PHYSICAL REVIEW E **82**, 015201(R) (2010)

Optimized synchronization of chaotic and hyperchaotic systems

Paul H. Bryant*

BioCircuits Institute (formerly Institute for Nonlinear Science), University of California, San Diego, La Jolla, California 92093, USA

(Received 19 February 2010; published 12 July 2010)

A method of synchronization is presented which, unlike existing methods, can, for generic dynamical systems, force all conditional Lyapunov exponents to go to $-\infty$. It also has improved noise immunity compared to existing methods, and unlike most of them it can synchronize hyperchaotic systems with almost any single coupling variable from the drive system. Results are presented for the Rossler hyperchaos system and the Lorenz system.

DOI: 10.1103/PhysRevE.82.015201 PACS number(s): 05.45.Xt

This Rapid Communication presents a method for the synchronization [1,2] of chaotic and hyperchaotic systems. This is a distinctly different approach from previous methods, including those intended for hyperchaos (see, e.g., Ref. [3]). It is capable of synchronizing almost all systems with almost any single coupling variable. The Rossler hyperchaos equations, for example, can be synchronized using any one of the four variables for coupling. In the presence of noise, it dramatically reduces the synchronization error compared to the standard continuous-feedback (or diffusive) method [2], in effect performing two functions at once: synchronization and noise reduction. In the absence of noise, it can force all of the conditional Lyapunov exponents (CLEs) [2] to go to $-\infty$. In practice this means that a (small) error in the state of the response system can be completely corrected in a single step, something that has previously been demonstrated only in special cases (though proven in theory for generic systems [4]). The properties of the method can be related to time-delay embedding [5] and also to shadowing [6].

We consider two identical systems with unidirectional coupling between the drive (or master) system and the response (or slave) system. The method introduces corrections to all of the response variables at specific correction times. The corrections may be of finite amplitude, in which case the response variables will be discontinuous at the correction times. Between correction times, the response system evolves on its own, completely unaffected by the drive system. These corrections are designed to minimize the mean-square error in matching the coupling signal over a finite time interval, T_{opt}. This interval typically starts at the correction time and extends a specified amount into the future, making this a type of "initial value" or "shooting" method. Such methods have been used for parameter estimation (PE) using a finite data set [7] but have not previously been used for synchronization. Some other PE methods [8,9] use standard synchronization terms in conjunction with optimization, but the optimization process itself is not used as a mechanism for synchronization as it is here. The need for future time data means that the drive system must be running slightly ahead of the response system. (If need be this latency of the response system can be eliminated by a modification of the method [10] but at a cost of degraded noise immunity.)

This process is repeated over successive intervals, and the calculated corrections go to zero on approach to exact synchronization.

The finite time interval of data can be considered to be the continuous-time limit of a time-delay embedding and thus contains information about the full state of the system. For a generic system, a perfect fit over the entire interval is only possible when the full states of the drive and response systems are exactly identical. A previous paper that used embedding [4] looked at directly inverting a time-delay embedding of the coupling data. This is a very difficult task in general, which is avoided here by the error minimization approach. Another paper [11] develops a time-delay method called "extended observers," which is applicable to some iterated maps but not continuous time systems. Two other papers [12,13], use "derivative" embedding [5] to generate a coupling strength vector for continuous feedback that is not constant but depends on the current state of the response system and obtain some interesting results.

In the presence of noise, the synchronization error is significantly smaller than can be achieved by standard methods. The error can, in some cases, be further improved by increasing T_{opt}. The method is, in effect, being used as a means of noise reduction by "shadowing," i.e., as a process for seeking a deterministic orbit within noisy data. Such methods have been known for some time [6], but they do not appear to have been used for synchronization before. The noise reduction properties of the method could be of interest in the fields of chaotic communication [14] and parameter estimation.

The variables of the drive and response systems are represented as $\mathbf{x}=(x_1,x_2,\ldots,x_d)$, and $\mathbf{y}=(y_1,y_2,\ldots,y_d)$, where d is the dimension of the system. These are assumed to be governed by identical sets of first-order ordinary differential equations (ODEs) $d\mathbf{x}/dt=\mathbf{F}(\mathbf{x})$ and $d\mathbf{y}/dt=\mathbf{F}(\mathbf{y})$ (modified as required by the coupling scheme). We will assume that x_α is the variable that will be used for coupling, where α is a particular index in the range 1 to d. For the current method, corrections to the response system variables are made periodically at discrete times $t_n=t_0+nT_{\text{cor}}$, where T_{cor} is the correction time period and t_0 is the starting time. The corrections can be made "between" integration steps, making the systems identical to the numerical integration. In contrast, the standard continuous-feedback method requires the addition of the term $u(x_\alpha-y_\alpha)$ to the equation for dy_α/dt, where u is the coupling strength. Note that T_{cor} is typically much larger than numerical integration time step, T_{step}, and is often

