Entropy **2013**, *15*, 767-788; doi:10.3390/e15030767

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Transfer Entropy for Coupled Autoregressive Processes

Daniel W. Hahs [1] and Shawn D. Pethel [2,*]

[1] Torch Technologies, Inc. Huntsville, AL 35802, USA; E-Mail: dan.hahs@torchtechnologies.com

[2] U.S. Army, Redstone Arsenal, Huntsville, AL 35898, USA

* Author to whom correspondence should be addressed; E-Mail*:* shawn.d.pethel.civ@mail.mil; Tel.: 256-842-9734; Fax: 256-842-2507.

Received: 26 January 2013; in revised form: 13 February 2013 / Accepted: 19 February 2013 / Published: 25 February 2013

Abstract: A method is shown for computing transfer entropy over multiple time lags for coupled autoregressive processes using formulas for the differential entropy of multivariate Gaussian processes. Two examples are provided: (1) a first-order filtered noise process whose state is measured with additive noise, and (2) two first-order coupled processes each of which is driven by white process noise. We found that, for the first example, increasing the first-order AR coefficient while keeping the correlation coefficient between filtered and measured process fixed, transfer entropy increased since the entropy of the measured process was itself increased. For the second example, the minimum correlation coefficient occurs when the process noise variances match. It was seen that matching of these variances results in minimum information flow, expressed as the sum of transfer entropies in both directions. Without a match, the transfer entropy is larger in the direction away from the process having the larger process noise. Fixing the process noise variances, transfer entropies in both directions increase with the coupling strength. Finally, we note that the method can be generally employed to compute other information theoretic quantities as well.

Keywords: transfer entropy; autoregressive process; Gaussian process; information transfer

---

1. Introduction

Transfer entropy [1] quantifies the information flow between two processes. Information is defined to be flowing from system X to system Y whenever knowing the past states of X reduces the uncertainty of one or more of the current states of Y above and beyond what uncertainty reduction is achieved by only knowing the past Y states. Transfer entropy is the mutual information between the current state of system Y and one or more past states of system X, conditioned on one or more past states of system Y. We will employ the following notation. Assume that data from two systems X and Y are simultaneously available at k timestamps: $t_{n-k+2:n+1} \equiv \{t_{n-k+2}, t_{n-k+2}, ..., t_n, t_{n+1}\}$. Then we express transfer entropies as:

$$TE_{x\to y}^{(k)} = I\left(y_{n+1}; x_{n-k+2:n} \mid y_{n-k+2:n}\right) = H\left(y_{n+1} \mid y_{n-k+2:n}\right) - H\left(y_{n+1} \mid y_{n-k+2:n}, x_{n-k+2:n}\right) \quad (1)$$

$$TE_{y\to x}^{(k)} = I\left(x_{n+1}; y_{n-k+2:n} \mid x_{n-k+2:n}\right) = H\left(x_{n+1} \mid x_{n-k+2:n}\right) - H\left(x_{n+1} \mid x_{n-k+2:n}, y_{n-k+2:n}\right) \quad (2)$$

Each of the two transfer entropy values $TE_{x\to y}$ and $TE_{y\to x}$ is nonnegative and both will be positive (and not necessarily equal) when information flow is bi-directional. Because of these properties, transfer entropy is useful for detecting causal relationships between systems generating measurement time series. Indeed, transfer entropy has been shown to be equivalent, for Gaussian variables, to Granger causality [2]. Reasons for caution about making causal inferences in some situations using transfer entropy, however, are discussed in [3–6]. A formula for normalized transfer entropy is provided in [7].

The contribution of this paper is to explicitly show how to compute transfer entropy over a variable number of time lags for autoregressive (AR) processes driven by Gaussian noise and to gain insight into the meaning of transfer entropy in such processes by way of two example systems: (1) a first-order AR process $X = \{x_n\}$ with its noisy measurement process $Y = \{y_n\}$, and (2) a set of two mutually-coupled AR processes. Computation of transfer entropies for these systems is a worthwhile demonstration since they are simple models that admit intuitive understanding. In what follows we first show how to compute the covariance matrix for successive iterates of the example AR processes and then use these matrices to compute transfer entropy quantities based on the differential entropy expression for multivariate Gaussian random variables. Plots of transfer entropies *versus* various system parameters are provided to illustrate various relationships of interest.

Note that Kaiser and Schreiber [8] have previously shown how to compute information transfer metrics for continuous-time processes. In their paper they provide an explicit example, computing transfer entropy for two linear stochastic processes where one of the processes is autonomous and the other is coupled to it. To perform the calculation for the Gaussian processes the authors utilize expressions for the differential entropy of multivariate Gaussian noise. In our work, we add to this understanding by showing how to compute these quantities analytically for higher time lags. We now provide a discussion of differential entropy, the formulation of entropy appropriate to continuous-valued processes as we are considering.

2. Differential Entropy

The entropy of a continuous-valued process is given by its differential entropy. Recall that the entropy of a discrete-valued random variable is given by the *Shannon entropy* $H = -\sum_i p_i \log p_i$ (we shall always choose log base 2 so that entropy will be expressed in units of bits) where p_i is the probability of the i^{th} outcome and the sum is over all possible outcomes.

Following [9] we derive the appropriate expression for differential entropies for conditioned and unconditioned continuous-valued random variables. When a process X is continuous-valued we may approximate it as a discrete-value process by identifying $p_i = f_i \Delta x$ where f_i is the value of the pdf at the i^{th} partition point and Δx is the refinement of the partition. We then obtain:

$$\begin{aligned} H(X) &= -\sum_i p_i \log p_i \\ &= -\sum_i f_i \Delta x \log f_i \Delta x \\ &= -\sum_i f_i \Delta x (\log f_i + \log \Delta x) \\ &= -\sum_i f_i \log f_i \Delta x - \sum_i \log \Delta x f_i \Delta x \\ &= -\int f \log f dx - \log \Delta x \int f dx \\ &= h(X) - \log \Delta x \end{aligned} \tag{3}$$

Note that since the X process is continuous-valued, then, as $\Delta x \to 0$, we have $H(X) \to +$ infinity. Thus, for continuous-valued processes, the quantity h(X), when itself defined and finite, is used to represent the entropy of the process. This quantity is known as the *differential entropy* of random process X.

Closed-form expressions for the differential entropy of many distributions are known. For our purposes, the key expression is the one for the (unconditional) multivariate normal distribution [10]. Let the probability density function of the n-dimensional random vector x be denoted f(x), then the relevant expressions are:

$$\begin{aligned} f(\bar{x}) &= \frac{\exp\left[-\frac{1}{2}(\bar{x}-\bar{\mu})^T C^{-1}(\bar{x}-\bar{\mu})\right]}{(2\pi)^{\frac{n}{2}}[\det C]^{\frac{1}{2}}} \\ h(\bar{x}) &= -\int f(\bar{x}) \log[f(\bar{x})] dx \\ &= \frac{1}{2}\log\left[(2\pi e)^n \det C\right] \end{aligned} \tag{4}$$

where detC is the determinant of matrix C, the covariance of x. In what follows, this expression will be used to compute differential entropy of unconditional and conditional normal probability density functions. The case for conditional density functions warrants a little more discussion.

Recall that the relationships between the joint and conditional covariance matrices, C_{XY} and $C_{Y|X}$, respectively, of two random variables X and Y (having dimensions n_x and n_y, respectively) are given by:

$$C_{XY} = \operatorname{cov}\left(\begin{bmatrix} X \\ Y \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$
$$\operatorname{cov}[Y \mid X = x] = C_{Y|X} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \tag{5}$$

Here blocks Σ_{11} and Σ_{22} have dimensions n_x by n_x and n_y by n_y, respectively. Now, using Leibniz's formula, we have that:

$$\det C_{XY} = \det\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} = \det\Sigma_{11}\det\left(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right) = \det C_X \det C_{Y|X}. \tag{6}$$

Hence the conditional differential entropy of Y, given X, may be conveniently computed using:

$$\begin{aligned} h(Y \mid X) &= \frac{1}{2}\log\left[(2\pi e)^{n_y}\det C_{Y|X}\right] \\ &= \frac{1}{2}\log\left[(2\pi e)^{n_y}\frac{\det C_{XY}}{\det C_X}\right] \\ &= \frac{1}{2}\log\left[(2\pi e)^{n_x+n_y}\det C_{XY}\right] - \frac{1}{2}\log\left[(2\pi e)^{n_x}\det C_X\right] \\ &= h(X,Y) - h(X). \end{aligned} \tag{7}$$

This formulation is very handy as it allows us to compute many information-theoretic quantities with ease. The strategy is as follows. We define $C^{(k)}$ to be the covariance of two random processes sampled at k consecutive timestamps $\{t_{n-k+2}, t_{n-k+1}, \ldots, t_n, t_{n+1}\}$. We then compute transfer entropies for values of k up to k sufficiently large to ensure that their valuations do not change significantly if k is further increased. For our examples, we have found k = 10 to be more than sufficient. A discussion of the importance of considering this sufficiency is provided in [11].

