Monte Carlo method for adaptively estimating the unknown parameters and the dynamic state of chaotic systems

Inés P. Mariño,[1] Joaquín Míguez,[2] and Riccardo Meucci[3]
[1]*Departamento de Física, Nonlinear Dynamics and Chaos Group, Universidad Rey Juan Carlos, 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*
[3]*Istituto Nazionale di Ottica Applicata, 50125 Firenze, Italy*
(Received 6 November 2008; revised manuscript received 28 February 2009; published 15 May 2009)

We propose a Monte Carlo methodology for the joint estimation of unobserved dynamic variables and unknown static parameters in chaotic systems. The technique is sequential, i.e., it updates the variable and parameter estimates recursively as new observations become available, and, hence, suitable for online implementation. We demonstrate the validity of the method by way of two examples. In the first one, we tackle the estimation of all the dynamic variables and one unknown parameter of a five-dimensional nonlinear model using a time series of scalar observations experimentally collected from a chaotic CO_2 laser. In the second example, we address the estimation of the two dynamic variables and the phase parameter of a numerical model commonly employed to represent the dynamics of optoelectronic feedback loops designed for chaotic communications over fiber-optic links.

DOI: 10.1103/PhysRevE.79.056218 PACS number(s): 05.45.Gg, 02.70.Uu, 05.45.Pq, 05.45.Tp

I. INTRODUCTION

Many complex physical systems with a rich dynamical behavior can be modeled by sets of nonlinear differential equations. A compact and common notation for this class of models is

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \boldsymbol{\theta}], \tag{1}$$

where $t \in \mathbb{R}^+$ is time, $\mathbf{x}(t) \in \mathbb{R}^{d_x}$ comprises of the time-varying system state, and $\mathbf{f}$ is a nonlinear function, parametrized by the fixed vector $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$. In many practical problems, it is desired to estimate both the static parameters in $\boldsymbol{\theta}$ and the dynamic variables in $\mathbf{x}(t)$ from the observation of a time series. Such series often consists of a subset of the state variables, possibly transformed and corrupted by noise.

The estimation problem becomes particularly attractive, yet harder, when the system exhibits chaotic dynamics. Many models described in the literature have been designed to provide a good representation of the dynamical features of interest in some physical system but cannot actually be used to estimate its evolution for a particular series of observations, in the sense that a parameter vector $\boldsymbol{\theta}$ and an initial condition $\mathbf{x}(0)$ cannot be found that follow closely a particular observed time series. This is due to the very definition of chaos, since arbitrarily small perturbations in $\mathbf{x}(0)$ or $\boldsymbol{\theta}$ quickly lead to radically different realizations of $\mathbf{x}(t)$.

Some approaches have been recently proposed for estimating either the model state, its fixed parameters, or both. Multiple shooting techniques [1,2] are offline (batch) methods to jointly estimate the evolution of $\mathbf{x}(t)$, in a finite-length time interval, and the value of $\boldsymbol{\theta}$ from a time series. They are computationally very demanding and cannot be adapted to the online estimation. The family of Kalman filtering methods can tackle the latter difficulty [3] but they imply approximations that result in estimation inaccuracy, even in much simpler problems [4]. Another approach is based on the synchronization properties of coupled chaotic systems. If two models—master and slave—are coupled in a way that ensures synchronization when they are identical then the parameters of the slave can be adjusted by minimizing the synchronization error [5–7]. This is a very appealing strategy because of the dimensionality reduction: only $\boldsymbol{\theta}$ has to be estimated and the estimation of $\mathbf{x}(t)$ is "automatic" from the synchronized slave model. Moreover, adaptive implementations are feasible. However, such methods are hard to apply in practice because they rely on the unrealistic assumption that the master and slave systems are identical [5,7]. Observational noise can also lead to great practical difficulties.

In this paper, we introduce a class of adaptive algorithms for the joint estimation of $\mathbf{x}(t)$ and $\boldsymbol{\theta}$ from an observed time series. Our approach is adapted from the sequential Monte Carlo optimization (SMCO) methodology in [8]. It requires the time discretization and randomization of the system model, in a way similar to [3], but it does not involve any approximations regarding the nonlinearity $\mathbf{f}(\cdot)$. Unlike in synchronization-oriented procedures, it is not necessary to explicitly plug the observations into the model to simulate any type of coupling. We provide a general description of the proposed methodology, in a form that makes it applicable to virtually any model of the form (1) and then consider two specific examples. In the first one, we derive a procedure to estimate the latent state variables and an unknown fixed parameter of the five-dimensional model in [9,10] using experimental data from a CO_2 laser in chaotic regime. This example enables us to demonstrate the ability to estimate the dynamic variables and the static parameters of a chaotic model online and using experimental data. For the second example, we address the estimation of the two dynamic variables and a fixed phase parameter of a numerical model that has often been employed to represent the nonlinear dynamics of optoelectronic delayed-feedback loops designed for chaotic communications over fiber-optic links [11].

The remaining of the paper is organized as follows. The proposed sequential estimation methodology is introduced in Sec. II. In Sec. III, we derive a specific algorithm for the estimation in a CO_2 laser in chaotic regime and show some results obtained with experimental data. In Sec. IV, we consider a delayed-feedback model and address a numerical study of the relationship between the proposed method performance and the maximum Lyapunov exponent (MLE) of the system. Finally, Sec. V is devoted to the conclusions and we provide the details of the method, as implemented for the two examples and including a brief discussion on computational complexity, in an Appendix.

II. SEQUENTIAL MONTE CARLO OPTIMIZATION

A. Discrete-time random model

We are concerned with nonlinear systems of the form of Eq. (1) and, more specifically, with those displaying a chaotic behavior, since their long-term unpredictability makes it very challenging to track their dynamic variables or estimate any unknown parameters. In order to obtain a more flexible model able to produce a larger set of state trajectories but still subject to the main dynamic features of the system of interest, we transform Eq. (1) into a discrete-time state-space random model. Discretization is easily carried out by Euler's method with time step T and it yields the difference equation

$$\mathbf{x}_n = \mathbf{x}_{n-1} + T\mathbf{f}(\mathbf{x}_{n-1}, \boldsymbol{\theta}), \tag{2}$$

where $\mathbf{x}_n \triangleq (x_{1,n}, \ldots, x_{d_x,n})$ is the d_xth dimensional state vector, with components $x_{i,n} \triangleq x_i(t=nT)$. Note that for sufficiently small T, this discrete-time approximation of Eq. (1) can be as accurate as we wish.

