NOISE REDUCTION FOR FLOWS USING NONLINEAR CONSTRAINTS*

Krzysztof Urbanowicz

Max Planck Institute for the Physics of Complex Systems
Nöthnitzer Str. 38, D–01187 Dresden, Germany
urbanow@pks.mpg.de

Janusz A. Hołyst

Faculty of Physics and Center of Excellence for Complex Systems Research
Warsaw University of Technology
Koszykowa 75, 00-662 Warsaw, Poland
jholyst@if.pw.edu.pl

(Received July 5, 2005)

On the basis of a local-projective with nonlinear constraints (LPNC) approach (see K. Urbanowicz, J.A. Hołyst, T. Stemler and H. Benner, *Acta Phys. Pol.* **B35**, 2175 (2004)) we develop a method of noise reduction in time series that makes use of constraints appearing due to the continuous character of flows. As opposed to local-projective methods in our method we do not need to determine the Jacobi matrix. The approach has been successfully applied for separating a signal from noise in the Lorenz model and in noisy experimental data obtained from an electronic Chua circuit. The method was then applied for filtering noise in human voice.

PACS numbers: 05.45.Tp, 05.40.Ca

1. Introduction

It is common that observed data are contaminated by noise (for a review of methods of nonlinear time series analysis see [2–4]). The presence of noise can substantially affect such system parameters as dimension, entropy or Lyapunov exponents [5]. In fact noise can completely destroy the fractal structure of a chaotic attractor [6] and even 2% of noise can make a dimension calculation misleading [7]. It follows that both from the theoretical as

* Presented at the Conference on Applications of Random Matrices to Economy and Other Complex Systems, Kraków, Poland, May 25–28, 2005.

well as from the practical point of view it is desirable to reduce the noise level. Thanks to noise reduction [1, 6, 8–19] it is possible *e.g.* to restore the hidden structure of an attractor which is smeared out by noise, as well as to improve the quality of predictions.

Every method of noise reduction assumes that it is possible to distinguish between noise and a *clean signal* on the basis of some objective criteria. Conventional methods such as linear filters use a power spectrum for this purpose. Low pass filters assume that a clean signal has some typical low frequency, respectively it is true for high pass filters. It follows that these methods are convenient for a regular source which generates a periodic or a quasi-periodic signal. In the case of chaotic signals linear filters cannot be used for noise reduction without a substantial disturbance of the clean signal. The reason is the broad-band spectrum of chaotic signals. It follows that for chaotic systems we make use of another generic feature of dissipative motion that is located on attractors consisting of subset of smooth manifolds of an admissible phase space. As result corresponding state vectors reconstructed from time delay variables are limited to geometric objects that can be locally linearized. This fact is a common background of all local projective (LP) methods of noise reduction.

Besides the LP approach there are also noise reduction methods that approximate an unknown equation of motion and use it to find corrections to state vectors. Such methods make use of neural networks [12] or a genetic programming [13] and one has to assume some basis functions *e.g.* radial basis functions [20] to reconstruct the equation of motion. Another group of methods are modified linear filters *e.g.* the Wiener filter [19], the Kalman filter [15], or methods based on wavelet analysis [16]. Applications of these methods are limited to systems with large sampling frequencies, and they are confined to the locally linear nearest neighborhood of every point in phase space.

The method described in this paper can be considered as an extension of a local-projective with nonlinear constraints (LPNC) approach that was introduced in Ref. [1]. We call our method the *local projection with nonlinear constraints for flows* (LPNCF). The method takes into account natural constraints that occur due to the continuous behavior of flows.

The paper is organized as follows. In the following section we shall present the LPNCF method and the comparison with LP methods is shown in Sec. 3. In Sec. 4 we present examples of noise reduction and application to human voice filtering.

2. The LPNCF method

In Ref. [1] the LPNC method of noise reduction of deterministic signal is presented. In this paper we introduce a method that is based on the formulation given in Ref. [1] but it brings much better results as compared to LP approach.

Let $\{x_i\}$ for $i = 1, 2, \ldots, N$ be the time series. The corresponding clean signal we denote as $\{\tilde{x}_i\}$, so when the measurement noise $\{\eta_i\}$ is present we come to the formula $x_i = \tilde{x}_i + \eta_i$ for $i = 1, 2, \ldots, N$. We can define the time delay vectors $\boldsymbol{x}_i = (x_i, x_{i-\tau}, \ldots, x_{i-(d-1)\tau})$ as our points in the reconstructed phase space. Then we can find two nearest neighbors $\boldsymbol{x}_k, \boldsymbol{x}_j \in \boldsymbol{X}_n^{\mathrm{NN}}$ to vector $\boldsymbol{x}_n$ ($\boldsymbol{X}_n^{\mathrm{NN}}$ is the set of nearest neighborhood of the point $\boldsymbol{x}_n$). Let us introduce the following function [1]

$$\begin{aligned} \boldsymbol{G}_n(s) \;=\;& x_{n-s}\left(x_{k+1-s} - x_{j+1-s}\right) \\ &+ x_{k-s}\left(x_{j+1-s} - x_{n+1-s}\right) + x_{j-s}\left(x_{n+1-s} - x_{k+1-s}\right), \end{aligned} \tag{1}$$

for $s = 0, 1, \ldots, d-1$. The function $\boldsymbol{G}_n(s)$ vanishes for clean one-dimensional systems because it appears as a constraint after eliminating a and b from the following equations:

$$\begin{aligned} \tilde{x}_{n+1} &= a\tilde{x}_n + b\,, \\ \tilde{x}_{k+1} &= a\tilde{x}_k + b\,, \\ \tilde{x}_{j+1} &= a\tilde{x}_j + b\,. \end{aligned} \tag{2}$$