3. Transfer Entropy Computation Using Variable Number of Timestamps

We wish to consider two example processes each of which conforms to one of the two model systems having the general expressions:

$$(1)\quad \begin{cases} x_{n+1} = a_0 x_n + a_1 x_{n-1} + \cdots + a_m x_{n-m} + w_n \\ y_{n+1} = c_{-1} x_{n+1} + v_n \\ v_n \sim N(0,R), w_n \sim N(0,Q) \end{cases} \tag{8}$$

and:

$$(2)\quad \begin{cases} x_{n+1} = a_0 x_n + a_1 x_{n-1} + \cdots + a_m x_{n-m} + b_0 y_n + b_1 y_{n-1} + \cdots + b_j y_{n-j} + w_n \\ y_{n+1} = c_0 x_n + c_1 x_{n-1} + \cdots + c_m x_{n-m} + d_0 y_n + d_1 y_{n-1} + \cdots + d_j y_{n-j} + v_n \\ v_n \sim N(0,R), w_n \sim N(0,Q). \end{cases} \tag{9}$$

Here, v_n and w_n are zero mean uncorrelated Gaussian noise processes having variances R and Q, respectively. For system stability, we require the model poles to lie within the unit circle.

The first model is of a filtered process noise X one-way coupled to an instantaneous, but noisy measurement process Y. The second model is a two-way coupled pair of processes, X and Y.

Transfer entropy (as defined by Schreiber [1]) considers the flow of information from past states (*i.e.*, state values having, timetags $t_{n-k+2:n} \equiv \{t_{n-k+2}, t_{n-k+2}, ..., t_n\}$) of one process to the present (t_{n+1}) state of another process. However, note that in the first general model (measurement process) there is an explicit flow of information from the present state of the X process; x_{n+1} determines the present state of the Y process y_{n+1} (assuming c_{-1} is not zero). To fully capture the information transfer from the X process to the current state of the Y process we must identify the correct causal states [4]. For the measurement system, the causal states include the current (present) state. This state is not included in the definition of transfer entropy, being a mutual information quantity conditioned on only past states. Hence, for the purpose of this paper, we will temporarily define a quantity, "information transfer," similar to transfer entropy, except that the present of the driving process, x_{n+1}, will be lumped in with the past values of the X process: $x_{n-k+2}:x_n$. For the first general model there is no information transferred from the Y to the X process. We define the (non-zero) information transfer from the X to the Y process (based on data from k timetags) as:

$$IT_{x\to y}^{(k)} = I\left(y_{n+1}; x_{n-k+2:n+1} \mid y_{n-k+2:n}\right) = H\left(y_{n+1} \mid y_{n-k+2:n}\right) - H\left(y_{n+1} \mid y_{n-k+2:n}, x_{n-k+2:n+1}\right). \quad (10)$$

The major contribution of this paper is to show how to analytically compute transfer entropy for AR Gaussian processes using an iterative method for computing the required covariance matrices. Computation of information transfer is additionally presented to elucidate the power of the method when similar information quantities are of interest and to make the measurement example more interesting. We now present a general method for computing the covariance matrices required to compute information-theoretic quantities for the AR models above. Two numerical examples follow.

To compute transfer entropy over a variable number of multiple time lags for AR processes of the general types shown above, we compute its block entropy components over multiple time lags. By virtue of the fact that the processes are Gaussian we can avail ourselves of analytical entropy expressions that depend only on the covariance of the processes. In this section we show how to analytically obtain the required covariance expressions starting with the covariance for a single time instance. Taking expectations, using the AR equations, we obtain the necessary statistics to characterize the process. Representing these expectation results in general, the process covariance matrix $C^{(1)}(t_n)$ corresponding to a single timestamp, t_n, is:

$$C^{(1)}(t_n) \equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}\right) = \begin{bmatrix} E[x_n^2] & E[x_n y_n] \\ E[y_n x_n] & E[y_n^2] \end{bmatrix}. \quad (11)$$

To obtain an expanded covariance matrix, accounting for two time instances (t_n and t_{n+1}), we compute the additional expectations required to fill in the matrix $C^{(2)}(t_n)$:

$$C^{(2)}(t_n) \equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \\ x_{n+1} \\ y_{n+1} \end{bmatrix}\right) = \begin{bmatrix} E[x_n^2] & E[x_n y_n] & E[x_n x_{n+1}] & E[x_n y_{n+1}] \\ E[x_n y_n] & E[y_n^2] & E[x_{n+1} y_n] & E[y_n y_{n+1}] \\ E[x_n x_{n+1}] & E[x_{n+1} y_n] & E[x_{n+1}^2] & E[x_{n+1} y_{n+1}] \\ E[x_n y_{n+1}] & E[y_n y_{n+1}] & E[x_{n+1} y_{n+1}] & E[y_{n+1}^2] \end{bmatrix}. \quad (12)$$

Because the process is stationary, we may write:

$$C^{(2)}(t_n) = C^{(2)} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{13}$$

where:

$$\begin{aligned} \Sigma_{11} &\equiv \begin{bmatrix} E[x_n^2] & E[x_n y_n] \\ E[x_n y_n] & E[y_n^2] \end{bmatrix} \\ \Sigma_{12} &\equiv \begin{bmatrix} E[x_n x_{n+1}] & E[x_n y_{n+1}] \\ E[x_{n+1} y_n] & E[y_n y_{n+1}] \end{bmatrix} \\ \Sigma_{21} &= \Sigma_{12}^T \\ \Sigma_{22} &= \Sigma_{11}. \end{aligned} \tag{14}$$

Thus we have found the covariance matrix $C^{(2)}$ required to compute block entropies based on two timetags or, equivalently, one time lag. Using this matrix the single-lag transfer entropies may be computed.

We now show how to compute the covariance matrices corresponding to any finite number of time stamps. Define vector $\bar{z}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$. Using the definitions above, write the matrix $C^{(2)}$ as a block matrix and, using standard formulas, compute the conditional mean and covariance C_c of $\bar{z}_{n+1}$ given $\bar{z}_n$:

$$\begin{aligned} C^{(2)} &= \mathrm{cov}\left(\begin{bmatrix} \bar{z}_n \\ \bar{z}_{n+1} \end{bmatrix}\right) = E\left[\left(\begin{bmatrix} \bar{z}_n \\ \bar{z}_{n+1} \end{bmatrix}\begin{bmatrix} \bar{z}_n & \bar{z}_{n+1} \end{bmatrix}\right)\right] = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \\ E[\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}] &= E[\bar{z}_n] + \Sigma_{21}\Sigma_{11}^{-1}[\bar{z} - E[\bar{z}_n]] \\ &= \bar{\mu}_{\bar{z}} + \Sigma_{21}\Sigma_{11}^{-1}[\bar{z} - \bar{\mu}_{\bar{z}}] \\ C_c &\equiv \mathrm{cov}[\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}] = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \end{aligned} \tag{15}$$

Note that the expected value of the conditional mean is zero since the mean of the $\bar{z}_n$ process, $\bar{\mu}_{\bar{z}}$, is itself zero.

With these expressions in hand, we note that we may view propagation of the state $\bar{z}_n$ to its value $\bar{z}_{n+1}$ at the next timestamp as accomplished by the recursion:

$$\begin{aligned} \bar{z}_{n+1} &= \bar{\mu}_z + D(\bar{z}_n - \bar{\mu}_z) + S\bar{u}_n : \bar{u}_n \sim N(0_2, I_2) \\ D &\equiv \Sigma_{21}\Sigma_{11}^{-1} \\ C_c &\equiv SS^T \equiv \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \end{aligned} \tag{16}$$

Here S is the principal square root of the matrix C_c. It is conveniently computed using the inbuilt Matlab function *sqrtm*. To see analytically that the recursion works, note that using it we recover at each timestamp a process having the correct mean and covariance:

$$E\{\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}\} = E\{\bar{\mu}_z + D(\bar{z}_n - \bar{\mu}_z) + S\bar{u}_n \mid \bar{z}_n = \bar{z}\} = \bar{\mu}_z + D(\bar{z} - \bar{\mu}_z) \tag{17}$$

and:

$$\bar{z}_{n+1} - E\{\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}\} = \bar{\mu}_z + D(\bar{z}_n - \bar{\mu}_z) + S\bar{u}_n - (\bar{\mu}_z + D(\bar{z} - \bar{\mu}_z)) = S\bar{u}_n + D(\bar{z}_n - \bar{z})$$

$$\begin{aligned} \operatorname{cov}(\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}) &= E\left\{[\bar{z}_{n+1} - E\{\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}\}][\bar{z}_{n+1} - E\{\bar{z}_{n+1} \mid \bar{z}_n = \bar{z}\}]^T \mid \bar{z}_n = \bar{z}\right\} \\ &= E\left\{[S\bar{u}_n + D(\bar{z}_n - \bar{z})][S\bar{u}_n + D(\bar{z}_n - \bar{z})]^T \mid \bar{z}_n = \bar{z}\right\} \\ &= E\left\{[S\bar{u}_n][S\bar{u}_n]^T\right\} = SE\{\bar{u}_n\bar{u}_n^T\}S^T = SS^T. \end{aligned} \tag{18}$$

Thus, because the process is Gaussian and fully specified by its mean and covariance, we have verified that the recursive representation yields consistent statistics for the stationary AR system. Using the above insights, we may now recursively compute the covariance matrix $C^{(k)}$ for a variable number (k) of timestamps. Note that $C^{(k)}$ has dimensions of 2k × 2k. We denote 2 × 2 blocks of $C^{(k)}$ as $C^{(k)}_{ij}$ for i, j = 1,2, ..., k , where $C^{(k)}_{ij}$ is the 2-by-2 block of $C^{(k)}$ consisting of the four elements of $C^{(k)}$ that are individually located in row 2i − 1 or 2i and column 2j − 1 or 2j.

