ELSEVIER

Physica D 121 (1998) 75–88

PHYSICA D

Dynamics from multivariate time series

Liangyue Cao *, Alistair Mees, Kevin Judd

Department of Mathematics, University of Western Australia, Nedlands, WA 6907, Australia

Received 31 January 1998; received in revised form 15 March 1998; accepted 29 March 1998
Communicated by A.M. Albano

Abstract

Multivariate time series data are common in experimental and industrial systems. If the generating system has nonlinear dynamics, we may be able to construct a model that reproduces the dynamics and can be used for control and other purposes. In principle, multivariate time series are not necessary for recovering dynamics: according to the embedding theorem, only one time series should be needed. However, for real data, there may be large gains in using all of the measurements. In this paper we examine the issues of how to use multiple data streams most effectively for modeling and prediction. For example, perhaps the data are redundant in that only a subset of the data streams is useful. And how should we embed the data, if indeed embedding is required at all? We show how these questions can be answered, and describe some numerical experiments which show that using multivariate time series can significantly improve predictability. We also demonstrate a somewhat surprising synchronization between different reconstructions. © 1998 Elsevier Science B.V.

Keywords: Multivariate time series; Variable relationships; Embedding; Prediction; Synchronization

1. Introduction

It is routine in the analysis of time series data from a nonlinear system to make a time-delay reconstruction of a phase space in which to view the dynamics of the underlying system. For a review, see for example [1,18]. This technique has led to a large number of applications which include nonlinear prediction and modeling (e.g., [5,6,15,23]), noise reduction (e.g., [8,13]), signal classification (e.g., [12]), and control (e.g., [17]).

Most of the work in the published literature concerns only scalar time series. In principle, because of the embedding theorem [22], scalar time series are *generically* sufficient to reconstruct the dynamics of the underlying systems if enough delayed coordinates are used. But in practice the situation may be different: for example, measurements of the z-coordinate of the Lorenz equations [14] cannot reconstruct the dynamics of Lorenz system because they do not resolve the x–y symmetry. We cannot therefore be sure in practical problems that any given scalar time series is sufficient to reconstruct the dynamics. Moreover, it would not be surprising if there were substantial advantages in using several different time series if they are available, especially if the system is noisy. For example, if we had two

* Corresponding author. Fax: + 618 9380 1028; e-mail: caoly@maths.uwa.edu.au.

PII S0167-2789(98)00151-1

measurements which were identical except for independent measurement noise processes, the (possibly weighted) sum of the two time series would be less noisy than either series on its own, and would therefore be more useful for prediction.

Multivariate time series data are available in many practical situations: for example, physiological data and economic data are usually multi-dimensional. In this paper, we discuss prediction of multivariate time series and some related issues. We examine the questions of how to choose which time series to use, and how to select embedding dimensions and the time delays. We show an example in which univariate time series give poor predictions, but multivariate time series give good predictions.

An important issue is the relationships between different measurements. Abarbanel [2] discussed this issue in the context of a non-predictive relationship. We will discuss how to identify both predictive and non-predictive functional relationships.

We also show that synchronization can occur between a reconstructed system built from multivariate time series and the original system with some common variable as a driving signal; this might be taken as an indication that our models capture important features of the dynamics.

2. Relationships between different time series

Given a set of multichannel data, the first question we might ask is whether all the channels are needed. If any channel can be predicted *exactly* from the others then it is not needed in any sufficiently sophisticated modeling technique, although given the limitations of current nonlinear modeling methods, it might be useful to keep it anyway unless the predictive relationship is rather simple.

This means that sometimes we can manage with a subset of the channels, or more likely with a smaller number of time series derived from the original ones. Questions such as this can be answered in the linear case by singular value decomposition [4,16] or by reduced autoregressive modeling [10]. If the dynamics is nonlinear, the problem is considerably harder.

Suppose we have an M-dimensional time series $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, where $\mathbf{X}_i = (x_{1,i}, x_{2,i}, \ldots, x_{M,i})$, $i = 1, 2, \ldots, N$. Naively, we might try to partition the given M time series, i.e., $\{x_{i,1}, x_{i,2}, \ldots, x_{i,N}\}$, $i = 1, 2, \ldots, M$, into subsets indexed by $I_1, I_2, \ldots, I_K$ where all the time series in, say, I_1 can be predicted from one another. We could then replace all the time series in I_1 with a single one, perhaps a weighted mean or something cleverer. Repeating this with the other subsets gives us $K < M$ time series that are pairwise independent to within our ability to model relationships.

