On noise reduction methods for chaotic data

Peter Grassberger, Rainer Hegger, Holger Kantz, Carsten Schaffrath, and Thomas Schreiber[a)]
Physics Department, University of Wuppertal, Gauss-Strasse 20, D-5600 Wuppertal 1, Germany

(Received 11 September 1992; accepted for publication 18 January 1993)

Recently proposed noise reduction methods for nonlinear chaotic time sequences with additive noise are analyzed and generalized. All these methods have in common that they work iteratively, and that in each step of the iteration the noise is suppressed by requiring locally linear relations among the delay coordinates, i.e., by moving the delay vectors towards some smooth manifold. The different methods can be compared unambiguously in the case of strictly hyperbolic systems corrupted by measurement noise of infinitesimally low level. It was found that all proposed methods converge in this ideal case, but not equally fast. Different problems arise if the system is not hyperbolic, and at higher noise levels. A new scheme which seems to avoid most of these problems is proposed and tested, and seems to give the best noise reduction so far. Moreover, large improvements are possible within the new scheme and the previous schemes if their parameters are not kept fixed during the iteration, and if corrections are included which take into account the curvature of the attracting manifold. Finally, the fact that comparison with simple low-pass filters tends to overestimate the relative achievements of these nonlinear noise reduction schemes is stressed, and it is suggested that they should be compared to Wiener-type filters.

I. INTRODUCTION

Recently, there has been an increased interest in nonlinear time series analysis (see, e.g., Ref. 1 for a review). In particular, algorithms have been given for constructing dynamical models generating measured time sequences,[2,3] and for forecasting such sequences.[4–9] Closely related to these is the problem of noise reduction.

Any noise reduction method must assume that the time series to be cleaned can be unambigously seperated into "noise" and "signal" on the basis of some objective criterion. Conventional methods like linear filters use for this the spectrum: Low- or high-pass filters assume that the signal has much lower resp. much higher typical frequencies. Alternatively, if such a distinction on the basis of characteristic time scales is not possible, broad band components of the spectra are assumed to be related to noise, while sharp lines are due to the signal. This is fine for signals from regular sources which give periodic or quasiperiodic signals, but if the signal is not (quasi-) periodic, linear filters cannot remove the noise without also distorting the signal. This is due to the fact that already pure signals from chaotic systems show broad band spectra, and no Fourier-based method is able to distinguish this from random noise. For the latter, one has somehow to use the fact that deterministically chaotic (and dissipative) motion takes place on attractors which are smooth submanifolds of the total available phase space. This implies that state vectors constructed from delay variables are constrained to fall onto geometrical objects which are locally linear.

When adding noise to an otherwise deterministic system, we have to distinguish between "dynamic" and "measurement" noise. Assume that the noise-free dynamics would be

$$\mathbf{y}_{n+1}=\mathbf{f}(\mathbf{y}_n). \tag{1.1}$$

We speak of measurement noise if there exists a trajectory satisfying this exact dynamics, but the measured trajectory is corrupted by additive (or possibly multiplicative) noise,

$$\mathbf{x}_n=\mathbf{y}_n+\mathbf{r}_n. \tag{1.2}$$

Dynamic noise, in contrast, is added already during the evolution,

$$\mathbf{x}_{n+1}=\mathbf{f}(\mathbf{x}_n)+\mathbf{r}_n \tag{1.3}$$

so that no near-by trajectory satisfying the exact dynamics needs to exist *a priori*. The "shadowing problem" deals just with the question whether such a trajectory does exist nevertheless, and how to find it.

Obviously the shadowing problem is harder than the problem of removing measurement noise. For a noise reduction scheme to be practically useful, one should demand that it works at least in the latter case, and that it at least does not produce nonsense in the case of dynamic noise. In particular, this is claimed for the noise reduction schemes recently proposed by several authors.[10–16] All these methods are supposed to work also when the data are sampled with too low frequency for conventional low-pass or Wiener[17] filters.

Let us assume that we have observed only a univariate time series corrupted by measurement noise, and construct state vectors by means of delay coordinates. Then the observed signal x_n is obtained from a "true" trajectory y_n by

$$x_n=y_n+r_n. \tag{1.4}$$

[a)]Current address: Niels Bohr Institute, Blegdamsvej 17, DK-2100 Copenhagen ø, Denmark.

1054-1500/93/020127-15$06.00

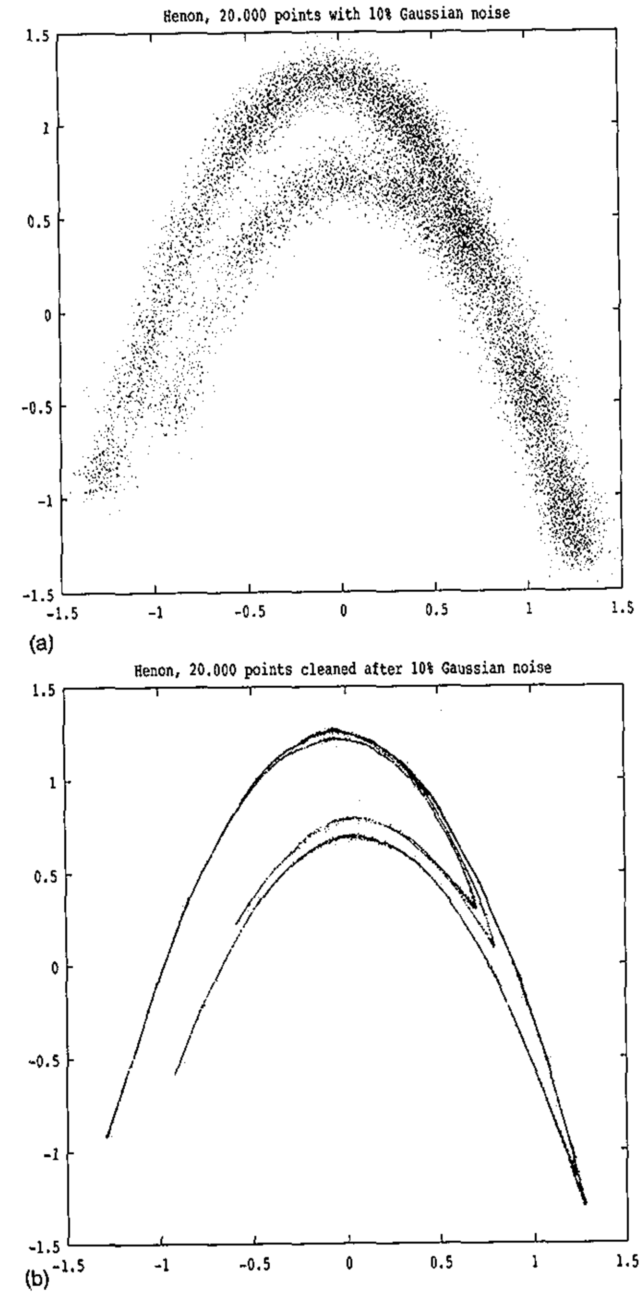

FIG. 1. Hénon attractor with 10% Gaussian white noise before (panel a) and after (panel b) noise reduction. Here, the exact form of the Hénon map was not assumed to be known, but was fitted from the noisy time series of 20 000 points shown in panel (a). The noise reduction algorithm used was that described in Secs. IV and V.

Let m be a positive integer. If it is sufficiently much larger than the attractor dimension,[18] then an m-dimensional embedding will give in general a faithful representation of it, and Eq. (1.1) can be written either as

$$y_{n+m}=f(y_n,y_{n+1},\dots,y_{n+m-1}) \qquad (1.5)$$

or implicitly as

$$F(y_n,y_{n+1},\dots,y_{n+m})=0. \qquad (1.6)$$

The embedding dimension m will however not be known *a priori*, and for practical reasons it will often be convenient to use an embedding dimension larger than strictly needed. In this case, there will not be a single relation of this type, but several constraints

$$F_q(y_n,y_{n+1},\dots,y_{n+m})=0, \quad q=1,2,\dots,Q\leqslant m, \qquad (1.7)$$

where $m-Q+1$ is the dimension of the manifold.

Most of the above authors assume that the true dynamics is sufficiently smooth so that the functions f and F can be approximated locally by linear functions. Thus, for points $\mathbf{y}_k=(y_k,\dots,y_{k+m})$ in the neighborhood of the point $\mathbf{x}_n=(x_n,x_{n+1},\dots,x_{n+m})$ [Notice that we change here the notation with respect to Eqs. (1.1)–(1.3). While there $\mathbf{x}_n$ was an m dimensional vector, now and in the following it will be $m+1$ dimensional.] they assume for the noise-free system either a single constraint

$$F^{(n)}(\mathbf{y}_k)=\sum_{i=0}^{m} a_i^{(n)} y_{k+i}+b^{(n)}=0 \qquad (1.8)$$

or, if the embedding dimension m is larger than the minimally necessary one, a set $\{F_q^{(n)}\,|\,q=1,2,\dots,Q\}$ of such constraints. In the former case, the point is confined to a hyperplane, in the latter it is in a lower-dimensional linear subspace.

The most straightforward attempt to use Eq. (1.8) would be to write it in an explicit "forward time" form, and to define a "corrected" time sequence as

$$x_{n+m}^{\mathrm{corr}}=\frac{1}{a_m^{(n)}}\left[\sum_{i=0}^{m-1} a_i^{(n)} x_{n+i}+b^{(n)}\right]. \qquad (1.9)$$

After this has been done for all n, we would replace x_n by the new x_n^{corr}, and repeat the procedure. Thus we would obtain an iterative scheme which hopefully would converge towards the correct trajectory y_n.

Unfortunately, this would not work for a chaotic system, since the "correction" would be ill defined along the unstable manifold. Any error in x_0 would induce a bigger error in x_1, etc. In order to stabilize the corrections in the direction of the unstable manifold, we must also use information from the future.

In Refs. 10–12 this is done in two separate steps. In the first step, for each point the local map (1.8) or a higher order polynomial is constructed. In the second step, a new trajectory is found which on the one hand satisfies this map (at least approximately) and on the other hand is close to $\{x_n\}$. In all three papers this is implemented differently, but we do not want to discuss the details here. We just point out that the second step is the crucial one and numerically nontrivial in all three approaches.

