Adaptive approximation method for joint parameter estimation and identical synchronization of chaotic systems

Inés P. Mariño[1] and Joaquín Míguez[2]

[1]*Nonlinear Dynamics and Chaos Group, Departamento de Matemáticas y Física Aplicadas y Ciencias de la Naturaleza, Universidad Rey Juan Carlos, C/ Tulipán s/n, 28933 Móstoles, Madrid, Spain*

[2]*Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain*

(Received 27 May 2005; revised manuscript received 19 August 2005; published 17 November 2005)

We introduce a numerical approximation method for estimating an unknown parameter of a (primary) chaotic system which is partially observed through a scalar time series. Specifically, we show that the recursive minimization of a suitably designed cost function that involves the dynamic state of a fully observed (secondary) system and the observed time series can lead to the identical synchronization of the two systems and the accurate estimation of the unknown parameter. The salient feature of the proposed technique is that the only external input to the secondary system is the unknown parameter which needs to be adjusted. We present numerical examples for the Lorenz system which show how our algorithm can be considerably faster than some previously proposed methods.

DOI: 10.1103/PhysRevE.72.057202 PACS number(s): 05.45.Xt, 05.45.Gg, 05.45.Pq, 47.52.+j

An important issue in time series analysis of a nonlinear system is the estimation of the parameters in the system model using scalar measurements from the system. Provided the functional form of the model is accurate, these estimates can subsequently be used to track the dynamic system state. This problem can be tackled in different ways, e.g., using multiple shooting methods [1,2] or some standard statistical procedures [3–7]. However, these methods are usually offline (the complete record of observations is iteratively processed to obtain a sequence of convergent solutions) and involve the solution of high-dimensional minimization problems, since not only the unknown parameters but also the initial values of the trajectory segments between the sampling times need to be estimated [1,8].

Alternatively, the new techniques for synchronization of coupled chaotic systems have been turned into promising parameter estimation methods by some authors [8–15]. This approach is appealing because the only unknowns are the parameters to be estimated, hence only low-dimensional optimization problems need to be tackled.

The vast majority of synchronization-based methods for parameter estimation that are found in the literature depend on the explicit coupling (either by the Pecora and Carroll technique [16] or by the so-called linear feedback coupling [17]) of the chaotic system that outputs the time series and a model system with the same functional form. In this way, it is ensured that the state of the model system does not arbitrarily deviate from the state of the original system, as long as the difference in the corresponding parameter sets is not too large [12].

A more challenging problem arises when there is not such a explicit coupling of the two systems, because the model system can only be controlled by dynamically adjusting the unknown parameters (and arbitrary divergence can be expected if control is not properly and carefully exercised). Only very recently, it has been shown that if the time series consists of the full system state and only one parameter in the model is to be adjusted, identical synchronization and parameter estimation can be achieved under certain conditions [18].

In this paper, we focus on the latter scenario (no explicit coupling) and propose an online numerical approximation algorithm that allows to achieve both accurate estimation of a single parameter and identical synchronization of the systems when only a scalar time series (instead of the full state) is observed. The proposed method aims at the minimization, with respect to the adjustable parameter in the model, of a suitably designed cost function that involves the scalar observations and the dynamics of the model. Simple analytical approximations allow to systematically derive online estimation algorithms for any unknown parameter.

To be specific, let

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, p) \tag{1}$$

represent the *primary* system with state variables $\mathbf{x} \in \mathbb{R}^n$, and an unknown parameter $p \in \mathbb{R}$. If the functional form of Eq. (1) is known, we can build the *secondary* system as

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, q) \tag{2}$$

with $\mathbf{g}$ identical to $\mathbf{f}$, where $\mathbf{y} \in \mathbb{R}^n$ is the time-varying vector that contains the state variables and $q \in \mathbb{R}$ is a free (adjustable) parameter. The system in Eq. (2) is fully observed, and we assume the ability to periodically change the value of q. Note that there is no explicit coupling between the dynamic variables of the primary and secondary systems. Finally, let $h(\mathbf{x}): \mathbb{R}^n \to \mathbb{R}$ be the scalar time series we observe from the primary system, which consists of a known transformation of a subset of the dynamic variables in Eq. (1).

Our aim is to devise an algorithm to adaptively adjust q until the secondary system variables and the parameter itself converge to their counterparts in the primary system, i.e., both $\mathbf{y}\rightarrow\mathbf{x}$ and $q\rightarrow p$. In this way, synchronization between both systems is achieved and the unknown parameter p is estimated.