*pbryant@ucsd.edu

a multiple thereof. The corrections at t_n optimize the fit to the coupling signal over a finite time interval which typically starts at time t_n and ends at time $t_n+T_{\rm opt}$. In general, $T_{\rm opt}$ must be long enough to unfold the full (time-delay) state of the system, yet short enough that errors are not magnified excessively across it. It should not be much larger than the inverse of the primary Lyapunov exponent. The ratio $T_{\rm opt}/T_{\rm cor}$ is roughly proportional to the computation rate required to maintain synchronization. If greater than one, the successive optimization intervals will overlap one another. Setting the ratio equal to one is a good starting point. The best value will generally have to be found empirically. Uncorrected response variables are identified with a superscript 0, i.e., $y_i^0(t)$. We use ξ_i to represent the corrections to be made at time t_n so that $y_i(t_n)=y_i^0(t_n)+\xi_i$. Corrections are applied to all of the response variables, not just y_α, and they will typically all be different. The analysis is easily generalized, e.g., to the case where more that one coupling variable is used, etc. For each correction time t_n, the goal is to find a correction that minimizes C, the square of the distance between x_α and y_α integrated over the optimization time interval, i.e.,

$$C=\int_{t_n}^{t_n+T_{\rm opt}}[y_\alpha(t)-x_\alpha(t)]^2dt. \tag{1}$$

One approach is to use a standard minimization algorithm [15] to find the optimal $\boldsymbol{\xi}$. However, if one is already close to synchronization, there is a preferable linear method which leads to an exact solution. This method can be used to maintain synchronization and, for the cases tried, usually achieves synchronization even when started far from it. We look for a minimum by taking the derivative of C with respect to each correction component ξ_i and setting the results to zero:

$$\int_{t_n}^{t_n+T_{\rm opt}}\frac{\partial y_\alpha(t)}{\partial \xi_i}[y_\alpha(t)-x_\alpha(t)]dt=0. \tag{2}$$

Assuming $\boldsymbol{\xi}$ is small we expand $y_\alpha(t)$ as

$$y_\alpha(t)=y_\alpha^0(t)+\sum_{j=1}^{d}\frac{\partial y_\alpha(t)}{\partial \xi_j}\xi_j. \tag{3}$$

Substituting this back into Eq. (2) we obtain

$$\sum_{j=1}^{d}A_{ij}\xi_j=b_i, \tag{4}$$

where

$$A_{ij}=\int_{t_n}^{t_n+T_{\rm opt}}\frac{\partial y_\alpha(t)}{\partial \xi_i}\frac{\partial y_\alpha(t)}{\partial \xi_j}dt \tag{5}$$

and

$$b_i=\int_{t_n}^{t_n+T_{\rm opt}}\frac{\partial y_\alpha(t)}{\partial \xi_i}[x_\alpha(t)-y_\alpha^0(t)]dt. \tag{6}$$

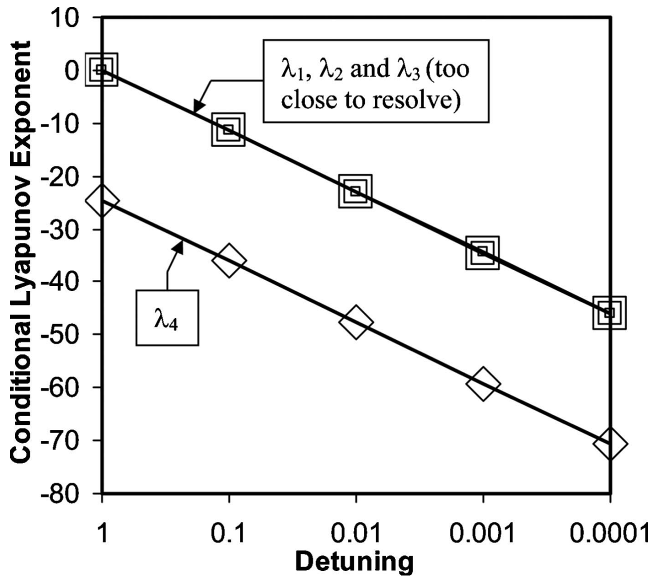

FIG. 1. Numerically calculated CLEs as a function of detuning ϵ, which shows them going to $-\infty$ as ϵ goes to zero (the normal mode of operation). Results shown are for the Rossler hyperchaos system, coupled through x_1, and using $T_{\rm cor}=0.2$ and $T_{\rm opt}=0.4$. $T_{\rm step}$ is small (0.002). At the left end of the graph, the detuning is 1.0 so that there is no coupling, and thus the CLEs are the same as the ordinary Lyapunov exponents (approximately 0.111, 0.021, 0, and -25.0). Synchronization is successful as soon as all CLEs are negative, i.e., for $0\leq\epsilon<0.978$.