In contrast to this, the authors of Refs. 13–16 virtually eliminate the second step by essentially projecting (in different ways) the vectors $\mathbf{x}_n$ onto the subspaces defined by the constraints $F_q^{(n)}=0$. Actually the latter is not precisely what they do. Since the delay vectors $\mathbf{x}_n$ for different n are not independent (they involve partly the same elements of the time series), projecting all of them exactly onto their hyperspaces would in general be mutually inconsistent. Therefore one has to find a compromise between the dif-

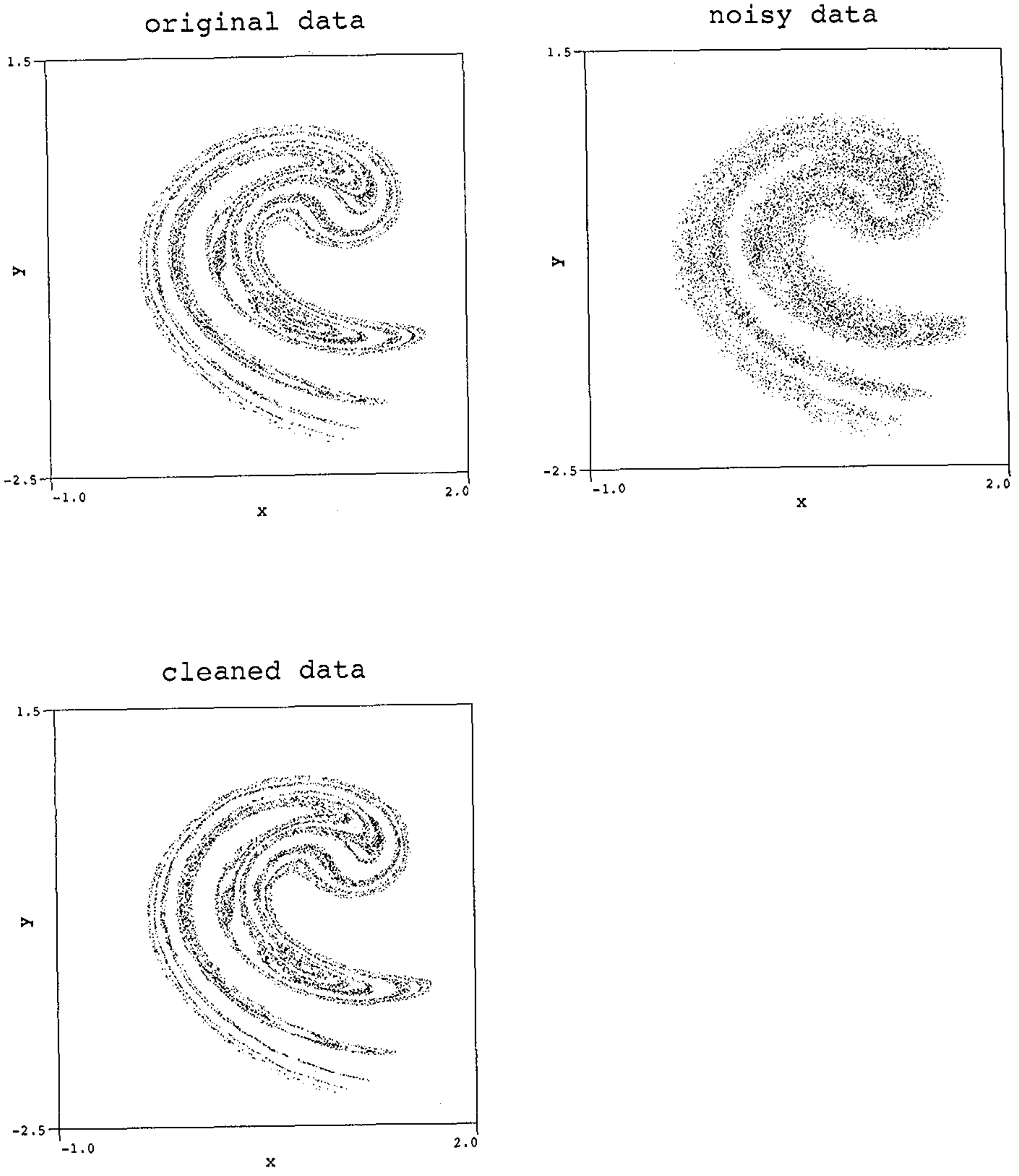

FIG. 2. Ikeda attractor $z' = 1 + 0.9ze^{0.4i-6i/(1+|z|^2)}$: clean (panel a), with 5% uniform white noise (panel b), and after noise reduction (panel c) using again only information from the noisy signal. The algorithm used here was indeed somewhat simpler than in Fig. 1, and used the two-variate time series of points $z_n = x_n + iy_n$ as described in Ref. 13.

ferent projections. Since the results of Refs. 13–16 are quite promising, in the following we shall concentrate on this class of noise reduction schemes.

In Sec. II, we shall analyze in more detail the method proposed independently by Cawley and Hsu[14,15] and by Sauer.[16] In Sec. III, we shall discuss that by Schreiber and Grassberger,[13] and in Sec. IV we shall discuss a method which can be understood as a compromise between both, and which we consider as optimal. The methods discussed in Secs. II and III can be obtained as special cases of the latter. Details of the implementation and numerical results are given in Sec. V.

Two typical examples of the performance of our algorithm are shown in Figs. 1 and 2. They used time sequences of 20 000 points of the well known Hénon and Ikeda maps. More details on these figures will also be given in later sections.

Before leaving this introduction, we should however stress that comparisons with linear filtering methods should be done in a fair way, by comparing with *optimal* linear filters. These are Wiener filters[17] and closely related "Wiener-type" filters using singular value decomposition (SVD).[19,20] In general, they can give much better results than the simple low-pass filters used for comparison in previous analyses.[10,13,14] Since they are useful also as preliminary first steps in nonlinear noise reduction methods, we discuss them in Appendix A. Another useful "zeroth order" method with which our results should be compared is obtained by correcting each point towards the center of gravity of its neighbors.[24] Results of this and of SVD-Wiener filters will be discussed also in Sec. V.

II. CAWLEY-HSU-SAUER METHOD

Apart from first applying a low-pass linear filter and some other tricks which we do not want to discuss here in detail (some will be discussed in Sec. V), the noise reduction method used in Refs. 14–16 consists essentially in a compromise between *orthogonal* projections onto the subspaces defined by the linear constraints $F_q^{(n)}=0$.

Let us be more precise and consider the correction of the nth value x_n in the case of embedding dimension m and a single constraint $F=0$. The discussion will be simpler if we assume for the moment that this constraint is not linear and thus only locally valid, but is globally valid and and hence nonlinear. The total time series consists of the T values $(x_1,...,x_T)$. Due to the noise, the measured time sequence does not satisfy Eq. (6), but rather

$$F(\mathbf{x}_j)=F(x_j,x_{j+1},...,x_{j+m})=\epsilon_j\neq 0,\quad 1\leqslant j\leqslant T-m. \tag{2.1}$$

If ϵ_j is small [or $F(\mathbf{x})$ is linear], then an orthogonal projection of $\mathbf{x}_j$ onto the surface $F(\mathbf{x}_j)=0$ would be given by

$$\mathbf{z}_j=\mathbf{x}_j+\boldsymbol{\theta}_j \tag{2.2}$$

with

$$\boldsymbol{\theta}_j=\text{const}\;\nabla_j F, \tag{2.3}$$

where $\nabla_j F=\nabla F(\mathbf{x}_j)$. Inserting this ansatz into Eq. (2.1) and Taylor expanding, we obtain

$$\epsilon_j=F(\mathbf{z}_j-\boldsymbol{\theta}_j)=0-\boldsymbol{\theta}_j\cdot\nabla_j F \tag{2.4}$$

which gives us finally

$$\boldsymbol{\theta}_j=-\epsilon_j\frac{\nabla_j F}{\|\nabla_j F\|^2}. \tag{2.5}$$

If the constraint were satisfied everywhere except for one time j so that all ϵ_j were zero except for one single ϵ_{j_0}, then Eq. (2.2) would give corrections only to x_{j_0}, $x_{j_0+1},...,x_{j_0+m}$. But this is not realistic, and we must expect that the corrections for x_n, say, obtained from projecting onto $F(\mathbf{x}_{n-m})=0$, $F(\mathbf{x}_{n-m+1})=0,...,F(\mathbf{x}_n)=0$ are all different. The natural assumption made in Refs. 16, 14, and 15 is that the finally accepted correction is proportional to the sum of all these corrections,

$$x'_n=x_n+\alpha\sum_{j=n-m}^{n}\theta_{j,n-j}. \tag{2.6}$$

Inserting this into $F(\mathbf{x}_k)$ will give us the new errors $\epsilon'_k=F(\mathbf{x}'_k)$ which are then used to get the second correction x''_k, etc. Again Taylor expanding $F(\mathbf{x}'_k)$, we obtain after some algebra our first result

$$\epsilon'_k=(1-\alpha)\epsilon_k-\alpha\sum_{j\neq k}\epsilon_j\frac{\nabla_k F\cdot\mathbf{B}^{j-k}\nabla_j F}{\|\nabla_j F\|^2}. \tag{2.7}$$

Here $\mathbf{B}$ is the $(m+1)\times(m+1)$ backshift matrix defined as $B_{ij}=\delta_{i,j+1}$. Its nth power is simply $(\mathbf{B}^n)_{ij}=\delta_{i,j+n}$.

Using the normalized vectors

$$\mathbf{e}_k=\frac{\nabla_k F}{\|\nabla_k F\|} \tag{2.8}$$

and normalized errors

$$\eta_k=\frac{\epsilon_k}{\|\nabla_k F\|}, \tag{2.9}$$

Eq. (2.7) can be written as

$$\eta'_k=(1-\alpha)\eta_k-\alpha\sum_{j\neq k}(\mathbf{e}_k\cdot\mathbf{B}^{j-k}\mathbf{e}_j)\eta_j. \tag{2.10}$$

If there are several constraints, then Eq. (2.3) has to be replaced by the ansatz $\boldsymbol{\theta}_j=\Sigma_{q=1}^{Q}\alpha_q\nabla_j F_q$. The generalization of Eq. (2.7) is straightforward but somewhat cumbersome, and will not be given here. For realistic numerical applications where the global constraints F_q are replaced by linear fits which hold only locally, we will not use it anyhow but will use instead the formalism of Sec. IV.

Let us now study in some examples the convergence of the iterative scheme based on Eq. (2.10).

