorithms used here apply only to the particular experimental paradigm presented in the example analysis. However, the flexibility of choice in these parameters means that the algorithms are very general in the types of repeated measures data to which they may be applied. Any continuous variable bivariate time series data in which an assumption of stationarity in linear association between the variables might be invalid is a candidate for analysis with these algorithms.

A second disadvantage is that the assumption of local stationarity may frequently be violated to some degree and this may produce downward bias in the estimates of the magnitude of the cross–correlations and the variances of the lags. While this remains an unsolved problem with our method, it produces errors in the conservative direction; underestimating the magnitudes of effects.

A third disadvantage is one that plagues all modeling methods: an unobserved variable may be the cause of the associations estimated between the observed variables. Although this method estimates time lagged associations, this does not necessarily imply causality between the observed variables.

Finally, the methods presented here estimate peak correlations and their associated lags but do not impose a testable model of the process that gave rise to the nonstationarity in the first place. An advance on these methods would be the creation of a multivariate model that would predict patterns of changes in peak cross–correlations among several variables in such a way that particular hypotheses concerning the short term dynamics of the processes involved could be rejected.

It seems unlikely that living, adapting individuals will always exhibit predictive associations between variables that are stable and unchanging over time. And yet this is exactly the assumption that is generally made in order to provide a statistically tractable estimate of those associations. It seems much more likely that in adaptive relations between variables sometimes person A will lead and person B will follow and sometimes the reverse; sometimes variable $\mathbf{X}$ will predict $\mathbf{Y}$ and sometimes the reverse; sometimes system J will drive system K and sometimes the reverse. We propose that the time has come to develop new methods that are able to relax the assumption of unchanging structure over time in the association between variables. It is our expectation that the development of such methods will lead to a deeper and more realistic understanding of human behavior.