In order to encompass a larger set of possible dynamical behaviors, we introduce randomness in the system dynamics by adding a sequence of statistically independent perturbations denoted $\mathbf{v}_n = (v_{1,n}, \ldots, v_{d_x,n}) \in \mathbb{R}^{d_x}$, hence we arrive at the model

$$\mathbf{x}_n = \mathbf{x}_{n-1} + T\mathbf{f}(\mathbf{x}_{n-1}, \boldsymbol{\theta}) + \mathbf{v}_n. \tag{3}$$

Note that we are interested in chaotic systems, which exhibit a very high sensitivity to small perturbations. Therefore, if we intend to take advantage of Eq. (3) as a useful approximation of the behavior given by Eq. (1), the variances of the perturbations $v_{i,n}$ must be small enough.

The state $\mathbf{x}_n$ and any unknown parameters in $\boldsymbol{\theta}$ are estimated from the observations related to the dynamic variables. In particular, we assume a scalar observation consisting of the ith variable, $x_{i,n}$, $i \in \{1, \ldots, d_x\}$, perturbed by observational noise, i.e.,

$$y_{rN} = x_{i,rN} + m_{rN}, \quad r = 0, 1, \ldots, \tag{4}$$

where N is the observation period [we collect one observation for every N time steps of model (3)] and m_{rN} is a zero-mean noise term not necessarily Gaussian. Equation (4) is rather specific and somewhat restrictive. In particular, the estimation of the state and the unknown parameters becomes much simpler when observation vectors $\mathbf{y}_{rN} \in \mathbb{R}^{d_y}$, with $d_y > 1$, are available for processing, instead of mere scalars. Therefore, the latter problem is harder and more interesting both practically and theoretically. Moreover, we also assume that observations may not be available at each time step, hence the period $N \geq 1$. This is a usual constraint in many experimental setups (see, e.g., the example in Sec. III) due to practical measurement constraints. Finally, we note that the proposed methodology is also easily adapted to scenarios where the observations are known transformations—possibly nonlinear—of a number of state variables.

B. Proposed method

We can now address the problem of sequentially estimating the system state, $\mathbf{x}_n$, $n \in \mathbb{N}$, and the unknown parameters using the sequence of observations $y_{N:rN} = \{y_N, y_{2N}, \ldots, y_{rN \leq n}\}$. In the sequel, we use notation $\boldsymbol{\phi}$ to denote the unknown parameters to be estimated. Note that $\boldsymbol{\phi}$ is a vector that consists of a subset of the elements of $\boldsymbol{\theta}$.

Since Eqs. (3) and (4) together yield a random state-space dynamical model, the immediate thought is to apply either Kalman filtering techniques (including extended and unscented Kalman filters) or sequential Monte Carlo filtering (SMCF) methods. The former approach has been investigated in [3] for numerical models (not experimentally) but it is well known to be suboptimal. SMCF algorithms [12–14] have been recognized to be more effective and robust than Kalman-type filters for many problems [4,15], but there are several reasons why they are not suitable for the problem described in this paper. First, Eqs. (3) and (4) are artificial. They are not intended to represent accurately the dynamics of the observed time series and the underlying physical system. Hence, there is no reason to claim that SMCF should work better than any other, even *ad hoc*, technique. Second, the estimation of static parameters using SMCF algorithms is an open problem. Although some specific techniques exist [16], there is no general method that can be applied to an arbitrary state-space model with a guarantee of convergence. Third, because of the sensitivity of chaos to small perturbations of the initial conditions, a stable optimal filter may not exist for many models of the forms (3) and (4) (see [17] for a technical exposition of this problem for generic discrete-time nonlinear dynamical systems). The latter problem becomes particularly severe when unknown static parameters exist, as in our case.

We propose to avoid the limitations of conventional statistical filtering techniques by adapting the SMCO methodology proposed in [8] to our problem. SMCO methods are aimed at the minimization of a time-varying cost function by generating discrete sets of particles that evolve with time and converge—under some mild conditions—toward the successive minimizers of the cost function. Each particle is a random sample in the space of the unknowns to be estimated (in our case, the space of $\{\mathbf{x}_n, \boldsymbol{\phi}\}$) with its associated cost. Particles are propagated from one time instant to the next using a Markov-chain scheme that depends on the costs (low cost particles have a bigger chance to evolve, while high cost particles have a bigger chance to die out) and involves particle interaction.

In order to describe the proposed technique formally, let $\mathbf{x}'_{rN}$ and $\boldsymbol{\phi}'_r$ denote candidate values of the state variables and

the unknown static parameters, both obtained at time rN (note that the parameters $\boldsymbol{\phi}$ are fixed: the subscript in $\boldsymbol{\phi}'_r$ indicates only that the candidate value is computed at time rN). Let us define function $\mathbf{F}^j$ as the result of iterating Eq. (3) j times in the absence of noise ($\mathbf{v}_n=\mathbf{0}$ for all n). Specifically, if $\boldsymbol{\theta}'_r$ is the complete set of parameters that includes $\boldsymbol{\phi}'_r$, we define $\mathbf{F}^0(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r)\triangleq\mathbf{x}'_{rN}$ and recursively construct

$$\mathbf{F}^{j+1}(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r)\triangleq\mathbf{F}^j(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r)+T\mathbf{f}[\mathbf{F}^j(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r),\boldsymbol{\theta}'_{rN}]. \quad (5)$$

If the dynamical noise in Eq. (3) is zero mean with a symmetric probability density function then it is apparent that $\mathbf{F}^{jN}(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r)$ is the expected value of $\mathbf{x}_{(r+j)N}$ conditional on $\mathbf{x}_{rN}=\mathbf{x}'_{rN}$ and $\boldsymbol{\phi}=\boldsymbol{\phi}'_r$. Therefore, given the observations $y_{rN:(r+j)N}$, we can compute a prediction error

$$\mathcal{E}_r^j(\mathbf{x}'_{rN},\boldsymbol{\phi}'_r)\triangleq\sum_{l=0}^{j}|y_{(r+l)N}-[\mathbf{F}^{lN}(\mathbf{x}'_{rN},\boldsymbol{\theta}'_r)]_1|^2, \quad (6)$$

where $[\mathbf{a}]_i$ denotes the ith element of vector $\mathbf{a}$. If we adopt the minimization of the prediction error $\mathcal{E}_r^j$ as the criterion for the sequential estimation of the dynamic variables and the unknown parameters then we need to solve the sequence of optimization problems

$$(\mathbf{x}^o_{rN},\boldsymbol{\phi}^o_r)=\arg\min_{\tilde{\mathbf{x}},\tilde{\boldsymbol{\phi}}}\mathcal{E}_r^j(\tilde{\mathbf{x}},\tilde{\boldsymbol{\phi}}). \quad (7)$$