(a) 1-dimensional maps: First we consider 1-d maps $F(y_n,y_{n+1})\equiv y_{n+1}-f(y_n)=0$. Here,

$$\nabla_j F=(-df(x_j)/dx_j,1) \tag{2.11}$$

and Eq. (2.10) reduces to

$$\eta'_k=(1-\alpha)\eta_k+\alpha[\beta_k\eta_{k-1}+\beta_{k+1}\eta_{k+1}] \tag{2.12}$$

with

$$\beta_k=-\mathbf{e}_{k-1}\cdot\mathbf{B}\mathbf{e}_k=\frac{f'_k}{\sqrt{(1+[f'_{k-1}]^2)(1+[f'_k]^2)}},$$

$$f'_k=df(x_k)/dx_k. \tag{2.13}$$

During the iteration of Eq. (2.12), the errors spread essentially like they do in a reactive diffusion process with frozen randomness. For a sawtooth map $f(x)=ax+c$ mod 1

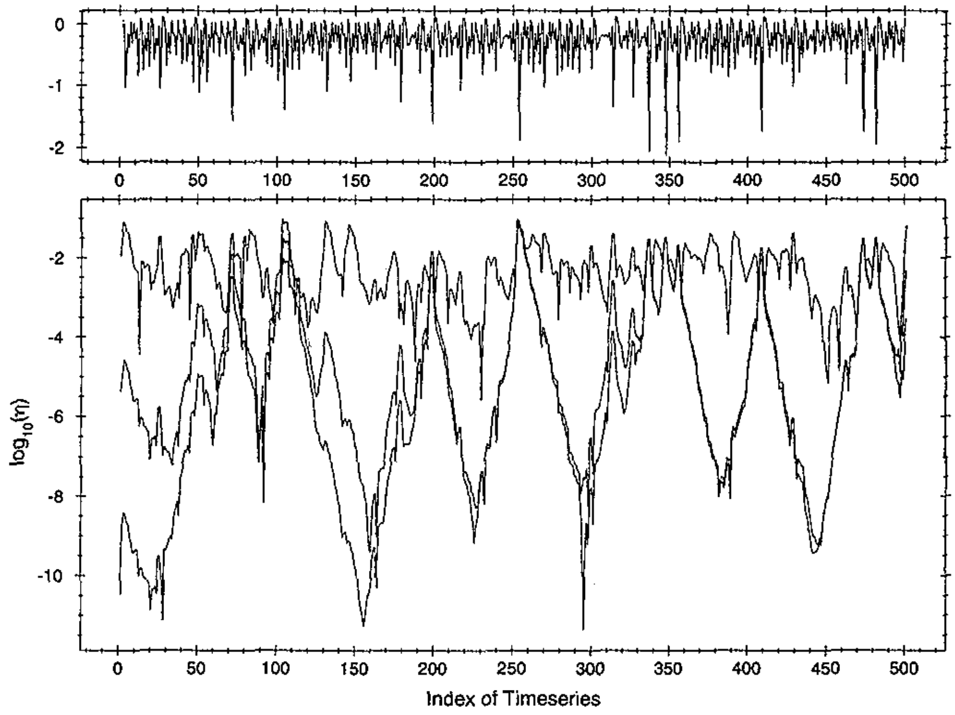

FIG. 3. Lower part: logarithm of η_k plotted against k after 50, 1000, and 2000 iterations of Eq. (2.14) for the Hénon map. The iteration had started with $\eta_k = 1 \forall k$. On the upper part, the distance of (x_k, x_{k+1}) from the nearest primary tangency point as determined in Ref. 22 is plotted on a logarithmic scale. We see that the convergence of the iteration becomes arbitrarily slow near these tangency points.

with $a > 1$, we can easily see that this diffusion process is damped for sufficiently small α, since the largest eigenvalue of Eq. (2.12) is $\mu = 1 - \alpha(a-1)^2/(1+a^2)$. Thus in this case the iteration converges exponentially, though less fast than with the trivial scheme $\eta_k' = (1-\alpha)\eta_k + \alpha\eta_{k+1}/a$ which can be considered as an application of the method discussed in the next section.

For maps with $f'(x) \neq \text{const}$ we cannot in general show convergence analytically, even if they are hyperbolic. But numerical simulations very clearly show that Eq. (2.12) always converged for chaotic maps to $\eta_k = 0$, for sufficiently small α. But for maps with a quadratic critical point (like the logistic map), the convergence is neither uniform nor exponential. More precisely, the convergence seems to be arbitrarily slow if x comes close to a critical point of $f(x)$. The average error seems to decrease only algebraically with the number of iterations.

Thus we conclude that the above represents indeed a viable noise reduction scheme for 1-d maps, though not necessarily a very efficient one.

(b) 2-d maps: This is very similar for 2-d maps $y_{n+2} = f(y_n, y_{n+1})$. There, the rhs of the analogon to Eq. (2.12) contains 4 terms,

$$\eta_k' = (1-\alpha)\eta_k + \alpha[\sigma_k\eta_{k-1} + \sigma_{k+1}\eta_{k+1} + \tau_{k-1}\eta_{k-2} + \tau_{k+1}\eta_{k+2}]. \tag{2.14}$$

Again we can show for maps with constant derivatives,

$$y_{n+2} = a_0 y_n + a_1 y_{n+1} + c \mod c', \tag{2.15}$$

that the iteration converges exponentially with an eigenvalue

$$\mu = 1 - \alpha \frac{(1-a_0-a_1)^2}{1+a_0^2+a_1^2}. \tag{2.16}$$

which is again larger than the analogous eigenvalue of the method discussed in the next section.

But for nonlinear maps like the Hénon[21] map $y_{n+2} = a + by_n - y_{n+1}^2$, exponential convergence seems again to be prevented by homoclinic tangencies. For the Hénon map, we have

$$\sigma_k = -\mathbf{e}_{k-1} \cdot \mathbf{B}\mathbf{e}_k = \frac{2(bx_k - x_{k+1})}{\sqrt{(1+b^2+4x_k^2)(1+b^2+4x_{k+1}^2)}}, \tag{2.17}$$

$$\tau_k = -\mathbf{e}_{k-1} \cdot \mathbf{B}^2\mathbf{e}_{k+1} = \frac{b}{\sqrt{(1+b^2+4x_k^2)(1+b^2+4x_{k+2}^2)}}. \tag{2.18}$$

In simulations for randomly chosen trajectories with the standard parameters $a = 1.4$, $b = 0.3$, Eq. (2.14) never diverged for $\alpha < 0.9$, but convergence was arbitrarily slow when the point (x_n, x_{n+1}) was near the line of "primary" tangencies[22] (see Fig. 3). This agrees with the fact that "realistic" application of the Cawley–Hsu–Sauer method (which involves also local linearizations, use of fitted maps instead of the exact map, and eventually linear prefiltering) always gave somewhat poorer results for the Hénon map[14] than the algorithm discussed in the next section. It seems to result from the fact that also the first and last components of delay vectors are corrected. As we had pointed out in the Introduction, we should expect that these components are hard to correct due to instabilities along the stable resp. unstable manifolds.

(c) Higher dimensions: For this same reason, we should expect also slow convergence when applying the method to higher dimensional systems. But applications of Sauer's algorithm to the Lorenz and Rössler models[16] gave very good results, much better, e.g., than linear filtering and the methods of Refs. 10 and 13.

This might be due to the fact that the time series tested in Ref. 16 were embedded in very high dimensions and measured with rather short delays. This could imply that the final slowness of convergence would become apparent only after very many iterations, much more than needed for realistically acceptable noise reduction. Another reason might be that in autonomous flows (as opposed to maps) there exists always one vanishing Lyapunov exponent. It corresponds to uncertainties in the flow direction, and implies that errors in this direction cannot be corrected very well anyhow. This affects strongly the methods of Refs. 10 and 13 which rely on hyperbolicity, but it might affect much less the Cawley–Hsu–Sauer method.

Thus we conclude that this method is very efficient for oversampled time series, in spite of its theoretical drawbacks. The latter can make it somewhat slower than the method discussed in Sec. III, but they make it also more robust. Indeed, we found that in many cases its performance can be improved by modifications which will be discussed in Sec. V together with other details of the implementation.

III. SCHREIBER–GRASSBERGER METHOD

Instead of projecting *orthogonally* onto the surfaces $F(\mathbf{x}_j)=0$, in Ref. 13 we projected by correcting only a single delay coordinate. In the examples shown in Ref. 13, we used even m, $m=2r$, and corrected the central coordinate x_{j+r}. More generally we write $m=r+s$ and correct x_{j+r}. The coordinates x_j, $x_{j-1},...,x_{j+r-1}$ are called "past coordinates," the coordinates $x_{j+r+1},...,x_{j+m}$ are called "future coordinates."

In this way, we avoided on the one hand the problem of conflicting projections discussed above. On the other hand, since this correction is well controlled both in the stable and unstable directions, we should have faster convergence when dealing with hyperbolic systems. The latter and the fact that the resulting correction is very simple and intuitive,

$$x_n' = x_n - \alpha \frac{F(\mathbf{x}_{n-r})}{\partial F(\mathbf{x}_{n-r})/\partial x_n} \tag{3.1}$$

are the main advantages.

Due to the simplicity of the method, it can also be applied directly to multivariate time sequences.[23] In the multivariate case, x_n in the above is a vector, F is a scalar function, and $\partial F/\partial x_n$ is its gradient. In practice, F will be taken as linear and its fit is straightforward along the lines of Ref. 13. Figure 4 was obtained in this way, with $m=2$ and $k=l=1$. More details about the multivariate method will be given in a forthcoming paper.[23]

The main disadvantages of Eq. (3.1) are the following:

(i) For each index n, we can take into account only one constraint, i.e., the attractor is always constrained to be on a codimension 1 manifold, even if the embedding dimension m is chosen large. *A priori*, one might have thought this to be a serious problem which should ruin the applicability of the method. To our surprise this was not the case. We obtained, e.g., very good results when embedding the Hénon map in 6 dimensions. In all cases where the exact dimension of the attractor was known, considerable "overembedding" improved the performance.[13] Nevertheless we have the suspicion that the mediocre performance for the Lorenz and Rössler systems might be due to this.