The above recursion is now used to compute the block elements of $C^{(3)}$. Then each of these block elements is, in turn, expressed in terms of block elements of $C^{(2)}$. These calculations are shown in detail below where we have also used the fact that the mean of the z_n vector is zero:

$$\begin{aligned} C^{(3)}_{ij} &= C^{(2)}_{ij} : i = 1,2;\ j = 1,2 \\ \bar{z}_{n+2} &= D\bar{z}_{n+1} + S\bar{u}_{n+1} \\ &= D[D\bar{z}_n + S\bar{u}_n] + S\bar{u}_{n+1} = D^2\bar{z}_n + DS\bar{u}_n + S\bar{u}_{n+1} \end{aligned} \tag{19}$$

$$\begin{aligned} C^{(3)}_{13} &= E[\bar{z}_n\bar{z}^T_{n+2}] = E\left[\bar{z}_n\left(D^2\bar{z}_n + DS\bar{u}_n + S\bar{u}_{n+1}\right)^T\right] = \Sigma_{11}[D^2]^T \\ C^{(3)}_{31} &= [C^{(3)}_{13}]^T \end{aligned} \tag{20}$$

$$\begin{aligned} C^{(3)}_{23} &= E[\bar{z}_{n+1}\bar{z}^T_{n+2}] = E\left[(D\bar{z}_n + S\bar{u}_n)\left(D^2\bar{z}_n + DS\bar{u}_n + S\bar{u}_{n+1}\right)^T\right] \\ &= D\Sigma_{11}[D^2]^T + C_cD^T = DC^{(3)}_{13} + C_cD^T \\ C^{(3)}_{32} &= [C^{(3)}_{23}]^T \end{aligned} \tag{21}$$

$$\begin{aligned} C^{(3)}_{33} &= E[\bar{z}_{n+2}\bar{z}^T_{n+2}] = E\left[\left(D^2\bar{z}_n + DS\bar{u}_n + S\bar{u}_{n+1}\right)\left(D^2\bar{z}_n + DS\bar{u}_n + S\bar{u}_{n+1}\right)^T\right] \\ &= D^2\Sigma_{11}[D^2]^T + DC_cD^T + C_c = DC^{(3)}_{23} + C_c. \end{aligned} \tag{22}$$

By continuation of this calculation to larger timestamp blocks (k > 3), we find the following pattern that can be used to extend (augment) $C^{(k-1)}$ to yield $C^{(k)}$. The pattern consists of setting most of the augmented matrix equal to that of the previous one, and then computing two additional rows and columns for $C^{(k)}$, k > 2, to fill out the remaining elements. The general expressions are:

$$\begin{aligned} C^{(k)}_{m,n} &= C^{(k-1)}_{m,n} : m,n = 1,2,\ldots,k-1 \\ C^{(k)}_{1k} &= \Sigma_{11}[D^{k-1}]^T \\ C^{(k)}_{ik} &= DC^{(k)}_{i-1,k} + C_c[D^{k-i}]^T : i = 2,3,\ldots k \\ C^{(k)}_{ki} &= [C^{(k)}_{ik}]^T : i = 1,2,\ldots,k. \end{aligned} \tag{23}$$

At this point in the development we have shown how to compute the covariance matrix:

$$C^{(k)} = \mathrm{cov}\left(\bar{z}^{(k)}\right) = \mathrm{cov}\left(\left[x_n \quad y_n \quad x_{n+1} \quad y_{n+1} \quad \cdots \quad x_{n+k-1} \quad y_{n+k-1}\right]^T\right) \tag{24}$$

Since the system is linear and the process noise w_n and measurement noise v_n are white zero-mean Gaussian noise processes, we may express the joint probability density function for the 2k variates as:

$$f\left(\bar{z}^{(k)}\right) = pdf\left(\bar{z}^{(k)}\right) = pdf\left(\left[x_n \quad y_n \quad x_{n+1} \quad y_{n+1} \quad \cdots \quad x_{n+k-1} \quad y_{n+k-1}\right]\right) = \frac{\exp\left\{-\frac{1}{2}\left[\bar{z}^{(k)}\right]^T\left[C^{(k)}\right]^{-1}\left[\bar{z}^{(k)}\right]\right\}}{(2\pi)^{\frac{n}{2}}\left(\det\left[C^{(k)}\right]\right)^{\frac{1}{2}}} \tag{25}$$

Note that the mean of all 2k variates is zero.

Finally, to obtain empirical confirmation of the equivalence of the covariance terms obtained using the original AR system and its recursive representation, numerical simulations were conducted. Using the example 1 system (below) 500 sequences were generated each of length one million. For each sequence the $C^{(3)}$ covariance was computed. The error for all $C^{(3)}$ matrices was then averaged, assuming that the $C^{(3)}$ matrix calculated using the method based on the recursive representation was the true value. The result was that for each of the matrix elements, the error was less than 0.0071% of its true value. We are now in position to compute transfer entropies for a couple of illustrative examples.

4. Example 1: A One-Way Coupled System

For this example we consider the following system:

$$\begin{aligned} x_{n+1} &= ax_n + w_n : w_n \sim N(0,Q) \\ y_{n+1} &= h_c x_{n+1} + v_n : v_n \sim N(0,R) \end{aligned} \tag{26}$$

Parameter h_c specifies the coupling strength of the Y process to the first-order AR process X, and R and Q are their respective (w_n and v_n) zero-mean Gaussian process noise variances. For stability, we require $|a| < 1$. Comparing to the first general representation given above, we have $m = 0$, $a_0 = a$, and $c_{-1} = h_c a$. The system models filtered noise x_n and a noisy measurement, y_n, of x_n. Thus the x_n sequence represents a hidden process (or model) which is observable by way of another sequence, y_n. We wish to examine the behavior of transfer entropy as a function of the correlation ρ between x_n and y_n. One might expect that the correlation ρ between x_n and y_n to be proportional of the degree of information flow; however, we will see that the relationship between transfer entropy and correlation is not quite that simple.

Both the X and Y processes have zero mean. Computing the joint covariance matrix $C^{(1)}$ for x_n and y_n and their correlation we obtain:

$$\begin{aligned} Var(x_n) &= \frac{Q}{1-a^2} \\ Var(y_n) &= h_c^2 Var(x_n) + R \\ E(x_n y_n) &= h_c Var(x_n) \\ \rho &\equiv \frac{E(x_n y_n)}{\sqrt{Var(x_n)Var(y_n)}} \end{aligned} \tag{27}$$

Hence the process covariance matrix $C^{(1)}$ corresponding to a single timestamp, t_n is:

$$C^{(1)} \equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}\right) = \begin{bmatrix} Var(x_n) & hVar(x_n) \\ h_c Var(x_n) & h_c^2 Var(y_n) + R \end{bmatrix}. \tag{28}$$

In order to obtain an expanded covariance matrix, accounting for two time instances (t_n and t_{n+1}) we compute the additional expectations required to fill in the matrix $C^{(2)}$:

$$C^{(2)} \equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \\ x_{n+1} \\ y_{n+1} \end{bmatrix}\right) = \begin{bmatrix} Var(x_n) & h_c Var(x_n) & aVar(x_n) & h_c aVar(x_n) \\ h_c Var(x_n) & h_c^2 Var(x_n) + R & h_c aVar(x_n) & h_c^2 aVar(x_n) \\ aVar(x_n) & h_c aVar(x_n) & Var(x_n) & h_c Var(x_n) \\ h_c aVar(x_n) & h_c^2 aVar(x_n) & h_c Var(x_n) & h_c^2 Var(x_n) + R \end{bmatrix}. \tag{29}$$

Thus we have found the covariance matrix $C^{(2)}$ required to compute block entropies based on a single time lag. Using this matrix the single-lag transfer entropies may be computed. Using the recursive process described in the previous section we can compute $C^{(1\circ)}$. We have found that using higher lags does not change the entropy values significantly.

To aid the reader in understanding the calculations required to compute transfer entropies using higher time lags, it is worthwhile to compute transfer entropy for a single lag. We first define transfer entropy using general notation indicating the partitioning of the X and Y sequences in to past and future $(\overleftarrow{x}, \vec{x})$ and $(\overleftarrow{y}, \vec{y})$, respectively. We then compute transfer entropy as a sum of block entropies:

$$\begin{aligned} TE_{x->y} &= I(\overleftarrow{x}; \vec{y} \mid \overleftarrow{y}) = h(\overleftarrow{x} \mid \overleftarrow{y}) + h(\vec{y} \mid \overleftarrow{y}) - h(\overleftarrow{x}; \vec{y} \mid \overleftarrow{y}) \\ &= [h(\overleftarrow{x}, \overleftarrow{y}) - h(\overleftarrow{y})] + [h(\vec{y}, \overleftarrow{y}) - h(\overleftarrow{y})] - [h(\overleftarrow{x}, \vec{y}, \overleftarrow{y}) - h(\overleftarrow{y})] \\ &= h(\overleftarrow{x}, \overleftarrow{y}) + h(\vec{y}, \overleftarrow{y}) - h(\overleftarrow{y}) - h(\overleftarrow{x}, \vec{y}, \overleftarrow{y}). \end{aligned} \tag{30}$$

Similarly:

$$TE_{y->x} = I(\overleftarrow{y}; \vec{x} \mid \overleftarrow{x}) = h(\overleftarrow{x}, \overleftarrow{y}) + h(\overleftarrow{x}, \vec{x}) - h(\overleftarrow{x}) - h(\overleftarrow{x}, \vec{y}, \vec{x}) \tag{31}$$

The Y states have no influence on the X sequence in this example. Hence $TE_{y\to x} = 0$. Since we are here computing transfer entropy for a single lag (*i.e.*, two time tags t_n and t_{n+1}) we have:

$$TE^{(2)}_{x->y} = I(x_n; y_{n+1} \mid y_n) = h(x_n, y_n) + h(y_n, y_{n+1}) - h(y_n) - h(x_n, y_n, y_{n+1}) \tag{32}$$