In the case of higher dimensional systems the function $\boldsymbol{G}_n(s)$ does not always vanish but is altering slowly in time for dense sampling. This is because the absolute value of the term $\boldsymbol{G}_n(s)$ is a function of difference of neighboring data $(x_{k+1-s} - x_{j+1-s})$ *etc.*, which evolve smoothly in time (near neighbors behave similar in consecutive time steps). Now one can check that for a highly sampled clean dynamics there can be derived such a constraint

$$\begin{aligned} \boldsymbol{C}_n^m \;=\;& \sum_{k=0}^{m-1} (-1)^l \boldsymbol{G}_n(k) \approx 0\,, \\ &\left(l \;=\; k + \sum_{s=1}^{\mathrm{int}(\log 2(k))} \mathrm{int}\left(\frac{k}{2^s}\right)\right), \end{aligned} \tag{3}$$

where $\mathrm{int}(z)$ is a integer part of z and $\log 2(z)$ is a logarithm with a base 2 from z. For example with $m = 8$ the formula (3) gives the following:

$$\begin{aligned} \boldsymbol{C}_n^8 \;=\;& \boldsymbol{G}_n(0) - \boldsymbol{G}_n(1) - \boldsymbol{G}_n(2) + \boldsymbol{G}_n(3) \\ &- \boldsymbol{G}_n(4) + \boldsymbol{G}_n(5) + \boldsymbol{G}_n(6) - \boldsymbol{G}_n(7)\,. \end{aligned} \tag{4}$$

Such a criterium for a constraint can be understood easily if we notice that all elements $\boldsymbol{G}_n(s)$ have almost the same value for clean data and small s. Using this we force the second element of constraint (4) $\boldsymbol{G}_n(1)$ to take the oppose sign as the first element $\boldsymbol{G}_n(0)$. Then the group consisting of the third $\boldsymbol{G}_n(2)$ and fourth $\boldsymbol{G}_n(3)$ elements should have the oppose sign to the group of the first and second element of the constraint (4). If we know that elements $\boldsymbol{G}_n(s)$ are slightly changing with increasing s the constraint (4) should vanish for clean data and for large enough m.

Similarly as in LP methods the constraints (3) are ensured in this approach by application of the method of Lagrange multipliers to an appropriate cost function. Since we expect that corrections to noisy data should be as small as possible, the cost function is assumed to be the sum of squared corrections $S = \sum_{s=1}^{N} (\delta x_s)^2$.

It follows that we are looking for the minimum of the functional

$$S = \sum_{n=1}^{N} (\delta x_n)^2 + \sum_{n=1}^{N} \lambda_n \boldsymbol{C}_n^m = \min . \tag{5}$$

After finding zero points of $2N$ partial derivatives one gets $2N$ equations with $2N$ unknown variables δx_n and λ_n. However, in such a case the derivatives of the functional (5) are nonlinear functions of these variables. For simplicity of computing we are interested to pose our problem in such a way that linear equations appear which can be solved by standard matrix algebra. To understand the role of nonlinearity let us write the terms $\boldsymbol{G}_n(s)$ in constraint $\boldsymbol{C}_n^m$ in such a way that an explicit dependence on the unknown variables is seen

$$\begin{aligned} \boldsymbol{G}_n(s) \;=\; & G(\boldsymbol{X}_{n-s}, \boldsymbol{X}_{n-s+1}) + G(\delta\boldsymbol{X}_{n-s}, \boldsymbol{X}_{n-s+1}) \\ & + G(\boldsymbol{X}_{n-s}, \delta\boldsymbol{X}_{n-s+1}) + G(\delta\boldsymbol{X}_{n-s}, \delta\boldsymbol{X}_{n-s+1}) \, . \end{aligned} \tag{6}$$

Here we introduced the following notation

$$\begin{aligned} G(\boldsymbol{X}_{n-s}, \boldsymbol{X}_{n-s+1}) \;\equiv\; & x_{n-s}(x_{k-s+1} - x_{j-s+1}) + x_{k-s}(x_{j-s+1} - x_{n-s+1}) \\ & + x_{j-s}(x_{n-s+1} - x_{k-s+1}) \, , \\ G(\delta\boldsymbol{X}_{n-s}, \boldsymbol{X}_{n-s+1}) \;\equiv\; & \delta x_{n-s}(x_{k-s+1} - x_{j-s+1}) \\ & + \delta x_{k-s}(x_{j-s+1} - x_{n-s+1}) \\ & + \delta x_{j-s}(x_{n-s+1} - x_{k-s+1}) \, , \\ G(\boldsymbol{X}_{n-s}, \delta\boldsymbol{X}_{n-s+1}) \;\equiv\; & x_{n-s}(\delta x_{k-s+1} - \delta x_{j-s+1}) \\ & + x_{k-s}(\delta x_{j-s+1} - \delta x_{n-s+1}) \\ & + x_{j-s}(\delta x_{n-s+1} - \delta x_{k-s+1}) \, , \end{aligned}$$