The SMCO algorithm sequentially generates sets $\Omega_r\triangleq\{\mathbf{x}^{(i)}_{rN},\boldsymbol{\phi}^{(i)}_r,\epsilon^{(i)}_r\}_{i=1}^M$ each of them containing M particles. The ith particle is the triple $(\mathbf{x}^{(i)}_{rN},\boldsymbol{\phi}^{(i)}_r,\epsilon^{(i)}_r)$, where $\mathbf{x}^{(i)}_{rN}$ and $\boldsymbol{\phi}^{(i)}_r$ are candidate values for the dynamic variables and the unknown parameters, respectively, and $\epsilon^{(i)}_r=\mathcal{E}_r^j(\mathbf{x}^{(i)}_{rN},\boldsymbol{\phi}^{(i)}_r)$ is the cost of the particle. Given Ω_r, the estimates of $\mathbf{x}_{rN}$ and $\boldsymbol{\phi}$ are the candidate values in the particle with the least cost, i.e., if we let $i_o=\arg\min_{i\in\{1,\ldots,M\}}\epsilon^{(i)}_r$, we calculate our estimates as $\hat{\mathbf{x}}_{rN}=\mathbf{x}^{(i_o)}_{rN}$ and $\hat{\boldsymbol{\phi}}_r=\boldsymbol{\phi}^{(i_o)}_r$. We can also predict the evolution of the state variables by simply propagating the latest estimate, i.e., $\hat{\mathbf{x}}_{rN+j}=\mathbf{F}^j(\hat{\mathbf{x}}_{rN},\hat{\boldsymbol{\phi}}_r)$.

To generate the sequence of particle sets, we need to choose a prior probability distribution at time $n=0$ for both the dynamic variables and the unknown parameters. We randomly draw the initial candidates from this prior and assign them equal costs (e.g., 0) to obtain the first set $\Omega_0=\{\mathbf{x}^{(i)}_0,\boldsymbol{\phi}^{(i)}_0,\epsilon^{(i)}_0=0\}_{i=1}^M$. Assume now that Ω_r is available. When the observations $y_{(r+1)N},\ldots,y_{(r+j+1)N}$ are collected, we build a probability distribution function in the joint space of $\mathbf{x}_{(r+1)N}$ and $\boldsymbol{\phi}$ conditional on Ω_r, denoted $\mu_{r+1}(\mathbf{x}_{(r+1)N},\boldsymbol{\phi}|\Omega_r)$, and draw M candidate values from it,

$$(\mathbf{x}^{(i)}_{(r+1)N},\boldsymbol{\phi}^{(i)}_{r+1})\sim\mu_{r+1}(\mathbf{x}_{(r+1)N},\boldsymbol{\phi}|\Omega_r),\quad i=1,\ldots,M. \quad (8)$$

We finally compute the costs (prediction errors) as $\epsilon^{(i)}_{r+1}=\mathcal{E}_{r+1}^j(\mathbf{x}^{(i)}_{(r+1)N},\boldsymbol{\phi}^{(i)}_{r+1})$ in order to obtain the set $\Omega_{r+1}=\{\mathbf{x}^{(i)}_{(r+1)N},\boldsymbol{\phi}^{(i)}_{r+1},\epsilon^{(i)}_{r+1}\}_{i=1}^M$.

If the distributions μ_r are adequately chosen, the convergence of the estimates toward the true minimizers of the prediction error can be assessed using simple induction procedures.

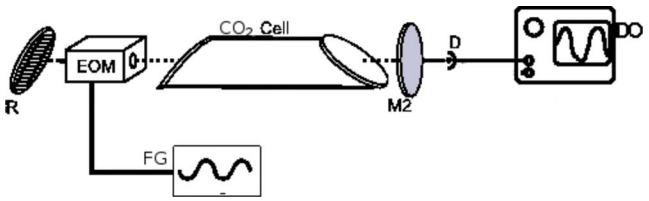

FIG. 1. (Color online) Experimental setup for a CO_2 laser with modulated losses. EOM: intracavity electro-optic modulator, R: reflecting grating, M2: partial reflecting mirror, D: fast infrared detector, FG: function generator, and DO: digital oscilloscope.

III. EXPERIMENTAL EXAMPLE: A CHAOTIC CO_2 LASER SYSTEM

A. Experimental setup

Figure 1 shows a diagram of the experimental setup. The laser cavity 1.45 m long is defined by a reflecting diffraction grating mounted in the autocollimation configuration and a partial reflecting outcoupler mirror (R and M2, respectively). The gain medium, a gas mixture containing CO_2, is pumped by an electric discharge. The discharge current value is set at 1.09 times the laser threshold (5.50 mA). An electro-optic modulator (EOM) is inserted in the laser cavity in order to control the cavity losses by a sinusoidal forcing provided by a function generator (FG). Such a signal has a frequency of 100 kHz, which is around the double of the relaxation frequency of the laser. By increasing the driving amplitude of the external periodic forcing signal, the system displays a transition to chaos through a sequence of subharmonic bifurcations. A further increase in this amplitude leads the system to an interior crisis characterized by a widening of the chaotic attractor (see Fig. 2 and its associated text in [9]). In the experiment, we set the system in the chaotic region just after the occurrence of the interior crisis.

B. Deterministic and random models

A deterministic model of the laser dynamics is given by the set of differential equations [9,18],

$$\begin{aligned}
\dot{x}_1&=-k_1(t)x_1+k_0x_1x_2,\\
\dot{x}_2&=-\gamma_1x_2-2k_0x_1x_2+gx_3+x_4+P,\\
\dot{x}_3&=-\gamma_1x_3+gx_2+x_5+P,\\
\dot{x}_4&=-\gamma_2x_4+zx_2+gx_5+zP,\\
\dot{x}_5&=-\gamma_2x_5+zx_3+gx_4+zP,
\end{aligned} \quad (9)$$

where $t\in\mathbb{R}^+$ denotes continuous time, $k_1(t)=k_0\{1+\alpha\sin^2[b_0+F(t)]\}$, $F(t)=A\sin(2\pi ft)$ is the external forcing signal, $\mathbf{x}(t)=(x_1,x_2,\ldots,x_5)\in\mathbb{R}^5$ is the system state (the time dependence $x_i=x_i(t)$ is omitted for conciseness), and $\boldsymbol{\theta}=(k_0,\alpha,b_0,\gamma_1,\gamma2,g,P,z)\in\mathbb{R}^8$ is the vector of fixed model parameters. The dynamic variable x_1 represents the laser output intensity, x_2 is the population inversion between the two resonant levels, and x_3, x_4, and x_5 account for molecular