By substitution of the expression for the differential entropy of each block we obtain:

$$\begin{aligned} TE^{(2)}_{x->y} &= \frac{1}{2}\log\left[(2\pi e)^2 \det C^{(2)}_{[1,2],[1,2]}\right] + \frac{1}{2}\log\left[(2\pi e)^2 \det C^{(2)}_{[2,4],[2,4]}\right] - \\ &\frac{1}{2}\log\left[(2\pi e)^1 \det C^{(2)}_{[2],[2]}\right] - \frac{1}{2}\log\left[(2\pi e)^3 \det C^{(2)}_{[1,2,4],[1:,2,4]}\right] \\ &= \frac{1}{2}\log\left[\frac{\det C^{(2)}_{[1,2],[1,2]} \det C^{(2)}_{[2,4],[2,4]}}{\det C^{(2)}_{[2],[2]} \det C^{(2)}_{[1,2,4],[1,2,4]}}\right]. \end{aligned} \tag{33}$$

For this example, note from the equation for y_{n+1} that state x_{n+1} is a causal state of X influencing the value of y_{n+1}. In fact, it is the most important such state. To capture the full information that is

transferred from the X process to the Y process over the course of two time tags we need to include state x_{n+1}. Hence we compute the information transfer from x → y as:

$$IT^{(2)}_{x->y} = I(x_n, x_{n+1}; y_{n+1} \mid y_n) = h(x_n, x_{n+1}, y_n) + h(y_n, y_{n+1}) - h(y_n) - h(x_n, x_{n+1}, y_n, y_{n+1}) \tag{34}$$

$$\begin{aligned} IT^{(2)}_{x->y} &= \frac{1}{2}\log\left[(2\pi e)^3 \det C^{(2)}_{[1,2,3],[1,2,3]}\right] + \frac{1}{2}\log\left[(2\pi e)^2 \det C^{(2)}_{[2,4],[2,4]}\right] - \\ &\frac{1}{2}\log\left[(2\pi e)^1 \det C^{(2)}_{[2],[2]}\right] - \frac{1}{2}\log\left[(2\pi e)^4 \det C^{(2)}_{[1:4],[1:4]}\right] \\ &= \frac{1}{2}\log\left[\frac{\det C^{(2)}_{[1:,2,3],[1,2,3]} \det C^{(2)}_{[2,4],[2,4]}}{\det C^{(2)}_{[2],[2]} \det C^{(2)}_{[1:4],[1:4]}}\right]. \end{aligned} \tag{35}$$

Here the notation $\det C^{(2)}_{[i],[i]}$ indicates the determinant of the matrix composed of the rows and columns of $C^{(2)}$ indicated by the list of indices i shown in the subscripted brackets. For example, $\det C^{(2)}_{[1:4],[1:4]}$ is the determinant of the matrix formed by extracting columns {1, 2, 3, 4} and rows {1, 2, 3, 4} from matrix $C^{(2)}$. In later calculations we will use slightly more complicated-looking notation. For example, $\det C^{(10)}_{[2:2:20],[2:2:20]}$ is the determinant of the matrix formed by extracting columns {2, 4 ,…, 18, 20} and the same-numbered rows from matrix $C^{(1\circ)}$. (Note $C^{(k)}_{[i],[i]}$ is not the same as $C^{(k)}_{ii}$ as used in Section 3).

It is interesting to note that a simplification in the expression for information transfer can be obtained by writing the expression for it in terms of conditional entropies:

$$IT^{(2)}_{x->y} = I(x_n, x_{n+1}; y_{n+1} \mid y_n) = h(y_{n+1} \mid y_n) - h(y_{n+1} \mid x_n, y_n, x_{n+1}) \tag{36}$$

From the fact that $y_{n+1} = x_{n+1} + v_{n+1}$ we see immediately that:

$$h(y_{n+1} \mid x_n, y_n, x_{n+1}) = h(v_{n+1}) = \frac{1}{2}\log(2\pi eR). \tag{37}$$

Hence we may write:

$$\begin{aligned} IT^{(2)}_{x->y} &= h(y_{n+1} \mid y_n) - h(y_{n+1} \mid x_n, y_n, x_{n+1}) \\ &= \frac{1}{2}\log\left[\frac{2\pi e \det C^{(2)}_{[2,4],[2,4]}}{\det C^{(2)}_{[2],[2]}}\right] - \frac{1}{2}\log[2\pi eR] \\ &= \frac{1}{2}\log\left[\frac{\det C^{(2)}_{[2,4],[2,4]}}{R \det C^{(2)}_{[2],[2]}}\right]. \end{aligned} \tag{38}$$

To compute transfer entropy using nine lags (ten timestamps) assume that we have already computed $C^{(10)}$ as defined above. We partition the sequence

$$\left\{\vec{z}^{\,T}_{n+i}\right\}_{i=0}^{9} = \{x_n, y_n, x_{n+1}, y_{n+1}, x_{n+2}, y_{n+2}, x_{n+3}, y_{n+3}, x_{n+4}, y_{n+4}, x_{n+5}, y_{n+5}, x_{n+6}, y_{n+6}, x_{n+7}, y_{n+7}, x_{n+8}, y_{n+8}, x_{n+9}, y_{n+9}\}$$

into three subsets:

$$\begin{aligned} \overleftarrow{x} &\equiv \{x_n, x_{n+1}, \ldots, x_{n+8}\} \\ \overleftarrow{y} &\equiv \{y_n, y_{n+1}, , \ldots, y_{n+8}\} \\ \vec{y} &\equiv \{y_{n+9}\}. \end{aligned} \tag{39}$$

Now, using these definitions, and substituting in expressions for differential block entropies we obtain:

$$\begin{aligned} TE^{(10)}_{x->y} &= I(\vec{x};\vec{y}\,|\,\vec{y}) = h(\vec{x},\vec{y}) + h(\vec{y},\vec{y}) - h(\vec{y}) - h(\vec{x},\vec{y},\vec{y}) \\ &= \frac{1}{2}\log\left[(2\pi e)^{18}\det C^{(10)}_{[1:18],[1:18]}\right] + \frac{1}{2}\log\left[(2\pi e)^{10}\det C^{(10)}_{[2:2:20],[2:2:20]}\right] - \\ &\quad \frac{1}{2}\log\left[(2\pi e)^{9}\det C^{(10)}_{[2:2:18],[2:2:18]}\right] - \frac{1}{2}\log\left[(2\pi e)^{19}\det C^{(10)}_{[1:18,20],[1:18,20]}\right] \\ &= \frac{1}{2}\log\left[\frac{\det C^{(10)}_{[1:18],[1:18]}\det C^{(10)}_{[2:2:20],[2:2:20]}}{\det C^{(10)}_{[2:2:18],[2:2:18]}\det C^{(10)}_{[1:18,20],[1:18,20]}}\right]. \end{aligned} \tag{40}$$

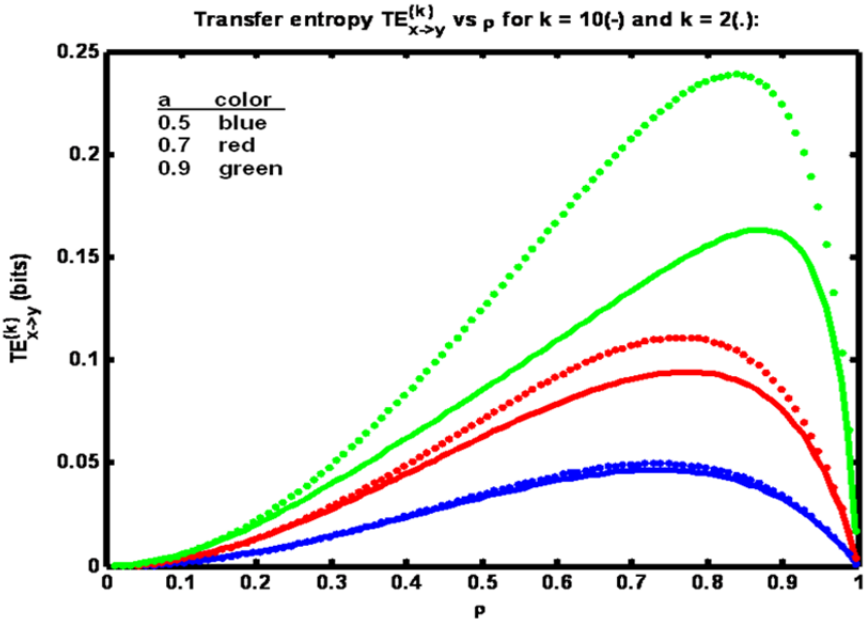
Similarly:

$$IT^{(10)}_{x->y} = h(\vec{y}\,|\,\vec{y}) - h(\vec{y}\,|\,\vec{y},\vec{x},x_{n+1}) = \frac{1}{2}\log\left[\frac{\det C^{(10)}_{[2:2:20],[2:2:20]}}{R\det C^{(10)}_{[2:2:18],[2:2:18]}}\right]. \tag{41}$$

As a numerical example we set h_c = 1, Q = 1, and for three different values of *a* (0.5, 0.7 and 0.9) we vary R so as to scan the correlation ρ between the x and y processes between the values of 0 and 1.