We propose a parameter adaptation procedure that is based on the optimization of a cost function J that involves the observed times series and the secondary system dynamics. In general, we consider functions of the form

$$J(t)=\int_0^t \|e(\tau)\|^a \lambda^{(t-\tau)/T} d\tau, \tag{3}$$

where $e(\tau)=h(\mathbf{x})-h(\mathbf{y})$ is an error signal, $\|\cdot\|$ denotes the vector norm, $a\geqslant 1, T$ is the adaptation period (i.e., we assume that q can be updated every T time units), and $0<\lambda<1$ is a *forgetting factor* used to guarantee that recent observations are emphasized over older ones. Obviously, $e(\tau)$ depends on q through the argument of $h(\mathbf{y})$ and, therefore, the adjustable parameter can be updated at time $t=nT$ as

$$q_n=\arg\min_q\{J(nT)\}, \quad n\in\mathbb{N}. \tag{4}$$

The feasibility of the method relies on a choice of J that is tractable to the extent of allowing the derivation of a practical and effective minimization algorithm. For example, we could choose the straightforward error signal $e(\tau)=x_i-y_i$ (where x_i and y_i are the ith components of $\mathbf{x}$ and $\mathbf{y}$) but, unfortunately, it is difficult to minimize $e^2(\tau)$ (furthermore its integral) with respect to q. As an alternative, we have found that particularly appealing results can be obtained when the error signal consists of the difference between derivatives of the dynamic variables. As an instance, let us consider the observation $h(\mathbf{x})=\dot{x}_i$, and the resulting error signal $e(\tau)=\dot{x}_i-\dot{y}_i$. In this case, we can build the quadratic cost function

$$J(t)=\int_0^t [\dot{x}_i(\tau)-\dot{y}_i(\tau)]^2 \lambda^{(t-\tau)/T} d\tau. \tag{5}$$

The use of the difference of derivatives of the state variables of the primary and secondary systems (instead of the straightforward difference between the state variables [19]) is advantageous because the analytical expression of $\dot{\mathbf{y}}$ is known from model (2) and, as a consequence, the signal $e(\tau)=\dot{x}_i(\tau)-\dot{y}_i(\tau)$ is an explicit function of the parameters in the ith dynamic equation. If the adjustable parameter appears in equations other than the ith, higher-order derivatives can be used to have q explicitly appear in $e(\tau)$.

The main limitation of our approach is that minimizing an error signal that involves derivatives alone may not necessarily lead to identical synchronization in general. Further research is needed to determine what type of synchronization (if any) can be expected from the minimization of a given cost function.

Let us illustrate the application of the proposed method by way of an example that involves the Lorenz system. Thus, we assume the primary system

$$\dot{x}_1=-\sigma_1(x_1-x_2), \quad \dot{x}_2=R_1x_1-x_2-x_1x_3,$$

$$\dot{x}_3=-b_1x_3+x_1x_2, \tag{6}$$

where $\mathbf{x}=(x_1,x_2,x_3)\in\mathbb{R}^3$ forms the state space. The secondary system is

$$\dot{y}_1=-\sigma_2(y_1-y_2), \quad \dot{y}_2=R_2y_1-y_2-y_1y_3,$$

$$\dot{y}_3=-b_2y_3+y_1y_2, \tag{7}$$

where $\mathbf{y}=(y_1,y_2,y_3)\in\mathbb{R}^3$ contains the state variables. We first consider the problem of estimating the parameter σ_1 assuming both b_1 and R_1 are *a priori* known. Therefore, $b_2=b_1$ and $R_2=R_1$ are set from the start and σ_2 needs to be adjusted. We assume that the observed time series from the primary system is $\dot{x}_1$ and, hence, the error signal is $e(\tau)=\dot{x}_1-\dot{y}_1$. The nth update of the adjustable parameter $q=\sigma_2$ is carried out by solving the equation

$$\frac{dJ}{d\sigma_2}=-\int_0^{nT} 2(\dot{x}_1-\dot{y}_1)\frac{d\dot{y}_1}{d\sigma_2}\lambda^{n-\tau/T}d\tau=0 \tag{8}$$

for σ_2. Clearly, there is a difficulty in the computation of the derivative $d\dot{y}_1/d\sigma_2$ because both y_1 and y_2 have an implicit dependence on σ_2 [see Eq. (7)]. However, if we consider only the explicit derivatives, the problem becomes very simple and Eq. (8) reduces to

$$\int_0^{nT}[\dot{x}_1+\sigma_2(y_1-y_2)](y_1-y_2)\lambda^{n-\tau/T}d\tau=0, \tag{9}$$

which yields the nth parameter estimate

$$\sigma_{2,n}=-\frac{\int_0^{nT}\dot{x}_1(y_1-y_2)\lambda^{n-\tau/T}d\tau}{\int_0^{nT}(y_1-y_2)^2\lambda^{n-\tau/T}d\tau}. \tag{10}$$