$$\begin{aligned} G\left(\delta \boldsymbol{X}_{n-s}, \delta \boldsymbol{X}_{n-s+1}\right) \equiv\ & \delta x_{n-s}\left(\delta x_{k-s+1}-\delta x_{j-s+1}\right) \\ & +\delta x_{k-s}\left(\delta x_{j-s+1}-\delta x_{n-s+1}\right) \\ & +\delta x_{j-s}\left(\delta x_{n-s+1}-\delta x_{k-s+1}\right) , \end{aligned} \tag{7}$$

where

$$\boldsymbol{X}_{n-s}=\left\{x_{n-s}, x_{k-s}, x_{j-s}\right\} , \quad \delta \boldsymbol{X}_{n-s}=\left\{\delta x_{n-s}, \delta x_{k-s}, \delta x_{j-s}\right\} ,$$

and $\boldsymbol{x}_k, \boldsymbol{x}_j$ are the nearest neighbors of $\boldsymbol{x}_n$. Indices are defined as

$$\left\{n, j, k : \boldsymbol{x}_n, \boldsymbol{x}_k, \boldsymbol{x}_j \in \boldsymbol{X}_n^{\mathrm{NN}}\right\} .$$

In the case of uncorrelated noise and under the assumption that the introduced corrections completely reduce the noise effect $\delta x_s=-\eta_s \quad (\forall_{s=1,\ldots,N})$ one can neglect the nonlinear terms in Eqs. (7) *i.e.*

$$\sum_{s=0}^{m} G\left(\delta \boldsymbol{X}_{n-s}, \delta \boldsymbol{X}_{n-s+1}\right) \cong 0 \quad \forall\, n=1, \ldots, N. \tag{8}$$

In the equation (8) we use the fact that $\langle\eta_i\rangle=0$ and $\langle\eta_i \eta_j\rangle \sim \delta_{ij}$.

Taking into account the assumption (8) one can write the following *linear equation* for the problem (5)

$$\boldsymbol{M} \cdot \delta \boldsymbol{X}=\boldsymbol{B}, \tag{9}$$

where $\boldsymbol{M}$ is a matrix containing constant elements, $\boldsymbol{B}$ is a constant vector, and $\delta \boldsymbol{X}^T=(\delta x_1, \delta x_2, \ldots, \delta x_N, \lambda_1, \lambda_2, \ldots, \lambda_N)$ are vector dependent variables (T — transposition). In practice it is very difficult or even impossible to find the solution of the equation (9) for large N. First, it is time consuming to solve a linear equation with a matrix $2N \times 2N$ matrix for $N>1000$. Second, when $\boldsymbol{M}$ becomes singular the estimation error of the inverse matrix M^{-1} is very large. Third, we cannot always find the true nearest neighbors (the set $\boldsymbol{X}_n^{\mathrm{NN}}$ for clean dynamics) from the noisy data $\{x_i\}$. Taking into account the above reasons it is useful to replace the global minimization problem (5) by N local minimization problems related to the nearest neighborhood $\boldsymbol{X}_n^{\mathrm{NN}}$. The corresponding local functionals to be minimized are

$$\begin{aligned} S_n^{\mathrm{NN}} &= \sum_s\left(\delta x_s\right)^2+\lambda_n \boldsymbol{C}_n^m=\min , \\ \forall\, n &= 1, \ldots, N , \quad \text{where} \quad \left\{s : x_s \in \boldsymbol{X}_n^{\mathrm{NN}} \text{ or } x_s \in \boldsymbol{X}_{n+1}\right\} . \end{aligned} \tag{10}$$

We can consider the minimization problem (10) as a certain approximation of (5). The global problem (5) is equivalent to Eq. (9) with $2N$ unknown variables that should be found single-time. The problem (10) is equivalent to a system of coupled equations that should be solved several times and as a result one gets an approximate global solution. Writing Eq. (10) in the linear form *i.e.* calculating the zeros of corresponding derivatives and using Eq. (8) one gets N linear equations as follows

$$\boldsymbol{M}_n \cdot \delta \boldsymbol{X}_n^{\lambda} = \boldsymbol{B}_n \,, \quad \forall\, n = 1, \ldots, N \,, \tag{11}$$

where $\left(\delta \boldsymbol{X}_n^{\lambda}\right)^T = (\delta x_n, \delta x_k, \delta x_j, \delta x_{n+1}, \delta x_{k+1}, \delta x_{j+1}, \lambda_n)$. The matrices $\boldsymbol{M}_n$ corresponding to (10) avoid the disadvantages of (9), *i.e.* they are not singular, their dimension is small and they do not substantially depend on the initial approximation of nearest neighbors. Matrix $\boldsymbol{M}_n$ for $m = 1$ is given by

$$\boldsymbol{M}_n = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & x_{k+1} - x_{j+1} \\ 0 & 2 & 0 & 0 & 0 & 0 & x_{j+1} - x_{n+1} \\ 0 & 0 & 2 & 0 & 0 & 0 & x_{n+1} - x_{k+1} \\ 0 & 0 & 0 & 2 & 0 & 0 & x_j - x_k \\ 0 & 0 & 0 & 0 & 2 & 0 & x_n - x_j \\ 0 & 0 & 0 & 0 & 0 & 2 & x_k - x_n \\ x_{k+1} - x_{j+1} & x_{j+1} - x_{n+1} & x_{n+1} - x_{k+1} & x_j - x_k & x_n - x_j & x_k - x_n & 0 \end{bmatrix} . \tag{12}$$