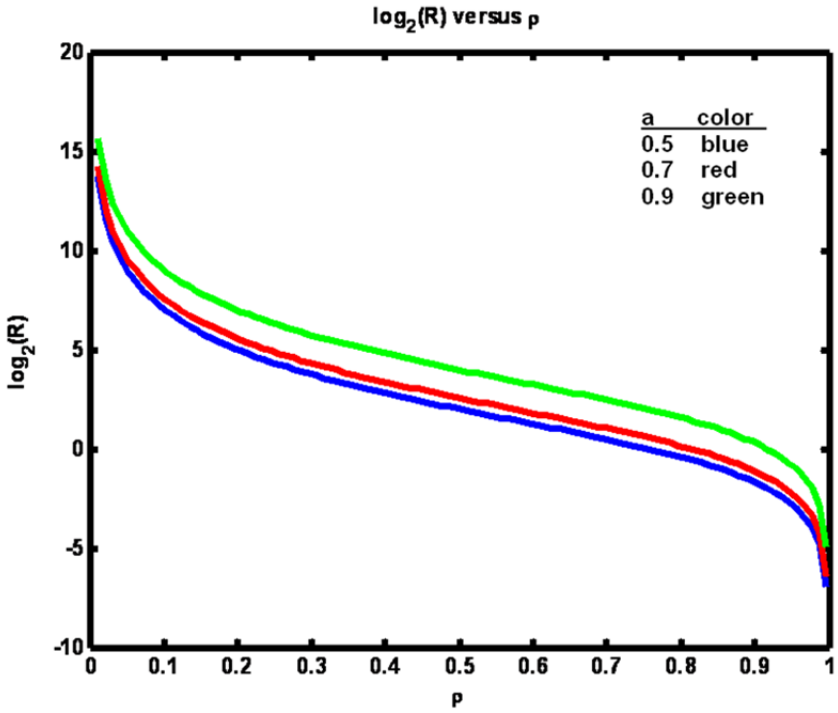
In Figure 1 it is seen that for each value of parameter *a* there is a peak in the transfer entropy $TE^{(k)}_{x\to y}$. As the correlation ρ between x_n and y_n increases from a low value the transfer entropy increases since the amount of information shared between y_{n+1} and x_n is increasing. At a critical value of ρ transfer entropy peaks and then starts to decrease. This decrease is due to the fact that at high values of ρ the measurement noise variance R is small. Hence y_n becomes very close to equaling x_n so that the amount of information gained (about y_{n+1}) by learning x_n, given y_n, becomes small. Hence $h(y_{n+1} \mid y_n) - h(y_{n+1} \mid y_n, x_n)$ is small. This difference is $TE^{(2)}_{x\to y}$.

Figure 1. Example 1: Transfer entropy $TE^{(k)}_{x\to y}$ *versus* correlation coefficient ρ for three values of parameter *a* (see legend). Solid trace: k = 10, dotted trace: k = 2.



The relationship between ρ and R is shown in Figure 2. Note that when parameter *a* is increased, a larger value of R is required to maintain ρ at a fixed value. Also, in Figure 1 we see the effect of including more timetags in the analysis. When k is increased from 2 to 10 transfer entropy values fall, particularly for the largest value of parameter *a*. It is known that entropies decline when conditioned on additional variables. Here, transfer entropy is acting similarly. In general, however, transfer entropy, being a mutual information quantity, has the property that conditioning could make it increase as well [12].

Figure 2. Example 1: Logarithm of R *versus* ρ for three values of parameter *a* (see legend).



The observation that the transfer entropy decrease is greatest for the largest value of parameter *a* is perhaps due to the fact that the entropy of the X process is itself greatest for the largest *a* value and therefore has more sensitivity to an increase in X data availability (Figure 3).

From Figure 1 it is seen that as the value of parameter *a* is increased, transfer entropy is increased for a fixed value of ρ. The reason for this increase may be gleaned from Figure 3 where it is clear that the amount of information contained in the x process, H_X, is greater for larger values of *a*. Hence more information is available to be transferred at the fixed value of ρ when *a* is larger. In the lower half of Figure 3 we see that as ρ increases the entropy of the Y process, H_Y, approaches the value of H_X. This result is due to the fact that the mechanism being used to increase ρ is to decrease R. Hence as R drops close to zero y_n looks increasingly identical to x_n (since $h_c = 1$).

Figure 3. Example 1: Process entropies H_X and H_Y *versus* correlation coefficient ρ for three values of parameter *a* (see legend).

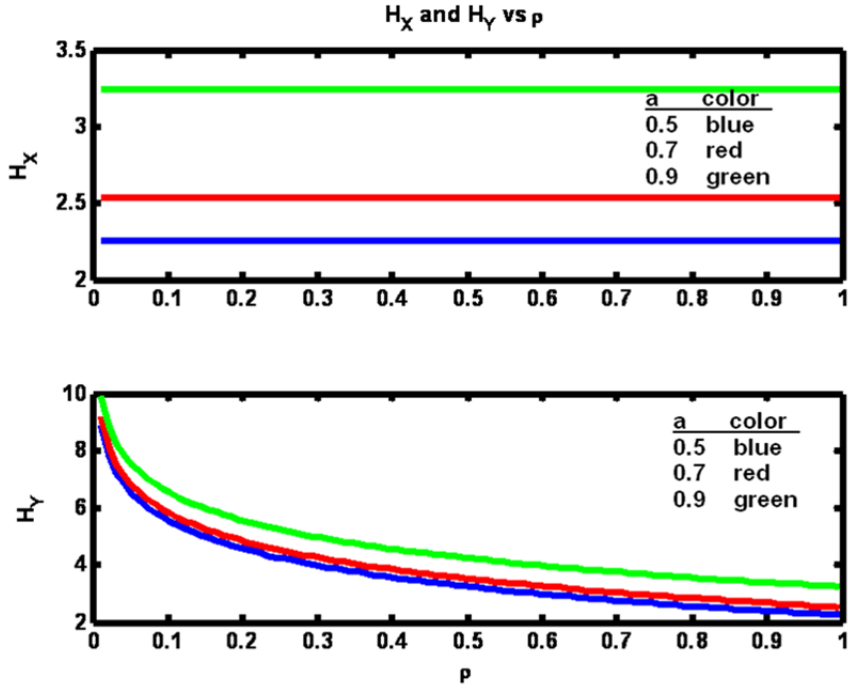
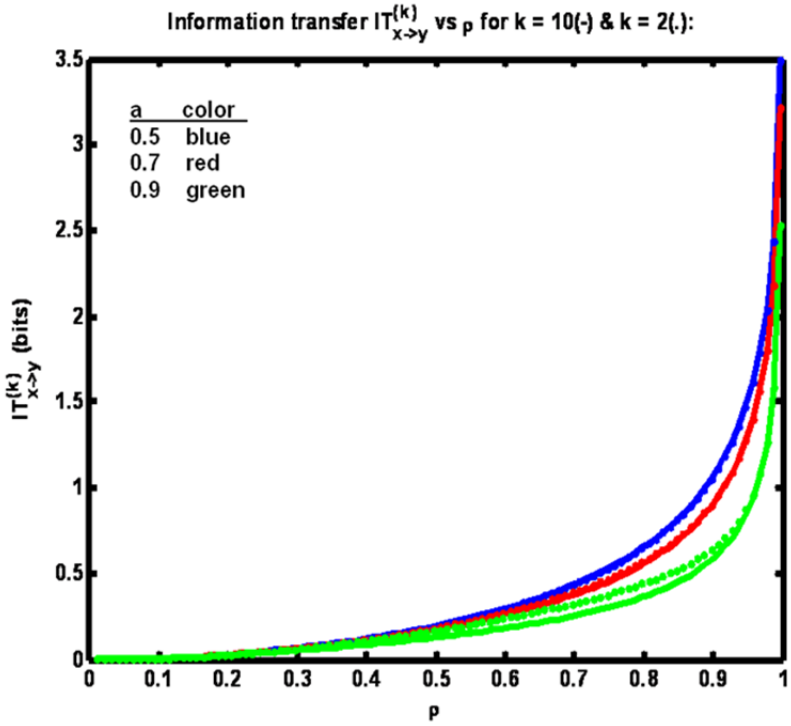


Figure 4 shows information transfer $IT^{(k)}_{x\to y}$ plotted *versus* correlation coefficient ρ. Now note that the trend is for information transfer to increase as ρ is increased over its full range of values. °

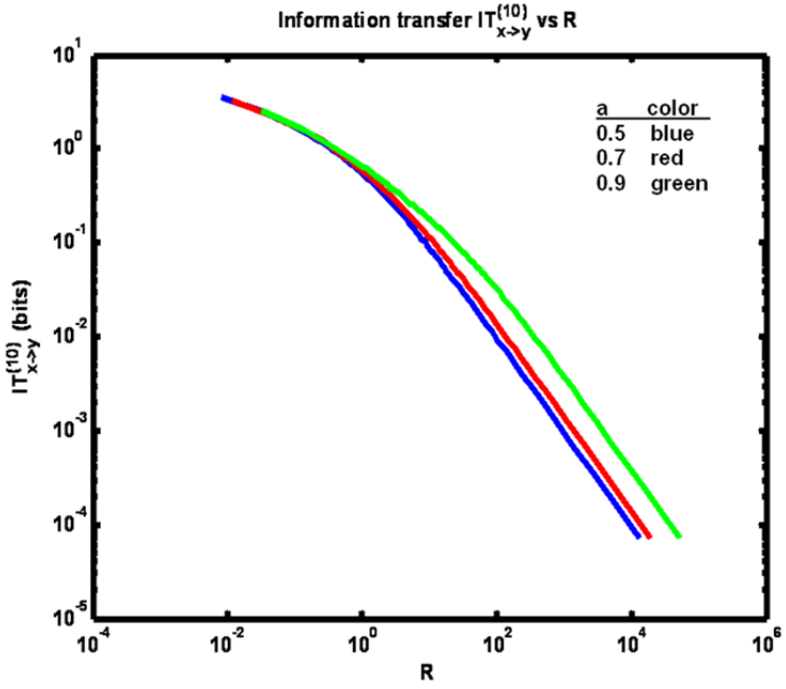
Figure 4. Example 1: Information transfer $IT^{(k)}_{x\to y}$ *versus* correlation coefficient ρ for three different values of parameter *a* (see legend) for k = 10 (solid trace) and k = 2 (dotted trace).