We have neglected some implicit derivatives in obtaining the updating rule (10) which, therefore, is only an approximate solution of Eq. (8). However, it has the virtues of being simple to compute (it is given in closed form), effective as a joint synchronization and parameter estimation algorithm (as will be subsequently shown by our numerical simulations) and general (the same approach can be applied for any other parameter in the system). Moreover, some algebraic manipulation allows to rewrite Eq. (10) as an adaptive algorithm where the nth parameter estimate $\sigma_{2,n}$ is computed as $\sigma_{2,n-1}$ plus an additive term. Specifically,

$$\sigma_{2,n}=\sigma_{2,n-1}+\frac{A_n-\sigma_{2,n-1}B_n}{\lambda C_{n-1}+B_n} \tag{11}$$

where

$$A_n=\int_{(n-1)T}^{nT}\dot{x}_1(y_2-y_1)\lambda^{n-\tau/T}d\tau,$$

$$B_n=\int_{(n-1)T}^{nT}(y_1-y_2)^2\lambda^{n-\tau/T}d\tau,$$

$$C_{n-1} = \int_0^{(n-1)T} (y_1 - y_2)^2 \lambda^{(n-1)-\tau/T} d\tau. \qquad (12)$$

The latter formulas are particularly suitable for the online application of the proposed technique.

We have carried out computer simulations to numerically demonstrate the performance of the algorithm given by Eq. (11) in terms of parameter estimation accuracy. The primary Lorenz system is assigned the standard parameter values, $(\sigma_1, R_1, b_1)=(10, 28, \frac{8}{3})$; hence we set $b_2=b_1$, $R_2=R_1$ (fixed), and take an initial value $\sigma_{2,0}=\sigma_1-5$ for the algorithm of Eq. (11) to start running. The equations of both the primary and the secondary systems have been numerically integrated using a fourth-order Runge-Kutta method with integration step $I=10^{-3}$ time units (t.u.). We have adopted the same value for the parameter adaptation period, $T=10^{-3}$ t.u., and the forgetting factor has been set as $\lambda=0.94$. For comparison, we have also applied to the same problem the algorithms proposed by Maybhate and Amritkar [11] (subsequently denoted as MA) and Sakaguchi [12]. The former is an adaptive method where the adjustable parameter is handled as a dynamic variable with an associated partial differential designed to ensure that it converges to the desired value (much in the spirit of the original paper by Parlitz [8]). The latter is an offline method based on a Monte Carlo optimization procedure and its computational complexity is much higher than that of the adaptive techniques (at each iteration, a complete simulation of the secondary system has to be run with the length of the available time series). Both MA's and Sakaguchi's algorithms involve a linear feedback coupling of the chaotic systems, which is not necessary for the technique proposed in this paper. To implement the MA procedure, we use Ref. [11], Eq. (18) with coupling and stiffness parameters $\epsilon=20$ and $\delta=1$, respectively. The Sakaguchi method is applied with coupling strength $D=9$, perturbation variance var$(r)=0.01$, and 1000 steps (iterations). The length of the observed time series which is fed to the three algorithms is $T_s=40$ t.u., and they all share the same initial condition for the secondary system $(y_1, y_2, y_3)=(10.2, 14.2, 20.9)$.

The results of the numerical simulation are shown in Figs. 1(a) and 1(b). Figure 1(a) depicts the temporal evolution of dynamic variables x_1 (from the primary system) and y_1 corresponding to the secondary system, as obtained for each one of the three algorithms. It is seen that identical synchronization is quickly achieved in all cases, after approximately 4 t.u. Note that synchronization is *aided* by the linear feedback coupling in the case of MA and Sakaguchi but not for our technique. Figure 1(b) shows the time evolution of the normalized absolute error between σ_1 and σ_2, defined as $\varepsilon_\sigma=|(\sigma_2-\sigma_1)/\sigma_1|$, for the MA and proposed techniques, as well as the value of ϵ for the final output of the Sakaguchi procedure. It is clearly observed that the convergence of the proposed algorithm is much faster (at the expense of a less "smooth" trajectory). The final estimates yielded by MA and Sakaguchi could be improved by using longer observation series. For $T_s=40$ t.u. and the considered setup, the Sakaguchi algorithm converges long before reaching the 1000th step; hence there is no significant improvement in running more iterations.