The partial derivatives above can be evaluated by central finite differencing [16] or by integration of the differential Jacobian [17] and the results then used to evaluate $\mathbf{A}$ and $\mathbf{b}$. We can then solve for the correction vector $\boldsymbol{\xi}$ by inverting the matrix $\mathbf{A}$:

$$\xi_j=\sum_{j=1}^{d}(\mathbf{A}^{-1})_{ij}b_j. \tag{7}$$

In the absence of noise and the limit of small synchronization error the solution is exact, i.e., the error will be completely eliminated in a single correction step. Note that rather than an integral over the optimization interval, the method can also be formulated as a sum over at least d time points within that interval for which coupling data is available, i.e., all of the integrals above would be replaced by the corresponding summations.

To demonstrate the effect of the method on the CLEs, a detuning parameter ϵ is introduced and the calculated corrections $\boldsymbol{\xi}$ are multiplied by $(1-\epsilon)$ so that $\epsilon=0$ is normal operation, and $\epsilon=1$ is complete decoupling. Results [18] are shown in Fig. 1 for the Rossler hyperchaos equations:

$$\frac{d}{dt}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=\begin{pmatrix}-(x_2+x_3)\\x_1+p_1x_2+x_4\\p_2+x_1x_3\\p_3x_4-p_4x_3\end{pmatrix}, \tag{8}$$

where the "standard" values are used for the parameters, i.e., $p_1=0.25$, $p_2=3$, $p_3=0.05$, and $p_4=0.5$, and the equations are

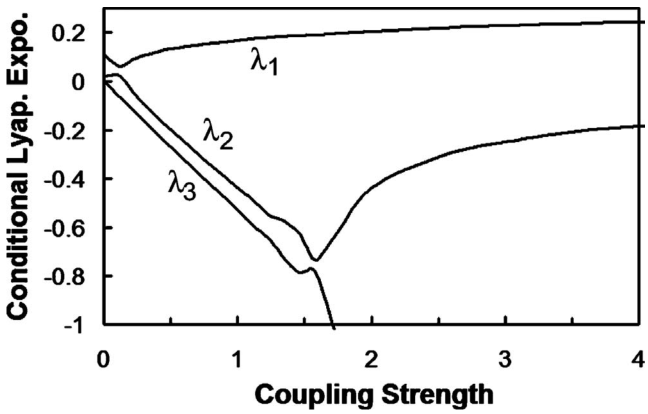

FIG. 2. Numerically calculated CLEs with standard continuous-feedback coupling for Rossler hyperchaos. Note that λ_4 is off scale near −25. Synchronization fails because λ_1 remains positive. Compare with Fig. 1 and also note the difference in vertical scale.

coupled to the response equations through the first variable. The figure shows all of the CLE's going to $-\infty$ as ϵ goes to zero. Surprisingly, the method achieves all negative CLEs and synchronization until the corrections are reduced to only a few percent of their normal values. For comparison, Fig. 2 shows results using standard continuous-feedback coupling. Synchronization cannot occur since there is always at least one positive CLE.

Noise in the coupling signal will cause problems when the matrix **A** is nearly singular. One method to deal with this problem is to use singular value decomposition (SVD) [15] to express **A** as the product of an orthogonal matrix **U**, a diagonal matrix **W**, and the transpose of another orthogonal matrix **V**, i.e., $\mathbf{A}=\mathbf{U}\cdot\mathbf{W}\cdot\mathbf{V}^T$. Since **W** is diagonal it can be expressed as $W_{ij}=\delta_{ij}w_j$, where δ_{ij} is the Kronecker delta and w_j are the singular values of **A**. The inverse of **W** is simply $(\mathbf{W}^{-1})_{ij}=\delta_{ij}/w_j$. Since the inverse of an orthogonal matrix is its transpose, we obtain $\mathbf{A}^{-1}=\mathbf{V}\cdot\mathbf{W}^{-1}\cdot\mathbf{U}^T$. The reason for using SVD is to be able to limit the singularity of **A** which otherwise in the presence of noise could result in exceedingly large errors in the correction values. The method involves setting a threshold or lower limit s for the acceptable values of w_j. For the results in this Rapid Communication, we define $v_j(s)$ as follows: if $w_j \geq s$ then $v_j(s)=1/w_j$ and if $w_j < s$ then $v_j(s)=1/s$. Other definitions are possible [19]. We then use $v_j(s)$ in $\mathbf{W}_{\mathrm{lsi}}$, a limited singularity inverse of **W** defined as $(\mathbf{W}_{\mathrm{lsi}})_{ij}(s)=\delta_{ij}v_j(s)$. This is used to define a limited singularity inverse of **A** to be used in Eq. (7):

