State and parameter estimation using unconstrained optimization

Jan Schumann-Bischoff[1] and Ulrich Parlitz[2,3]

[1]*Drittes Physikalisches Institut, Georg-August-Universität Göttingen, Friedrich-Hund-Platz 1, D-37077 Göttingen, Germany*

[2]*Max Planck Research Group Biomedical Physics, Max Planck Institute for Dynamics and Self-Organization, Am Fassberg 17, D-37077 Göttingen, Germany*

[3]*Institute for Nonlinear Dynamics, Georg-August-Universität Göttingen, Am Fassberg 17, D-37077 Göttingen, Germany*

(Received 20 May 2011; revised manuscript received 15 July 2011; published 23 November 2011)

We present an efficient method for estimating variables and parameters of a given system of ordinary differential equations by adapting the model output to an observed time series from the (physical) process described by the model. The proposed method is based on (unconstrained) nonlinear optimization exploiting the particular structure of the relevant cost function. To illustrate the features and performance of the method, simulations are presented using chaotic time series generated by the Colpitts oscillator, the three-dimensional Hindmarsh-Rose neuron model, and a nine-dimensional extended Rössler system.

DOI: 10.1103/PhysRevE.84.056214 PACS number(s): 05.45.Xt, 05.45.Tp, 07.05.Kf

I. INTRODUCTION

Quantitative models describing the temporal evolution of dynamical processes are essential for physics and many other scientific disciplines. Often, such models can be derived from first principles, but they contain parameters whose values are not known at all or are only partially known and depend on the physical context. To identify these parameters, experimental data are required that consist of observed time series of finite length. To adapt the model to the data, not only do (unknown) parameters have to be adjusted, but also model variables that have not been measured but represent state variables that determine the temporal evolution of the model. This data-driven estimation task for model parameters and state variables has been addressed by many authors who suggested various methods to fit the model to the observed data. The challenge of this task depends on the complexity of the model (e.g., dimensionality, type of functional relations), the kind of underlying dynamics (periodic, chaotic, etc.), and (last but not least) on the quality and quantity of available data. Among the more difficult cases are chaotic dynamical systems given by nonlinear ordinary differential equations (ODE's) in which a sensitive dependence on variables and parameters may lead to a nontrivial estimation problem. To cope with this identification task, several methods have been proposed in the past, including synchronization-based methods [1–9], adaptive observers and control system [10–15], optimization-based methods [16–20], probabilistic and geometric approaches [21,22], path-integral methods [23,24], and a reformulation of the problem as a boundary-value problem [25,26].

II. ESTIMATION METHOD

In the following, the proposed state and parameter estimation method will be described. We start with a given (experimental) R-dimensional (i.e., multivariate) time series $\{\boldsymbol{\eta}(t_n)\}$ consisting of N samples $\boldsymbol{\eta}(t_n)$ measured at times t_n $(n = 1, \ldots, N)$. Furthermore, we assume that the model we want to adapt to the data is known (except for some unknown parameters) [27]. This model can be a discrete (iterated map) or continuous dynamical system. Here we shall focus on the latter case and consider D-dimensional models consisting of coupled ODE's,

$$\frac{d\,\boldsymbol{y}(t)}{dt} = \boldsymbol{F}(\boldsymbol{y}(t), \boldsymbol{p}, t). \tag{1}$$

The state vector(s) $\boldsymbol{y}(t) = (y_1, \ldots, y_D)^{\mathrm{T}}$ and U model parameters $\boldsymbol{p} = (p_1, \ldots, p_U)^{\mathrm{T}}$ are unknown and have to be estimated from the time series $\{\boldsymbol{\eta}(t_n)\}$. To do this, a measurement function

$$\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{y}(t), \boldsymbol{q}, t) \tag{2}$$

is required describing the relation between model states $\boldsymbol{y}(t)$ and resulting time-series values $\boldsymbol{z}(t)$ corresponding to the observations $\boldsymbol{\eta}(t)$. This measurement function may contain V additional unknown parameters $\boldsymbol{q} = (q_1, \ldots, q_V)^{\mathrm{T}}$ that also have to be estimated using information from the given time series $\{\boldsymbol{\eta}(t_n)\}$.

A. The cost function

The goal of the estimation process is to find a set of values for all unknown variables and parameters such that the model ODE's (when integrated with these values) provide via measurement function (2) a model time series $\{\boldsymbol{z}(t_n)\}$ that fits to the experimentally obtained time series $\{\boldsymbol{\eta}(t_n)\}$. In other words, the average difference between $\boldsymbol{\eta}(t_n)$ and $\boldsymbol{z}(t_n)$ should be as small as possible. This goal can be achieved by minimizing the cost function (also called *tracking error* [25]),

$$C(\{\boldsymbol{y}(t_n)\}, \boldsymbol{p}, \boldsymbol{q}) = \sum_{t_n \in \mathcal{T}_M} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)]^{\mathrm{T}} \boldsymbol{A}\, [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)], \tag{3}$$

where $\mathcal{T}_M$ is the set of instants of time where measurement values are available and $\boldsymbol{A}$ is a weighting matrix whose exact form will be discussed in Sec. II B. To make sure that the solution of this minimization problem also fulfills the model equations (1), one can consider the system's ODE's as constraints [16,18] and use a suitable method for constrained optimization [28,29]. Here we follow a different approach in which the (average) deviation from the model equations (1) is added to the cost functions. Technically, this can be done by approximating the first (temporal) derivative $d\,\boldsymbol{y}(t)/dt$ by finite

differences evaluated at equidistant times $t_m = (m-1)\Delta t$ $(m = 1, \ldots, M)$ constituting the set $\mathcal{T}_S$ of points in time where approximations of the solution of the system's equations are available. Note that $\mathcal{T}_M$ and $\mathcal{T}_S$ are not necessarily identical. For example, experimental sampling times may be larger than the time steps used for numerical integration of the ODE's. To avoid additional interpolations, we shall assume from now on that $\mathcal{T}_M$ is a subset of $\mathcal{T}_S$ [30].

Using a finite-difference approximation of the derivatives, the model equations read

$$\left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_m} \approx \boldsymbol{F}(\boldsymbol{y}(t_m), \boldsymbol{p}, t_m), \tag{4}$$

where the symbol $\frac{\Delta y}{\Delta t}|_{t_m}$ stands for a finite-difference approximation of $\frac{dy(t)}{dt}$ at time t_m. The exact form of the approximation used in our simulations is given in Appendix.

The difference between both sides of Eq. (4) can be expressed as a residue $\boldsymbol{u}(t_m)$ given by

$$\boldsymbol{u}(t_m) = \left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_m} - \boldsymbol{F}(\boldsymbol{y}(t_m), \boldsymbol{p}, t_m) \tag{5}$$

and the goal of the adaption process is to minimize (on average) the residues *and* the differences between $\boldsymbol{\eta}(t_n)$ and $\boldsymbol{z}(t_n)$. Therefore, the cost function is extended by a term accounting for the *modeling error* [25],

$$C(\{\boldsymbol{y}(t_m)\}, \boldsymbol{p}, \boldsymbol{q}) = \sum_{t_n \in \mathcal{T}_M} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)]^{\mathrm{T}} \boldsymbol{A} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)] + \sum_{m=1}^{M} \boldsymbol{u}(t_m)^{\mathrm{T}} \boldsymbol{B} \boldsymbol{u}(t_m), \tag{6}$$

where $\boldsymbol{B}$ is another weighting matrix whose form will be discussed in Sec. II B.

When using the cost function (6), we observed suboptimal solutions $\hat{\boldsymbol{y}}(t_m)$ oscillating close to the true solution, as shown in Fig. 1. Such oscillating solutions can be avoided by adding a penalty term to the cost function punishing nonsmooth $\hat{\boldsymbol{y}}(t_m)$. Technically, this is implemented as the deviation of the solution $\hat{\boldsymbol{y}}(t_m)$ from an interpolation at time t_m using neighboring points. Since derivatives $\boldsymbol{F}(\boldsymbol{y}(t_m), \boldsymbol{p}, t_m)$ are also known, Hermite interpolation turns out to be a suitable scheme providing the approximation

$$\begin{aligned}\boldsymbol{y}_{\mathrm{apr}}(t_m) = {} & \frac{11}{54}[\boldsymbol{y}(t_{m-2}) + \boldsymbol{y}(t_{m+2})] + \frac{8}{27}[\boldsymbol{y}(t_{m-1}) + \boldsymbol{y}(t_{m+1})] \\ & + \frac{\Delta t}{18}[\boldsymbol{F}(\boldsymbol{y}(t_{m-2}), \boldsymbol{p}, t_{m-2}) - \boldsymbol{F}(\boldsymbol{y}(t_{m+2}), \boldsymbol{p}, t_{m+2})] \\ & + \frac{4\Delta t}{9}[\boldsymbol{F}(\boldsymbol{y}(t_{m-1}), \boldsymbol{p}, t_{m-1}) - \boldsymbol{F}(\boldsymbol{y}(t_{m+1}), \boldsymbol{p}, t_{m+1})]\end{aligned} \tag{7}$$

of $\hat{\boldsymbol{y}}(t_m)$. The averaged deviation of $\boldsymbol{y}_{\mathrm{apr}}(t_m)$ from $\boldsymbol{y}(t_m)$,

$$\sum_{m=3}^{M-2} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)]^{\mathrm{T}} \boldsymbol{E} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)], \tag{8}$$

is then used to add to the cost function (6) a term enforcing smoothness,

$$\begin{aligned}C(\{\boldsymbol{y}(t_m)\}, \boldsymbol{p}, \boldsymbol{q}) = {} & \sum_{t_n \in \mathcal{T}_M} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)]^{\mathrm{T}} \boldsymbol{A} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)] + \sum_{m=1}^{M} \boldsymbol{u}(t_m)^{\mathrm{T}} \boldsymbol{B} \boldsymbol{u}(t_m) \\ & + \sum_{m=3}^{M-2} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)]^{\mathrm{T}} \boldsymbol{E} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)],\end{aligned} \tag{9}$$

where $\boldsymbol{E}$ is a weighting matrix to be specified in Sec. II B.