This would not be too computationally burdensome if M were not large: one could try each time series in turn and try to use it to predict all of the others, giving $(M-1)M$ modeling attempts for the first partitioning. Things are not quite that simple, however, since it may be that the time series X_1 and X_2 together can predict the time series X_3 but neither can do so on its own. We therefore have $O(2^M)$ models to build in the general case, which will almost always be prohibitive. In practice, we have to resort to simplifying the process, but the general idea is the motivation for the work in this paper.

Because we are interested in dynamical relationships we cannot simply find a static relationship defining (say) $x_{1,n}$ as a function of $x_{2,n}, x_{3,n}, \ldots, x_{M,n}$. Rather, we have to find $x_{1,n+1}$ as a function of $z_{1,n}, z_{2,n}, \ldots, z_{M,n}$ where

$$z_{i,n} = (x_{i,n-\tau_i}, x_{i,n-2\tau_i} \ldots, x_{i,n-(d_i-1)\tau_i});$$

that is, we use time delay embeddings. Thus the complexity of an exhaustive modeling approach is even worse than indicated above. This also means that we cannot simply find the mutual information [7] between different time

series as an indicator of dependence: if the mutual information between any pair of time series is significant, this only tells us that there is a static relationship. We would have to estimate the mutual information of a large number of pairs of variables with different lags, and given the expense of calculating mutual information and the likely unreliability of the estimates, it seems that a different approach is required.

3. Embedding multivariate time series

Recall that we are considering an M-dimensional time series $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, where $\mathbf{X}_i = (x_{1,i}, x_{2,i}, \ldots, x_{M,i})$, $i = 1, 2, \ldots, N$. As in the case of scalar time series (where $M = 1$), let us start by making a time-delay reconstruction:

$$\begin{aligned}\mathbf{V}_n = (&x_{1,n}, x_{1,n-\tau_1}, \ldots, x_{1,n-(d_1-1)\tau_1},\\ &x_{2,n}, x_{2,n-\tau_2}, \ldots, x_{2,n-(d_2-1)\tau_2},\\ &\cdots \quad \cdots \quad ,\\ &x_{M,n}, x_{M,n-\tau_M}, \ldots, x_{M,n-(d_M-1)\tau_M}),\end{aligned} \tag{1}$$

where τ_i, d_i, $i = 1, \ldots, M$, are the time-delays and the embedding dimensions, respectively. Following the embedding theorem [21,22] (we are not aware of a written-out proof for multivariate time series but we are taking the natural generalization), there exists in the generic case a function $F : \mathbb{R}^d \to \mathbb{R}^d$ ($d = \sum_{i=1}^{M} d_i$) such that

$$\mathbf{V}_{n+1} = F(\mathbf{V}_n), \tag{2}$$

if d or each d_i is sufficiently large. We can also write the equivalent form of (2):

$$\begin{aligned} x_{1,n+1} &= F_1(\mathbf{V}_n),\\ x_{2,n+1} &= F_2(\mathbf{V}_n),\\ &\vdots \qquad\quad \vdots\\ x_{M,n+1} &= F_M(\mathbf{V}_n).\end{aligned} \tag{3}$$

Then the problem is how to choose the time delays τ_i and embedding dimensions d_i, $i = 1, \ldots, M$, so that Eq. (2) or Eq. (3) holds.

There are several methods for choosing the time delay for a scalar time series, such as mutual information [7] and autocorrelation [3]. In the present problem we can use those methods to find τ_i separately for each scalar time series $x_{i,1}, x_{i,2}, \ldots, x_{i,N}$. One could also use some visual method such as plotting $x_{i,m+\tau}$ against $x_{i,m}$ ($m = 1, \ldots, N-\tau$). It is important to choose a good time-delay for several reasons, and in particular because a good choice can make the necessary embedding dimension lower. In this paper we use mutual information to choose the time-delay τ_i for each scalar time series.

The issue we are concerned with is to choose embedding dimensions from multivariate time series data. There are several possible extensions of methods for scalar time series to work with multivariate time series: in this paper we propose a general approach which is applicable to any kinds of time series, either univariate or multivariate, and in particular to the special class of multivariate time series which are known as input/output time series.