$$\mathbf{A}_{\mathrm{lsi}}(s) = \mathbf{V}\cdot\mathbf{W}_{\mathrm{lsi}}(s)\cdot\mathbf{U}^T. \tag{9}$$

The value of s which minimizes the synchronization error is often found to be relatively large. It appears that the SVD process is, in effect, identifying directions that are associated approximately with the Lyapunov direction vectors [20]. As s is increased from zero, the corrections associated with the most negative exponents are limited first (and these are the least important). This suggests an alternate approach in which these directions are obtained directly and used as a basis for the correction vector. Another interesting result is that the optimal value of s does not tend to zero with noise amplitude.

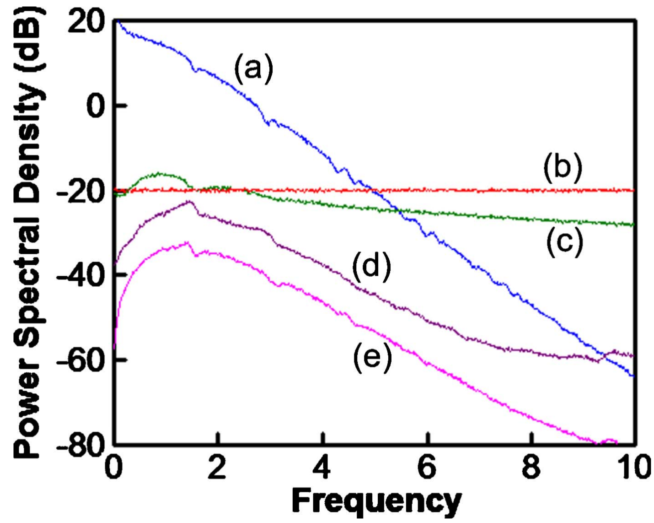

FIG. 3. (Color online) Calculated PSD from the Lorenz system. (a) The first variable x_1 of the drive system, which is the coupling signal. (b) The added noise, showing that it is extremely flat (or white) and has the intended value of 0.01 or −20 dB. (c) The synchronization error using the standard continuous-feedback method with the optimal coupling strength (see text). (d) The synchronization error using the method of this Rapid Communication with $T_{\mathrm{cor}}=0.25$, $T_{\mathrm{opt}}=0.5$, and $s=5.0$. (e) The same as (d) except T_{opt} increased to 4.0. For all results, T_{step} is small (0.005).

To study noise sensitivity, independent and identically distributed (iid) Gaussian deviates [15] were added to the coupling signal. This behaves like white noise that cuts off above the Nyquist frequency, i.e., above $1/(2T_{\mathrm{step}})$. For a desired noise power spectral density P_N, the rms amplitude A_N is set to $A_N=\sqrt{P_N/(2T_{\mathrm{step}})}$. Here we used the Lorenz equations:

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \sigma(x_2-x_1) \\ x_1(\rho-x_3)-x_2 \\ x_1x_2-\beta x_3 \end{pmatrix}, \tag{10}$$

where the standard values are used for the parameters, i.e., $\sigma=10$, $\rho=28$, and $\beta=8/3$. Figure 3 shows results for the power spectral density (PSD) of the synchronization error, y_1-x_1, for the optimized synchronization method of this Rapid Communication compared to that of the standard continuous-feedback method. Also shown are the PSD of x_1 and of the added noise. The results are given in dB, i.e., as $10\log_{10}(\mathrm{PSD})$. Coupling strength for the continuous-feedback case (c) was adjusted to minimize the mean-square synchronization error (mse), which for the first variable occurs for a coupling strength $u=27.5$. The mse corresponding to cases (c), (d), and (e) of the figure are 0.080, 0.0070, and 0.000 84, respectively. These values can also be obtained by integrating the PSD over all frequencies. The noncoupling variables follow a similar pattern of decreasing mse: 0.36, 0.017, and 0.0020 for y_2 and 0.34, 0.027, and 0.0029 for y_3.