When minimizing the cost function, it is useful to incorporate preknowledge in terms of relevant ranges of parameters and variables (i.e., *bound constraints*). Here, however, we shall employ a minimization method that is not able to handle bound constraints directly. Therefore, we shall force parameters and variables to stay close to predefined bounds by adding a penalty function to the cost function, which is zero within the bounds and increases quadratically outside the specified ranges. Let $\boldsymbol{w} = (\{\boldsymbol{y}(t_m)\}, \boldsymbol{p}, \boldsymbol{q}) = (w_1, \ldots, w_L)$ be a vector of length $L = MD + U + V$ containing all quantities to be estimated. To force $\boldsymbol{w}$ to stay between the lower and upper bounds $\boldsymbol{w}_{\mathrm{l}}$ and $\boldsymbol{w}_{\mathrm{u}}$, respectively, a penalty function

$$Q(\boldsymbol{w}, \boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{u}}) = \boldsymbol{q}(\boldsymbol{w}, \boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{u}})^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{q}(\boldsymbol{w}, \boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{u}}) \tag{10}$$

is introduced, whereas $\boldsymbol{q}(\boldsymbol{w}, \boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{u}}) = (q_1, \ldots, q_L)^{\mathrm{T}}$ and

$$q_i(w_i, w_{\mathrm{l},i}, w_{\mathrm{u},i}) = \begin{cases} w_{\mathrm{u},i} - w_{\mathrm{i}} & \text{for} \quad w_i \geqslant w_{\mathrm{u},i}, \\ 0 & \text{for} \quad w_{\mathrm{l},i} < w_{\mathrm{i}} < w_{\mathrm{u},i}, \\ w_{\mathrm{l},i} - w_i & \text{for} \quad w_i \leqslant w_{\mathrm{l},i}. \end{cases} \tag{11}$$

q_i is zero if the value of w_i lies in its bounds. The weighting matrix $\boldsymbol{G}$ sets the strength of the penalty if some w_i are outside their bounds (see Sec. II B). To punish values outside their bounds, the penalty function $Q(\boldsymbol{w}, \boldsymbol{w}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{u}})$ has to be added to

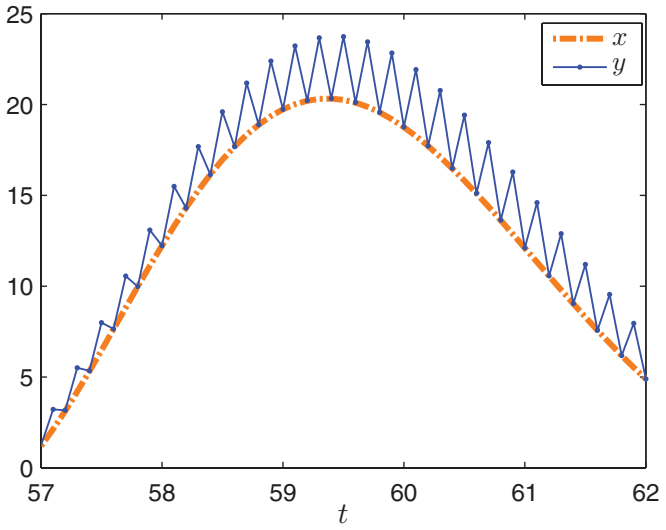

FIG. 1. (Color online) Example of a nonsmooth solution $y(t_m)$ (solid curve) oscillating close to the true solution $x(t_m)$ (dashed curve) due to an instability that leads to an alternating sequence of integration values.

the cost function (9). This leads to the final form of the cost function,

$$\begin{aligned} C(\boldsymbol{w}) = & \sum_{t_n \in \mathcal{T}_M} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)]^{\mathrm{T}} \boldsymbol{A} [\boldsymbol{\eta}(t_n) - \boldsymbol{z}(t_n)] \\ & + \sum_{m=1}^{M} \boldsymbol{u}(t_m)^{\mathrm{T}} \boldsymbol{B} \boldsymbol{u}(t_m) \\ & + \sum_{m=3}^{M-2} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)]^{\mathrm{T}} \boldsymbol{E} [\boldsymbol{y}_{\mathrm{apr}}(t_m) - \boldsymbol{y}(t_m)] \\ & + \boldsymbol{q}(\boldsymbol{w}, \boldsymbol{w}_1, \boldsymbol{w}_{\mathrm{u}})^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{q}(\boldsymbol{w}, \boldsymbol{w}_1, \boldsymbol{w}_{\mathrm{u}}), \end{aligned} \tag{12}$$

which can be summarized as a single sum,

$$C(\boldsymbol{w}) = \sum_{j=1}^{J} H_j(\boldsymbol{w})^2 = \|\boldsymbol{H}(\boldsymbol{w})\|_2^2, \tag{13}$$

whereas $\boldsymbol{H} = (H_1, \ldots, H_J)^{\mathrm{T}}$ are the summands of Eq. (12) and $J = NR + MD + (M-4)D + L$. The task is to find the minimum of this cost function with L unknown quantities to be estimated. The values of the variables and parameters where the cost function is minimal provide the solutions of the estimation problem, and they will be marked in the following with a hat [i.e., the cost function is minimal at $C(\{\hat{\boldsymbol{y}}(t_m)\}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{q}})$]. The smaller the minimum is, the better the time series $\boldsymbol{\eta}(t_n)$ can be described by the given model.

The cost function (12) is minimized by a numerical optimization routine that exploits the specific structure of Eq. (13). In general, an effective and well-explored algorithm to minimize such least-squares problems was developed by Levenberg and Marquardt [31,32], and there exist many implementations of this algorithm for general nonlinear optimization problems. Here we shall use an algorithm called sparseLM [33] that makes use of the fact that our cost function Eq. (13) is given as a sum of squared terms $H_j(\boldsymbol{w})$ and requires the vector $\boldsymbol{H}(\boldsymbol{w})$ and the sparse Jacobian [34] of the function $\boldsymbol{H}(\boldsymbol{w})$ as input.

B. Weighting in the cost function

All weighting matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{E}$, and $\boldsymbol{G}$ have to be chosen so that the optimization problem is well conditioned and can be properly solved by the optimization algorithm. To control their relative weights [i.e., the influence of the corresponding terms of the cost function (12)], a homotopy parameter α is introduced to switch from an emphasis on agreement with observations ($\alpha = 1$) to an emphasis on correct model dynamics ($\alpha = 0$). Another parameter β determines the penalties for leaving the prescribed parameter ranges. In general, diagonal matrices are sufficient, and therefore we used the following matrices:

$$\begin{aligned} \boldsymbol{A} &= \frac{\alpha}{N} \mathrm{diag}(\boldsymbol{a}), \quad \boldsymbol{B} = \frac{1-\alpha}{M} \mathrm{diag}(\boldsymbol{b}), \\ \boldsymbol{E} &= \frac{1-\alpha}{M} \mathrm{diag}(\boldsymbol{e}), \quad \boldsymbol{G} = \frac{\beta}{L} \mathbb{1}(L), \end{aligned} \tag{14}$$

where $\mathbb{1}(d)$ denotes the d-dimensional unity matrix and $\mathrm{diag}(\boldsymbol{v})$ denotes a diagonal matrix with the elements of $\boldsymbol{v}$ as diagonal elements. $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{e}$ are R-, D-, and D-dimensional vectors, respectively, and can set the individual weighting for certain time series or model variable dimensions. In general, all weighting matrices may also depend on time (for example, to give more recent samples a higher impact), but we shall consider here constant values only. If $\alpha = 1$ [i.e., $\boldsymbol{A} = \frac{1}{N}\mathrm{diag}(\boldsymbol{a})$ and $\boldsymbol{B} = \boldsymbol{E} = 0$], the minimization of the cost function results in a solution $\hat{\boldsymbol{z}}(t_n)$ that coincides with the observed time series $\boldsymbol{\eta}(t_n)$ but does not necessarily fulfill the model equations. If $\alpha = 0$, the opposite is true and the solution fulfills the model equations exactly but the output of the measurement function $\boldsymbol{z}(t_n)$ may not fit to the observed data $\boldsymbol{\eta}(t_n)$. Practically, a good choice is to start with α close to 1 because this yields (for suitable measurement functions $\boldsymbol{h}$) a cost function with a single minimum and leads to a solution close to the observed time series. This solution is then used as an initial value for another run of the optimization procedure with a slightly decreased α. In general, the smaller α is, the more complex becomes the cost function landscape with an increasing number of additional local minima. However, using optimal values from previous runs of the optimization procedure (with slightly larger α), we can "track" the relevant global minimum. The reduction step is repeated until a suitable value for α is reached. The technique of changing a certain parameter, rerun the algorithm with the result of the previous run as an initial guess and repeat this procedure until the parameter reaches a desired value, is also called *continuation* and turned out to be crucial for solving the estimation problem. As a criterion for stopping the variation of α, we suggest to monitor the standard deviation of parameter estimations based on different segments of the given time series. This approach will be discussed in detail in Sec. IV B.

In all cases considered in this work, β was set to a large number, $\beta = 1000L$. Note that the choice of β should only affect the search trajectory in the cost function during optimization but not the final values of the estimated quantities. The values of the estimated quantities should be in their predefined bounds, and with that the part of the cost function punishing quantities outside their bounds should be zero. Another question is how to choose the weighting vectors $\boldsymbol{b}$ and $\boldsymbol{e}$ in Eq. (14). Remember that $\boldsymbol{e}$ affects the weighting of the Hermite interpolation of each variable to estimate. A high weight only leads to a smooth trajectory of $\boldsymbol{y}(t_m)$, but the main dynamics will not be influenced by the Hermite interpolation because (8) will only increase (decrease) if $\boldsymbol{y}(t_m)$ becomes less smooth (smoother). A change in the dynamics will not change the costs (8) if $\boldsymbol{y}(t_m)$ stays smooth. Hence we set all elements on $\boldsymbol{e}$ to a large number, say 1000. Later for $\boldsymbol{b}$ we will only present an idea of how to choose it.