(ii) As seen from Eq. (3.1), we run into problems if $\partial F/\partial x_n$ is very small, i.e., if the manifold is roughly parallel to the x_n axis. In this case, the proposed correction is very large [unless the error $F(\mathbf{x}_{n-r})$ happens to be small too], and we should not accept it. Indeed, we simply discarded such corrections completely in Ref. 13, by not correcting x_n at all if the proposed correction was larger than some given threshold. In the case of linear hyperbolic mappings like Eq. (2.15), the iteration of Eq. (3.1) converges indeed exponentially with leading eigenvalues $\mu=1-\alpha \pm\alpha(1-a_0)/a_1$.[13] For nonlinear maps the situation is again less simple, even if they are hyperbolic. Let us assume for simplicity a 2-d map of the Hénon type. Thus, for any pair (r,s) of delays the exact dynamics can be represented by a constraint

$$F(y_{n-r},y_n,y_{n+s})=0 \tag{3.2}$$

which we can also write explicitly as

$$y_n=f(y_{n-r},y_{n+s}). \tag{3.3}$$

For the Hénon map with $r=s=1$, this reads for instance

$$y_n=\pm\sqrt{a+by_{n-1}-y_{n+1}}. \tag{3.4}$$

Equation (3.1) reduces then to

$$x_n'=(1-\alpha)x_n+\alpha f(x_{n-r},x_{n+s}). \tag{3.5}$$

Assume now that the values of x_{n-r} and x_{n+s} had errors. [For some maps such as, e.g., the Hénon map, it may then happen that Eq. (3.3) has no real-valued solutions if x_{n-r} and x_{n+s} have errors. In such cases we do not correct at all.] These errors are then multiplied by $\partial f/\partial x_{n-r}$ resp. $\partial f/\partial x_{n+s}$ in the correction of x_n. For r, $s\rightarrow\infty$, one easily sees that these partial derivatives behave for nearly all x's as

$$\partial f/\partial x_{n-r}\sim e^{-r|\lambda_2|}, \quad \partial f/\partial x_{n+s}\sim e^{-s\lambda_1}, \tag{3.6}$$

where $\lambda_1>0>\lambda_2$ are the Lyapunov exponents. Indeed, using, e.g., x_{n-r} and x_{n-r+1} as independent variables we can write

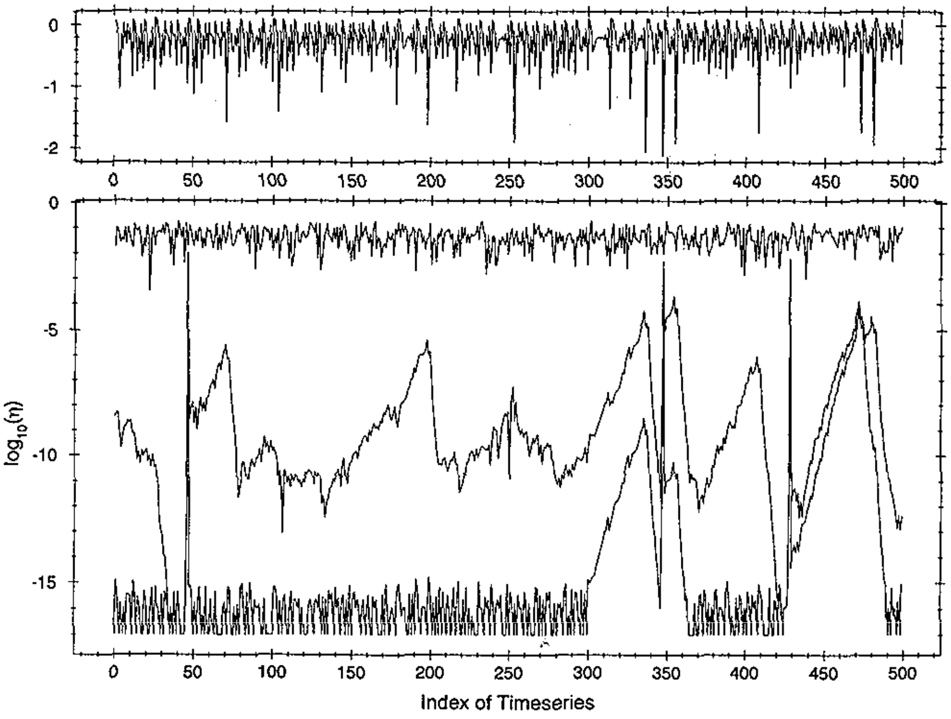

FIG. 4. Similar to Fig. 3, but using the noise reduction scheme given by Eq. (3.5) with $r=s=1$. In the lower curves, the logarithm of $x_{n+1}-a-bx_{n-1}+x_n^2$ is plotted after 0, 100, and 500 iterations. The iteration started with measurement errors distributed uniformly in [−0.05,0.05]. As in Fig. 3, the exact form of the Hénon map was assumed to be known, and the sign in $x_n'=\frac{1}{2}(x_n \pm \sqrt{a+bx_{n-1}-x_{n+1}})$ was chosen as that of x_n. This cannot be applied in points where the square root would become imaginary, and in such points nothing was done at all. The iteration converges to machine precision except at these points. In the upper curve, we also plotted the logarithms of the distances from the nearest primary tangency point, as in Fig. 3.

$$\frac{\partial f}{\partial x_{n+s}}=\frac{\partial x_n}{\partial x_{n+s}}\bigg|_{x_{n-r}}=\frac{\partial x_n/\partial x_{n-r+1}|_{x_{n-r}}}{\partial x_{n+s}/\partial x_{n-r+1}|_{x_{n-r}}}\sim\frac{e^{r\lambda_1}}{e^{(r+s)\lambda_1}}, \tag{3.7}$$

and similarly for $\partial f/\partial x_{n-r}$.

For sufficiently large delays k and l and for sufficiently low noise level, our method should thus converge. But for finite k and l the above partial derivatives can be larger than unity even for hyperbolic systems, and we must expect problems. This was indeed seen in the Hénon case with r and s up to 2. There neither Eq. (3.5) nor its linearized version

$$\theta_n'=(1-\alpha)\theta_n+\alpha\left[\theta_{n-r}\frac{\partial f}{\partial y_{n-r}}+\theta_{n+s}\frac{\partial f}{\partial y_{n+s}}\right] \tag{3.8}$$

converged exponentially. But unlike Sauer's method discussed in Sec. II, it seemed that tangencies between the stable and unstable manifold are not the main reason when using the exact nonlinear equation (3.5) (see Fig. 4). Instead, there the lack of convergence happens only at points where we do not correct at all since the square root in Eq. (3.5) would be negative [we have not found a better method to correct such points; for the linearized version, Eq. (3.8), the iteration even diverged at these points].

We should finally point out that the choice $r=s$ used in Ref. 13 is in general not optimal. Since in general $\lambda_{\text{unstable}}<|\lambda_{\text{stable}}|$ (since the system is dissipative), Eqs. (3.6) and (3.8) suggest that in general $r\leqslant s$ should be optimal.

IV. OPTIMIZED METHOD

After the preceding discussions, the points to be improved should be obvious: in order to avoid the drawbacks of both Secs. II and III, we should neither make large corrections for the outer components of the delay vectors, nor should we project by changing only a single coordinate.

Let us first discuss the construction of the "proposal vectors" $\boldsymbol{\theta}_k$, and deal later with the problem of properly weighting them.

One possibility consists in making projections orthogonal to the attractor *in the subspace spanned by some of the central coordinates*. Thus we would choose a pair of positive integers (r,s) with $p:=m+1-r-s>1$, and project in the subspace spanned by the p coordinates $x_{n+r},\dots,x_{n+m-s}$ of the $(m+1)$-dimensional delay vector $(x_n,\dots,x_{n+m})$. Technically, this would be obtained from the formalism of Sec. II by replacing Eq. (2.3) by

$$\boldsymbol{\theta}_j=\text{const}\,\mathbf{P}\nabla F(\mathbf{x}_j), \tag{4.1}$$

where $\mathbf{P}$ is the projection operator onto the subspace spanned by the coordinates $x_{j+r},\dots,x_{j+m-l}$,

$$\mathbf{P}_{ii'}=\begin{cases}\delta_{ii'}, & \text{for } r\leqslant i\leqslant m-s\\ 0, & \text{else.}\end{cases} \tag{4.2}$$

Actually we shall stay more general and simply assume Eq. (4.1) with some non-negative symmetric matrix $\mathbf{P}$ which will in general not be a projection matrix. We only require that for nearly all $\mathbf{x}_j$ the corrections $\boldsymbol{\theta}_j$ will be transverse to the manifold $F(\mathbf{x}_j)=0$. If we have Q such constraints, i.e., a codimension Q manifold, then $\mathbf{P}$ should be of rank $\geqslant Q$. In practice, we shall use diagonal matrices, $P_{ik}=\delta_{ik}P_i$, with P_i maximal in the middle of the delay window and decreasing towards its ends.

From the ansatz Eq. (4.1) we obtain

$$\boldsymbol{\theta}_j=-\epsilon_j\frac{\mathbf{P}\nabla_jF}{(\nabla_jF\cdot\mathbf{P}\nabla_jF)}. \tag{4.3}$$

Since the different correction proposals $\theta_{n,i}$ for the same coordinate x_i will have different errors we can also generalize the ansatz Eq. (2.6) so that not all proposed corrections $\theta_{j,k}$ are weighted equally:

$$x_n'=x_n+\sum_{j=n-m}^{n}(\mathbf{W}\boldsymbol{\theta}_j)_{n-j}, \tag{4.4}$$

where $\mathbf{W}$ is another diagonal non-negative $(m+1)\times(m+1)$ matrix. With this ansatz, we obtain

$$\epsilon_k'=\epsilon_k-\alpha\sum_j\epsilon_j\frac{(\nabla_kF\cdot\mathbf{B}^{j-k}\mathbf{W}\mathbf{P}\nabla_jF)}{(\nabla_jF\cdot\mathbf{P}\nabla_jF)}. \tag{4.5}$$

The algorithm thus depends on two non-negative diagonal matrices $\mathbf{P}$ and $\mathbf{WP}$ which should be chosen such that convergence of Eq. (4.5) is fastest. In the following we shall argue for a certain choice of $\mathbf{P}$ which is close to optimal for most purposes. Since we found no similiar general results for the matrix $\mathbf{W}$ we fix it to the unit matrix.

A. *Locally linear constraints.* In realistic cases we use locally linear constraints of the form of Eq. (1.8) which have to be fitted from the very time series to be cleaned. Instead of a single constraint at each point, we also should expect $Q>1$ such constraints F_q, so that the attractor is constrained to be on a manifold of codimension Q. One could modify Eqs. (4.3) and (4.5) such that they could still be used in this case. But the resulting formalism would be somewhat inconvenient, and we use a different approach.