Vector $\boldsymbol{B}_n$ has the form $\boldsymbol{B}_n^T = \{0, 0, 0, 0, 0, 0, -\boldsymbol{G}_n(0)\}$. Note that this matrix in one-dimensional case is the same for LPNC method given in Ref. [1], but constraints and matrix $\boldsymbol{M}_n$ will essentially differ in higher dimensions for both methods.

3. Comparison to standard LP methods

Minimizations problems used in standard LP methods and in this method Eq. (3) are not equivalent because in our case we do not have to estimate the Jacobi matrix at all. These differences in practice are as follows (a) Eq. (3) is nonlinear against corrections δx_i. The approximation in this case means a corresponding linearization. (b) For constraints in standard LP methods we do not know the exact values of Jacobi matrix $\boldsymbol{A}$. The approximation means that Jacobi matrix $\boldsymbol{A}$ is estimated from noisy data. The LP methods look for subsequent corrections to noisy data by finding of a subspace tangent to an unknown attractor corresponding to the clean dynamics and projecting noisy data on this subspace. If one tries to estimate the position of the tangent subspace, what is equivalent to estimation of the Jacobi matrix $\boldsymbol{A}$ from noisy data, the range of the neighborhood should be larger than the magnitude of noise. Such a procedure should allow to distinguish between the dominating direction (connected with system dynamics) and random directions connected with a noise (see Figs. 1(a) and 1(b)). If the noise level is very high it is not possible to use the tangent subspace to find the attractor of the clean dynamics since the range of the necessary nearest neighborhood would be very large and the linear approximation would be invalid (see Fig. 1(c)) [21]. On the other hand, if we consider the minimal-

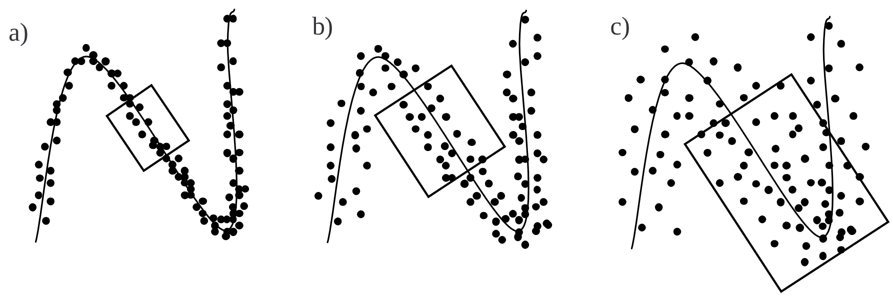

Fig. 1. The plot of the clean attractor (continuous line), noisy data connected to this attractor (black dots) as well as range of the nearest neighborhood taken under consideration to determine tangent subspace (rectangle). The level of noise for the case (c) is so high that a linear approximation is no longer valid.

ization problem (5) we do not need to find the Jacobi matrix $\boldsymbol{A}$ but only to take into account the constraint equation (3). Such an approach makes it possible to use a neighborhood smaller than the noise magnitude and in our approach the corresponding number of nearest neighbors equals 2. Note that here we encounter a flow, so nearest neighbors searching is not so biased as local projection. To find two nearest neighbors $\boldsymbol{X}_n^{\mathrm{NN}}$ to $\boldsymbol{x}_n$ we use the Delaunay triangulation [22] and the method to find is given in Ref. [1]. Searching nearest neighbors by Delaunay triangulation is very time consuming. That is why we first look for the N_{nn} nearest neighbors by means of Euclidian distance minimization and then perform the Delaunay triangulation only on this nearest neighborhood. This approach is as fast as standard nearest neighbors searching. Accordingly to the needs, *e.g.* in online noise reduction of human voice, one can think about speeding up the method. As we mention in our approach we need only two nearest neighbors as opposed to standard LP methods, so looking only for the two nearest neighbors close in time would make the searching for closest neighbors very fast and the method robust against high non-stationarity.

One can ask on the smallest sampling rate per cycle $\mathcal{RS}$ which makes the method applicable. As we have checked the method works even for the rate $\mathcal{RS}$ comparable with the m parameter used in Eq. (3) but then the efficiency is smaller than for the standard LP method. The method works the best for more than $2\,m$ samples per cycle. We have taken the embedding window

$d\,\tau$ used in nearest neighbor searching as long as one cycle. The method is robust against changing the number of taken nearest neighbors, as opposed to standard LP methods what will be seen in the next section.