III. APPLICATIONS AND EXAMPLES

In the following, the algorithm presented in the previous section will be used to adapt a model to a (noisy) time series in order to obtain unobserved states and model as well as measurement function parameters.

A. Measurement noise

In all examples considered here, the time series $\boldsymbol{\eta}(t_n)$ is not observed from a "real" experiment, but created numerically. To show that our method works with noisy time series, a noise

signal $\boldsymbol{\eta}_{\rm n}(t_n)$ will be added to the "clean" time series $\boldsymbol{\eta}_{\rm ts}(t_n)$. A common type of noise in experimentally observed time series is white noise, which is given by normally distributed random numbers with a variance σ^2 and a mean that is zero in all examples. Adding the noise to the clean time series, we obtain a noisy time series,

$$\boldsymbol{\eta}(t_n) = \boldsymbol{\eta}_{\rm ts}(t_n) + \boldsymbol{\eta}_{\rm n}(t_n), \quad \boldsymbol{\eta}_{\rm n}(t_n) = \mathcal{N}_n(0,\sigma^2), \tag{15}$$

which is typical for experiments with measurement noise. To quantify the power of the clean signal and the noise, one can define the signal-to-noise ratio (SNR in dB) as

$$\begin{aligned} \mathrm{SNR} &= 10\log_{10}\left(\frac{\sum_{t_n\in\mathcal{T}_M}\left[\boldsymbol{\eta}_{\rm ts}(t_n)-\overline{\boldsymbol{\eta}}_{\rm ts}\right]^2}{\sum_{t_n\in\mathcal{T}_M}\boldsymbol{\eta}_n(t_n)^2}\right) \\ &= 10\log_{10}\left(\frac{\sum_{t_n\in\mathcal{T}_M}\left[\boldsymbol{\eta}_{\rm ts}(t_n)-\overline{\boldsymbol{\eta}}_{\rm ts}\right]^2}{\sum_{t_n\in\mathcal{T}_M}\mathcal{N}_n(0,\sigma^2)^2}\right), \end{aligned} \tag{16}$$

which gives a logarithmic relation between the power of $\boldsymbol{\eta}_{\rm ts}(t_n)$ and the power of the noise $\mathcal{N}_n(0,\sigma^2)$ (the overbar denotes the mean). The smaller the SNR, the more measurement noise is present.

B. The Colpitts oscillator

Our first example for state and parameter estimation is based on the Colpitts oscillator [35]. The Colpitts oscillator is an electronic circuit, and its (normalized) model ODE's (describing voltages and currents of the circuit) are given by

$$\begin{aligned} \frac{dy_1(t)}{dt} &= p_1 y_2(t), \\ \frac{dy_2(t)}{dt} &= -p_2[y_1(t)+y_3(t)] - p_3 y_2(t), \\ \frac{dy_3(t)}{dt} &= p_4[y_2(t)+1+e^{-y_1(t)}]. \end{aligned} \tag{17}$$

To generate the time series, the following ODE's (18) were integrated:

$$\begin{aligned} \frac{dx_1(t)}{dt} &= 5x_2(t), \\ \frac{dx_2(t)}{dt} &= -0.08[x_1(t)+x_3(t)] - 0.7x_2(t), \\ \frac{dx_3(t)}{dt} &= 6.3[x_2(t)+1+e^{-x_1(t)}]. \end{aligned} \tag{18}$$

As an integration method, the Euler method with time step 0.02 was chosen because the inaccuracy of this method for large time steps leads to an integration error that results in some dynamical noise in the time series. Furthermore, white noise $[\mathcal{N}_n(0,\sigma^2)]$ was added to show that the model can be adapted to noisy time series. Observations $\eta(t_n)$,

$$\eta(t_n) = x_1(t_n) + x_3(t_n) + 1.3 + \mathcal{N}_n(0,1), \tag{19}$$

were sampled at $t_n \in \mathcal{T}_M = \{0,0.02,0.04,\ldots,150\}$ including an offset of 1.3 and measurement noise with SNR = 10.9 dB.

The model consists of Eq. (17), and the measurement function for the estimation procedure is given by

$$h(\boldsymbol{y}(t),q) = y_1(t) + y_3(t) + q\ . \tag{20}$$

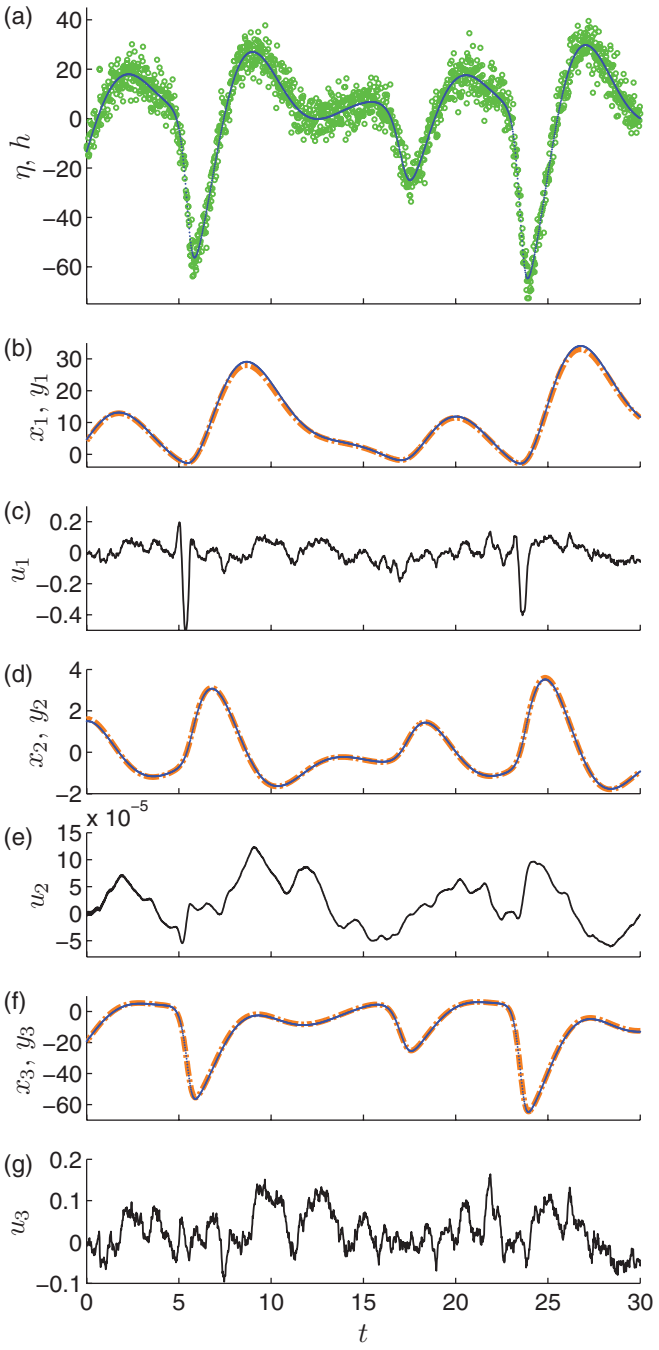

FIG. 2. (Color online) Adaption of the Colpitts oscillator model (17) to the time series $\eta(t_n)$ [Eq. (19)]. (a) The model output given by the measurement function h [Eq. (20); dark blue dots] was adapted to the time series $\eta(t_n)$ (bright green circles). (b), (d), and (f) The estimated variables y_1, y_2, and y_3 (blue dots) and original ("true") variables x_1, x_2, and x_3 (orange dashed line; not directly observed) used to generate η. (c), (e), and (g) Residues [see Eq. (5)] u_i, $i = 1,2,3$ representing deviations from the model ODE's. In addition to the model variables, the model parameters are estimated at $\boldsymbol{p} = (5.23,0.0777,0.677,6.28)$ whereas the parameters used to integrate (18) and to generate the time series $\eta(t_n)$ are $(5,0.08,0.7,6.3)$. The parameter of the measurement function is estimated at $q = 1.29$ [while 1.3 was used in Eq. (19) to generate $\eta(t_n)$]. For better visibility, the results are only shown from $t = 0$ to 30 while the model time series (20) was adapted to the given data (19) from $t = 0$ to 150.

Note that the parameter q of the measurement function will be estimated in addition to the four model parameters $p_1, \ldots, p_4$. For this example, state variables $\boldsymbol{y} = \boldsymbol{y}(t_m)$ will be estimated at $t_m = t_n \in \mathcal{T}_S = \mathcal{T}_M$, i.e., at the same times that observations were done. The homotopy parameter α was decreased in steps from 0.9 to 0.5 and finally to 0.1. The weighting vector for the ODE approximation was set to $\boldsymbol{b} = (1,1000,1)$. That means that the final approximation of the second ODE of the model will be more accurate than for the first and third model ODE. This $\boldsymbol{b}$ was chosen because the measurement function (20) is independent of $y_2(t)$, and it turned out that giving a high weight to the ODE approximation of variables that are not part of the measurement function provides better results. Figure 2 shows the estimated variables and parameters. Despite the integration errors in $x_1(t),x_2(t),x_3(t)$, the estimated variables and the parameters of the model and the measurement function parameters coincide very well with the ones used to generate $\eta(t_n)$.