The basic idea is to formulate a minimization problem, from which both the linear constraints and the corrections are obtained in one step. We first define for each point $\mathbf{x}_n$ a neighborhood $\mathscr{U}_n$ on which the linearized constraints are supposed to be valid.

Each constraint can be written as $(\mathbf{a}_q^{(n)}\cdot\mathbf{y})+b_q^{(n)}=0$, where $\mathbf{y}$ is again the noise-free signal corresponding to $\mathbf{x}$, and $\mathbf{a}_q^{(n)}=\nabla_nF_q$. Assume there are $|\mathscr{U}_n|$ points in the neighborhood $\mathscr{U}_n$. For each of them, the constraints can be written as

$$(\mathbf{a}_q^{(n)}\cdot(\mathbf{x}_k+\boldsymbol{\theta}_k))+b_q^{(n)}=0,\quad q=1,\ldots,Q,\quad \mathbf{x}_k\in\mathscr{U}_n, \tag{4.6}$$

where the Q vectors $\mathbf{a}_q^{(n)}$ should be linearly independent and properly normalized. The latter can be achieved by requiring

$$(\mathbf{a}_q^{(n)}\cdot\mathbf{P}\mathbf{a}_{q'}^{(n)})=\delta_{qq'}. \tag{4.7}$$

With these two requirements we guarantee the consistency of the corrections with the dynamics. Finally, as the new trajectory should be as close as possible to the old one, we have to impose

$$\epsilon=\sum_{k:\mathbf{x}_k\in\mathscr{U}_n}(\boldsymbol{\theta}_k\cdot\mathbf{P}^{-1}\boldsymbol{\theta}_k)\overset{!}{=}\min. \tag{4.8}$$

[Instead of Eq. (4.8), we could also use a weighted sum where each contribution is multiplied by a weight factor w_k with $w_k>0$ and $\Sigma_k w_k=1$. We then would have to replace Eqs. (4.9) and (4.10) below by $\xi_i^{(n)}=\Sigma_{\mathbf{x}_k\in\mathscr{U}_n}w_kx_{k+i}$ resp. $C_{i,j}^{(n)}=\Sigma_{\mathbf{x}_k\in\mathscr{U}_n}w_kx_{k+i}x_{k+j}-\xi_i^{(n)}\xi_j^{(n)}$. In Ref. 14 it is claimed that distance-dependent weights $w_k\propto|\mathbf{x}_n-\mathbf{x}_k|^{-2}$ are indeed better than the uniform weight we use in this paper.] Here we have of course assumed that $\mathbf{P}^{-1}$ exists, i.e., that all P_i are different from zero. If not, then the corresponding component of $\boldsymbol{\theta}_k$ has to vanish, which would be formally obtained by using $1/P_i=\infty$ in Eq. (4.8). Notice that Eq. (4.8) indeed forces the ith component of $\boldsymbol{\theta}_k$ to be proportional to P_i, as we want it to be.

The minimization problem defined by Eqs. (4.8), (4.6), and (4.7) is solved in Appendix B, with the following result.

From the points $\mathbf{x}_k\in\mathscr{U}_n$ we first construct the average

$$\xi_i^{(n)}=\frac{1}{|\mathscr{U}_n|}\sum_{k:\mathbf{x}_k\in\mathscr{U}_n}x_{k+i},\quad i=0,1,\ldots,m \tag{4.9}$$

and the $(m+1)\times(m+1)$ covariance matrix

$$C_{i,j}^{(n)}=\frac{1}{|\mathscr{U}_n|}\sum_{\mathbf{x}_k\in\mathscr{U}_n}x_{k+i}x_{k+j}-\xi_i^{(n)}\xi_j^{(n)}. \tag{4.10}$$

We then define

$$R_i=1/\sqrt{P_i} \tag{4.11}$$

and

$$\Gamma_{ij}^{(n)}=R_iC_{i,j}^{(n)}R_j. \tag{4.12}$$

The Q orthonormal eigenvectors of the matrix $\boldsymbol{\Gamma}^{(n)}$ with smallest eigenvalues are called $\mathbf{e}_q^{(n)}$, $q=1,\ldots,Q$. The projector onto the subspace spanned by these vectors is denoted as $\mathscr{Q}^{(n)}$, i.e.,

$$\mathscr{Q}_{ij}^{(n)}=\sum_{q=1}^{Q}e_{q,i}^{(n)}e_{q,j}^{(n)}. \tag{4.13}$$

Then the ith component of $\boldsymbol{\theta}_n$ is given by

$$\theta_{n,i}=\frac{1}{R_i}\sum_{j=0}^{m}\mathscr{Q}_{ij}^{(n)}R_j(\xi_j^{(n)}-x_{n+j}). \tag{4.14}$$

Intuitively, the $\mathbf{a}_q^{(n)}$ are orthogonal to a linear subspace, in which the neighborhood of the considered point is embedded. Formally this subspace is given by the leading eigenvectors of the covariance matrix. The correction simply consists in projecting the point onto this hyperplane of codimension Q, but in a metric given by $\mathbf{P}$ rather than orthogonally.

We notice that the cases studied in Secs. II and III are just special cases of the present algorithm, and are most easily implemented using Eq. (4.14). In the case of Sec. II, we have simply $R_i=1$ for all i. The case of Sec. III is obtained formally from the above (with $Q=1$) by using $R_i=1$ for $i=r$ and $R_i=R\gg 1$ for all $i\neq r$. In the limit $R\to\infty$, a single eigenvalue of $\Gamma^{(n)}$ will stay finite, and Eq. (4.14) gives exactly the correction proposed in Ref. 13.

Before leaving this section, we should point out that a simplified version of the methods of this and the previous sections consists in replacing $\theta_{n,i}$ by $P_i(\xi_i^{(n)}-x_{n+i})$. In this way we would correct each point such that the coordinates for which $P_i\neq 0$ move towards the center of mass of its neighbors. Just like the original algorithms correspond to locally linear fits, this corresponds to assuming locally constant fits. It will in general give worse results, but it is much simpler to implement and might thus be worth while. A detailed description (including a complete FORTRAN code) and some numerical results are given in Ref. 24.

In Ref. 12 a second order polynomial was fitted to the local dynamics. Especially if the density of points is low the radii of the neighborhoods may be so large that a higher than first order approximation could be reasonable. In our algorithm one could include nonlinear terms in Eq. (4.6). But unfortunately then the corrections would appear in a nonlinear way in the minimization problem. The criterion for the constraints to be independent would be much more complicated than Eq. (4.7), and finally we would have to determine much more parameters of the dynamics. Instead, we restrict ourselves to linear constraints and in real applications subtract a local trend of the suggested corrections, which should correct curvature effects (see Sec. V).

V. IMPLEMENTATION AND RESULTS

In this last section, we shall present details of the implementation and some more numerical results. We compare in particular the algorithms based on Secs. II–IV. We also give results obtained from two simpler algorithms: SVD-Wiener filtering as described in Appendix A, and a fit with locally constant functions instead of locally linear functions.

In all test cases considered below, we only use white measurement noise with Gaussian or uniform distribution, i.e., the noisy signal is written as $x_n=y_n+\epsilon_n$, where ϵ_n is Gaussian or uniform white noise. The noise level is defined as $\sqrt{\langle\epsilon_n^2\rangle/\langle y_n^2\rangle}$. Furthermore, in all cases the exact deterministic model is known for performance estimation, though it is not used during noise reduction (for an application to signals with unknown underlying dynamics, see Ref. 13). The final trajectory after noise reduction is written $x_n^{\text{corr}} = y_n + \epsilon_n^{\text{red}}$. In this case, we have (at least) two measures for performance:

- The increase of the signal-to-noise ratio,

$$r_0=\sqrt{\frac{\Sigma_n\,(\epsilon_n)^2}{\Sigma_n\,(\epsilon_n^{\text{red}})^2}}. \tag{5.1}$$

The logarithmic improvement of the signal-to-noise ratio, measured in decibels (dB), is defined as

$$\delta SNR=20\log_{10} r_0. \tag{5.2}$$

- The decrease of the violation of determinism. Assume the signal obeys the equation $y_n=f(y_{n-1},...)$. Then we define

$$r_{\text{dyn}}=\sqrt{\frac{\Sigma_n\,(x_n-f(x_{(n-1)},...))^2}{\Sigma_n\,(x_n^{\text{corr}}-f(x_{(n-1)}^{\text{corr}},...))^2}}. \tag{5.3}$$

Though this could, in principle, be estimated for all our examples, its evaluation is easiest for a series whose dynamics is already formulated in terms of delay coordinates, as in the Hénon map—or in the case when all coordinates are dealt with simultaneously. Thus we evaluated it only in the latter cases.

A. SVD-Wiener filtering

Let us first discuss the results of SVD-Wiener filters as defined in Appendix A, case (c). In applying these filters, we need an estimate for the noise spectrum. The most natural assumption is that it is white (which is indeed true in our examples). We pretend however that we do not know the correct noise level. Thus we assume in applying Wiener filters that $S^{\text{noise}}=\min_k S_k$, i.e., that the minimum of S^{signal} is zero.

For low noise level and low sampling frequency it does not give any improvements since the distortion it involves is more important than the noise reduction. Thus for the Hénon map, we do not get any improvement in r_0 for a noise level less than ca. 45%, for any embedding dimension (we would get noise reduction for arbitrarily small noise level if we would use knowledge of this level). For higher noise levels, the best results are obtained with embedding dimension $m\approx 17$, but the results depend only weakly on m. For 100% noise level, we obtain $r_0=1.49$.

The situation is very different for frequently sampled flows. In Fig. 5 we show results for the x and z components of the Lorenz model[25] (with parameters $\sigma=10$, $\rho=28$, $\beta=8/3$), and for the x component of the Rössler model[26] (with parameters $a=0.36$, $b=0.4$, $c=4.5$). In panel (a), the noise level is 10%, while in panel (b) it is 100%. We see that the results improve greatly with the sampling frequency. We also see that the method works much better for the Rössler model, reflecting the rather sharply peaked spectrum in this model. On the other hand, there is little difference between the results for the x and z coordinates of the Lorenz attractor, in spite of the fact that the spectrum of the z component is much larger at large frequencies, and a simple low-pass filter would do much worse for the z component than for the x component.

The results shown in Fig. 5 were obtained with embedding dimensions $m=40$, but again they depended very little on the precise value of m. Practically the same results were obtained for all m between 30 and 50.