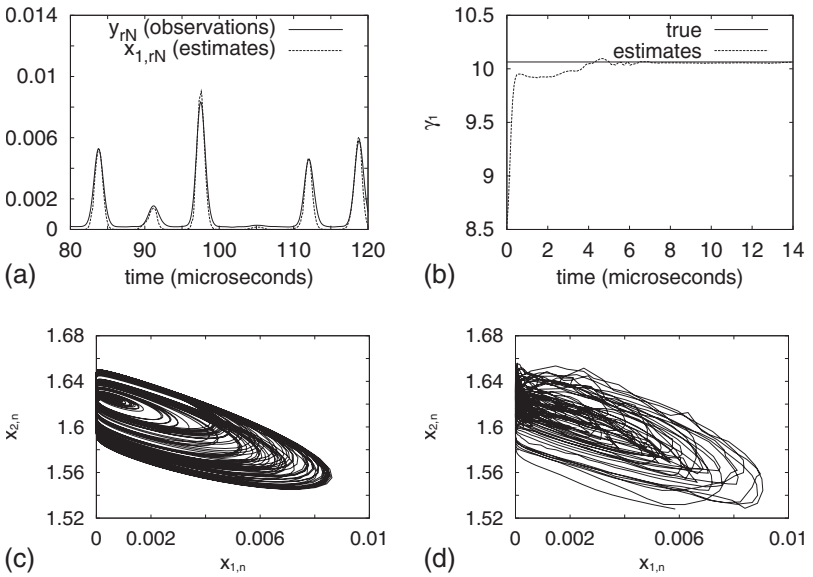

FIG. 2. (a) Experimentally observed laser intensity and the estimate of $x_{1,n}$. (b) Convergence of the estimated parameter $\hat{\gamma}_1$ toward the correct value $\gamma_1=10.043$. (c) Phase-space diagram obtained by integrating the model equations (in the absence of noise). (d) The same phase-space diagram obtained from the estimates of $x_{2,n}$ versus the estimates of $x_{1,n}$. The number of particles in the applied SMCO algorithm was $M=2000$.

exchanges between the two levels resonant with the radiation field and the other rotational levels of the same vibrational band. The static parameters include the unperturbed cavity loss parameter $k_0=29.55$, a coupling constant $g=0.05$, the population relaxation rates $\gamma_1=10.064\ 3$ and $\gamma_2=1.064\ 3$, the effective number of rotational levels $z=10$, the efficiency of the electro-optic modulator $\alpha=4$, and the "pump" rate $P=0.026\ 3$. The remaining parameters are related to the external periodic forcing: $f=1/7$ is the signal frequency, $b_0=0.403$ is the bias voltage, and $A=0.060\ 3$ is the amplitude of forcing. All the parameters in the model are adimensional and obtained from the experimental ones with an accuracy better than 5%. They lead to a dynamical behavior that shares many properties exhibited by the experiment (see, e.g., [9]), but the model does not yield a representation of the physical system accurate enough to estimate or predict the time evolution of the dynamic variables of interest in the actual experiment. In other words, it is not possible to choose $\mathbf{x}(0)$ and $\boldsymbol{\theta}$ to guarantee that the trajectory of x_1 generated by the model and the experimental laser intensity remain correlated (they diverge very quickly).

The five equalities in Eq. (9) yield, in an obvious way, a specific form of the vector-valued nonlinearity $\mathbf{f}[\mathbf{x}(t),\boldsymbol{\theta}]$ of Eq. (1). If we select a time step of $T=5\times10^{-3}$ time units and define the state vector $\mathbf{x}_n \triangleq (x_{1,n},x_{2,n},\ldots,x_{5,n}) \in \mathbb{R}^5$, with $x_{i,n} \triangleq x_i(t=nT)$, we can readily build the discrete-time approximation of Eq. (2). We additionally choose the dynamical noise process $\mathbf{v}_n=(v_{1,n},\ldots,v_{5,n}) \in \mathbb{R}^5$ to consist of a degenerate component, $v_{1,n}=0$ for all n with probability 1, and four independent and identically distributed Gaussian random variables $v_{i,n}$, $i=2,\ldots,5$, with zero mean and variance $\sigma_v^2=10^{-8}$

The only observable in our experimental setup is the laser output intensity, which corresponds to variable x_1. Hence, following Eq. (4), we model the available time series as

$$y_{rN}=x_{1,rN}+m_{rN}, \quad r \in \mathbb{N}, \tag{10}$$

where $N=14$ time steps and m_{rN} is the observational noise. We do not need to select any particular probability distribution for m_{rN}.

C. Experimental results

In order to illustrate the validity of the method, we have measured a sequence of experimental laser intensities and applied the sequential optimization algorithm to track the value of the dynamic variables $x_{1,n},\ldots,x_{5,n}$ and estimate the value of parameter γ_1. The details of the applied algorithm are given in the Appendix for reproducibility. In order to obtain the discrete-time model for Eq. (9), we have selected a step-size $T=0.005$ time units. In the experiment, the laser intensity is measured every 0.1 μs and we have verified that a good approximation of this period between consecutive observations is achieved in the discrete-time model by choosing $N=14$ in Eq. (10), i.e., if we let t_d denote the (real) time (in μs) per time unit in model (9), then $NTt_d \approx 0.1$ μs and we obtain $t_d \approx 0.1/NT=1.43$ μs. The amplitude of the experimental observations has been normalized to a maximum value of 0.09, in order to put them in the amplitude scale of the numerical model (9).

Figure 2(a) shows the normalized experimental intensity values and the values of the dynamic variable $x_{1,n}$ estimated using the proposed algorithm once convergence is achieved. It can be seen that the algorithm tracks the intensity peaks tightly. In the intervals between consecutive peaks, the experimental observations remain at a "floor" level which is due to the limited accuracy of the measurement instruments. The estimated signal, however, exhibits the typical dynamical behavior of model (9) and takes values which are positive but close to 0.

Figure 2(b) depicts the evolution of the parameter estimate. We recall that the value $\gamma_1=10.064\ 3$ (indicated by the straight horizontal line in the plot) yields the best approximation of the laser dynamics by the model. Given an initial estimate $\hat{\gamma}_1=8.45$ at time $n=0$, the proposed algorithm yields converged estimates after ≈ 11.4 μs equivalent to ≈ 1600 discrete-time steps and the processing of ≈ 115 experimental observations. The normalized mean-square error (MSE) of the parameter estimate after convergence is $(\gamma_1-\hat{\gamma}_1)^2/\gamma_1^2 \approx 10^{-6}$.

Finally, plots (c) and (d) in Fig. 2 show the phase-space diagram of variable $x_{2,n}$ versus $x_{1,n}$ for a series of noiseless

data obtained by integrating Eq. (9) [graph (c)] and the estimates of $x_{1,n}$ and $x_{2,n}$ computed by the proposed algorithm using the experimental observations [graph (d)]. It can be seen that diagram (d) is a noisy version of diagram (c), showing the ability of the proposed algorithm to follow the experimental time series using the randomized model (3). Let us remark that it is not possible to observe the physical magnitudes represented by the dynamic variables x_i, $i=2,\ldots,5$, directly in the experiment, so the proposed method provides means to "observe" them indirectly, by fitting them to the measured laser intensity.