C. The Hindmarsh-Rose model

Our second example is based on the Hindmarsh-Rose neuron model [36], which generates typical neuronal activity such as spiking and bursting governed by dynamics on clearly separated time scales. The model equations read

$$\begin{aligned}\frac{dy_1(t)}{dt} &= -y_1(t)^3 + p_1 y_1(t)^2 + y_2(t) - y_3(t),\\ \frac{dy_2(t)}{dt} &= 1 - p_2 y_1(t)^2 - y_2(t),\\ \frac{dy_3(t)}{dt} &= p_3\{y_1(t) + 0.25[p_4 - y_3(t)]\},\end{aligned} \tag{21}$$

where $y_1(t)$ denotes the membrane potential whereas $y_2(t)$ and $y_3(t)$ describe slow and fast ion current rates, respectively. To generate the time series $\eta(t_n)$, the system

$$\begin{aligned}\frac{dx_1(t)}{dt} &= -x_1(t)^3 + 3x_1(t)^2 + x_2(t) - x_3(t),\\ \frac{dx_2(t)}{dt} &= 1 - 5x_1(t)^2 - x_2(t),\\ \frac{dx_3(t)}{dt} &= 0.004\{x_1(t) + 0.25[3.19 - x_3(t)]\}\end{aligned} \tag{22}$$

was integrated (Runge-Kutta method, time step of 0.1). For this parameter set, one observes chaotic bursting where a sequence of spikes is interrupted by periods of time with a slow and smooth variation of the membrane potential [37]. A noisy time series $\eta(t_n)$ was generated by

$$\eta(t_n) = x_1(t_n) + 1.8 + \mathcal{N}_n(0,0.13) \tag{23}$$

with SNR = 14.03 dB, $t_n \in \mathcal{T}_M = \{0,0.5,1,\ldots,800\}$, and an offset of 1.8. The measurement function used for estimation is

$$h(\boldsymbol{y}(t),q) = y_1(t) + q, \tag{24}$$

whereas $y_1(t)$ is the first variable of the Hindmarsch-Rose model (21). In addition to the four model parameters

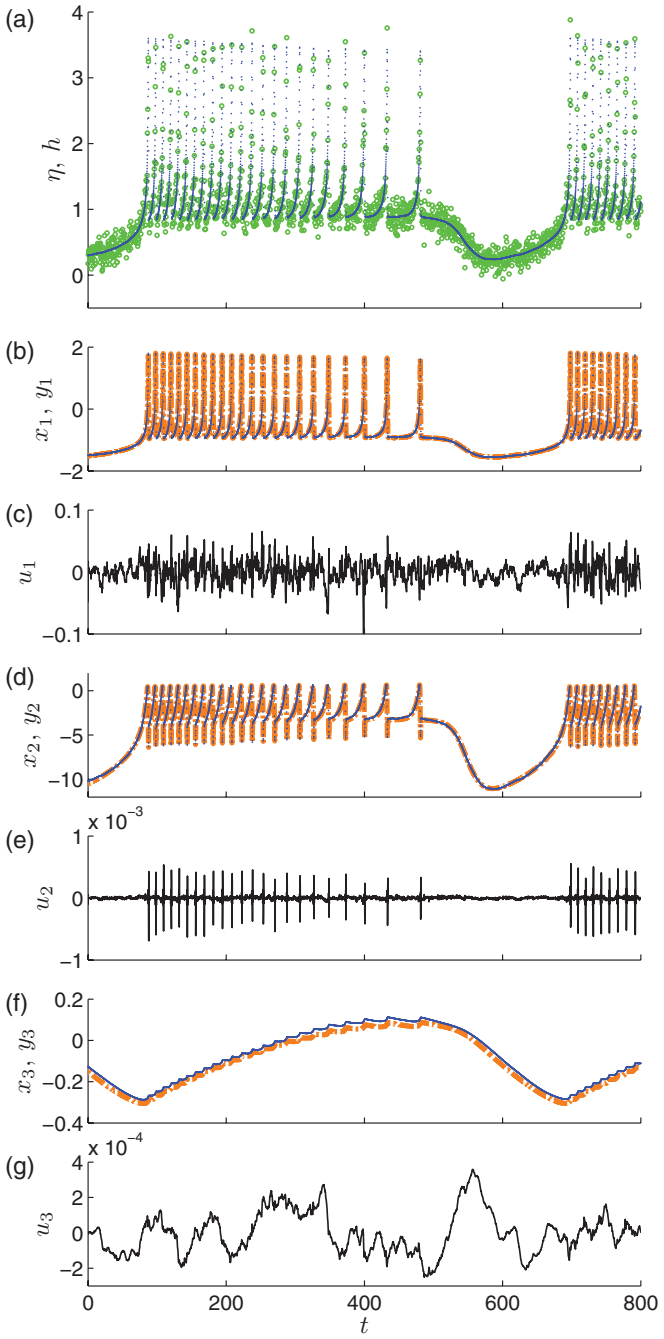

FIG. 3. (Color online) Adaption of the Hindmarsh Rose model (21) to the time series $\eta(t_n)$ [Eq. (23)]. (a) The model output given by the measurement function h [Eq. (24); dark blue dots] was adapted to the time series $\eta(t_n)$ (bright green circles). (b), (d), and (f) The estimated variables y_1, y_2, and y_3 (blue dots) and original ("true") variables x_1, x_2, and x_3 (orange dashed line; not directly observed) used to generate η. (c), (e), and (g) Residues [see Eq. (5)] u_i, $i = 1,2,3$ representing deviations from the model ODE's. In addition to the model variables, the model parameters are estimated at $\boldsymbol{p} = (2.99,4.98,0.003\,97,3.19)$ whereas the parameters used to integrate (22) and to generate the time series $\eta(t_n)$ are (3,5,0.004,3.19). The parameter of the measurement function is estimated at $q = 1.791$ [while 1.8 was used in Eq. (23) to generate $\eta(t_n)$].

$p_1, \ldots, p_4$, the parameter q of the measurement function will be estimated, too. The model variables $\boldsymbol{y}(t_m)$ will be estimated at $t_m \in \mathcal{T}_S = \{0, 0.1, 0.2, \ldots, 800\}$. This means that for this example, the state variables $\boldsymbol{y}(t_m)$ will also be estimated at times where no observations are available. The parameter α was decreased in steps from 0.9 to 0.5 to 0.1. The weighting of the model ODE's was set to $\boldsymbol{b} = (1, 1000, 1000)$, because the observations (24) depend on $y_1(t)$ only. The results of the estimation procedure are shown in Fig. 3. The estimated parameters and variables coincide very well with the corresponding values used to generate the time series $\eta(t_n)$.

D. A hyperchaotic extension of the Rössler system

The previous examples exhibited low-dimensional chaos. Now we shall consider a dynamical system with a hyperchaotic attractor. This system belongs to a family of extensions of the well-known Rössler system [38], which was introduced by Baier and Sahle in 1995 [39]. This D-dimensional model is given by the following set of ODE's:

$$\begin{aligned} \frac{dy_1(t)}{dt} &= p_1 x_1(t) - x_2(t), \\ \frac{dy_i(t)}{dt} &= x_{i-1}(t) - x_{i+1}(t), \\ \frac{dy_D(t)}{dt} &= p_2 + p_3 x_D(t)(x_{D-1} - p_4), \end{aligned} \tag{25}$$

where $i = 2, \ldots, D-1$. For our simulations, we employed the nine-dimensional case, so $D = 9 \Rightarrow i = 2, \ldots, 8$. To obtain a time series $\eta(t_n)$, the model

$$\begin{aligned} \frac{dx_1(t)}{dt} &= 0.3 x_1(t) - x_2(t), \\ \frac{dx_i(t)}{dt} &= x_{i-1}(t) - x_{i+1}(t), \\ \frac{dx_D(t)}{dt} &= 0.1 + 4 x_D(t)(x_{D-1} - 2) \end{aligned} \tag{26}$$