Appendix A

```
# File name: peakPick.S
# Author: Steven M. Boker, Minquan Xu
# University of Notre Dame
# Date: Feb 09, 2001

# input data structure(Splus object matrix):
#    rows: elapsed time
#    columns: cross correlations
# output: a list of local peak indices and values

# Parameters:
# ----------------------------------------------------------------------
# tAllCor:   Splus object matrix, this matrix is created from the output
#            file of windcross program using Splus function such as
#            "scan". ex. tAllCor <- matrix(scan("windcross.dat"), ncol=n, byrow=T),
#            where n is the number of columns in windcross.dat file.
# Lsize:     local search region, the value should be larger than 0 and
#            less than 1/2 length of one row
# graphs:    number of graphs to draw for one input data object, the
#            value should be larger than 0 and less than number of rows
#            of one input data object
# pspan:     see help(loess) for span
# type:      local maximum or local minimum, valid values are: "Min" and "Max"
# tFileName: root characters for .eps file
#----------------------------------------------------------------------

peakpick<- function(tAllCor, Lsize=8, graphs=0, pspan=.25,
                    type="Max",tFileName="peak") {
    #----------------check for validity of parameters -------------------
    colLen <- length(tAllCor[1,]) # col length --- number of columns
    rowLen <- length(tAllCor[,1]) # row length --- number of rows
    tLsize <- floor((1/2)*colLen) # maximum local search region

    if(Lsize<1 || Lsize>tLsize) { # Lsize too small or too large
          errorStr<- paste("Lsize should be >0 and <= ", tLsize, sep="")
          stop(errorStr) # print error message and stop the program
    }
    if(graphs<0||graphs>rowLen) { # num of graphics to print is too small or large
           errorStr <- paste("graphs should be >=0 and <= ", rowLen, sep="")
           stop(errorStr) # print error message and stop the program
    }
    if(pspan<0 || pspan>1) { # invalid pspan value
           stop("pspan should be >0 and <1\n") # print error message and stop
    }
    if(type!="Max"&&type!="Min"&&type!="max"&&type!="min"){ # only two types
           stop("valid types are: max|Max or Min|min \n") # print message and stop
    }
    #-----------Initialization-----------------------------------------------
    drawgraph <- 0 # graphics drawn
```

```
colLen <- length(tAllCor[1,]) # col length
rowLen <- length(tAllCor[,1]) # row length
xSequence <- seq(-(colLen-1), (colLen-1), by=1) # X axis for each graph
mx<- rep(NA, (2*colLen-1)) #vector for keeping temp peak value for a row
                            #data points will be 2*colLen-1 after smooth
tIndex <- rep(NA, rowLen) #vector of peak index---one peak index for each row
tValue <- rep(NA, rowLen) #vector of peak value---one peak value for each row

#------------ type == max or Max -------------------------------------
if(type=="Max"||type=="max") { #compute local maximum
      for(rowNo in c(1: rowLen)) { #acess each row
            #eliminate missing value
            miss <- is.na(tAllCor[rowNo, ]) #a row is T is "NA" or F is not "NA"
            #initialize the position of NA in a row
            missposition <- 0
            for(mIndex in c(1:colLen)) { #evaluate an entire row
                  missposition <- mIndex #the position of NA
                  if(miss[missposition]) break #find one
                  missposition <- missposition+1 #increase count
                  if(missposition==colLen+1) break #No NA in this row
            }
            #cat("missposition=", missposition, "\n")
            #if has missing value
            if(missposition <= colLen) next #skip a row with NA
            else { # no missing value
                  drawgraph <- drawgraph+1 #number of graph to draw
                  tCor <- tAllCor[rowNo, ] #number of columns
                  #smooth
                  t1 <- loess(tCor~c(1:colLen), degree=2,
                              span=pspan, )$fitted.values
                  #data points is set to n
                  t2 <- spline(c(1:colLen), t1, n=(2*colLen-1))$y

                  # show calculate progress
                  # cat("row=", rowNo, "\n")
                  #----- process a row --find max value and max index------
                  windowWidth <- 0  # searched region
                  lookAhead <- 0  # look ahead data points
                  for(j in 1:(colLen-1)) {  # search from 1 to colLen -1
                        windowWidth <- windowWidth+1 # increase search ed region
                        # select the search region, the center of search region
                        # is in the middle of t2, notice that t2 has 2*colLen-1
                        # data points.
                        tSelect <- (colLen - windowWidth):(colLen+windowWidth)
                        mx[j] <- max(t2[tSelect], na.rm=T) # store temp max value
                        if(j==1) mmx <- mx[j] # mmx is final local max value
                        #remember current max
                        else { # if j != 1
                              if(mx[j]>mmx) { # new temp max value, only one
                                              # max value in tSelect
                                    lookAhead <- 0 # set stop criterion = 0,
```

```
                                # the criterion is that if we find
                                # Lsize data points less than current
                                # max, then stop and the current max
                                # value is the local maximum we wanted
                        mmx <- mx[j] # update new max value
                  }
            else if(mx[j]<=mmx) { # if other values are less than