IV. NUMERICAL EXAMPLE: A DELAYED-FEEDBACK SYSTEM

A. Deterministic and random models

We tackle now the estimation of the two dynamic variables and a fixed phase parameter of a numerical model that has been employed to represent the nonlinear dynamics of optoelectronic delayed-feedback loops designed for chaotic communications over fiber-optic links [11,19]. It is in general hard to forecast the time evolution of this class of systems because they can exhibit a high Lyapunov dimension [11]. Here, we focus on the problem of tracking the system state and recursively estimating static parameters. See [11] for a prediction method based on synchronization, applicable to experimental time series, but not aimed at the online parameter estimation.

When the low- and high-pass filters of the actual system in [11] are approximated by single-pole responses, the associated numerical model becomes

$$\tau_L \dot{x}_1 = -\left(1+\frac{\tau_L}{\tau_H}\right)x_1 - x_2 - \beta\cos^2[x_1(t-\tau_D)+\psi_0],$$

$$\tau_H \dot{x}_2 = x_1, \tag{11}$$

where t is continuous time, $x_1(t)$ is the system voltage output, $x_2(t)$ is the integrated output, the dimensionless parameter β determines the strength of the feedback in the loop, τ_D is the delay of the feedback signal, τ_L and τ_H are the time constants of the (single-pole approximations of the) low- and high-pass filters in the loop, and the phase ψ_0 is the bias point of a Mach-Zehnder modulator (see [11] for details). Note that we have skipped the time dependence of the dynamic variables x_1 and x_2, except to make explicit the delay of the feedback signal $x_1(t-\tau_D)$.

The discrete-time approximation of Eq. (11) is readily carried out if we choose a time step $T=\tau_D/\kappa$, where κ is some positive integer. In such case, and denoting $x_{i,n}=x_i(t=nT)$, $i\in\{1,2\}$, we obtain the discrete-time approximation of the system of interest

$$x_{1,n} = x_{2,n-1} - T\left[\left(1+\frac{\tau_L}{\tau_H}\right)x_{1,n-1} + x_{2,n-1} + \beta\cos^2(x_{1,n-\kappa-1}+\psi_0)\right]/\tau_L,$$

$$x_{2,n} = x_{2,n-1} + Tx_{1,n-1}/\tau_H. \tag{12}$$

Equation (12) can be easily assimilated to the generic model (3) if we introduce the $(\kappa+2)$-dimensional state vector $\mathbf{x}_n=(x_{1,n-\kappa},\ldots,x_{1,n},x_{2,n})$ and we let the dynamic noise term $\mathbf{v}_n=(v_{1,n},v_{2,n})$ consist of two independent and identically distributed Gaussian variables with zero mean and variance $\sigma_v^2=10^{-5}$. It is remarkable that the dimension of the state of the discrete-time model is actually dependent on the feedback delay.

In this type of system, the "natural" observable is the output voltage x_1, hence we assume the observations to have the form

$$y_{rN} = x_{1,rN} + m_{rN}, \quad r\in\mathbb{N}, \tag{13}$$

where N is the discrete observation period and m_{rN} is a noise term with arbitrary (but zero-mean) distribution.

B. Numerical results

We have applied the proposed SMCO algorithm to jointly track the state variables x_1, x_2 and estimate the phase parameter ψ_0 of system (11). We have assumed null initial conditions for x_1 and x_2, i.e., $x_1(t)=x_2(t)=0$ for all $t<\tau_D$. As for the model parameters, we have set the time constants as $\tau_L=\frac{10^{-2}}{2\pi}$ μs and $\tau_H=\frac{1}{2\pi}$ μs, which correspond to cut-off frequencies of 100 MHz and 1 MHz in the low-pass and high-pass filters, respectively. The feedback delay has been set to $\tau_D=22.5$ ns and the step size for the time discretization of the model has been selected as $T=0.1$ ns (as a result, the discrete feedback delay becomes $\kappa=225$ time steps and the dimension of the state vector $\mathbf{x}_n$ is $\kappa+2=227$). The true (assumed unknown) value of the phase parameter is $\psi_0=\frac{\pi}{9}$ rad and we have used different values of the feedback strength parameter β to illustrate the behavior of the estimation algorithm. The details of the latter can be found in the Appendix. All the simulations in this section have been carried out using $M=2000$ particles.

Figure 3 shows the results obtained when the SMCO method is applied to the considered system with $\beta=3$ and we collect one observation every $N=10$ time steps of the numerical model (i.e., one observation per ns). In particular, Fig. 3(a) shows the evolution of variable x_1 and the noisy observation y, in the upper axes, together with the estimated signal $\hat{x}_1$, in the lower axes, in the interval $1800\leq t\leq 2100$ ns. It is apparent that the signal is tracked very accurately despite the relatively low observation rate of $1/N=0.1$. This is also the case for the second dynamic variable x_2, as shown in Fig. 3(b). The true signal x_2 is represented in the upper axes, for $22.5\leq t\leq 2977.5$ ns, while its estimates are plot in the lower axes. We observe how this unobserved signal is also tracked accurately, with only small-magnitude zero-mean fluctuations due to the observational noise in y.

Figure 4 shows the normalized square error defined as $(\psi_0-\hat{\psi}_{0,r})^2/\psi_0^2$ attained by the SMCO algorithm in the same simulation of Fig. 3. We observe that the algorithm converges very quickly and yields accurate estimates of ψ_0 after a few time steps. Then, the error progressively reduces with time, although at a slower rate.

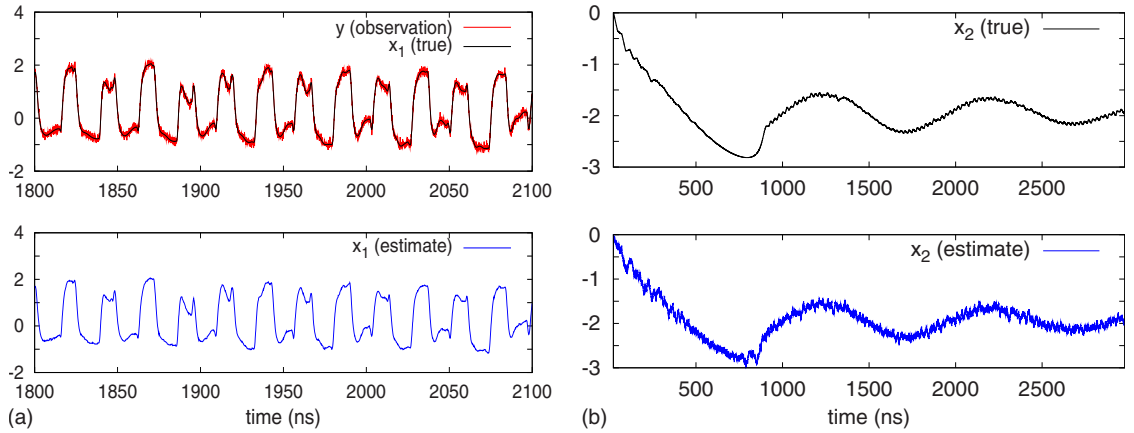

FIG. 3. (Color online) (a) The upper axes depict the evolution of variable x_1 and the associate observation $y=x_1+m$, where m is a Gaussian zero-mean random process, for $1800 \leq t \leq 2100$ ns. The variance of the Gaussian perturbations in y is chosen to ensure a signal-to-noise ratio of 20 dB and the discrete-time data rate is $1/N=0.1$ (one observation per ns). The estimate of x_1 computed recursively by the SMCO algorithm is shown in the lower axes. (b) The upper axes depict the evolution of the unobserved dynamic variable x_2 for $22.5 \leq t \leq 2977.5$ ns, while the lower plot shows its estimate.