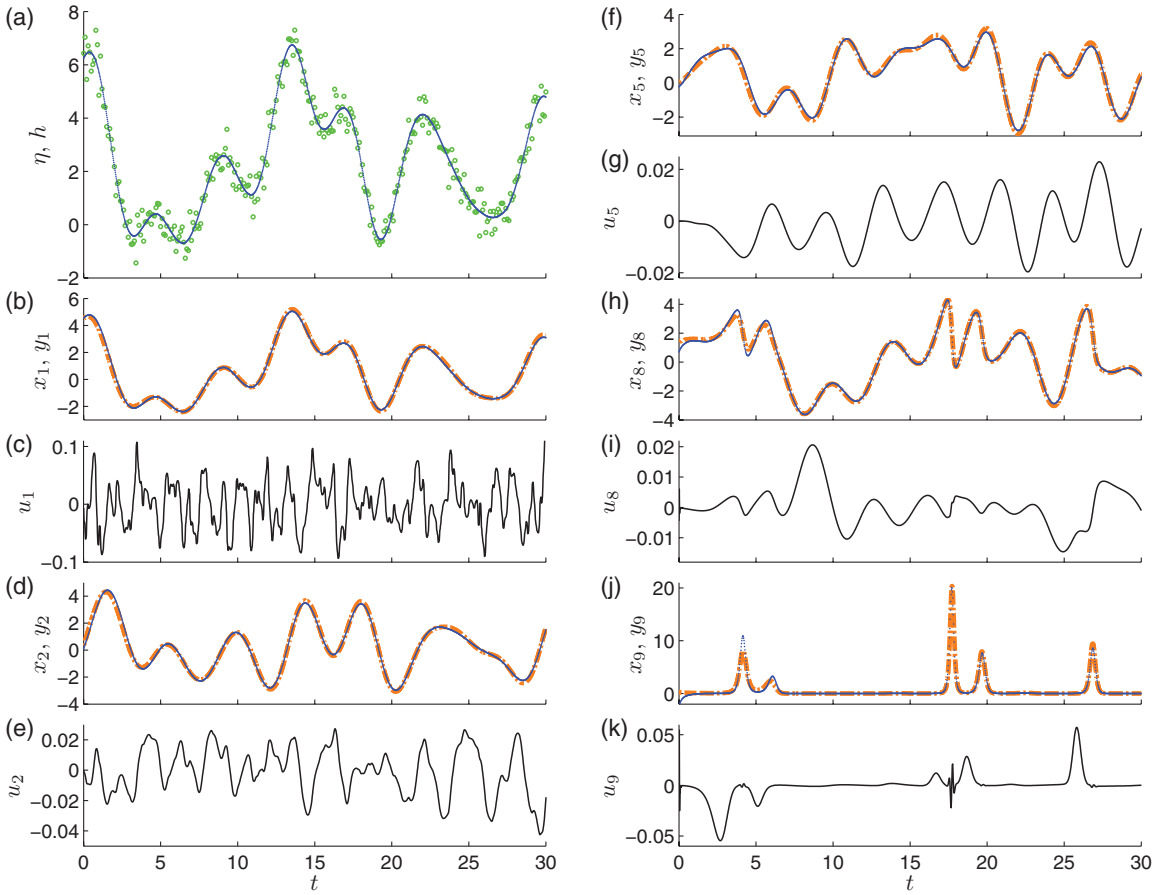

FIG. 4. (Color online) Adaption of the extended Rössler system (25) to the time series $\eta(t_n)$ given by Eq. (27). (a) The model output given by the measurement function h [Eq. (28); dark blue dots] was adapted to the time series $\eta(t_n)$ (bright green circles). (b), (d), (f), (h), and (j) The estimated variables y_1, y_2, y_5, and y_9 (blue dots) and corresponding original ("true") variables x_1, x_2, x_5, and x_9 (orange dashed line; not directly observed) used to generate η (variables y_3, y_4, y_6, and y_7 not shown here exhibit a similar dynamics and coincide with the corresponding x variables as well. (c), (e), (g), (i), and (k) Residues [see Eq. (5)] u_i, $i = 1,2,5,8,9$ representing deviations from the model ODE's. In addition to the model variables, the model parameters are estimated to be $\boldsymbol{p} = (0.293, 0.0801, 4.08, 1.96)$ whereas the parameters used to integrate (26) and to generate the time series $\eta(t_n)$ are $(0.3, 0.1, 4, 2)$. The parameter of the measurement function is estimated at $q = 1.69$ [while 1.7 was used in Eq. (27) to generate $\eta(t_n)$]. For better visibility, the results are only shown from $t = 0$ to 30 while the model time series (28) was adapted to the given data (27) from $t = 0$ to 100.

was integrated with the Euler method (time step 10^{-3}). According to Ref. [39], this system has Lyapunov exponents (0.078,0.066,0.057,0.043,0.027,0.010,0,−0.024, −9.63). Note that six of them are positive. The time series was generated by

$$\eta(t_n) = x_1(t_n) + 1.7 + \mathcal{N}_n(0,0.4) \tag{27}$$

with SNR = 13.9 dB. Observations were sampled at $t_n \in \mathcal{T}_M = \{0,0.1,\ldots,100\}$. For system identification, the measurement function

$$h(\boldsymbol{y}(t),q) = y_1(t) + q \tag{28}$$

is used to adapt the generated time series $h(\boldsymbol{y}(t),q)$ to the "observed" time series generated with Eq. (27). The model variables $y_1(t_m),\ldots,y_9(t_m)$ will be estimated at $t_m \in \mathcal{T}_S = \{0,0.02,\ldots,100\}$ and hence also at times where no observations are available. As with the previous examples, not only the model variables and the model parameters $p_1,\ldots,p_4$ will be estimated but also the parameter q of the measurement function. As shown in Fig. 4, the estimated parameters and variables coincide with the corresponding values used to generate $\eta(t_n)$. This is also true for the other variables not shown here.

IV. PARAMETER ESTIMATION

In the previous subsections, we presented examples in which the estimated parameters are close to the ones used to generate the time series $\boldsymbol{\eta}(t_n)$. The question is, "Is this always the case?" or more precisely, "When does the observed time series $\boldsymbol{\eta}(t_n)$ contain enough information so that all parameters can be estimated with the necessary accuracy?" Furthermore, "How do we determine proper values of the continuation parameter?" and last but not least, "Is there some evidence that the model (architecture) is (not) suitable for describing the given data?" These important questions will be discussed in the following subsections.

A. Observability of parameters and the shape of the cost function

Consider the common situation in which the state vector $\boldsymbol{x}(t)$ is not directly available. In a "real" experiment in which just the time series $\boldsymbol{\eta}(t_n)$ is observed, we usually do not know $\boldsymbol{x}(t)$ and the true parameters of the model describing the experiment. Therefore, comparing the estimated parameters and variables with the true values is not a feasible way to evaluate the accuracy of the estimated quantities.

The value of the cost function at the optimum $C(\hat{\boldsymbol{y}}(t_m),\hat{\boldsymbol{p}},\hat{\boldsymbol{q}})$ has only limited significance for the accuracy of the estimated quantities. To illustrate this fact, we shall consider now two examples based on noise-free time series from the Colpitts oscillator. Equation (18) was integrated (Runge-Kutta integration scheme, time step 0.1), which gave $\boldsymbol{x}(t_n)$ with $t_n \in \mathcal{T}_M = \{0,0.1,\ldots,100\}$. The (observed) time series that is used as input for state and parameter estimation is given by the first state variable,

$$\eta(t_n) = x_1(t_n)\,. \tag{29}$$

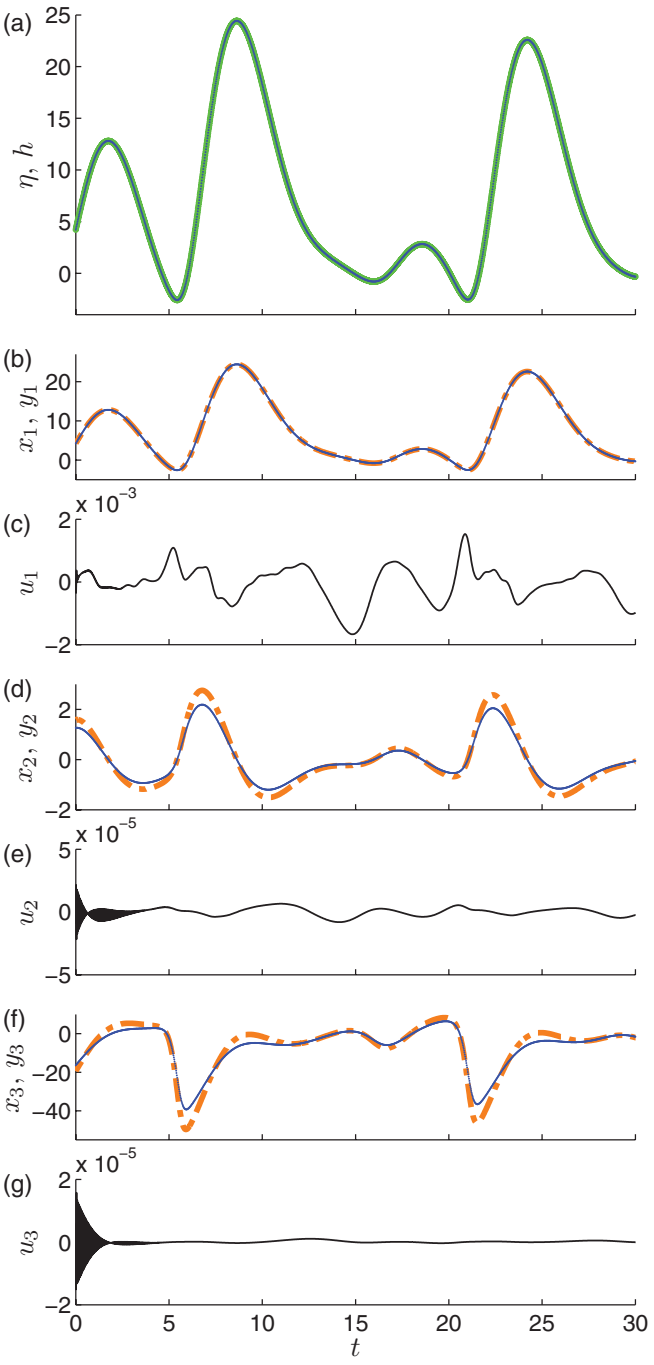

FIG. 5. (Color online) The Colpitts oscillator model (17) is adapted to the time series (29). p_2 and p_3 are fixed to $p_2 = 0.08$ and $p_3 = 0.7$, p_1 and p_4 are estimated in addition to $\boldsymbol{y}(t_m)$ to $\hat{p}_1 = 6.33$ and $\hat{p}_4 = 4.97$ [$p_1 = 5$ and $p_3 = 6.3$ are used to integrate (18)]. The meaning of the line styles and colors is the same as in Fig. 2.

The model ODE's are again given by Eqs. (17), but in contrast to Sec. III B the measurement function is now a function of y_1 only,

$$h[\boldsymbol{y}(t)] = y_1(t)\,. \tag{30}$$

$\boldsymbol{y}(t_m)$ will be estimated at $t_m \in \mathcal{T}_S = \mathcal{T}_M$ and hence at the same times at which observations are available. Due to the absence of any noise in $\eta(t_n)$, there must exist a solution $\{\hat{\boldsymbol{y}}(t_m)\}$, $\hat{\boldsymbol{p}}$,

and $\hat{\boldsymbol{q}}$, where $C(\{\hat{\boldsymbol{y}}(t_m)\},\hat{\boldsymbol{p}},\hat{\boldsymbol{q}})$ is zero. Such a solution is given, for example, by $\boldsymbol{p} = (5,0.08,0.7,6.3)$, where the model (17) is identical with Eqs. (18), which were used to generate $\eta(t_n)$.

We shall now consider two cases. In the first example, the parameters p_2 and p_3 are fixed to $p_2 = 0.08$ and $p_3 = 0.7$, i.e., the same values as used in Eq. (18) to generate $x_1(t)$. The other two parameters, p_1 and p_4, are estimated. Figure 5 shows the results of the estimation procedure. The estimated parameters $\hat{p}_1 = 6.33$ and $\hat{p}_4 = 4.97$ are *not* identical with the values $p_1 = 5$ and $p_4 = 6.3$ used to generate the time series $\eta(t_n) = x_1(t_n)$, although $x_1(t_n)$ and $y_1(t_n)$ coincide very well and $\boldsymbol{u}(t_m)$ is almost zero (i.e., the model ODE's are fulfilled very well). Furthermore, $x_2(t_n)$ and $x_3(t_n)$ are not identical with $y_2(t_n)$ and $y_3(t_n)$, respectively.