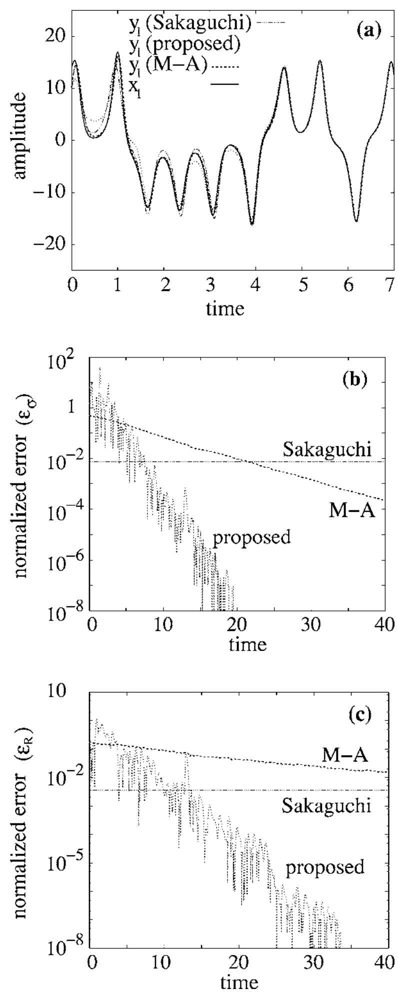

FIG. 1. Estimation of σ_1 and R_1 using the proposed technique, the Sakaguchi Monte Carlo algorithm, and the method of Maybhate and Amritkar. (a) Temporal evolution of the scalar observed time series of the primary Lorenz system (solid lines) and the corresponding variable of the secondary system when σ_1 is estimated. (b) Normalized absolute error attained by the three estimation algorithms with respect to σ_1. Plot (c) is the same as (b), but R_1 is estimated instead of σ_1.

Next, we consider the problem of estimating R_1 when $(\sigma_1, b_1)=(10, \frac{8}{3})$ are known and the time series $\ddot{x}_1$ is available (again, $\ddot{x}_1$ can be easily obtained from x_1). In this case, we set $(\sigma_2, b_2)=(\sigma_1, b_1)$ and define the error signal $e(\tau)=\ddot{x}_1-\ddot{y}_1$ in order to adjust $q=R_2$. The second derivative is necessary because R_2 does not appear in the equation of $\dot{x}_1$. The nth update of the unknown parameter is carried out by solving

$$\frac{dJ}{dR_2} = -2\int_0^{nT} (\ddot{x}_1 - \ddot{y}_1)\frac{d\ddot{y}_1}{dR_2}\lambda^{n-\tau/T} d\tau = 0, \qquad (13)$$

which can be handled in the same manner as Eq. (8). In particular, by taking only explicit derivatives, solving for R_2 and adequately reordering terms, we obtain the adaptive update rule

$$R_{2,n} = R_{2,n-1} + \frac{A_n - R_{2,n-1}B_n}{\lambda C_{n-1} + B_n} \qquad (14)$$

where

$$A_n = \int_{(n-1)T}^{nT} [\ddot{x}_1 + \sigma_2(\dot{y}_1 + y_2 + y_1 y_3)]\sigma_2 y_1 \lambda^{n-\tau/T} d\tau,$$

$$B_n = \int_{(n-1)T}^{nT} (\sigma_2 y_1)^2 \lambda^{n-\tau/T} d\tau,$$

$$C_{n-1} = \int_{0}^{(n-1)T} (\sigma_2 y_1)^2 \lambda^{(n-1)-\tau/T} d\tau. \qquad (15)$$

The simulation setup we have considered for assessing the performance of the algorithm given by Eq. (14) is almost identical to the one described for Figs. 1(a) and 1(b). Specifically, we have kept $\lambda=0.94$ and $T=I=10^{-3}$ for the proposed estimation algorithm, and we have used the same observed time series (of length $T_s=40$ t.u.) and initial values for **y**. The MA algorithm is implemented with Ref. [11], Eq. (21) [$\epsilon=20$ and $\delta=1$] and the Sakaguchi method has been run through 1000 steps with var$(r)=0.4$ [20]. The initial parameter value for all methods was set as $R_{2,0}=R_1+5$.