Since the proposed estimation algorithm is based on the minimization of a prediction error, there is a relevant question of how much the predictability of the system may affect its performance. In particular, the ability to predict the behavior of a chaotic system decreases (and, therefore, so does the relevance of the prediction error as defined in this paper) as its MLE increases. Therefore, we have studied the performance of the SMCO method as we let the feedback strength grow from $\beta=1$ to $\beta=5$, which also leads to a significant increase in the MLE.

In particular, Fig. 5(a) shows the evolution of the MLE as β increases. It is seen that we move from a nonchaotic scenario when $\beta=0.5$ (for which the MLE ≈ 0) to a highly unpredictable system for $\beta=5$, when the MLE ≈ 0.077 (ns^{-1}). Indeed, if we apply the SMCO algorithm to systems with $\beta=1,2,\ldots,5$, the attained error increases clearly with the MLE. Fig. 5(b) shows the normalized MSE of the estimates $\hat{x}_1, \hat{x}_2, \hat{\psi}_0$ for the different feedback strengths. The normalized MSE puts together the error in the estimation of the three unknowns. Specifically, it is the value

$$\frac{(x_1-\hat{x}_1)^2+(x_2-\hat{x}_2)^2+(\psi_0-\hat{\psi}_0)^2}{x_1^2+x_2^2+\psi_0^2} \tag{14}$$

averaged over time and over several independent simulations (40 for this figure). We observe that this joint error is well

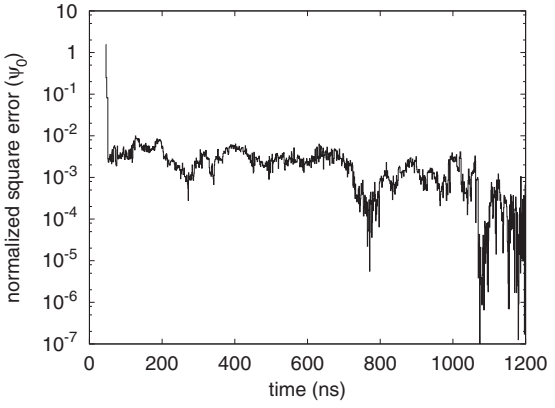

FIG. 4. Square error in the estimation of the phase parameter ψ_0 in the same simulation of Fig. 3. The feedback strength is $\beta=3$ and the observation rate is $1/N=0.1$.

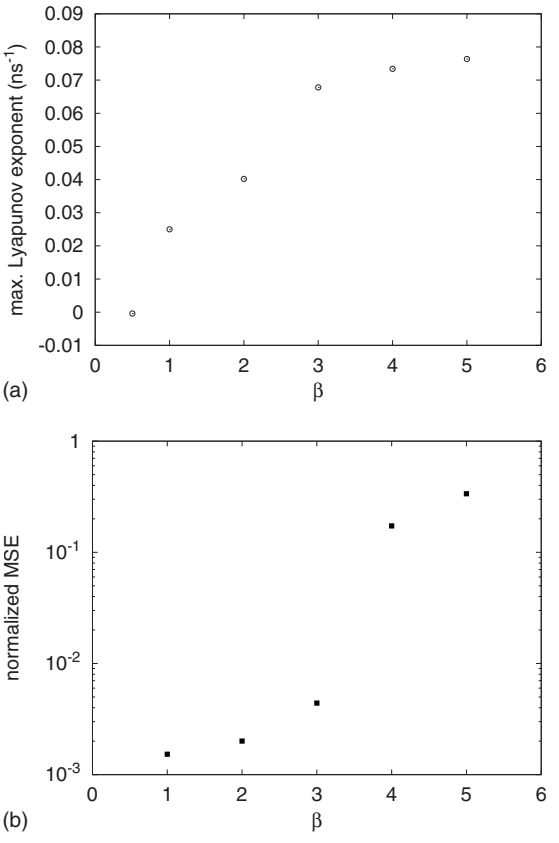

FIG. 5. (a) MLE of the delay-feedback system versus the feedback strength β. (b) Normalized MSE attained by the SMCO algorithm with observation rate $1/N=0.1$.

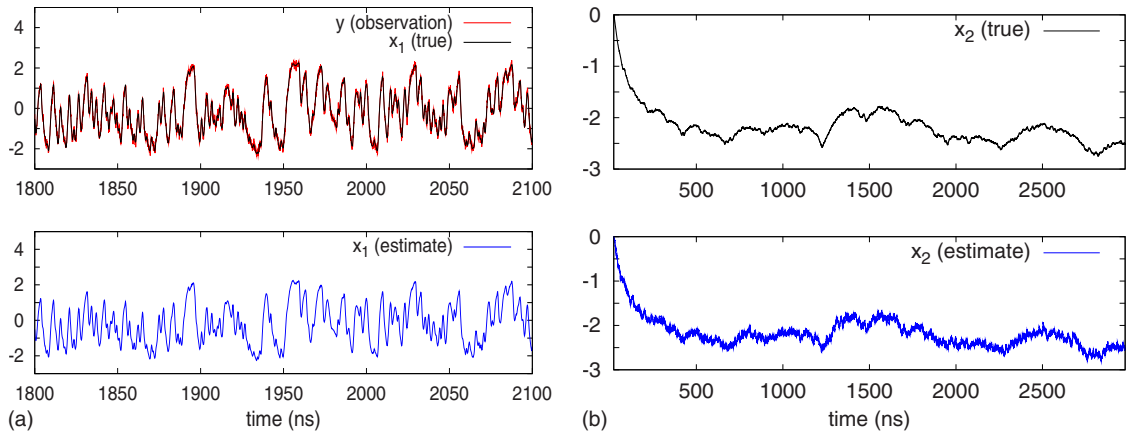

FIG. 6. (Color online) (a) The upper axes depict the evolution of variable x_1 and the associate observation $y=x_1+m$, where m is a Gaussian zero-mean random process, for $1800 \leq t \leq 2100$ ns. The variance of the Gaussian perturbations in y is chosen to ensure a signal-to-noise ratio of 20 dB and the discrete-time data rate is $1/N=0.5$ (five observations per ns). The estimate of x_1 computed recursively by the SMCO algorithm is shown in the lower axes. (b) The upper axes depict the evolution of the unobserved dynamic variable x_2 for $22.5 \leq t \leq 2977.5$ ns, while the lower plot shows its estimate.

below 0.01 for $\beta=1,2,3$ and then grows drastically for $\beta=4,5$.