In the second example, we repeat the estimation with fixed $p_1 = 5$ and $p_4 = 6.3$ and estimate p_2 and p_3. In this case, the estimated parameters $\hat{p}_2 = 0.0799$ and $\hat{p}_3 = 0.699$ are (almost) identical with the values $p_2 = 0.08$ and $p_3 = 0.7$ used to generate the time series (29), and $\boldsymbol{x}(t_n)$ and $\boldsymbol{y}(t_n)$ are also identical (not shown here). These examples show that in general, it is possible that the measured time series does not contain enough information to estimate all desired parameters adequately.

Whether parameters (and variables) are *observable* [40] depends on the shape of the cost function [41]. To illustrate this aspect for the current examples, we shall now consider simulations in which all four parameters are fixed to certain values and only the state variables are estimated with $\alpha = 0.5$ In the first example, p_1 and p_4 take values on a grid within ranges $2 \leqslant p_1 \leqslant 10$ and $2 \leqslant p_4 \leqslant 10$ for fixed values $p_2 = 0.08$ and $p_3 = 0.7$. For each (p_1,p_4) on the grid, the cost function was minimized, and the minimal vales is denoted as $C(\hat{\boldsymbol{y}}(t_m),(p_1,p_4))$. More precisely, for each pair (p_1,p_4) of parameter values, the optimal values $\hat{\boldsymbol{y}}(t_m)$ of the variables are determined providing $C(\hat{\boldsymbol{y}}(t_m),(p_1,p_4))$.

Figure 6(a) shows the dependence of the cost function $C(\hat{\boldsymbol{y}}(t_m),(p_1,p_4))$ on (p_1,p_4) in terms of contour lines. One can see that in the middle there exists a long, thin expanded area where the cost function is (almost) zero, $C(\hat{\boldsymbol{y}}(t_m),p_1,p_4) < 0.003$. This area will be denoted as the set

$$A_C = \{(p_1,p_4) \,|\, C(\hat{\boldsymbol{y}}(t_m),(p_1,p_4)) \text{ is minimal}\}. \quad (31)$$

Here the minimum is not clearly located at a single point, but there exists a direction where the cost function stays (almost) constant and does not increase significantly. This means that for all parameter sets of $(p_1,p_4) \in A_C$, $h[\boldsymbol{y}(t_n)]$ is (almost) identical with $\eta(t_n)$ *and* the residue $\boldsymbol{u}(t_m)$ is (almost) zero, i.e., the model perfectly describes the time series $\eta(t_n)$. So if one would estimate (p_1,p_4) (in addition to the model variables), the optimizer might stop at different $(\hat{p}_1,\hat{p}_4) \in A_C$ in different runs (for example, with different initial conditions). Of course, for the different $(\hat{p}_1,\hat{p}_4)$, usually the unobserved variables $x_2(t_n)$ and $x_3(t_n)$ will *not* be identical to the estimates $\hat{y}_2(t_n)$ and $\hat{y}_3(t_n)$. For comparison, the same experiment was done with fixed p_1 and p_4 ($p_1 = 5$, $p_4 = 6.3$), and now p_2 and p_3 are varied [Fig. 6(b)]. Here we can see a sharp minimum with a cost function that is (almost) uniformly increasing in all directions. This means that p_2 and p_3 can be estimated simultaneously, or in other words p_2 and p_3 are observable using the given time series.

The time series used for the estimates shown in Figs. 6(a) and 6(b) is sampled after transients decayed and represents the dynamics on the attractor. Additional information about the system can in principle be obtained from (transient) trajectories and resulting time series. Therefore, we have repeated the same computation with a time series including transient dynamics. This time series was generated using a trajectory starting from initial conditions (44.22, 25.59, −69.46) far off the attractor, and its first 30%;–40%; samples are clearly visibly (macroscopic) transient. The resulting graphs of the cost functions are shown in Figs. 6(c) and 6(d) and possess very similar shapes compared to Figs. 6(a) and 6(b). In particular, the long valley in the p_1-p_4 plane still exists. This result indicates that including transients does not provide a means to cope with (efficiently) unobservable parameters here.

For the diagrams shown in Fig. 6, we used $\alpha = 0.5$ and zero initial values for all variables $\boldsymbol{y}(t_n)$ of the trajectory to be reconstructed. We also repeated the computation of the cost function landscape for smaller values of the homotopy parameter α (e.g., $\alpha = 0.1$). In that case, plateaus at relatively high values of the cost function occurred in some parts of the parameter plane due to convergence of the variables $\boldsymbol{y}(t_n)$ toward coexisting local minima. This observation was confirmed by simulations with different initial conditions that yielded similar but different plateau structures. Starting with larger values of α and solving a sequence of optimization problems with decreasing α values (see Sec. II B) turned out to be a suitable approach to avoid convergence to local minima (corresponding to poor estimates).

One way to obtain information about the uniqueness of the U estimated parameters is to consider the form of the graph of the function $C(\hat{\boldsymbol{y}}(t_m),\hat{\boldsymbol{p}})$ at the estimated parameters $\hat{\boldsymbol{p}}$. This shape is characterized by the Hessian matrix given by the second derivatives of the cost function with respect to the parameters

$$H_{ij}(\hat{\boldsymbol{p}}) = \left.\frac{\partial C(\hat{\boldsymbol{y}}(t_m),\boldsymbol{p})}{\partial p_i \partial p_j}\right|_{\boldsymbol{p}=\hat{\boldsymbol{p}}}. \quad (32)$$

The Hessian matrix provides information whether the minimum is flat (small second derivative) or unique (large second derivative). An eigenvalue analysis $\boldsymbol{H}\boldsymbol{h}_k = \lambda_k \boldsymbol{h}_k$ ($k = 1,\ldots,U$) gives the eigenvectors $\boldsymbol{h}_k$ with the corresponding eigenvalues λ_k. Due to the symmetry $H_{ij} = H_{ji}$, all eigenvectors are orthogonal and the corresponding eigenvalues are real. The eigenvector with the smallest eigenvalue points in the direction with the smallest slope. The smaller the smallest eigenvalue is, the less accurate the parameter estimation would be. In Fig. 6, the eigenvectors of the Hessian of the corresponding cost function are shown where eigenvalues have been ordered with increasing magnitude ($\lambda_1 \leqslant \lambda_2 \leqslant \cdots$), i.e., eigenvector $\boldsymbol{h}_1$ points in the direction with the smallest second derivative. For the first example shown in Fig. 6(a), the smallest eigenvalue is $\lambda_1 \approx 0$ and therefore attempts to simultaneously estimate p_1 and p_4 may not give results coinciding with the parameter values used to generate the data because the minimum is flat at least in one direction. On the other hand, simultaneously estimating p_2 and p_3 [second example, Fig. 6(b)] provides the expected results because the smallest