Figure 1(c) shows the normalized absolute error, defined as $\varepsilon_R=|(R_2-R_1)/R_1|$, that the three algorithms attain in the estimation of R_1 and, as was the case with σ_1, the proposed technique turns out to achieve considerably faster convergence. Variable y_1 also converges to x_1 rapidly for the three methods (not shown) in a way very similar to Fig. 1(a).

To conclude the simulation study, we have considered the case in which the observed time series is contaminated with Gaussian noise, i.e., the observations have the form $\ddot{x}_1+\eta$, where η is a zero-mean white Gaussian random process with power spectral density $P_\eta=1$. Otherwise, the simulation setup is the same as in Fig. 1(c). The results are depicted in Fig. 2, where it is seen that convergence is attained, although there is a floor of approximately 10^{-2} for the normalized error ε_R.

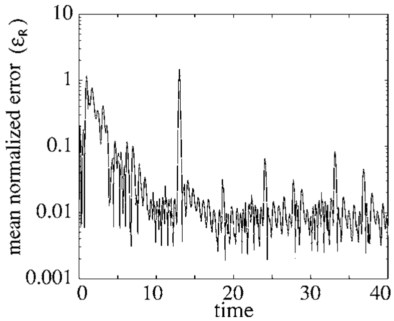

FIG. 2. Estimation of R_1 and identical synchronization of the Lorenz systems by adaptively adjusting R_2 when the observed signal is $\ddot{x}_1+\eta$, where η is a white and Gaussian random process.

In summary, we have introduced an online method for estimating an unknown parameter of a chaotic system with known functional form using a time series of scalar observations. The salient feature of our technique is that the adjustable parameter is the only control input of the secondary system. The application of the method is systematic and it does not require the explicit coupling (neither linear feedback nor a Pecora-Carrol type of coupling) of the primary and the secondary chaotic systems. Compared to some existing methods based on synchronization (and aided by linear feedback coupling), the proposed algorithm attains much faster convergence.

This work has been supported by the Spanish Ministry of Science and Technology (Project No. BFM2003-03081) and Universidad Rey Juan Carlos (Project No. PPR-2004-03).

[1] E. Baake, M. Baake, H. G. Bock and K. M. Briggs, Phys. Rev. A **45**, 5524 (1992).
[2] A. Ghosh, V. R. Kumar, and B. D. Kulkarni, Phys. Rev. E **64**, 056222 (2001).
[3] V. Petridis, E. Paterakis, and A. Kehagias, IEEE Trans. Neural Netw. **9**, 862 (1998).
[4] J. Timmer, Chaos, Solitons Fractals **11**, 2571 (2000).
[5] H. Singer, J. Comput. Graph. Stat. **11**, 972 (2002).
[6] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss, Phys. Rev. E **66**, 016210 (2002).
[7] V. F. Pisarenko and D. Sornette, Phys. Rev. E **69**, 036122 (2004).
[8] U. Parlitz, L. Junge, and L. Kocarev, Phys. Rev. E **54**, 6253 (1996).
[9] U. Parlitz, Phys. Rev. Lett. **76**, 1232 (1996).
[10] C. Zhou and C-H Lai, Phys. Rev. E **59**, 6629 (1999).
[11] A. Maybhate and R. E. Amritkar, Phys. Rev. E **59**, 284 (1999).
[12] H. Sakaguchi, Phys. Rev. E **65**, 027201 (2002).
[13] R. Konnur, Phys. Rev. E **67**, 027204 (2003).
[14] D. Huang, Phys. Rev. E **69**, 067201 (2004).
[15] C. Tao, Y. Zhang, G. Du, and J. J. Jiang, Phys. Rev. E **69**, 036204 (2004).
[16] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **64**, 821 (1990).
[17] K. Pyragas, Phys. Lett. A **170**, 421 (1992).
[18] U. S. Freitas, E. E. N. Macau and C. Grebogi, Phys. Rev. E **71**, 047203 (2005).
[19] We note that if x_i is the actual observable, it is straightforward to obtain $\dot{x}_i$ with either analog or digital circuitry.
[20] We have found that for lesser values of the variance, and $T_s=40$, the algorithm gets easily stuck in local minima of the cost function $\int_0^{T_s}|x_1(\tau)-y_1(\tau)|^2 d\tau$.