The latter results do not imply that the SMCO method cannot be used for $\beta \geq 4$. Its performance can be improved, at the expense of a higher computational complexity, either by increasing the number of particles in the algorithm M or decreasing the observation period in the model N. When $N=2$, and we get an observation rate of $1/N=0.5$, it is possible to track x_1 and x_2, as well as to estimate ψ_0, accurately without any other modifications in the algorithm. Figure 6(a) depicts in the upper axes, the evolution of x_1 and the associated observations y, together with the estimated signal $\hat{x}_1$ in the lower axes. It is clearly seen that it fluctuates more rapidly (and also more chaotically, as indicated by the MLE) but we still obtain a tight estimate of it. The unobserved signal x_2 is also tracked properly, as shown by Fig. 6(b).

V. CONCLUSIONS

We have proposed a sequential Monte Carlo method for the joint estimation of unobserved dynamic variables and unknown static parameters in chaotic systems. The procedure aims at the minimization of prediction errors and it updates the variable and parameter estimates recursively. It is, therefore, suitable for an online implementation. We have given a general description of the method for a general system model and then applied it in two examples. In the first one, we have used the five-dimensional model of a CO_2 laser and have shown the validity of the proposed approach by jointly estimating the five dynamic variables and one static parameter using experimental observations from the laser. An online parameter estimation algorithm is demonstrated to work effectively with an experimental time series from a real-world chaotic system. For the second example, we have considered a delayed-feedback loop. The resulting system is interesting because the state of the discretized model grows with the feedback delay (for the specific example, the discrete-time state vector consists of 227 variables) and the dynamic behavior changes significantly with the feedback strength. With this setup, we have numerically studied the accuracy of the proposed estimation method and its relationship with the MLE of the model. In particular, we have shown that the estimation error grows with the MLE (as the system becomes more unpredictable) but accurate results can still be achieved if the observation rate (the number of observations per time unit) is high enough.

ACKNOWLEDGMENTS

I.P.M. acknowledges financial support by the Spanish Ministry of Science and Innovation of Spain (Project No. FIS2006-08525). J.M. acknowledges the support of the Ministry of Science and Innovation of Spain (project MONIN, Project No. TEC2006-13514-C02-01/TCM and program Consolider-Ingenio 2010 CSD-2008-00010 COMONSENS), and the Comunidad de Madrid (project PRO-MULTIDIS, Project No. S0505/TIC/0223). R.M. thanks Kais Al Naimee and F. Tito Arecchi for useful discussions. We thank the two anonymous reviewers for their constructive comments on the first version of this work. The example in Sec. IV was developed following a suggestion of one of the referees.

APPENDIX: SMCO ALGORITHMS

The implementation of the SMCO algorithms for the examples in Secs. III and IV are similar and differ significantly only in the choice of prior probability distributions for their initialization. In this appendix, we briefly introduce some notations needed for the concise presentation of the algorithms. Then we specify the initialization stages for the two examples and describe the details of the recursive step in the SMCO procedure employed in the examples. Finally, we briefly comment on the complexity of the algorithm.

1. Notations

Let $\mathcal{N}(\boldsymbol{\mu},\mathbf{C})$ denote a Gaussian probability distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$ and let $\mathcal{U}(a,b)$ denote

the uniform probability distribution in the interval $[a,b]$. The operation of drawing a random sample $x^{(i)}$ from a distribution $\mathcal{P}$ is denoted as $x^{(i)} \sim \mathcal{P}$.

The evolution function $\mathbf{F}^k$ defined recursively in Eq. (5) for a general setting is used in the algorithm descriptions below. For the CO_2 laser example, $\mathbf{F}^k$ corresponds to the iterative application of the discrete-time version (via Euler's method) of Eq. (9), while for the delayed-feedback loop in Sec. IV, it corresponds to the iteration of model (12). The prediction errors $\mathcal{E}_r^j$, with general definition in Eq. (6), are constructed similarly for the two examples because the observations correspond to the first dynamic variable in both cases. Finally, we use $\mathbf{I}$ to denote the identity matrix.

2. Initialization for the CO_2 laser system

The SMCO procedure requires the choice of prior distributions for the signals of interest: both dynamic variables and static parameters. For this example, we have set $\mathcal{N}(\bar{\mathbf{x}}_0, \sigma_0^2\mathbf{I})$, where $\bar{\mathbf{x}}_0 = [1.077\times10^{-3}, 1.621, 1.672, 1.630\times10, 1.672\times10]^\top$ and $\sigma_0^2 = 10^{-6}$, for $\mathbf{x}_0$ and $\mathcal{N}(8.55, 0.40)$ for γ_1. Then, we proceed with the following initialization step:

(i) at time $n=0$, draw

$$\mathbf{x}_0^{(i)} \sim \mathcal{N}(\bar{x}_0, \sigma_0^2\mathbf{I}),$$

$$\gamma_{1,0}^{(i)} \sim \mathcal{N}(8.55, 0.40),$$

(ii) and set $\epsilon_0^{(i)} = 0$, for $i = 1, \ldots, M$. Then build the particle set $\Omega_0 = \{\mathbf{x}_0^{(i)}, \gamma_{1,0}^{(i)}, \epsilon_0^{(i)}\}_{i=1}^M$ and set the variance parameters $\sigma_p^2 = 10^{-5}$, $\sigma_v^2 = 10^{-8}$, and the prediction error lag $j=2$ (all to be used in the recursive step).

3. Initialization for the delayed-feedback model

In this case, the delayed-feedback loop is initialized with $x_{k,n} = 0$ for $k = 1, 2$ and $n = 0, \ldots, \kappa$ (with probability 1), while the unknown phase parameter ψ_0 is assumed to have the uniform prior pdf $\mathcal{U}(0, \frac{\pi}{2})$. The initialization step is detailed below.

For $i = 1, \ldots, M$,

(i) set $\mathbf{x}_n^{(i)} = \mathbf{0}$ for $n = 0, \ldots, \kappa$,

(ii) then draw $\psi_{0,0}^{(i)} \sim \mathcal{U}(0, \frac{\pi}{2})$ and set $\psi_{0,r}^{(i)} = \psi_{0,0}^{(i)}$ for all r such that $rN < \kappa$, and

(iii) fix the initial costs to null $\epsilon_0^{(i)} = 0$.