4. Noise reduction: examples and applications

Let us define the noise level $\mathcal{N}$

$$\mathcal{N} = \frac{\sigma}{\sigma_{\mathrm{DATA}}}\,, \tag{13}$$

where σ is a standard deviation of noise and σ_{DATA} is the standard deviation of data. The efficiency of noise reduction we calculate by means of the gain parameter which is defined as

$$\mathcal{G} = 10 \log \left(\frac{\sigma^2_{\mathrm{noise}}}{\sigma^2_{\mathrm{red}}} \right)\,, \tag{14}$$

where $\sigma^2_{\mathrm{noise}}$ is the variance of added noise and σ^2_{red} is the variance of noise left after noise reduction. The gain parameter can be transformed into another parameter: $\%R$, which says how much noise is reduced: $\%R = (1 - \sigma^2_{\mathrm{red}}/\sigma^2_{\mathrm{noise}})\,100\%$. In Fig. 2 the dependence of the $\%R$ parameter on the basic parameter $\mathcal{G}$ is presented. The gain parameter $\mathcal{G}$ is commonly used in the evaluation of the noise reduction because it gives more relevant information especially in the regime $\%R > 90\%$. We use the standard Lorenz

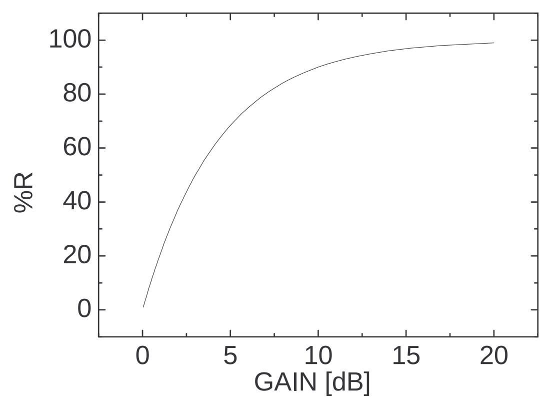

Fig. 2. The plot of the $\%R$ parameter on the gain parameter $\mathcal{G}$.

model to evaluate the performance of noise reduction methods. The Lorenz system is described by a system of three coupled differential equations

$$\begin{aligned} \dot{x} &= \xi \cdot (y - x)\,, \\ \dot{y} &= \rho x - y - xz\,, \\ \dot{z} &= xy - \beta z\,. \end{aligned} \tag{15}$$

The model is widely used for a description of Rayleigh–Benard instabilities in fluids [23] and in quantum optics for laser dynamics [24]. We use standard parameters for this system, *i.e.* $\xi = 10; \rho = 28; \beta = 8/3$, for which the standard "butterfly" attractor can be observed. To verify our method in a real experiment we have performed analysis of data generated by a nonlinear electronic circuit. The Chua circuit is one of the simplest electronic nonlinear system that exhibits chaotic behavior [25, 26]. The nonlinearity comes from two parallel connected negative resistors which are realized by amplifiers with a corresponding feedback. The Chua circuit has been studied in the presence of noise added to the outcoming signal. The noise (white and Gaussian) has been generated by an electronic noise generator. The LPNCF scheme of noise reduction improves estimations of invariant parameters. Figs. 3–5 present calculations of the correlation dimension D_2 *versus* threshold ε for the clean Lorenz system, the Lorenz system with noise and the latter parameter after noise reduction, respectively. Using a standard procedure one looks for a plateau in an intermediate threshold region. In fact one can observe that the plateau $D_2 \approx 2$ cannot be found in Fig. 4 corresponding to the noisy Lorenz system with the noise level $\mathcal{N} = 48\%$ but is well seen after the noise reduction with the LPNCF method (see Fig. 5).

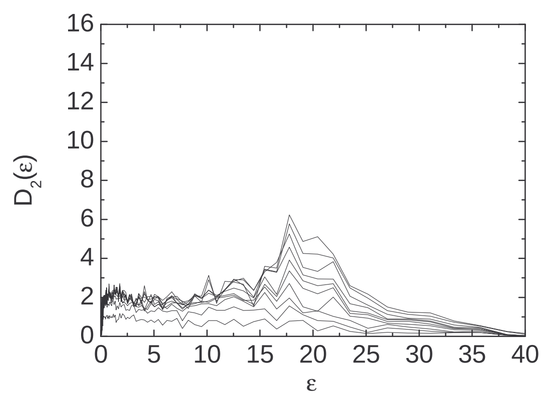

Fig. 3. The plot of the correlation dimension D_2 *versus* the threshold ε for the clean Lorenz system.

We have made a quantitative comparison between LPNCF and GHKSS method. GHKSS method [19] is the implementation of the standard LP approach that we think are optimal and most efficient. In case of both methods we use the same scheme of neighbors searching, *i.e.* the minimization of Eucleadian distance, in addition for LPNCF method we apply then the Delaunay search which is not needed for GHKSS method. For large sampling rate we did always some time averaging with curvature correc-

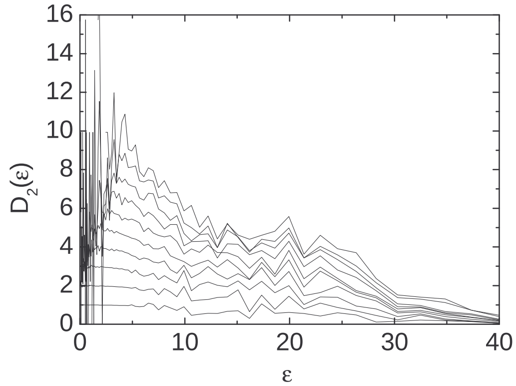

Fig. 4. The plot of the correlation dimension D_2 *versus* the threshold ε for the Lorenz system with noise $\mathcal{N} = 48\%$.