When comparing Fig. 5 with the results of Ref. 14, we see that there is very little difference at high sampling frequencies (> 100 points per cycle). Indeed, for such high sampling rates we found an improvement of the SNR even for very low-level noise with SNR < 1%.

We should finally point out that Wiener and SVD-Wiener filters based on finite window lengths m should in principle not increase the dimension of chaotic attractors,

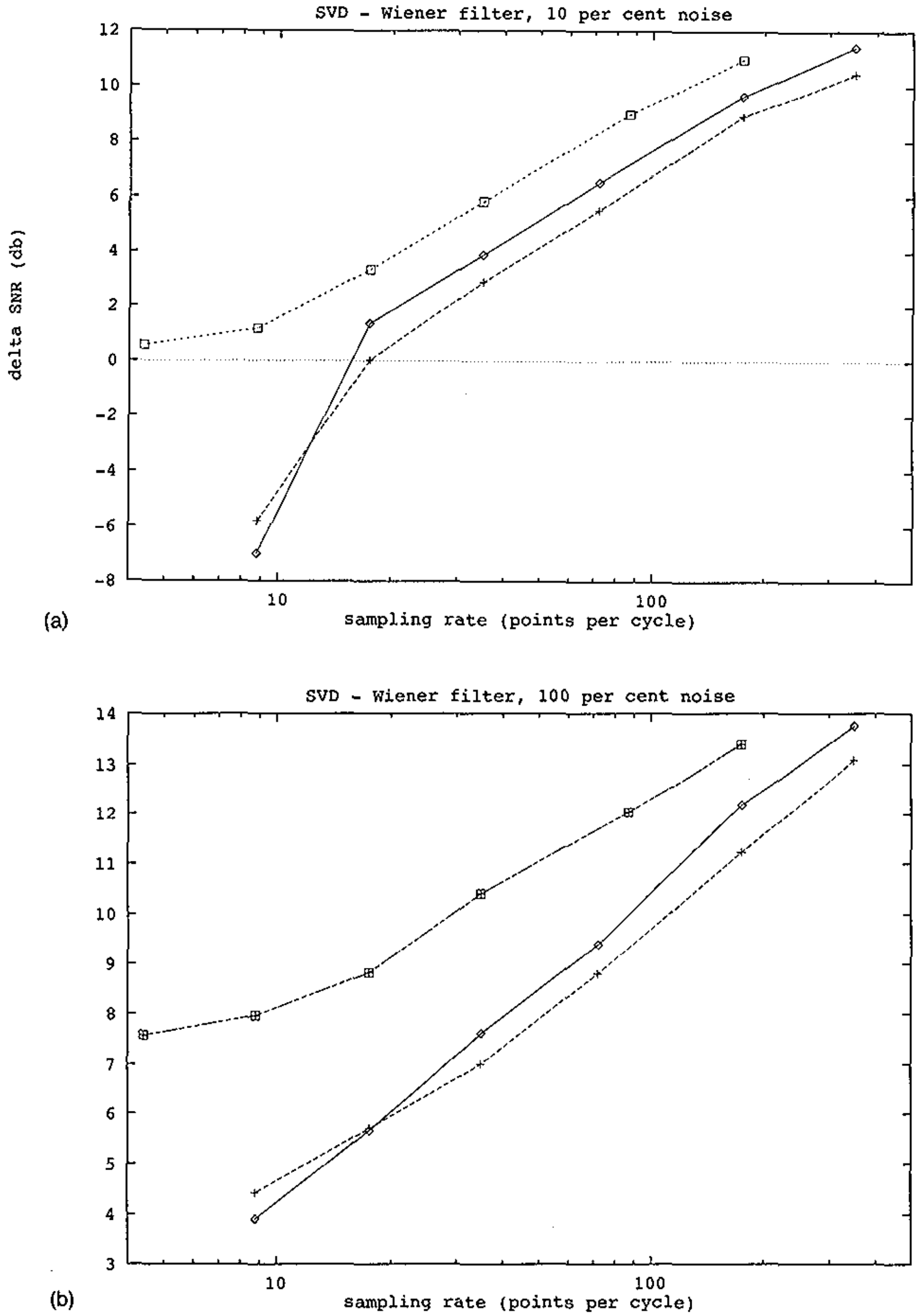

FIG. 5. Improvements of the signal-to-noise ratio, measured in decibels, and achieved by the SVD-Wiener method described in Appendix A. Each data point is based on a time series of 20 000 points. Embedding dimension was $m=40$. In panel (a), the noise level is 10%, while it is 100% in panel (b). Squares are for the x coordinate of the Rössler system, diamonds for the x coordinate of the Lorenz model, and crosses for the z coordinate of the latter.

in contrast to autoregressive filters.[27] In practice, they can however make dimension estimates harder when applied to too coarsely sampled time series.

B. Implementation of locally linear noise reduction

From the derivation in Sec. IV A it should be clear how to proceed:

● Embed the time series in an $m+1$-dimensional phase space using delay coordinates.

● For each embedding vector $\mathbf{x}_n$ find a neighborhood containing at least K points. For efficiency it is advisable to use a fast neighbor search algorithm, e.g., as described in Ref. 1.

● Compute the center of mass $\xi^{(n)}$, Eq. (4.9), the covariance matrix $\mathbf{C}^{(n)}$, Eq. (4.10), and its weighted form $\mathbf{\Gamma}^{(n)}$, Eq. (4.12).

● Determine all eigenvectors of $\mathbf{\Gamma}^{(n)}$. The correction is then obtained by projecting $\xi^{(n)}-\mathbf{x}_n$ into the subspace spanned by the Q eigenvectors with the smallest eigenvalues according to Eq. (4.14). Note that each element of the time series appears as a component in $m+1$ delay vectors. Therefore its correction is built up by several contributions.

- When all corrections are added the time series is replaced by the corrected one and the procedure is repeated.

When applying this scheme to a given time series one has to choose a suitable set of parameters, which are discussed in the following.

- For maps, the embedding dimension should be roughly twice the dimension for which the dynamics in delay coordinates gets deterministic. For finely sampled flows it could even be much higher because of redundancy of the information. Alternatively, data could be compressed as suggested in Ref. 16.

- The matrix **P** is conveniently chosen diagonal giving very small weights to the first and last components of the delay vectors and weight one to the center components (see discussion in Sec. IV).

- The number of constraints can be either kept fixed or increased during the iterations as the separation between signal and noise gets clearer. The remaining subspace has to be of a dimension large enough to cover the attractor.

- For optimal results the proper choice of the neighborhoods is essential. It has to be a compromise between the need for good statistics for the linear fits (large K), and the approximate linearity in the neighborhood (small radius). Assume that the dimension of the attractor is D, its overall size is normalized to 1, and that the entire time series contains T points. Then a box with K points will have a size $\approx (K/T)^{1/D}$. Assuming typical curvatures to be also of order 1, the nonlinearity will thus give errors of order $(K/T)^{2/D}$. Statistical fluctuations in the fitted vectors $\mathbf{a}^{(n)}$ on the other hand will be of order $\langle\epsilon\rangle/\sqrt{K-m}$, where $\langle\epsilon\rangle$ is the noise level. For the optimal value of K, both errors should be comparable. For large noise level, we get $K \gg m$ and thus

$$K \approx \{\langle\epsilon\rangle T^{2/D}\}^{2D/(4+D)}. \tag{5.4}$$

For vanishing noise level, K should decrease to $m+2-Q$, in which case the linear subspace is defined by a simplex. Since the noise level decreases during the iterations, K should be decreased as well. In regions of the phase space with higher than average density of points the estimate (5.4) leads to a neighborhood with a radius smaller than the noise level. In this case we take all neighbors closer than the assumed noise level.

- After a number of iterations the noise reduction does not improve further and the nonlinearities give cumulative systematic errors which distort the attractor. This can be partly avoided by shrinking the neighborhoods during iterations. However in spite of this, systematic shifts of the orbit due to nonlinearity of the attractor remain. In the noise-free case, points typically would be shifted perpendicularly to the manifold by an amount proportional to its curvature. This can be corrected by a trick suggested in Ref. 16. It is based on the assumed smoothness of the attracting manifold, due to which all neighboring points will feel essentially the same shift. This shift can thus be greatly reduced by subtracting from the proposed corrections θ_n the average of the corrections for the neighboring points.

- For a proper choice of the matrix **P** and the number Q of constraints, large spurious corrections occur less frequently than in the algorithm described in Sec. III. It turned out that the distribution of the magnitude of the corrections was Poissonian in all cases. We interpret the tail as spurious and discard the corresponding corrections.

The following prefiltering may be useful, especially in cases with high noise level and high sampling frequencies (small delays): When searching for the neighbors defining the linearization neighborhood, filtered data are used (SVD filtering or the method described in Ref. 24). The corrections are added to the raw data and not to the prefiltered data. Thus the local dynamics may be defined in a better way, but on the other hand no information about the original time series is lost.

C. Numerical results

For a detailed comparison of the different algorithms we have chosen several typical test cases. The time series of the Hénon, the Ikeda, and the Lorenz system were contaminated with different noise levels. The performance is reported in Table I. The parameters we have used for the SVD method are quoted in Sec. V A. The other methods are special cases of the general formalism of Sec. IV. Therefore for a better comparison apart from the matrices **P** all the parameters were the same. For the embedding dimension we have chosen 5 for the Hénon, 7 for the Ikeda map, and 7, respectively, 13 for low and high sampled Lorenz data. In all cases we required between 20 and 50 neighbors taking into account the estimate in Eq. (5.4). The curvature effects were corrected. Furthermore, too big correction vectors θ were rescaled to the average magnitude of corrections, using as a criterion the distribution of their size. This cutoff is essential in the case of small noise, as "wrong corrections" have a big destructive effect in this case. For the Cawley–Hsu–Sauer algorithm the matrix **P** is unity, whereas for the new proposal we have chosen $P_i = 0.001$ for the two outer entries and $P_i = 1$ for all others. In the Schreiber–Grassberger case **P** contains only a single 1 in the center, all other entries on the diagonal are 0.001. In all cases we increased Q during the iteration, namely, from 2 to 3 for the Hénon, from 3 to 5 for the Ikeda, from 2 to 4 for the Lorenz with low sampling rate and from 6 to 10 with high sampling rate. In contrast to Ref. 13 and Sec. III, where only one constraint is possible, in the Schreiber–Grassberger limit of the generalized algorithm we used the above values of Q, which improved the performance. (See Fig. 6.)