                  # current maximum
                  # increase the count---how many neighbor data
                  # point are less than current maximum
                  lookAhead <- lookAhead+1
                  if(lookAhead>=Lsize) break # meet criterion
            }
            }#else
      }#for j--max value and index for each row
      #use match function to find the index
      Index <- match(mmx, t2[tSelect])+tSelect[1]-1
      # tSelect[1] is the first index of the selected window

      # relative position to the middle point
      position <- Index -colLen

      #according to the local maximum definition
      if (position >(colLen - Lsize - 1) ||
          position < (-(colLen - Lsize -1))) { # fail
            tIndex[rowNo] <- NA
            tValue[rowNo] <- NA
      }
      else { # found a local maximum
            tIndex[rowNo] <- position
            tValue[rowNo] <-mmx
      }

      #draw plots
      if(drawgraph <= graphs) {
           # define graphic file name
           tepsfile <- paste(tFileName, "Max", rowNo, ".eps", sep="")
           # title of the graph
           tmain <- paste("max Index", tFileName, "r", rowNo,"w",
                          Lsize, sep="")
           # write to postscript format
           postscript(tepsfile, height=6.4, horizontal=F)

           # draw borders and their labels
           plot(c(-(colLen-1), (colLen-1)), c(-1,1), xlab="Lag",
                ylab="Cross Correlation", main=tmain, type="n")

           # draw the curve
           lines(xSequence, t2, type="l")

           # draw the local maximum
```

```
                    lines(c(position,position), c(-1,1), type="l", lty=8)

                    # draw the axes
                    lines(c(0,0), c(-1,1), type="l", lty=4)
                    lines(c(-(colLen-1), (colLen-1)), c(0,0), type="l", lty=4)
                    dev.off() # term off plot device to generate graph
                } # if drawgraph
            } #else no missing value
      } # for rowNo-- process each row
#end of process a row
return(list(maxIndex=tIndex, maxValue=tValue))
}#if type=max

#--------------------------------------------------------------------------------
# type == Min or min
#--------------------------------------------------------------------------------
else if(type=="Min"||type=="min") {
      for(rowNo in c(1: rowLen)) {
            #eliminate missing value
            miss <- is.na(tAllCor[rowNo, ])
            missposition <- 0
            for(mIndex in c(1:colLen)) {
                  missposition <- mIndex #the position of NA
                  if(miss[missposition]) break #find one
                  missposition <- missposition+1
                  if(missposition==colLen+1) break
            }
            #cat("missposition=", missposition, "\n")
            #if has missing value
            if(missposition <= colLen) next #skip a row with NA

            else { # no missing value
                  drawgraph <- drawgraph+1
                  tCor <- tAllCor[rowNo, ]
                  t1 <- loess(tCor~c(1:colLen), degree=2, span=pspan, )$fitted.values
                  t2 <- spline(c(1:colLen), t1, n=(2*colLen-1))$y
                  # show calculate progress
                  #cat("row=", rowNo, "\n")

                  #process a row --find max value and max index
                  #-----------------------------------------------------------
                  windowWidth <- 0
                  lookAhead <- 0
                  for(j in 1:(colLen-1)) {
                      windowWidth <- windowWidth+1
                      tSelect <- (colLen - windowWidth):(colLen+windowWidth)
                      mx[j] <- min(t2[tSelect], na.rm=T)
                      if(j==1) mmx <- mx[j]
                      #remember current max
                      else {
                      if(mx[j]<mmx) { #only one value in tSelect
```

```
                        lookAhead <- 0
                        mmx <- mx[j]
                  }
                  else if(mx[j]>=mmx) {
                        lookAhead <- lookAhead+1
                        if(lookAhead>=Lsize) break
                  }
                  }#else
              }#for j--max value and index for each row
              #use match function to find the index
              Index <- match(mmx, t2[tSelect])+tSelect[1]-1
              position <- Index -colLen

              if (position >(colLen - Lsize -1) ||
                  position < (-(colLen -Lsize -1))) { # fail
                    tIndex[rowNo] <- NA
                    tValue[rowNo] <- NA
              }
              else {
                    tIndex[rowNo] <- position
                    tValue[rowNo] <-mmx
              }

              #draw first 10 plots
              if(drawgraph <= graphs) {
                    tepsfile <- paste(tFileName, "Min", rowNo, ".eps", sep="")
                    tmain <- paste("min Index", tFileName, "r",
                                   rowNo,"w", Lsize, sep="")
                    postscript(tepsfile, height=6.4, horizontal=F)
                    plot(c(-(colLen-1), (colLen-1)), c(-1,1),
                         xlab="Lag", ylab="Cross Correlation",
                         main=tmain, type="n")

                    lines(xSequence, t2, type="l")
                    lines(c(position,position), c(-1,1), type="l", lty=8)
                    lines(c(0,0), c(-1,1), type="l", lty=4)
                    lines(c(-(colLen-1), (colLen-1)), c(0,0), type="l", lty=4)
                    dev.off()
              } # if drawgraph
        } #else no missing value
  } # rowNo-- process each row
 #end of process a row
 #------------------------------------------------------------------------

return(list(minIndex=tIndex, minValue=tValue))
}#if type=mix
}
```