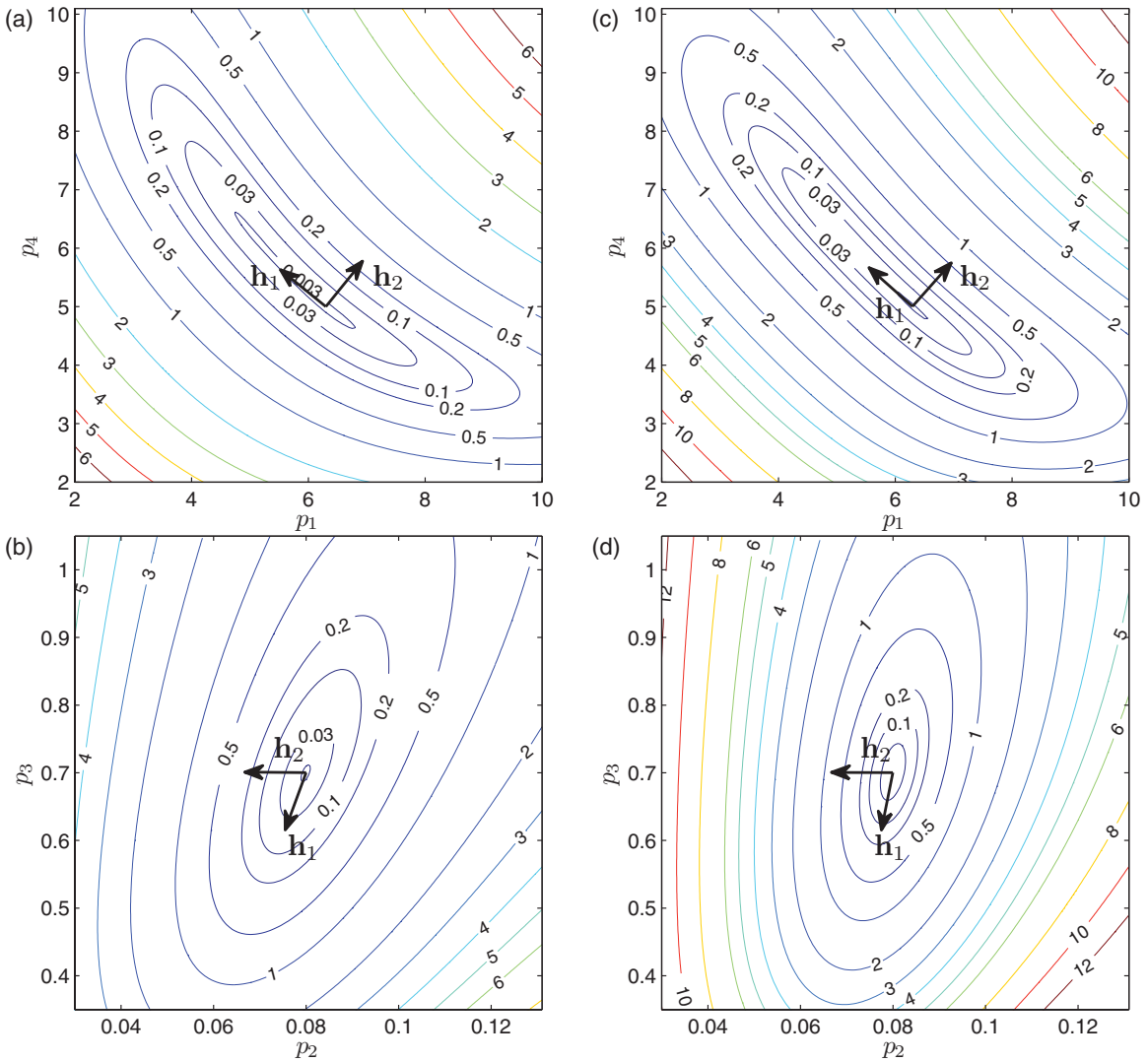

FIG. 6. (Color online) Contour lines of the minimized cost function (12) of the Colpitts oscillator model (17) when adapted to an x_1 time series (29) after transients decayed [(a), (b)] and including transient [(c), (d)]. In (a) and (c), the parameters p_2 and p_3 are fixed to $p_2 = 0.08$ and $p_3 = 0.7$ and the cost function is optimized for different combinations of p_1 and p_4 values resulting in a function $C(\hat{\mathbf{y}}(t_m), p_1, p_4)$ (visualized by means of contour lines). The eigenvectors of the Hessian at the optimum $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_4) = (6.29, 5.00)$ are $\boldsymbol{h}_1 = (-0.780, 0.63)^{\mathrm{T}}$ and $\boldsymbol{h}_2 = (0.63, 0.780)^{\mathrm{T}}$ with eigenvalues $\lambda_1 = 4.5 \times 10^{-3}$ and $\lambda_2 = 0.547$, respectively. In (b) and (d), parameters p_1 and p_4 are fixed to $p_1 = 5$ and $p_4 = 6.3$, the cost function is minimized on a grid of p_2-p_3 values, and the contour plot shows $C(\hat{\mathbf{y}}(t_m), p_2, p_3)$. The minimum is located at $\hat{\boldsymbol{p}} = (\hat{p}_2, \hat{p}_3) = (0.080, 0.70)$ and the eigenvectors of the Hessian at $\hat{\boldsymbol{p}}$ are $\boldsymbol{h}_1 = (-0.0534, -0.9986)^{\mathrm{T}}$ and $\boldsymbol{h}_2 = (-0.9986, 0.0534)^{\mathrm{T}}$ with eigenvalues $\lambda_1 = 10.4$ and $\lambda_2 = 2.56 \times 10^3$, respectively. Note that $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ do not appear orthogonal in the plots due to the different scaling of the axes.

eigenvalue $\lambda_1 = 10.4$ is relatively large and the minimum of the cost function is uniquely defined (not a valley).

Although the shapes of the cost functions shown in Figs. 6(a) and 6(b) are different, both cost functions are smooth and exhibit no local minima close to the optimum. These features are very advantageous for the optimization process.

B. Selecting the continuation parameter

Numerical simulations showed that the shape of the cost function (given by the Hessian matrix) is very robust with respect to noise in the data and variations of the continuation parameter α. Therefore, it cannot be employed for selecting "optimal" values of the (meta) parameter α. In order to find a suitable criterion for stopping the lowering of α mentioned in Sec. II B, we propose to split the observed time series into several segments and to monitor the variance of the parameter values estimated for the different segments. Figure 7 illustrates this approach for the Colpitts example where the given time series was split into 15 segments, and for each segment parameters and variables have been estimated.

In Figs. 7(a), 7(c), and 7(e) for 15 segments of the time series the standard deviations of estimated parameters p_1, p_2, and p_3 are given as a function of the continuation parameter α for different noise levels and $B = (1, 1000, 1000)$. With increasing noise amplitude, the minimum of the standard deviation moves toward smaller values of α (i.e., deviations from the given

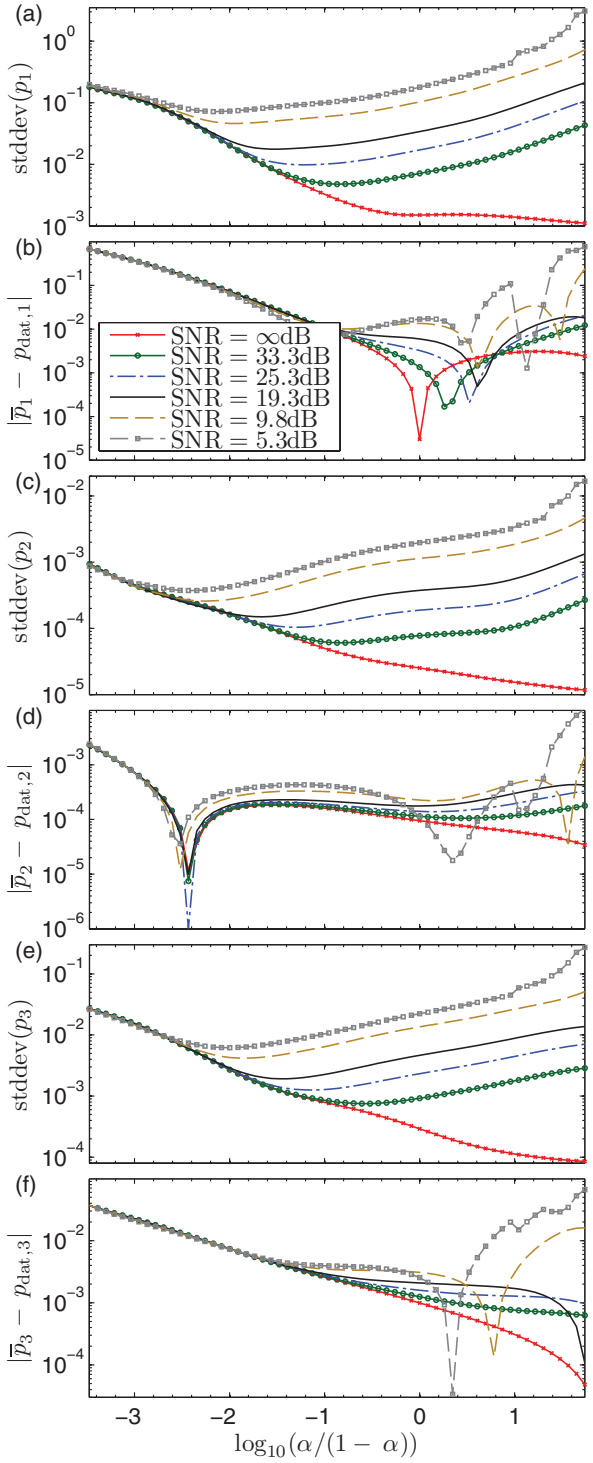

FIG. 7. (Color online) Parameter estimation results for the Colpitts oscillator (17) based on noisy observed x_1 time series (29) consisting of $15.150 = 2250$ samples. Six noise levels ranging from the noiseless case (∞db) to 5.3 dB are considered as indicated in the legend in (b) that holds for all subfigures. The given time series was split into 15 parts, and for each segment parameters have been individually estimated. (a), (c), (e) Variance of the 15 estimated parameters logarithmically plotted vs. $\log_{10}(\alpha/(1-\alpha))$, where α denotes the continuation parameter. (b), (d), (f) Deviations of the mean values of the estimated parameters p_1, p_2, and p_3 from the values used for generating the data.

noisy time series are weighted less compared to deviations from model ODE's). For comparison, Figs. 7(b), 7(d), and 7(f) show the absolute value of the difference between the estimated parameters from the true values (used upon time-series generation). For parameter p_2, minima of the standard deviations coincide with minima of the parameter estimation errors. Unfortunately, this is not always the case, as can be seen in diagrams for p_1 and p_3. Still, choosing a value of the continuation parameter α where the standard deviation of multiple parameter estimations is small seems to be a useful criterion.

C. Evaluating the consistency of the model and the data

To evaluate the suitability of the chosen model for describing the given data, we follow Abarbanel *et al.* [18] and define a relative measure of the deviation of the vector field (or ODE) components from the (approximated) derivatives [see Eq. (5)], which is given by

$$R_i^2(t_m) = \frac{F_i^2(t_m)}{F_i^2(t_m) + \left[\frac{\Delta y_i}{\Delta t}\Big|_{t_m} - F_i(t_m)\right]^2}, \tag{33}$$

where $F_i(t_m)$ is an abbreviation for $F_i(\boldsymbol{y}(t_m), \boldsymbol{p}, t_m)$. If the model is consistent with the data, the characteristics R_i will tend to 1, whereas low values of R_i indicate a poor agreement of the ith component of the vector field $\mathbf{F}$ with the reconstructed trajectory. The temporal evolution of $R_i(t)$ may additionally provide information where in state space the data are (not) well explained by the model. Here, we shall consider temporal averages only, providing the consistency measures

$$\bar{R}_i = \frac{1}{M}\sum_{m=1}^{M} R_i(t_m) \tag{34}$$

for the individual model ODE's. Averaging these characteristics yields a consistency measure for the full model,

$$\bar{R} = \frac{1}{D}\sum_{i=1}^{D} \bar{R}_i. \tag{35}$$

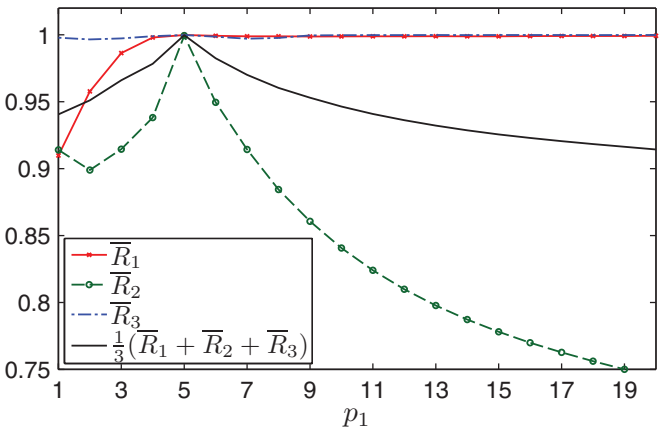

FIG. 8. (Color online) Consistency measures vs fixed (detuned) parameter p_1 of the Colpitts oscillator (17). State variables and parameters p_2, p_3, and p_4 are estimated. A high consistency of model and data is indicated by R values close to 1. The indicator $\bar{R}$ given by Eq. (35) uniquely selects the value $p_1 = 5$ that was also used for generating the (observed) time series.