The above parameters were in general not those which gave optimal performance for the specific time sequence. Instead, we preferred to quote results for a "compromise" set of parameters. Systematic studies of the dependence on single parameters were made in Refs. 14 and 15, but we found such studies not very useful due to the strong correlations between optimal parameters. Also, sets of param-

TABLE I. Numerical results of the different algorithms introduced in the paper. For explanation of the parameters see Sec. V.

			SVD	Sec. II		Sec. III		Sec. IV		0th order		Ref. 24	
System	T	ϵ	r_0	r_0	r_{dyn}	r_0	r_{dyn}	r_0	r_{dyn}	r_0	r_{dyn}	r_0	r_{dyn}
Hénon	5 000	100%	1.5	1.7	3.6	1.7	4.1	1.9	8.3	1.7	4.0		
		10%	···	2.6	4.5	3.4	7.5	4.2	10.7	2.8	5.6		
		1%	···	2.9	5.7	3.4	5.3	3.4	6.6	1.7	2.5		
	20 000	1%	···	3.2	6.5	4.1	6.8	4.2	8.4	2.4	3.8		
	100 000	0.3%	···	2.9	4.0	4.7	6.8	5.0	8.3	2.4	4.4		
Ikeda	20 000	10%	···	3.4	···	2.2	···	3.7	···	2.5	4.4[c]	3.6	6.6
(x coord.)		1%	···	1.5	···	1.7	···	2.1	···	1.2	1.7[c]	1.8	2.6
Lorenz[a]	20 000	10%	1.2	3.5	···	2.9	···	3.3	···	2.0	7.2[c]	3.1	8.1
	20 000	1%	···	2.3	···	1.8	···	2.3	···	1.1	1.3[c]	1.9	2.3
Lorenz[b]	5 000	10%	···	1.8	···	1.5	···	1.8	···	1.5	4.3[c]	2.4	4.8

[a] Delay $\tau=0.05$, x coordinate.

[b] Delay $\tau=0.2$, -"-.

[c] Obtained from the vector-valued time series using the 0th order multivariate version (Ref. 24).

eters optimal for one time series often gave mediocre results for other time series. Such differences occurred even for time series belonging to the same model but using different noise realizations. The above set of parameters gave consistently good results. Therefore the numbers reported in Table I must not be taken literally, but their relative magnitude is characteristic for the different algorithms. The only quantitative comparison with a noise reduction algorithm different from this class can be obtained from Ref. 10: For a Hénon trajectory of length 32 000 and 1% additive noise the authors of Ref. 10 find $r_{\mathrm{dyn}}=4.8$.

By "0th order" we denote the algorithm explained in Ref. 24, i.e., substituting each point of the noisy trajectory by the average value of its neighbors. The optimal parameters are reported in Ref. 24.

For applications to data sets with unknown dynamics (experimental data) one first has to have at least a vague idea about the correct embedding dimension. Then the parameters should be chosen following the above examples. Our experience with "toy models" shows that suboptimal choices of the parameters simply lead to suboptimal results but do not destroy the data structures. One possible diagnostics is a statistical analysis of the corrections made. Especially they should not be strongly correlated to the signal.

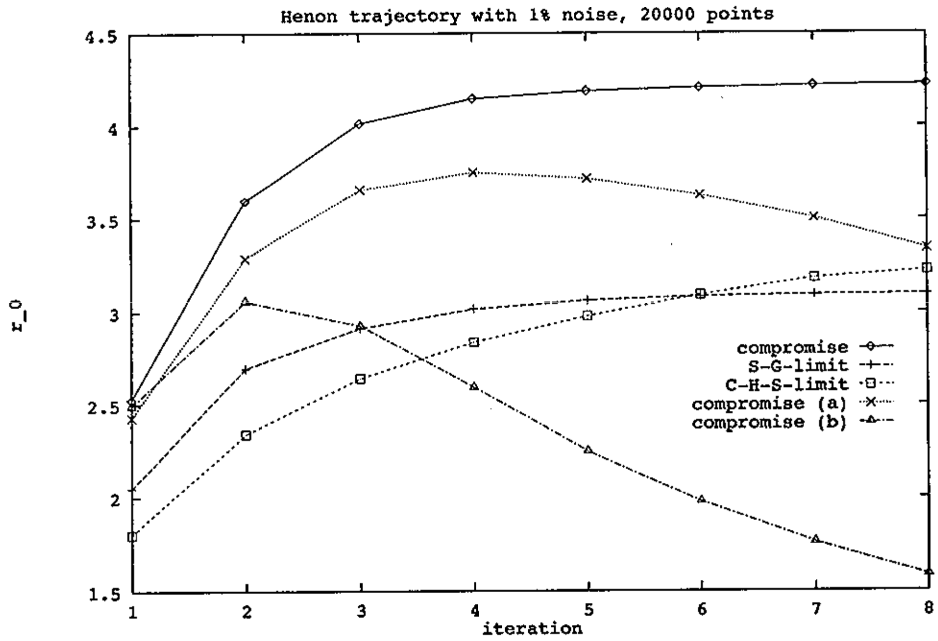

FIG. 6. Improvements of r_0 vs the number of iterations for three different choices of the matrix **P**: Cawley–Hsu–Sauer limit, Schreiber–Grassberger limit, and compromise with $P_i=0.001$ for $i=1$ and $i=5$ and $P_i=1$ else. The original series consisted of 20 000 points on the Hénon attractor, corrupted with 1% uniformly distributed measurement noise, and embedded with $m=4$. The two nonmonotonous curves are the results of the compromise without cutoff of too large corrections (a) and, in addition, without subtraction of the curvature effects (b). Both are crucial in this case [and (b) even more so for shorter trajectories], as the noise level is so small. For large noise levels and long trajectories both effects become negligible.

VI. CONCLUSIONS

We have seen that the two seemingly rather different classes of locally linear noise reduction methods developed by Cawley, Hsu, and Sauer[14–16] on the one hand, and by Schreiber and Grassberger[13] on the other, are indeed closely related. The formalism of Sec. IV is a generalization of both methods. It supplies the possibility to find a compromise between these two algorithms avoiding their disadvantages.

A basic problem with any noise reduction scheme—and thus also with the present ones—is that we have to *assume* that a separation into noise plus signal is meaningful. In real-life applications this need not always be the case. Also, no clear-cut criterion for the amount of noise to be subtracted can be given in such cases, and no performance measure is available. In view of this, robustness of the algorithm is of utmost importance.

The main problem with respect to applications is the choice of the different parameters. However, a quite robust scheme is obtained with the parameters suggested in Sec. V. The tabular presentation of the results shows the overall superiority of the new scheme for maps. For high sampled flows the performance of our compromise is as good as the Cawley–Hsu–Sauer limit. Nevertheless, the optimal algorithm and the best parameters depend on the noise level, the sampling rate, and the length of the time series.

For high noise levels and for large sampling rates, linear prefiltering as used by Sauer gives good results. We found that filtering based on SVD is for that purpose about as efficient as Fourier-based Wiener filtering, and usually better than simple low-pass filtering. For maps, sparsely sampled flows, or low noise levels, the distortions involved in linear prefiltering overwhelm its advantages.

Another trick (due to Sauer) which is not fundamental for the method to work but which can improve results considerably, is a correction which takes into account curvature effects.

As implemented in Sec. V, our algorithm is rather robust and not too time consuming. Typically, time sequences of ca. 10^4 points can be handled easily on work stations. It is much slower than conventional (linear) filters but it gives much better results, in particular for low noise levels.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschft, SFB 237. T.S. receives a European Communities grant within the framework of the SCIENCE programme, Contract No. B/SCI*-900557.

APPENDIX A: WIENER AND SVD–WIENER FILTERING

In this appendix we shortly discuss Wiener filtering[17] and its generalization[19,20] obtained by replacing the Fourier basis by the SVD basis.

In the following we assume that the time series is a randomly chosen realization of a stochastic stationary process $X = X_1 X_2 \cdots$ with zero mean.

(a) Let us first write the discrete Fourier transform of the noisy time series as

$$\tilde{x}_k = \sum_{j=1}^{T} x_j e^{2\pi ikj/T}. \tag{A1}$$

The spectrum S_k of the random process X is defined as

$$S_k = \langle |\tilde{x}_k|^2 \rangle. \tag{A2}$$

We assume it to be a sum of two contributions,

$$S_k = S_k^{(y)} + S_k^{\text{noise}} \tag{A3}$$

corresponding to the clean signal and the noise, resp. (we leave out the exact normalization of the spectrum for which several conventions exist in the literature). Notice that there is no cross term from the additive ansatz Eq. (2) for the *amplitudes* since we assume that signal and noise are statistically independent. We also assume for the moment that we know (or can guess) both S_k and the noise spectrum S_k^{noise}.

We look for a linear filter, i.e., for a linear relationship between the random variables X_n (whose realization are the x_n) and a filtered random variables X_n^{corr}, such that the spectrum of X_n^{corr} is $S_k - S_k^{\text{noise}}$ and such that

$$\langle |X_n^{\text{corr}} - X_n|^2 \rangle \tag{A4}$$

is minimal. This filter is uniquely given by[17]

$$\tilde{x}^{k,\text{corr}} = \left(1 - \frac{S_k^{\text{noise}}}{S_k}\right) \tilde{x}_k. \tag{A5}$$

If S_k^{noise} is white and S_k is monotonously decreasing, then this is essentially a low-pass filter. On the other hand, if S_k is not monotonous then it mainly filters away the regions where S_k is small (since it should be noise dominated there), and leaves the peak regions essentially unchanged. With such a filter, one can, e.g., clean completely periodic and quasiperiodic signals, provided the time series is sufficiently long.