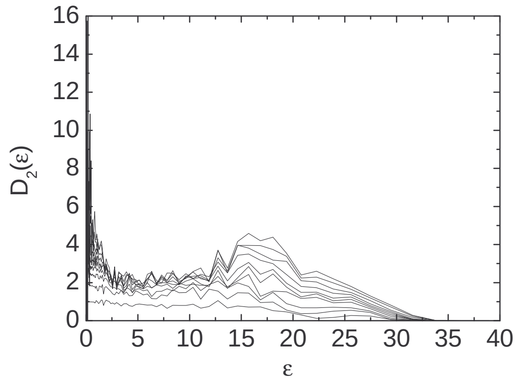

Fig. 5. The plot of the correlation dimension D_2 *versus* the threshold ε for the Lorenz system with noise after noise reduction.

tion [19] that improve the gain in both methods. For all the calculations we use 8 iterations of both methods. The projection dimension for GHKSS 3–12 and m for LPNCF method are in the range 4–64. We can say that the complexity in programming of the both methods is comparable. The LPNCF method implemented at 2.5 GHz computer is approximately 2–3 times too slow to be used for on-line noise reduction in voice and to receive a substantial improvement of the voice recognition.

Figs. 6–8 show the clean Chua attractor, with measurement noise $\mathcal{N} = 46\%$ and after noise reduction, respectively. The LPNCF approach is used for $m = 3{-}12$ when sampling ratio was about 50 per full cycle. The efficiency was $\mathcal{G} = 9.48$ ($\%R = 88.7\%$) when the GHKSS method did $\mathcal{G} = 9.33$ ($\%R = 88.3\%$).

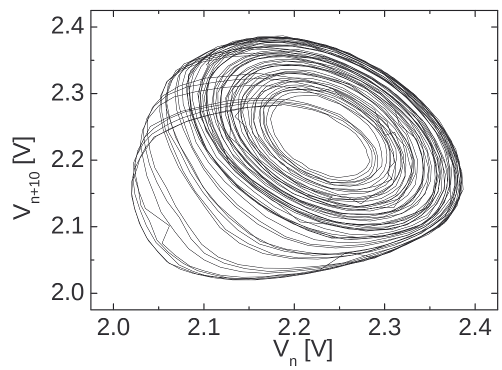

Fig. 6. The sampling corresponding to a clean trajectory in the Chua circuit (real experiment).

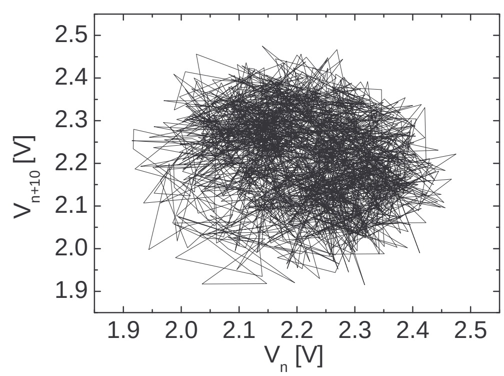

Fig. 7. The sampling received from the Chua circuit in the presence of a measurement noise $\mathcal{N} = 46\%$. Note the difference in scale.

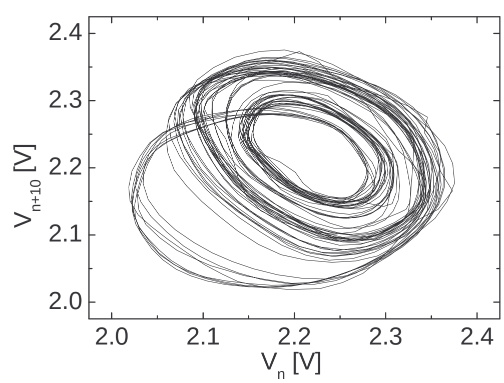

Fig. 8. The sampling of Chua circuit received after the noise reduction applied to data presented in Fig. 7.

We analyze the behavior of noise reduction by the LPNCF and the GHKSS method against increasing sampling rate per cycle $\mathcal{RS}$ (see Fig. 9). It is clear that the efficiency of LPNCF method should increase for larger sampling rate $\mathcal{RS}$. In the figure one can see for large $\mathcal{RS}$ that the LPNCF method is more efficient than GHKSS method while it is less efficient for small $\mathcal{RS}$. We compare the efficiency of these two methods for various noise levels $\mathcal{N}$ (see Fig. 10). One can see that LPNCF method in this case is more efficient for high noise levels starting from 30%. This is because for large noise levels it is difficult in GHKSS method to determine properly the tangent subspace as it was explained in the previous section. The last

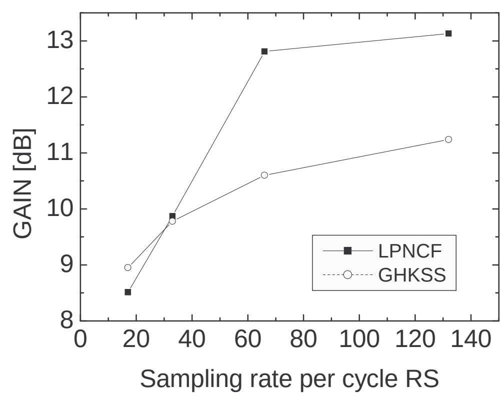

Fig. 9. The efficiency of noise reduction by LPNCF and GHKSS method for different sampling rate $\mathcal{RS}$. Here the Lorenz system [23] was used ($\mathcal{N} = 48\%$, $N = 5000$, $N_{\rm nn} = 20$).