This result is obtained since as ρ is increased y_{n+1} becomes increasingly correlated with x_{n+1}. Also, for a fixed ρ, the lowest information transfer occurs for the largest value of parameter *a*. We obtain this result since at the higher *a* values x_n and x_{n+1} are more correlated. Thus the benefit of learning the value of y_{n+1} through knowledge of x_{n+1} is relatively reduced, given that y_n (itself correlated with x_n) is presumed known. Finally, we have $IT^{(10)}_{x\to y} < IT^{(2)}_{x\to y}$ since conditioning the entropy quantities comprising the expression for information transfer with more state data acts to reduce their difference. Also, by comparison of Figure 2 and Figure 4, it is seen that information transfer is much greater than transfer entropy. This relationship is expected since information transfer as defined herein (for k = 2) is the amount of information that is gained about y_{n+1} from learning x_{n+1} and x_n, given that y_n is already known. Whereas transfer entropy (for k = 2) is the information gained about y_{n+1} from learning only x_n, given that y_n is known. Since the state y_{n+1} in fact equals x_{n+1}, plus noise, learning x_{n+1} is highly informative, especially when the noise variance is small (corresponding to high values of ρ). The difference between transfer entropy and information transfer therefore quantifies the benefit of learning x_{n+1}, given that x_n and y_n are known (when the goal is to determine y_{n+1}).

Figure 5 shows how information transfer varies with measurement noise variance R. As R increases the information transfer decreases since measurement noise makes determination of the value of y_{n+1} from knowledge of x_n and x_{n+1} less accurate. Now, for a fixed R, the greatest value for information transfer occurs for the greatest value of parameter *a*. This is the opposite of what we obtained for a fixed value of ρ as shown in Figure 4. The way to see the rationale for this is to note that, for a fixed value of information transfer, R is highest for the largest value of parameter *a*. This result is obtained since larger values of *a* yield the most correlation between states x_n and x_{n+1}. Hence, even though the measurement y_{n+1} of x_{n+1} is more corrupted by noise (due to higher R), the same information transfer is achieved nevertheless, because x_n provides a good estimate of x_{n+1} and, thus, of y_{n+1}.

Figure 5. Example 1: Information transfer $IT^{(10)}_{x\to y}$ *versus* measurement error variance R for three different values of parameter a (see legend).



5. Example 2: Information-theoretic Analysis of Two Coupled AR Processes.

In example 1 the information flow was unidirectional. We now consider a bidirectional example achieved by coupling two AR processes. One question we may ask in such a system is how transfer entropies change with variations in correlation and coupling coefficient parameters. It might be anticipated that increasing either of these quantities will have the effect of increasing information flow and thus transfer entropies will increase.

The system is defined by the equations:

$$\begin{aligned} x_{n+1} &= ax_n + by_n + w_n : w_n \sim N(0,Q) \\ y_{n+1} &= cx_n + dy_n + v_n : v_n \sim N(0,R). \end{aligned} \tag{42}$$

For stability, we require that the eigenvalues of the constant matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ lie in the unit circle. The means of processes X and Y are zero. The terms w_n and v_n are the X and Y processes noise terms respectively. Using the following definitions:

$$\begin{aligned} \lambda_0 &\equiv 1 + ad - bc \\ \lambda_1 &\equiv 1 - ad - bc \\ \psi_a &\equiv (1-ad)(1-a^2) - bc(1+a^2) \\ \psi_d &\equiv (1-ad)(1-d^2) - bc(1+d^2) \\ \tau &\equiv \psi_a \psi_d - b^2 c^2 \lambda_0^2 \\ \eta_{x1} &\equiv \lambda_1 \psi_d / \tau \\ \eta_{x2} &\equiv b^2 \lambda_0 \lambda_1 / \tau \\ \eta_{y1} &\equiv c^2 \lambda_0 \lambda_1 / \tau \\ \eta_{y2} &\equiv \lambda_1 \psi_a / \tau \end{aligned} \tag{43}$$

we may solve for the correlation coefficient ρ between x_n and y_n to obtain:

$$\begin{bmatrix} Var(x_n) \\ Var(y_n) \end{bmatrix} = \begin{bmatrix} \eta_{x1} & \eta_{x2} \\ \eta_{y1} & \eta_{y2} \end{bmatrix} \begin{bmatrix} Q \\ R \end{bmatrix}. \tag{44}$$

$$\begin{aligned} C_{[xy]} &\equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \end{bmatrix} \right) = \begin{bmatrix} Var(x_n) & \xi \\ \xi & Var(y_n) \end{bmatrix} \\ \xi &\equiv E[x_n y_n] = \frac{b(d\psi_a + abc\lambda_0)R + c(a\psi_d + bcd\lambda_0)Q}{\psi_a \psi_d - b^2 c^2 \lambda_0^2} \\ \rho &= \frac{\xi}{\sqrt{Var(x_n)Var(y_n)}}. \end{aligned} \tag{45}$$

Now, as we did previously in example 1 above, compute the covariance $C^{(2)}$ of the variates obtained at two consecutive timestamps to yield:

$$C^{(2)} \equiv \mathrm{cov}\left(\begin{bmatrix} x_n \\ y_n \\ x_{n+1} \\ y_{n+1} \end{bmatrix}\right) = \begin{bmatrix} Var(x_n) & \xi & aVar(x_n)+b\xi & cVar(x_n)+d\xi \\ \xi & Var(y_n) & bVar(y_n)+a\xi & dVar(y_n)+c\xi \\ aVar(x_n)+b\xi & bVar(y_n)+a\xi & Var(x_n) & \xi \\ cVar(x_n)+d\xi & dVar(y_n)+c\xi & \xi & Var(y_n) \end{bmatrix}. \quad (46)$$

At this point the difficult part is done and the same calculations can be made as in example 1 to obtain $C^{(k)}$; k = 3,4, …, 10 and transfer entropies. For illustration purposes, we define the parameters of the system as shown below, yielding a symmetrically coupled pair of processes. To generate a family of curves for each transfer entropy we choose a fixed coupling term ε from a set of four values. We set Q = 1000 and vary R so that ρ varies from about 0 to 1. For each ρ value we compute the transfer entropies. The relevant system equations and parameters are:

$$\begin{aligned} x_{n+1} &= \left(\frac{1}{2}-\varepsilon\right)x_n + \varepsilon y_n + w_n : w_n \sim N(0,Q) \\ y_{n+1} &= \varepsilon x_n + \left(\frac{1}{2}-\varepsilon\right)y_n + v_n : w_n \sim N(0,R) \\ \varepsilon &\in \{0.1, 0.2, 0.3, 0.4\} \\ Q &= 1000. \end{aligned} \quad (47)$$

Hence, we make the following substitutions to compute $C^{(2)}$:

$$\begin{aligned} a &= \left(\frac{1}{2}-\varepsilon\right) \\ b &= \varepsilon \\ c &= \varepsilon \\ d &= \left(\frac{1}{2}-\varepsilon\right). \end{aligned} \quad (48)$$

For each parameter set {ε, Q, R} there is a maximum possible ρ, ρ_∞ obtained by taking the limit as R→ ∞ of the expression for ρ given above. Doing so, we obtain:

$$\rho_\infty = \frac{\phi_1\phi_2 + \phi_3}{\sqrt{\phi_1(\phi_1\mu_1 + 1)}} \quad (49)$$

where:

$$\begin{aligned} \phi_1 &\equiv \frac{2ab^2d + b^2\lambda_1}{(1-a^2-b^2\mu_1)\lambda_1 - 2ab(ac+bd\mu_1)} \\ \phi_2 &\equiv \frac{ac+bd\mu_1}{\lambda_1} \\ \phi_2 &\equiv \frac{bd}{\lambda_1} \end{aligned} \quad (50)$$

$$\begin{aligned} \lambda_1 &\equiv 1-ad-bc \\ \mu_1 &\equiv \frac{c^2\lambda_1 + 2ac^2d}{(1-d^2)\lambda_1 - 2bcd^2}. \end{aligned} \quad (51)$$