(b) Assume we have only a single time series available, without any prior estimate of the spectrum, and we want to apply Fourier–Wiener filtering. Thus we have first to estimate the Fourier spectrum S_k from $\tilde{x}_k$. As is well known, we cannot simply use $S_k \approx |\tilde{x}_k|^2$ as an estimator since the statistical fluctuations of $|\tilde{x}_k|^2$ do not decrease in the limit $T \to \infty$. One way to circumvent this problem consists in modifying the above procedure as follows. We Fourier transform only short pieces $\mathbf{x}_n = (x_n, \ldots, x_{n+m})$ of length $m+1$, calling the results $\tilde{x}_k^{(n)}$, $k = 0, \ldots, m$, and estimate a coarse grained spectrum as

$$S_k^{(\text{coarse})} \approx \frac{1}{T-m} \sum_{n=1}^{T-m} |\tilde{x}_k^{(n)}|^2. \tag{A6}$$

In the filtering procedure it is then natural to apply Eq. (A5) to the same pieces $\mathbf{x}_n$, which gives $m+1$ "clean" candidates $z_j^{(n)}$ for each x_j,

$$z_j^{(n)} = \frac{1}{m+1} \sum_{k=0}^{m} \left(1 - \frac{S_k^{\text{noise,coarse}}}{S_k^{\text{coarse}}}\right) \tilde{x}_k^{(n)} e^{2\pi ikj/(m+1)}. \tag{A7}$$

In general, they will not be independent of n, i.e., Eq. (A7) would give $m+1$ mutually inconsistent "cleaned" values for each j. We propose that a natural compromise between them is just their arithmetic average,

$$x_j^{\text{corr}}=\frac{1}{m+1}\sum_{n=j-m}^{j} z_j^{(n)}. \tag{A8}$$

(c) In the SVD case, we proceed exactly as in case (b). We first form the $(m+1)\times(m+1)$ covariance matrix

$$C_{jk}=\frac{1}{T-m}\sum_{n=1}^{T-m} x_{n+j}x_{n+k}, \tag{A9}$$

its eigenvalues μ_i, and its (orthonormal) eigenvectors $\mathbf{e}_i$ $(i=0,...,m)$. We then assume that the eigenvalues (which correspond to the spectrum in the Fourier case) are additively composed of signal and noise contributions,

$$\mu_i=\mu_i^{(y)}+\mu_i^{\text{noise}}, \tag{A10}$$

and we form the matrix

$$P_{jk}=\sum_{i=0}^{m}\left(1-\frac{\mu_i^{\text{noise}}}{\mu_i}\right)e_{i,j}e_{i,k}. \tag{A11}$$

If some of the "signal" eigenvalues $\mu_i^{(y)}$ are zero, then $\mathbf{P}$ projects into the orthogonal subspace. If all μ_i^{noise} were zero, then $\mathbf{P}$ would be just the unit matrix. The method proposed in Ref. 20 is obtained by assuming that the smallest μ_i's are pure noise and the others are pure signal, in which case $\mathbf{P}$ is a projection matrix. Notice however that in general $\mathbf{P}^2\neq\mathbf{P}$, i.e., $\mathbf{P}$ is in general not a projection matrix. From each delay vector $\mathbf{x}_n=(x_n,...,x_{n+m})$ we can form candidates $z_j^{(n)}$ for cleaned values of x_j with $j=n,...,n+m$ as

$$z_j^{(n)}=\sum_{k=n}^{n+m} P_{j-n,k-n}x_k. \tag{A12}$$

Again, they will not be independent of n in general. We again propose as a compromise their arithmetic average,[20]

$$x_j^{\text{corr}}=\frac{1}{m+1}\sum_{n=j-m}^{j} z_j^{(n)}=\sum_{k=0}^{m} x_k\frac{1}{m+1}\sum_n P_{j-n,k-n}. \tag{A13}$$

Just like Wiener filtering, this method gives perfect cleaning for periodic and quasiperiodic signals, provided the time series and the window length m are long enough. Since the eigenvectors $\mathbf{e}_i$ become sines resp. cosines in the limit $m\rightarrow\infty$,[19] in this limit both methods essentially coincide.

APPENDIX B: A MINIMIZATION PROBLEM

In this appendix we solve the minimization problem defined by Eqs. (4.8), (4.6), and (4.7). We also show that its solution is a superposition of the form

$$\boldsymbol{\theta}_n=\sum_q \mu_{nq}\mathbf{P}\nabla F_q^{(n)}, \tag{B1}$$

i.e., it is exactly of the form of Eq. (4.1), generalized to several constraints.

We use a Lagrange multiplier formalism, with multipliers $\mu_{k,q}$ for the constraints (4.6) and with multipliers $\lambda_{qq'}$ for the constraints (4.7). The Lagrange function thus reads

$$\begin{aligned}L=&\frac{1}{2}\sum_{k:\mathbf{x}_k\in\mathscr{U}_n}(\boldsymbol{\theta}_k\cdot\mathbf{P}^{-1}\boldsymbol{\theta}_k)\\&-\sum_{\substack{k:\mathbf{x}_k\in\mathscr{U}_m\\ q\leqslant Q}}\mu_{kq}[\mathbf{a}_q^{(n)}\cdot(\mathbf{x}_k+\boldsymbol{\theta}_k)+b_q^{(n)}]\\&+\frac{1}{2}\sum_{q,q'\leqslant Q}\lambda_{qq'}[\delta_{qq'}-(\mathbf{a}_q^{(n)}\cdot\mathbf{P}\mathbf{a}_{q'}^{(n)})].\end{aligned} \tag{B2}$$

Varying L with respect to $\boldsymbol{\theta}_k$, we obtain first

$$\boldsymbol{\theta}_k=\sum_q \mu_{kq}\mathbf{P}\mathbf{a}_q^{(n)}. \tag{B3}$$

Observing that $\mathbf{a}_q^{(n)}$ is just the gradient of the qth constraint function, this proves already our first claim. Multiplying Eq. (B3) with $\mathbf{a}_{q'}^{(n)}$ gives next

$$\mu_{kq}=\mathbf{a}_q^{(n)}\cdot\boldsymbol{\theta}_k=-b_q^{(n)}-(\mathbf{a}_q^{(n)}\cdot\mathbf{x}_k). \tag{B4}$$

Minimizing L with respect to $b_q^{(n)}$ gives $\Sigma_k\mu_{kq}=0$ or

$$b_q^{(n)}=-(\mathbf{a}_q^{(n)}\cdot\boldsymbol{\xi}^{(n)}). \tag{B5}$$

Finally, varying L with respect to $\mathbf{a}_q^{(n)}$ and multiplying the derivative by $\mathbf{a}_{q''}^{(n)}$, we find that $\lambda_{qq'}=0$, as might have been anticipated. Dropping thus the term $\propto\lambda_{qq'}$ in this derivative and using Eqs. (B3), (B4), and (B5), we obtain

$$\mathbf{C}^{(n)}\mathbf{a}_q^{(n)}=\mathbf{P}\sum_{q'}M_{qq'}\mathbf{a}_{q'}^{(n)}. \tag{B6}$$

Here we have defined $M_{qq'}=\Sigma_k\mu_{kq}\mu_{kq'}$, and the matrix $\mathbf{C}^{(n)}$ is defined in Eq. (4.10). The matrix $\mathbf{M}$ is symmetric and can thus be diagonalized as $\mathbf{V}^{-1}\widetilde{\mathbf{M}}\mathbf{V}$ with orthogonal $\mathbf{V}$. Defining $\widetilde{\mathbf{a}}_q^{(n)}=\Sigma_{q'}V_{qq'}\mathbf{a}_{q'}^{(n)}$, and inserting this into Eq. (B6), it simplifies to

$$\mathbf{C}^{(n)}\widetilde{\mathbf{a}}_q^{(n)}=\mathbf{P}\widetilde{M}_{qq}\widetilde{\mathbf{a}}_q^{(n)}. \tag{B7}$$

Since $\mathbf{P}$ is symmetric and positive, $R_i=P_i^{-1/2}$ exists. We can define $\boldsymbol{\Gamma}^{(n)}$ as in Eq. (4.12), find that the vectors $\mathbf{e}_q^{(n)}=\sqrt{\mathbf{P}}\widetilde{\mathbf{a}}_q^{(n)}$ are its orthonormal eigenvectors, and see that $\boldsymbol{\theta}_k$ is given by Eq. (4.14). Since the covariance matric C has rank $m+1$, each set of Q of the $m+1$ eigenvectors is a solution of this problem, but the absolute minimum is reached when choosing the vectors with the smallest eigenvalues.

[1] P. Grassberger, T. Schreiber, and C. Schaffrath, Int. J. Bifurcation Chaos **1**, 521 (1991).

[2] J. P. Crutchfield and B. S. McNamara, Complex Systems **1**, 417 (1987).

[3] J. Cremers and A. Hübler, Z. Naturforsch. **42a**, 797 (1987).

[4] P. Grassberger, talk at Fritz Haber Conference on "Chaos and Related Non-Linear Phenomena," Kiryat Anavim, 1986; Physica Scripta **40**, 346 (1989).

[5] J. D. Farmer and J. Sidorowich, Phys. Rev. Lett. **59**, 845 (1987).

[6] J. D. Farmer and J. Sidorowich, "Exploiting Chaos to Predict the Future and Reduce Noise," in *Evolution, Learning and Cognition*, edited by Y. C. Lee (World Scientific, Singapore, 1988).

[7] M. Casdagli, Physica D **35**, 335 (1989).
[8] K. Stockbro, D. K. Umberger, and J. Hertz, J. Complex Systems (1991).
[9] K. Stockbro, "Predicting chaos with weighted maps," NORDITA preprint (1990).
[10] E. J. Kostelich and J. A. Yorke, Phys. Rev. A **38**, 1649 (1988).
[11] S. M. Hammel, Phys. Lett. A **148**, 421 (1990).
[12] J. D. Farmer and J. Sidorowich, Physica D **47**, 373 (1991).
[13] T. Schreiber and P. Grassberger, Phys. Lett. A **160**, 411 (1991).
[14] R. Cawley and G.-H. Hsu, Phys. Rev. A **46**, 3057 (1992).
[15] R. Cawley and G.-H. Hsu, Phys. Lett. A **166**, 188 (1992).
[16] T. Sauer, Physica D **58**, 193 (1992).
[17] W. H. Press, B. P. Flannery, S. A. Teukolski, and W. T. Vetterling, *Numerical Recipes* (Cambridge University, Cambridge, 1988).
[18] T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. **65**, 579 (1991).
[19] R. Vautard and M. Ghil, Physica D **35**, 395 (1989).
[20] R. Vautard, P. Yiou, and M. Ghil, Physica D **58**, 95 (1992).
[21] M. Hénon, Commun. Math. Phys. **50**, 69 (1976).
[22] P. Grassberger and H. Kantz, Phys. Lett. A **113**, 235 (1985).
[23] R. Hegger and T. Schreiber, Phys. Lett. A **170**, 305 (1992).
[24] T. Schreiber, Phys. Rev. E (in press) (1993).
[25] E. N. Lorenz, J. Atmos. Sci. **20**, 130 (1963).
[26] O. E. Rössler, Phys. Lett. **57**, 397 (1976).
[27] R. Badii *et al.*, Phys. Rev. Lett. **60**, 979 (1988).