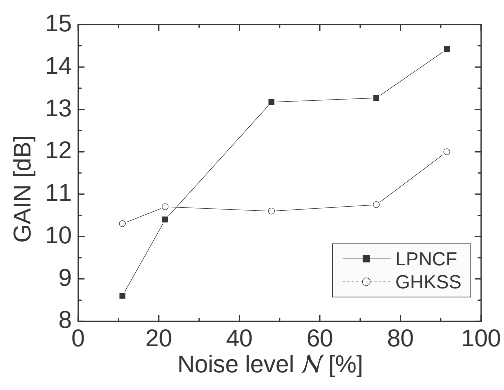

Fig. 10. The efficiency of noise reduction by the LPNCF and GHKSS method for different noise level $\mathcal{N}$. Here the Lorenz system [23] was used ($\mathcal{RS} = 66$, $N = 5000$, $N_{\rm nn} = 20$).

comparison of these two methods shows the dependence on the number of regarded neighbors $N_{\rm nn}$ (see Fig. 11). In the LPNCF method the parameter $N_{\rm nn}$ describes the number of neighbors that are used in preliminary search for candidates to the Delaunay procedure. It is shown in this figure that for small number of neighbors the LPNCF method can be used without the loss of efficiency. Such a behavior is very useful for non-stationary data, when the large number of neighbors could not be found or when because of correlated noise one should omit neighbors close in time. The GHKSS method did some noise reduction for small $N_{\rm nn}$ because here most corrections come from the time averaging which alone made $\mathcal{G} = 7.2$ ($\%R = 80.9\%$).

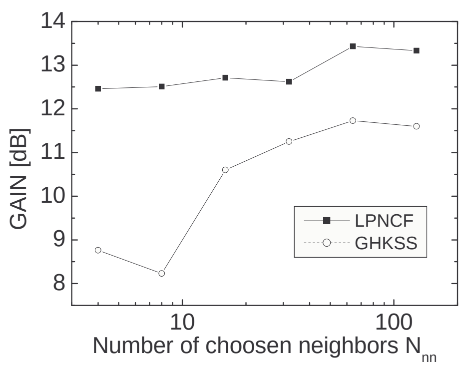

Fig. 11. The efficiency of noise reduction by the LPNCF and GHKSS method for various number of neighbors $N_{\rm nn}$. Here the Lorenz system [23] was used ($\mathcal{N} = 48\%$, $\mathcal{RS} = 66$, $N = 5000$).

We applied successfully our method to noise reduction from human voice [28]. In Fig. 12 we show a clean time series of the recorded sentence "Hello world, my name is Krzysztof Urbanowicz" (upper panel), this time series with temporally decreasing measurement noise (middle panel) and after noise reduction (bottom panel). Noise reduction was made in windows of length $N = 5000$, with parameter $m = 3$–12. The voice was recorded with sampling 22050 Hz what gives $\mathcal{RS} \approx 120$. The embedding window $d\,\tau = 100$, so it covers almost the whole cycle. Fig. 13 presents the efficiency of LPNCF and GHKSS methods. As it is suspected the LPNCF method did less for small noise levels (see for the comparison Fig. 10). Note that here we use larger number of neighbors than in Fig. 10 *i.e.* $N_{\rm nn} = 60$. For large noise level both methods work comparably. The gain of the noise reduction for whole data set shown in Fig. 13 is $\mathcal{G} = 11.4$ ($\%R = 92.8\%$) for LPNCF and $\mathcal{G} = 11.7$ ($\%R = 93.2\%$) for GHKSS. Such values of noise reduction improve significantly voice recognition for intermediate noise levels. After the performed noise reduction the background noise is not heard in the recorded signal.

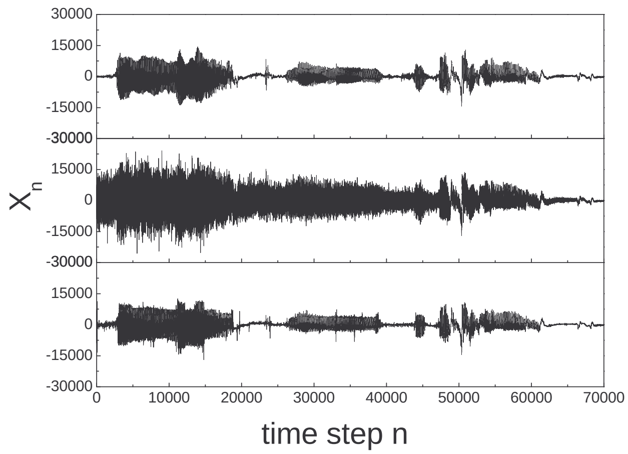

Fig. 12. The voice time series of sentence "Hello world, my name is Krzysztof Urbanowicz". From the upper panel to the bottom there are clean signals then a series with decreasing measurement noise and a noisy signal after noise reduction, respectively.