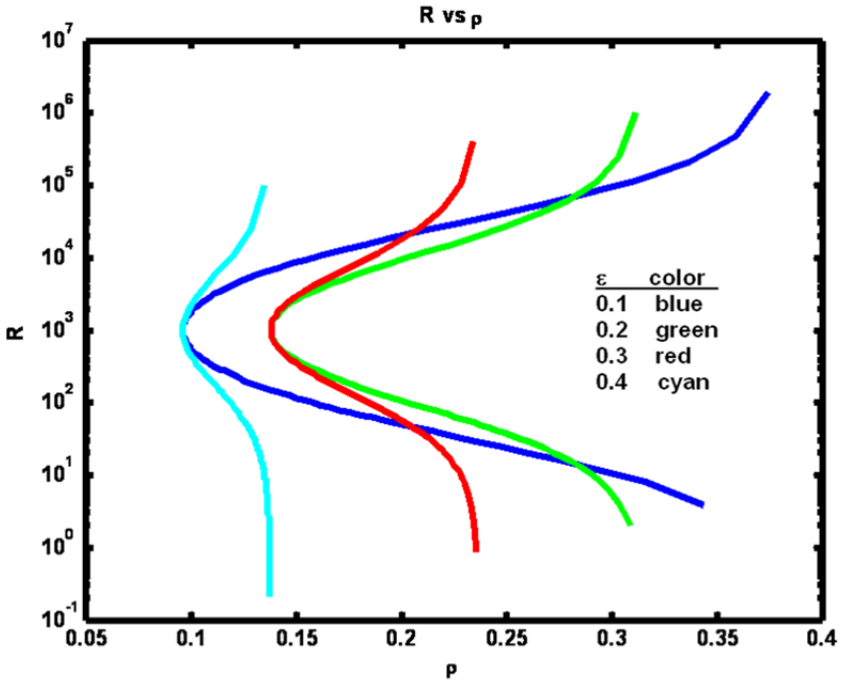
There is a minimum value of ρ also. The corresponding value for R, R_{min}, was found by means of the inbuilt Matlab program *fminbnd*. This program is designed to find the minimum of a function in this case ρ(*a*, *b*, *c* ,*d*, R, Q)) with respect to one parameter (in this case R) starting from an initial guess (here, R = 500). The program returns the minimum functional value (ρ_{min}) and the value of the parameter at which the minimum is achieved (R_{min}). After identifying R_{min} a set of R values were computed so that the corresponding set of ρ values spanned from ρ_{min} to the maximum ρ_∞ in fixed increments of Δρ (here equal to 0.002). This set of R values was generated using the iteration:

$$R_{new} = R_{old} + \Delta R = R_{old} + \left(\frac{\partial \rho}{\partial R}\right)^{-1}\Bigg|_{R=R_{old}} \Delta\rho \tag{52}$$

For the four selections of parameter ε we obtain the functional relationships shown in Figure 6.

From Figure 6 we see that for a fixed ε, increasing R increases (or decreases) ρ depending on whether R is less than (or greater than) Q (Q = 1000). Note that large increases in R > Q are required to marginally increase ρ when ρ nears its maximum value. The reason that the minimum ρ value occurs when Q equals R is because whenever they are unequal one of the processes dominates the other, leading to increased correlation. Also, note that if R << Q, then increasing ε will cause ρ to decrease since increasing the coupling will cause the variance of the y process $Var(y_n)$, a term appearing in the denominator of the expression for ρ, to increase. If Q << R, a similar result is obtained when ε is increased.

Figure 6. Example 2: Process noise variance R *versus* correlation coefficient ρ for a set of ε parameter values (see figure legend).



Transfer entropies in both directions are shown in Figure 7. Fixing ε, we note that as R is increased from a low value both ρ and $TE_{x->y}$ initially decrease while $TE_{y->x}$ increases. Then for further

increases of R, ρ reaches a minimum value then begins to increase, while $TE_{x\to y}$ continues to decrease and $TE_{y\to x}$ continues to increase.

Figure 7. Example 2: Transfer entropy values *versus* correlation ρ for a set of ε parameter values (see figure legend). Arrows indicate direction of increasing R values.

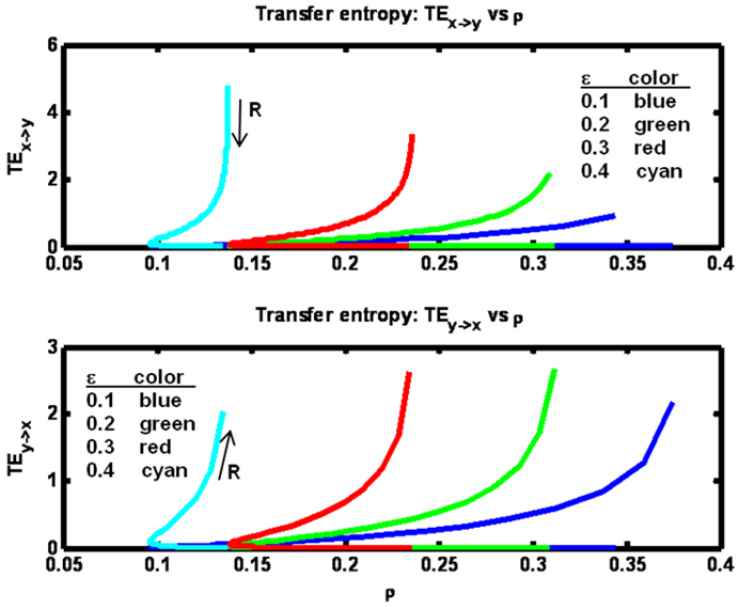
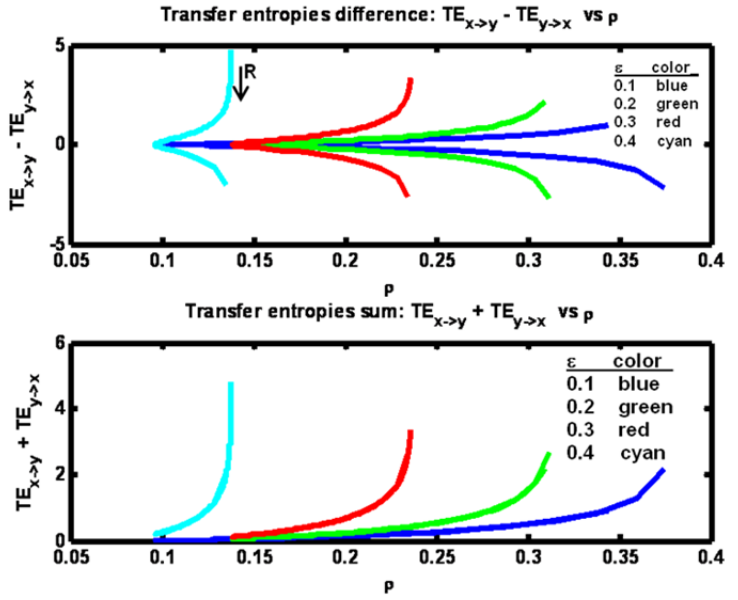


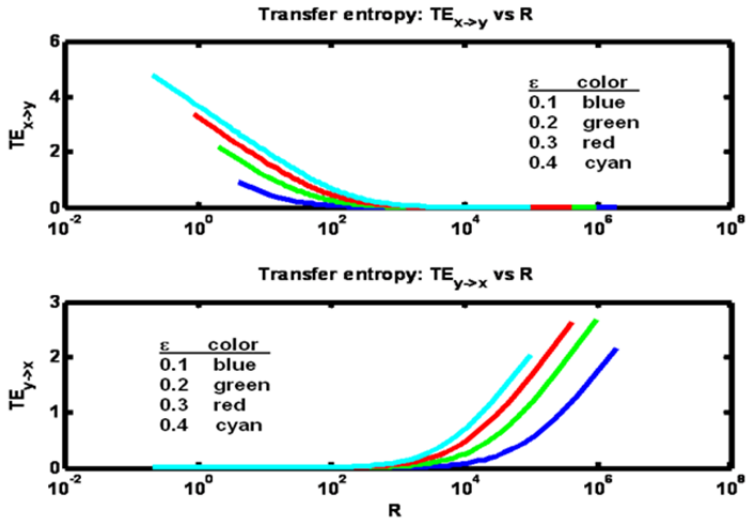
Figure 8. Example 2: Transfer entropies difference ($TE_{x->y} - TE_{y->x}$) and sum ($TE_{x->y} + TE_{y->x}$) *versus* correlation ρ for a set of ε parameter values (see figure legend). Arrow indicates direction of increasing R values.



By plotting the difference $TE_{x\to y} - TE_{y\to x}$ in Figure 8 we see the symmetry that arises as R increases from a low value to a high value. What is happening is that when R is low, the X process dominates the Y process so that $TE_{x\to y} > TE_{y\to x}$. As R increases, the two entropies equilibrate. Then, as R rises above Q, the Y process dominates giving $TE_{x\to y} < TE_{y\to x}$. The sum of the transfer entropies

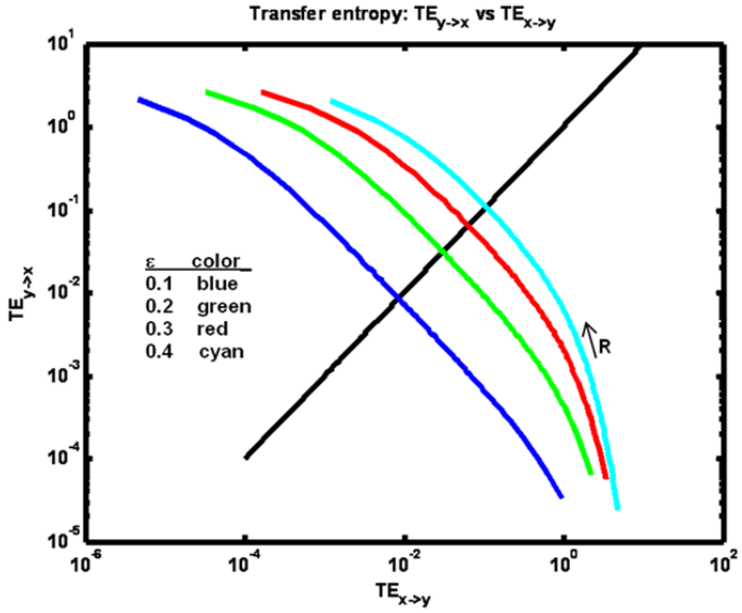
shown in Figure 8 reveal that the total information transfer is minimal at the minimum value of ρ and increases monotonically with ρ. The minimum value for ρ in this example occurs when the process noise variances Q and R are equal (matched). Figure 9 shows the changes in the transfer entropy values explicitly as a function of R. Clearly, when R is small (as compared to Q = 1000), $TE_{x\to y} > TE_{y\to x}$. Also it is clear that at every fixed value of R, both transfer entropies are higher at the larger values for the coupling term ε.