Appendix B

1. A search region size R_{size} is defined as the smallest integer such that $R_{size} \geq (L_{size}/2)$

2. A search region R is defined as a sequential and increasing set of integers of size R_{size} centered on c, the index corresponding to a lag of zero between the two cross–correlated data vectors as follows: $R = \{c - R_{size}, \ldots, c-1, c, c+1, \ldots, , c + R_{size}\}$.

3. Two local regions are defined: $L1$ as the first L_{size} integers in R and and $L2$ as the last L_{size} integers in R.

4. The maximum value $L1_{max}$ of the cross–correlations in $\mathbf{V}$ with indices contained in the local search region $L1$ is calculated.

5. If the index of $L1_{max}$ is the center element of $L1$ and the values from $\mathbf{V}$ on either side of $L1_{max}$ are monotonically decreasing, then a peak has been found. In this case, the value $L1_{max}$ and its index are returned and the algorithm stops. Otherwise, the algorithm goes to the next step.

6. The maximum value $L2_{max}$ of the cross–correlations in $\mathbf{V}$ with indices contained in the local search region $L2$ is calculated.

7. If the index of $L2_{max}$ is the center element of $L2$ and the values from $\mathbf{V}$ on either side of $L2_{max}$ are monotonically decreasing, then a peak has been found. In this case, the value $L2_{max}$ and its index are returned and the algorithm stops. Otherwise, the algorithm goes to the next step.

8. R_{size} is incremented by one and then if $R_{size} \leq \tau_{max}$ the process is repeated from step 2.

9. If $R_{size} > \tau_{max}$ then no viable peak was found and the algorithm returns a missing value indicator for both the peak value and peak index and then stops.

References

Abarbanel, H. D. I. (1996). *Analysis of observed chaotic data.* New York: Springer–Verlag.

Aborn, M., Rubenstein, H., & Sterling, T. D. (1959). Sources of contextual constraint upon words in sentences. *Journal of Experimental Psychology*, *57*, 171–180.

Bloomfield, P. (1976). *Fourier analysis of time series.* New York: Wiley-Interscience.

Boker, S. M., & Rotondo, J. L. (in press). Symmetry building and symmetry breaking in synchronized movement. In M. Stamenov & V. Gallese (Eds.), *Mirror neurons and the evolution of brain and language.* Amsterdam: John Benjamins.

Boker, S. M., Schreiber, T., Pompe, B., & Bertenthal, B. I. (1998). Nonlinear analysis of perceptual–motor coupling in the development of postural control. In H. Kantz, J. Kurths, & G. Mayer-Kress (Eds.), *Nonlinear analysis of physiological data* (pp. 251–270). Berlin: Springer Verlag.

Box, J. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). *Time series analysis: forecasting and control.* Englewood Cliffs, NJ: Prentice Hall.

Bregman, A. S. (1990). *Auditory scene analysis : the perceptual organization of sound.* Cambridge, MA: MIT Press.

Cattell, R. B. (1963). The structuring of change by P–technique and incremental R–technique. In C. W. Harris (Ed.), *Problems in measuring change* (pp. 167–198). Madison, WI: University of Wisconsin Press.

Cherry, C. (1961). *Two ears — but one world.* Cambridge, MA, New York: MIT Press and Wiley.

Cleveland, W. S., & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, *83*, 596–610.

Condon, W. S., & Ogston, W. D. (1966). Sound film analysis of normal pathological behavior patterns. *Journal of Nervous and Mental Disease*, *143*, 338-347.

Cook, T. D., Dintzer, L., & Mark, M. M. (1980). The causal analysis of concomitant time series. In L. Bickman (Ed.), *Applied social psychology annual (volume 1)* (pp. 93–135). Beverly Hills: Sage.

Dittman, A. T., & Llewellyn, L. G. (1969). Body movement and speech rhythm in social conversation. *Journal of Personality and Social Psychology*, *11*, 98-106.

Dixon, P., & Di Lolo, V. (1994). Beyond visible persistence: An alternative account of temporal integration and segregation in visual processing. *Cognitive Psychology*, *26*, 33-63.

Dodwell, P. (1971). On the perceptual clarity. *Psychological Review*, *78*, 275-289.