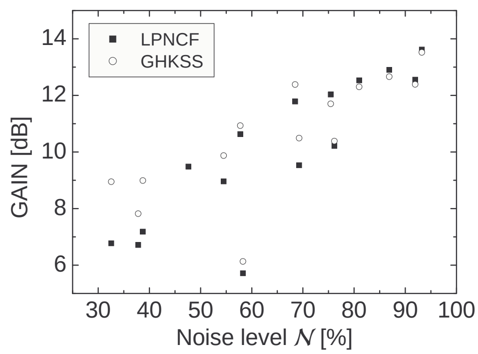

Fig. 13. The efficiency of noise reduction of human voice by the LPNCF and GHKSS method for different noise level $\mathcal{N}$ ($N = 5000$).

5. Conclusions

In conclusion, we developed the method of noise reduction design for flows. It uses a nonlinear constraints that appear due to the continuous behavior of flows. To efficiently perform the noise reduction one needs to find only two nearest neighbors. The method is robust against input parameters estimation as well as for highly non-stationarity data. We applied with success the method for noise from human voice separating.

We acknowledge Holger Kantz for fruitful discussions. The work of JAH was supported by the grant *Dynamics of Complex Systems* of Warsaw University of Technology and by the COST Action P10 *Physics of Risk*.

REFERENCES

[1] K. Urbanowicz, J.A. Hołyst, T. Stemler, H. Benner, *Acta Phys. Pol. B* **35**, 2175 (2004), `cond-mat/0308554`.

[2] H. Kantz, T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge 1997.

[3] H.D.I. Abarbanel, *Analysis of Observed Chaotic Data*, Springer, New York 1996.

[4] T. Kapitaniak, *Chaos in Systems with Noise*, World Scientific, Singapore 1990.

[5] K. Urbanowicz, J.A. Hołyst, *Phys. Rev.* **E67**, 046218 (2003).

[6] E.J. Kostelich, T. Schreiber, *Phys. Rev.* **E48**, 1752 (1993).

[7] T. Schreiber, *Phys. Rev.* **E48**, R13(4) (1993).

[8] T. Schreiber, *Phys. Rev.* **E47**, 2401 (1992).

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

[10] S.M. Hammel, *Phys. Lett.* **A148**, 421 (1990).

[11] M.E. Davies, *Physica D* **79**, 174 (1994).

[12] X.P. Zhang, *IEEE Trans. Neural Networ.* **12(3)**, 567 (2001).

[13] X.R. Chen, S. Tokinaga, *IEICE Trans. Fund. Electr. E* **85A**, 2107 (2002).

[14] P. Grassberger, I. Procaccia, *Phys. Rev. Lett.* **50**, 346 (1983).

[15] S.J. Julier, J.K. Uhlmann, H.F. Durrans-Whyte, *IEEE Trans. on Autom. Contr.* **5(3)**, 477 (2000).

[16] A. Effern, K. Lehnertz, T. Schreiber, T. Grunwald, P. David, C.E. Elger, *Physica D* **140**, 247 (2000).

[17] R. Cawley, G.H. Hsu, *Phys. Rev.* **A46**, 3057 (1992).

[18] T. Sauer, *Physica D* **58**, 193 (1994).

[19] P. Grassberger, R. Hegger, H. Kantz, C. Schaffrath, T. Schreiber, *Chaos* **3**, 127 (1993).

[20] D. Broomhead, D. Lowe, *Complex Syst.* **2**, 321 (1988).

[21] L. Matassini, H. Kantz, J. Holyst, R. Hegger, *Phys. Rev.* **E65**, 021102 (2002).

[22] S. Allie, A. Mees, K. Judd, D. Watson, *Phys. Rev.* **E55**, 87 (1997).

[23] E.N. Lorenz, *J. Atmos. Sci.* **20**, 130 (1963).

[24] G. Kociuba, N.R. Heckenberg, *Phys. Rev.* **E68**, 066212 (2003).

[25] S. Wu, Proceedings of the IEEE, vol. 75, No. 8 (1987).

[26] L.O Chua, G.-N. Lin, *IEEE Trans. Circuits Syst.* **37**, 885 (1990).

[27] V.S. Anishchenko, M.A. Safonova, L.O. Chua, *Journ. Circuits Syst. Comp.* **3**, 553 (1993).

[28] R. Hegger, H. Kantz, L. Matassini, *Phys. Rev. Lett.* **84**, 3197 (2000).

[29] C. Diks, *Phys. Rev.* **E53**, R4263(4) (1996).

[30] D. Yu, M. Small, R.G. Harrison, C. Diks, *Phys. Rev.* **E61**, 3750 (2000).

[31] H. Oltmans, P.J.T. Verheijen, *Phys. Rev.* **E56**, 1160 (1997). (1997).