Figure 9. *Example 2:* Transfer entropies $TE_{x\to y}$ and $TE_{y\to x}$ *versus* process noise variance R for a set of ε parameter values (see figure legend).



Another informative view is obtained by plotting one transfer entropy value *versus* the other as shown in Figure 10.

Figure 10. Example 2: Transfer entropy $TE_{x->y}$ plotted *versus* $TE_{y->x}$ for a set of ε parameter values (see figure legend). The black diagonal line indicates locations where equality obtains. Arrow indicates direction of increasing R values.



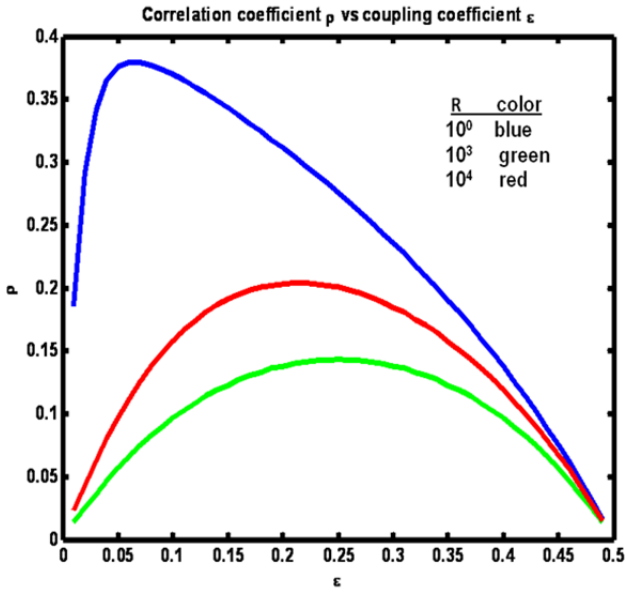
Here it is evident how $TE_{y\to x}$ increases from a value less than $TE_{x\to y}$ to a value greater than $TE_{x\to y}$ as R increases. Note that for higher coupling values ε this relative increase is more abrupt.

Finally, we consider the sensitivity of the transfer entropies to the coupling term ε. We reprise example system 2 where now ε is varied in the interval (0, ½) and three values of R (somewhat arbitrarily selected to provide visually appealing figures to follow) are considered:

$$\begin{aligned} x_{n+1} &= \left(\frac{1}{2}-\varepsilon_x\right)x_n + \varepsilon_x y_n + w_n : w_n \sim N(0,Q) \\ y_{n+1} &= \varepsilon_y x_n + \left(\frac{1}{2}-\varepsilon_y\right)y_n + v_n : w_n \sim N(0,R) \\ R &\in \left\{10^0, 10^3, 10^4\right\} \\ Q &= 10^3. \end{aligned} \tag{53}$$

Figure 11 shows the relationship between ρ and ε, where $\varepsilon_x = \varepsilon_y = \varepsilon$ for the three R values. Note that for the case R = Q the relationship is symmetric around ε = ¼. As R departs from equality more correlation between x_n and y_n is obtained.

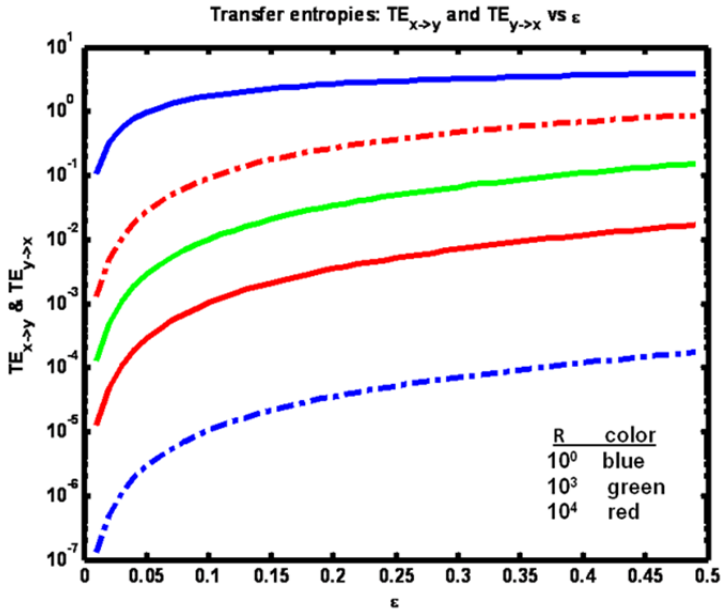
Figure 11. Example 2: Correlation coefficient ρ vs coupling coefficient ε for a set of R values (see figure legend).



The reason for this increase is that when the noise driving one process is greater in amplitude than the amplitude of the noise driving the other process, the first process becomes dominant over the other. This domination increases as the disparity between the process noise variances increases (R *versus* Q). Note also that as the disparity increases, the maximum correlation occurs at increasingly lower values of the coupling term ε. As the disparity increases at fixed ε = ¼ the correlation coefficient ρ increases. However, the variance in the denominator of ρ can be made smaller and thus ρ larger, if the variance of either of the two processes can be reduced. This can be accomplished by reducing ε.

The sensitivities of the transfer entropies to changes in coupling term ε are shown in Figure 12. Consistent with intuition, all entropies increase with increasing ε. Also, when R < Q (blue trace) we have $TE_{x->y} > TE_{y->x}$ and the reverse for R > Q. (red). For R = Q, $TE_{x->y} = TE_{y->x}$ (green).

Figure 12. Example 2: Transfer entropies $TE_{x\to y}$ (solid lines) vs $TE_{y\to x}$ (dashed lines) vs coupling coefficient ε for a set of R values (see figure legend).



Finally, it is interesting to note that whenever we define three cases by fixing Q and varying the setting for R (one of R_1, R_2 and R_3 for each case) such that $R_1 < Q$, $R_2 = Q$ and $R_3 = Q^2/R_1$ (so that $R_{i+1} = QR_i/R_1$ for i = 1 and i = 2) we then obtain the symmetric relationships $TE_{x->y}(R_1) = TE_{y->x}(R_3)$ and $TE_{x->y}(R_3) = TE_{y->x}(R_1)$ for all ε in the interval (1, ½). For these cases we also obtain $\rho(R_1) = \rho(R_3)$ on the same ε interval.

6. Conclusions

It has been shown how to compute transfer entropy values for Gaussian autoregressive processes for multiple timetags. The approach is based on the iterative computation of covariance matrices. Two examples were investigated: (1) a first-order filtered noise process whose state is measured with additive noise, and (2) two first-order symmetrically coupled processes each of which is driven by independent process noise. We found that, for the first example, increasing the first-order AR coefficient at a fixed correlation coefficient, transfer entropy increased since the entropy of the measured process was itself increased.

For the second example, it was discovered that the relationships between the coupling and correlation coefficients and the transfer entropies is more complicated. The minimum correlation coefficient occurs when the process noise variances match. It was seen that matching of these variances results in minimum information flow, expressed as the sum of both transfer entropies. Without a match, the transfer entropy is larger in the direction away from the process having the larger

process noise. Fixing the process noise variances, transfer entropies in both directions increase with coupling strength ε.

Finally, it is worth noting that the method for computing covariance matrices for a variable number of timetags as presented here facilitates the calculation of many other information-theoretic quantities of interest. To this purpose, the authors have computed such quantities as crypticity [13] and normalized transfer entropy using the reported approach.

References

1. Schreiber, T. Measuring information transfer. *Phys. Rev. Lett.* **2000**, *85*, 461–464.
2. Barnett, L.; Barrett, A.B.; Seth, A.K. Granger causality and transfer entropy are equivalent for Gaussian variables. *Phys. Rev. Lett.* **2009**, *103*, 238701.
3. Ay, N.; Polani, D. Information Flows in Causal Networks. *Adv. Complex Syst.* **2008**, *11*, 17–41.
4. Lizier, J.T.; Prokopenko, M. Differentiating information transfer and causal effect. *Eur. Phys. J. B* **2010**, *73*, 605-615.
5. Chicharro, D.; Ledberg, A. When two become one: the limits of causality analysis of brain dynamics. *PLoS One* **2012**, *7*, e32466.
6. Hahs, D.W.; Pethel, S.D. Distinguishing anticipation from causality: anticipatory bias in the estimation of information flow. *Phys. Rev. Lett.* **2011**, *107*, 128701.
7. Gourevitch, B.; Eggermont, J.J. Evaluating information transfer between auditory cortical neurons. *J. Neurophysiol.* **2007**, *97*, 2533–2543.
8. Kaiser, A.; Schreiber, T. Information transfer in continuous processes. *Physica D* **2002**, *166*, 43–62.
9. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley Series in Telecommunications, Wiley: New York, NY, USA, 1991.
10. Kotz, S.; Balakrishnan, N.; Johnson, N.L. *Continuous Multivariate Distributions, Models and Applications*; 2nd ed.; John Wiley and Sons, Inc.: New York, NY, USA, 2000; Volume 1.
11. Lizier, J.T.; Prokopenko, M.; Zomaya, A.Y. Local information transfer as a spatiotemporal filter for complex systems. *Phys. Rev. E* **2008**, *77*, 026110.
12. Williams, P.L.; Beer, R.D. Nonnegative decomposition of multivariate information. **2010**, arXiv:1004:2515.
13. Crutchfield, J.P; Ellison, C.J.; Mahoney, J.R. Time's barbed arrow: irreversibility, crypticity, and stored information. *Phys. Rev. Lett.* **2009**, *103*, 094101.