Ekman, P., & Friesen, W. V. (1969). The repertoire of nonverbal behaviour: categories, origins, usage and coding. *Semiotica*, *1*, 49-98.

Glass, L., & Switkes, E. (1976). Pattern recognition in humans: Correlations which cannot be perceived. *Perception*, *5*(1), 67–72.

Gregson, R. A., & Harvey, J. P. (1992). Similarities of low-dimensional chaotic auditory attractor sequences to quasirandom noise. *Perception & Psychophysics*, *51*(3), 267–278.

Hamagami, F., McArdle, J., & Cohen, P. (2000). A new approach to modeling bivariate dynamic relationships applied to evaluation of comorbidity amoung DSM–III personality disorder symptoms. In V. J. Molfese (Ed.), *Temperament and personality development across the life span.* Mahwah, NJ: Lawrence Erlbaum Associates.

Heath, R. A. (2000). *Nonlinear dynamics: Techniques and applications in psychology.* Mahwah, NJ: Lawrence Erlbaum.

Hendry, D., & Juselius, K. (2000). Explaining cointegration analysis: Part 1. *Energy Journal*, *21*, 1-42.

Itô, K. (Ed.). (1993). *Encyclopedic dictionary of mathematics.* Cambridge, MA: MIT Press.

Jones, M. R. (1993). Dynamics of musical patterns: How do melody and rhythm fit together? In T. J. Tighe & W. J. Dowling (Eds.), *Psychology and music: The understanding of melody and rhythm* (pp. 67–92). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Kantz, H., & Schreiber, T. (1997). *Nonlinear time series analysis.* Cambridge: Cambridge University Press.

Kendon, A. (1981). Geography of gesture. *Semiotica*, *37*, 129-163.

Licklider, J. (1959). *Three auditory theories.* New York: McGraw-Hill.

Miller, G. A., & Chomsky, N. (1963). Finitary models of language users. In R. D. Luce, R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology (vol 2)* (pp. 419–491). New York: Wiley.

Molenaar, P. C. M. (1985). A dynamic factor model for the analysis of multivariate time series. *Psychometrika*, *50*, 181–202.

Nesselroade, J. R., & Featherman, D. L. (1991). Intraindividual variability in older adults' depression scores: Some implications for longitudinal research. In D. Magnusson, L. Bergman, G. Rudinger, & B. Torestad (Eds.), *Stability and change: Methods and models for data treatment.* London: Cambridge University Press.

Nesselroade, J. R., & Ford, D. H. (1985). P–technique comes of age: Multivariate, replicated, single–subject designs for research on older adults. *Research on Aging*, *7*, 46–80.

Reichardt, W. (1961). *Autocorrelation, a principle for the evaluation of sensory information by the central nervous system.* Cambridge, MA, NEW York: MIT Press and Wiley.

Rotondo, J. L. (2000). *Dominance and gender in conversational interaction.* Unpublished doctoral dissertation, University of Notre Dame. (Unpublished doctoral dissertation)

Rotondo, J. L., & Boker, S. M. (in press). Behavioral synchronization in human conversational interaction. In M. Stamenov & V. Gallese (Eds.), *Mirror neurons and the evolution of brain and language.* Amsterdam: John Benjamins.

Santen, J. P. H. van, & Sperling, G. (1985). Elaborated Reichardt detectors. *Journal of Optical Society of America*, *2A*, 300-320.

Scheflen, A. E. (1964). The significance of posture in communication systems. *Psychiatry*, *27*, 316-331.

Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, *85*(2), 461–464.

Shao, X. M., & Chen. (1987). Normalized auto– and cross–covariance fucntions for neuronal spike train analysis. *International Journal of Neuroscience*, *34*, 85-95.

Siegler, R. S., & Jenkens, E. (1989). *How children discover new strategies.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Warner, R. M. (1998). *Spectral analysis of time–series data.* New York: Guilford Press.

West, S. G., & Hepworth, J. T. (1991). Statistical issues in the study of temporal data: Daily experiences. *Journal of Personality*, *59*(3), 609–662.