The results presented have shown that the optimized synchronization method can easily synchronize systems that were previously found difficult or impossible and can simultaneously reduce synchronization error in the presence of noise, perhaps 20 dB or more, compared to the standard synchronization methods. Note that the calculation is easily parallelized, making it easier to run in real time. One direction for further research may be an application to parameter estimation which can be achieved by a simple modification of the method [23].

The author gratefully acknowledges helpful discussions with L. Kocarev, N. Rulkov, L. Pecora, and H. Abarbanel.

[1] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **64**, 821 (1990).

[2] L. M. Pecora, T. L. Carroll, G. A. Johnson, D. J. Mar, and J. F. Heagy, Chaos **7**, 520 (1997).

[3] J. H. Peng, E. J. Ding, M. Ding, and W. Yang, Phys. Rev. Lett. **76**, 904 (1996).

[4] T. Stojanovski, U. Parlitz, L. Kocarev, and R. Harris, Phys. Lett. A **233**, 355 (1997).

[5] F. Takens, Lect. Notes Math. **898**, 366 (1981).

[6] J. D. Farmer and J. J. Sidorowich, Physica D **47**, 373 (1991).

[7] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Phys. Rev. A **45**, 5524 (1992).

[8] D. R. Creveling, P. E. Gill, and H. D. I. Abarbanel, Phys. Lett. A **372**, 2640 (2008).

[9] J. C. Quinn, P. H. Bryant, D. R. Creveling, S. R. Klein, and H. D. I. Abarbanel, Phys. Rev. E **80**, 016201 (2009).

[10] A separate nondelayed copy of the response system can be run in parallel with the calculational copy. When a new adjustment to the calculational copy is made, it is then run faster than real time to catch up with and then update the nondelayed response system.

[11] H. J. C. Huijberts, T. Lilge, and N. Nijmeijer, Int. J. Bifurcation Chaos Appl. Sci. Eng. **11**, 1997 (2001).

[12] L. Kocarev, U. Parlitz, and B. Hu, Chaos, Solitons Fractals **9**, 1359 (1998).

[13] X. F. Wang and Z. Q. Wang, IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **45**, 1101 (1998).

[14] K. M. Cuomo and A. V. Oppenheim, Phys. Rev. Lett. **71**, 65 (1993).

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, England, 2007).

[16] Introduce a small offset to one of the variables y_j at time t_n and integrate forward in time to $t_n + T_{\text{opt}}$ using an ODE integration method such as the fourth-order Runge-Kutta method [15], producing an offset version of $\mathbf{y}(t)$. Repeat for the negative of that offset. The difference between the two results for $\mathbf{y}(t)$ divided by twice the offset yields the jth column in the Jacobian matrix $\mathbf{J}(t)$ which includes $\partial y_\alpha(t)/\partial \xi_j$. This must be repeated for each j. An offset of 0.0001 was used for the Lorenz results and 0.000 01 for the Rossler hyperchaos results.

[17] The Jacobian map $\mathbf{J}(t)$ evolves according to $d\mathbf{J}/dt = \mathbf{D}(\mathbf{y}) \cdot \mathbf{J}$, where $D_{ij} = \partial F_i / \partial y_j$ is the differential Jacobian. Initialize $\mathbf{J}$ to the identity matrix and integrate it simultaneously with $\mathbf{y}$.

[18] The CLEs were calculated by the QR decomposition method [21]. Software implementing this method is available [22]. It requires a Jacobian map of the response system to be determined across regular time intervals (we use T_{cor}). For the current method, this must include the effect of the corrections.

[19] One alternate method is to zero the inverse of those singular values that are below threshold, i.e., define $v_j(s)$ as follows: if $w_j \geq s$ then $v_j(s) = 1/w_j$ and if $w_j < s$ then $v_j(s) = 0$.

[20] R. Brown, P. Bryant, and H. D. I. Abarbanel, Phys. Rev. A **43**, 2787 (1991), see discussion starting on p. 2796.

[21] J.-P. Eckmann and D. Ruelle, Rev. Mod. Phys. **57**, 617 (1985), see discussion starting on p. 650.

[22] Lyapunov exponent software is available from Paul Bryant (email: pbryant@ucsd.edu).

[23] One can treat the parameters like they are variables that do not change in time, i.e., the right-hand side of the corresponding ODEs are set to zero.