To demonstrate possible applications of this measure, we applied it to parameter estimations with a Colpitts oscillator (17) where parameter p_1 was "detuned" from the true value $p_1 = 5$. Figure 8 shows $\bar{R}_i$ $(i = 1,2,3)$ and $\bar{R}$ versus p_1 for $\alpha = 0.1$ and $B = (1,1,1)$. Only at the true value $p_1 = 5$ are all three individual measures $\bar{R}_1$, $\bar{R}_2$, and $\bar{R}_3$ close to 1, and therefore the consistency measure $\bar{R}$ for the full model uniquely selects the proper value for p_1.

V. CONCLUSION

The task of state and parameter estimation for a given set of model ODE's using an observed time series was revisited and solved using an efficient unconstrained optimization method exploiting the particular structure of the cost function. This algorithm was successfully applied to numerical examples, including a hyperchaotic system and (spiking) dynamics on different time scales. Although convergence is achieved typically within a few seconds of CPU time, the estimated parameter values do not always coincide with those used to generate the (synthetic) time series serving as input to the estimation process. Measurement noise may introduce a bias, although in the examples considered here, the impact of stochastic components was relatively small. More important seem to be features of the model and its functional dependence of parameters. If in a multiparameter estimation problem several combinations of parameter values provide (almost) the same dynamical output, then these parameters are redundant and (individually) not observable. This phenomenon was discussed and illustrated for the Colpitts oscillator. Typical indicators are long valleys in the shape of the cost function without a pronounced minimum. To detect such cases, the eigenvalues of the Hessian matrix (given by the second derivatives of the cost function with respect to parameters) can be used, because small eigenvalues correspond to eigendirections where the cost function is poorly localized. Another important task is the proper choice of meta parameters of the algorithm, such as the continuation parameter α controlling the relative weights of deviations from the observed time series and from the model ODE's, respectively. As a useful criterion, we suggested splitting the available time series into segments and select an α that minimizes the standard deviation of parameter values estimated with different segments. Finally, a characteristic was presented that may be used to evaluate the consistency of the obtained model with the given data.

Comparison with other estimation methods [18,25] indicates that the algorithm presented here is highly competitive [42].

ACKNOWLEDGMENTS

The research leading to the results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement No. HEALTH-F2-2009-241526, EUTrigTreat. We thank H. D. I. Abarbanel, J. Bröcker, and S. Luther for interesting and inspiring discussions on data assimilation and continuous support.

APPENDIX: APPROXIMATION OF DERIVATIVES

In the cost function (12), the temporal derivatives of the variables $\boldsymbol{y}(t_m)$ are needed. A numerically efficient way to approximate the derivatives at times $t_m = m\Delta t$ $(m = 3,\ldots,M-2)$ (for a given time step Δt) is given by the central difference

$$\left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_m} = \frac{-\boldsymbol{y}(t_{m+2}) + 8\boldsymbol{y}(t_{m+1}) - 8\boldsymbol{y}(t_{m-1}) + \boldsymbol{y}(t_{m-2})}{12\Delta t} + O(\Delta t^4)\,. \tag{A1}$$

For $m = 2$ and $m = M-1$, the central difference

$$\left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_m} = \frac{\boldsymbol{y}(t_{m+1}) - \boldsymbol{y}(t_{m-1})}{2\Delta t} + O(\Delta t^2) \tag{A2}$$

is used, and for $m = 1$ and $m = M$ the forward and backward Euler method is employed, respectively, with

$$\left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_1} = \frac{\boldsymbol{y}(t_2) - \boldsymbol{y}(t_1)}{\Delta t} + O(\Delta t) \tag{A3}$$

and

$$\left.\frac{\Delta \boldsymbol{y}}{\Delta t}\right|_{t_M} = \frac{\boldsymbol{y}(t_M) - \boldsymbol{y}(t_{M-1})}{\Delta t} + O(\Delta t)\,. \tag{A4}$$

Note the error of order $O(\Delta t^4)$ in the approximation by the central difference in Eq. (A1). This accuracy is valid at all times t_m except the two first and last.

[1] U. Parlitz, L. Junge, and L. Kocarev, Phys. Rev. E **54**, 6253 (1996).
[2] D. Ghosh and S. Banerjee, Phys. Rev. E **78**, 056211 (2008).
[3] H. D. I. Abarbanel, D. R. Creveling, and J. M. Jeanne, Phys. Rev. E **77**, 016208 (2008).
[4] F. Sorrentino and E. Ott, Chaos **19**, 033108 (2009).
[5] I. G. Szendro, M. A. Rodríguez, and J. M. López, J. Geophys. Res. **114**, D20109 (2009).
[6] R. Konnur, Phys. Lett. A **346**, 275 (2005).
[7] U. S. Freitas, E. E. N. Macau, and C. Grebogi, Phys. Rev. E **71**, 047203 (2005).
[8] M. Chen and J. Kurths, Phys. Rev. E **76**, 027203 (2007).
[9] P. H. Bryant, Phys. Rev. E **82**, 015201(R) (2010).
[10] W. Yu, G. Chen, J. Cao, J. Lü, and U. Parlitz, Phys. Rev. E **75**, 067201 (2007).
[11] D. Yu and U. Parlitz, Phys. Rev. E **77**, 066221 (2008).
[12] D. Ghosh, Europhys. Lett. **84**, 40012 (2008).
[13] D. Fairhurst, I. Tyukin, H. Nijmeijer, and C. van Leeuwen, Math. Model. Nat. Phenom. **5**, 146 (2010).
[14] D. Huang, G. Xing, and D. W. Wheeler, Chaos **17**, 023118 (2007).
[15] R. E. Amritkar, Phys. Rev. E **80**, 047202 (2009).
[16] D. R. Creveling, P. E. Gill, and H. D. I. Abarbanel, Phys. Lett. A **372**, 2640 (2008).

[17] J. C. Quinn, P. H. Bryant, D. R. Creveling, S. R. Klein, and H. D. I. Abarbanel, Phys. Rev. E **80**, 016201 (2009).
[18] H. D. I. Abarbanel, D. R. Creveling, R. Farsian, and M. Kostuk, SIAM J. Appl. Dyn. Syst. **8**, 1341 (2009).
[19] J. Bröcker and U. Parlitz, Chaos **11**, 319 (2001).
[20] D. Ridout and K. Judd, Physica D **165**, 26 (2002).
[21] J. Bröcker, D. Engster, and U. Parlitz, Chaos **19**, 043130 (2009).
[22] L. A. Smith, M. C. Cuellar, H. Du, and K. Judd, Phys. Lett. A **374**, 2618 (2010).
[23] J. C. Quinn and H. D. I. Abarbanel, Q. J. R. Meteorol. Soc. **136**, 1855 (2010).
[24] H. D. I. Abarbanel, M. Kostuk, and W. Whartenby, Q. J. R. Meteorol. Soc. **136**, 769 (2010).
[25] J. Bröcker, Q. J. R. Meteorol. Soc. **136**, 1906 (2010).
[26] J. Bröcker and I. Szendro, doi: 10.1002/qj.940.
[27] In principle, the presented approach could also be used to estimate parameters of black-box models.
[28] A. Wächter and L. T. Biegler, Math. Program. **106**, 25 (2006).
[29] P. E. Gill, W. Murray, and M. A. Saunders, SIAM J. Optim. **12**, 979 (2002).
[30] For example, due to nonuniform sampling in time it may be necessary to interpolate the model variables to obtain model output $\{\boldsymbol{z}(t_n)\}$ at times $t_n \in \mathcal{T}_M$, where observations $\{\boldsymbol{\eta}(t_n)\}$ are available. Since derivatives are known, Hermite interpolation (7) can be used to approximate $\boldsymbol{y}(t_n)$, which is needed to compute $\{\boldsymbol{z}(t_n)\} = \boldsymbol{h}(\boldsymbol{y}(t_n),\boldsymbol{q},t_n)$.
[31] K. Levenberg, Q. Appl. Math. **2**, 164 (1944).
[32] D. Marquardt, SIAM J. Appl. Math. **11**, 431 (1963).
[33] M. I. A. Lourakis, European Conference on Computer Vision **6312**, 43 (2010), http://dx.doi.org/10.1007/978-3-642-15552-9_4, http://www.ics.forth.gr/∼lourakis/sparseLM/.
[34] Due to the special structure of the cost function, its Jacobian matrix is sparse, i.e., it contains many zeros. sparseLM [33] was developed for problems with a sparse Jacobian and uses internal matrix operations that are optimized for sparse matrices.
[35] M. P. Kennedy, IEEE Trans. Circuits Syst. I **41**, 771 (1994).
[36] J. L. Hindmarsh and R. M. Rose, Proc. R. Soc. London, Ser. B **221**, 87 (1984).
[37] X.-J. Wang, Physica D **62**, 263 (1993).
[38] O. E. Rössler, Phys. Lett. A **57**, 397 (1976).
[39] G. Baier and S. Sahle, Phys. Rev. E **51**, 2712(R) (1995).
[40] H. U. Voss, J. Timmer, and J. Kurths, Int. J. Bif. Chaos **14**, 1905 (2004).
[41] A. Raue, V. Becker, U. Klingmüller, and J. Timmer, Chaos **20**, 045105 (2010).
[42] J. Schumann-Bischoff and U. Parlitz (unpublished).