Then build the particle sets $\Omega_r = \{\mathbf{x}_{rN}^{(i)}, \psi_{0,r}^{(i)}, \epsilon_r^{(i)}\}_{i=1}^M$, for all r such that $rN < \kappa$, and set the variance parameters $\sigma_p^2 = 10^{-6}$, $\sigma_v^2 = 10^{-5}$, and the prediction error lag $j=3$.

4. Recursive update

This stage is common for the two examples. Let ϕ denote the unknown parameter, i.e., either γ_1 or ψ_0 for the examples in Secs. III and IV, respectively. The covariance matrix of the noise vector $\mathbf{v}_n$ denoted $\mathbf{C}$ is easily obtained from the descriptions of the discrete-time random models in the two examples. At time $n = rN$, assume that $\Omega_r = \{\mathbf{x}_{rN}^{(i)}, \phi_r^{(i)}, \epsilon_r^{(i)}\}_{i=1}^M$ is available. The SMCO algorithm proceeds recursively as follows at time $n+N = (r+1)N$, when the next observation is received.

(i) Draw trial parameter values $\phi_{r+1}^{(M+i)} \sim \mathcal{N}(\phi_r^{(i)}, \sigma_p^2)$ and compute $\epsilon_r^{(M+i)} = \mathcal{E}_r^j(\mathbf{x}_{rN}^{(i)}, \phi_r^{(M+i)})$ for $i = 1, \ldots, M$ and $j=2$.

(ii) Construct the discrete probability measure

$$p_r(i) \propto \left(\epsilon_r^{(i)} - \min_{j\in\{1,\ldots,2M\}} \epsilon_r^{(j)} + \frac{1}{M} \right)^{-2},$$

for each $i \in \{1, \ldots, 2M\}$, and use it to draw a set of indices $i^{(k)} \sim p_r(i)$, where $k = 1, \ldots, M$.

(iii) Draw state values $\mathbf{x}_{(r+1)N}^{(k)} \sim \mathcal{N}[\mathbf{F}^N(\mathbf{x}_{rN}^{(i^{(k)})}, \phi_r^{(i^{(k)})}), \mathbf{C}]$, and set $\phi_{r+1}^{(k)} = \phi_r^{(i^{(k)})}$, for $k = 1, \ldots, M$.

(iv) Compute new costs $\epsilon_{r+1}^{(i)} = \mathcal{E}_{r+1}^j(\mathbf{x}_{(r+1)N}^{(i)}, \phi_{r+1}^{(i)})$ and build $\Omega_{r+1} = \{\mathbf{x}_{(r+1)N}^{(i)}, \phi_{r+1}^{(i)}, \epsilon_{r+1}^{(i)}\}_{i=1}^M$.

(v) Estimation: let $i_{\min} \in \{1, \ldots, M\}$ be the index of the smallest cost, i.e., $\epsilon_{r+1}^{(i_{\min})} \le \epsilon_{r+1}^{(k)}$ for $k = 1, \ldots M$. Then we choose estimates $\hat{\mathbf{x}}_{(r+1)N} = \mathbf{x}_{(r+1)N}^{(i_{\min})}$ and $\hat{\phi}_{r+1} = \phi_{r+1}^{(i_{\min})}$. Optionally, predict $\hat{\mathbf{x}}_{(r+1)N+k} = \mathbf{F}^k[\hat{\mathbf{x}}_{(r+1)N}, \hat{\phi}_{r+1}]$ for $k = 1, \ldots, N-1$.

5. Computational complexity

The complexity of the proposed procedure grows linearly with the number of particles M and, as a consequence, it can be computationally intensive. An advantage of the particle approach, however, is the feasibility of parallel implementations that ensure a fast processing of the data and enable real-time operation. Indeed, the processing of a single particle (or a relatively small group of them) is computationally inexpensive. It involves sampling from simple Gaussian distributions and straightforward computations for the costs. These tasks can be carried out by very simple processing elements.

The only relevant difficulty for a parallel implementation is the selection of particles performed in the second and third items of the recursive step because it requires the interaction of the whole population of samples. However, this problem can be avoided using the methods recently developed for the parallelization of the resampling procedures in standard sequential Monte Carlo filtering algorithms [20,21]. We also note that it is not necessary to carry out the selection of particles at each time step, although we have not addressed this issue in this paper.

[1] A. Ghosh, V. R. Kumar, and B. D. Kulkarni, Phys. Rev. E **64**, 056222 (2001).
[2] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Phys. Rev. A **45**, 5524 (1992).
[3] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss, Phys. Rev. E **66**, 016210 (2002).
[4] M. F. Bugallo, S. Xu, and P. M. Djurić, Digit. Signal Process. **17**, 774 (2007).
[5] U. Parlitz, Phys. Rev. Lett. **76**, 1232 (1996).
[6] A. Maybhate and R. E. Amritkar, Phys. Rev. E **59**, 284 (1999).
[7] D. Huang, Phys. Rev. E **69**, 067201 (2004).
[8] J. Míguez, Digit. Signal Process. **17**, 787 (2007).
[9] I. P. Mariño, E. Allaria, M. A. F. Sanjuán, R. Meucci, and F. T. Arecchi, Phys. Rev. E **70**, 036208 (2004).
[10] I. P. Mariño, S. Zambrano, M. A. F. Sanjuán, F. Salvadori, R. Meucci, and F. T. Arecchi, Int. J. Bifurcation Chaos Appl. Sci. Eng. **17**, 3639 (2007).
[11] A. B. Cohen, B. Ravoori, T. E. Murphy, and R. Roy, Phys. Rev. Lett. **101**, 154102 (2008).
[12] *Sequential Monte Carlo Methods in Practice*, edited by A. Doucet, N. de Freitas, and N. Gordon (Springer, New York, 2001).
[13] H. R. Künsch, Ann. Stat. **33**, 1983 (2005).
[14] J. S. Liu and R. Chen, J. Am. Stat. Assoc. **93**, 1032 (1998).
[15] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, IEEE Trans. Signal Process. **50**, 174 (2002).
[16] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić, Proc. IEEE **92**, 423 (2004).
[17] A. Papavasiliou, Stochastic Proc. Appl. **116**, 1048 (2006).
[18] M. Ciofini, A. Labate, R. Meucci, and M. Galanti, Phys. Rev. E **60**, 398 (1999).
[19] A. Argyris *et al.*, Nature (London) **438**, 343 (2005).
[20] M. Bolić, P. M. Djurić, and S. Hong, IEEE Trans. Signal Process. **53**, 2442 (2005).
[21] J. Míguez, Signal Process. **87**, 3155 (2007).