The basic idea is to utilize the continuity of the function F, of (2) or the functions $F_1, F_2, \ldots, F_M$ of (3) if they exist. Intuitively, for d_i, $i = 1, \ldots, M$, to qualify as embedding dimensions means that, if $\mathbf{V}_n$ and $\mathbf{V}_l$ are close, then $x_{i,n+1}$ and $x_{i,l+1}$ are close as well for each $i = 1, 2, \ldots, M$, because the functions $F_1, F_2, \ldots, F_M$ are continuous.

In practical calculations it is better to consider each equation in (3) separately, because different F_i may need different *minimum* embedding dimensions. (Think about the Lorenz system: $\dot{x}$ only depends on y and x but not on z, while $\dot{y}$ depends on all the coordinates x, y and z.)

Without loss of generality, we consider the problem of finding the embedding dimensions of F_1 in (3), having already chosen the time delays $\tau_i, i = 1, \ldots, N$. For any given set of dimensions $d_1, \ldots, d_M$, we get a series of delay vectors $\mathbf{V}_n$ defined in (1), where $n = \max_{1\le i\le M}(d_i - 1)\tau_i + 1, \ldots, N$. For each $\mathbf{V}_n$ we find its nearest neighbor $\mathbf{V}_{\eta(n)}$, i.e.,

$$\eta(n) = \operatorname{argmin}\left\{\|\mathbf{V}_n - \mathbf{V}_j\| : j = \max_{1\le i\le M}(d_i - 1)\tau_i + 1, \ldots, N,\ j \neq n\right\}, \tag{4}$$

where we might use either the Euclidean norm $\|\mathbf{a}\| = \|(a_1, a_2, \ldots, a_d)\| = (\sum_{i=1}^{d} a_i^2)^{1/2}$ or the infinity norm $\|\mathbf{a}\| = \max_i |a_i|$. In this paper we have used the Euclidean norm.

Next, we are going to calculate the mean one-step prediction error that would be obtained from a simple nearest-neighbor predictor, which can be thought of as a piecewise constant (or locally constant) model of the dynamics, by analogy to piecewise linear (or locally linear) models. The reason we choose this very simple predictor is, as mentioned earlier, that it is necessary that its error be small if the functions F_i are to exist and be continuous. (We discuss this further in Remark 1 below.) The error is

$$E(d_1, d_2, \ldots, d_M) = \frac{1}{N - J_0 + 1}\sum_{n=J_0}^{N} |x_{1,n+1} - x_{1,\eta(n)+1}|,\quad J_0 = \max_{1\le i\le M}(d_i - 1)\tau_i + 1. \tag{5}$$

We could also consider the maximum error (for low-noise data only) or the root-mean-square error. The error measure E depends on the dimensions $d_1, d_2, \ldots, d_M$. We want to choose the embedding dimensions $d_{1_e}, d_{2_e}, \ldots, d_{M_e}$ which minimize E, i.e.,

$$(d_{1_e}, d_{2_e}, \ldots, d_{M_e}) = \operatorname{argmin}\left\{E(d_1, d_2, \ldots, d_M) :\ (d_1, d_2, \ldots, d_M) \in \mathrm{Z}^M,\ \sum_{i=1}^{M} d_i \neq 0\right\}, \tag{6}$$

where $\mathrm{Z}^M = \prod_{i=1}^{M} \mathrm{Z}$, and Z denotes all non-negative integers.

The reasons for choosing the embedding dimensions so as to minimize E are essentially the same as one would argue in the scalar case. For low values of $(d_1, d_2, \ldots, d_M)$ we do not have a proper embedding and the prediction is poor, so we get large values of E. As $d_1, d_2, \ldots, d_M$ are increased to the minimum values which give a proper embedding, we make it possible to define the functions F_i and E decreases. If the system has any positive Lyapunov exponents and there is any noise, E increases when we increase the dimensions $d_1, d_2, \ldots, d_M$ significantly beyond their minimal acceptable values, because the prediction is attempting to use data in the far past which has lost correlation with the present.

Remarks.

1. We used a nearest neighbor ("locally constant") predictor to define E. Other predictors will perform at least as well, so we are being pessimistic here: if we choose an embedding that makes this error small, better modeling methods should also give small errors with that embedding. In particular, we use locally linear models in this paper for the final predictions once the embedding has been chosen. (We have experimented with using locally

linear models to determine the embedding, but for the examples we tried, the results were the same and the computation was much more burdensome.)

2. The main advantage of the locally constant method is that it is extremely fast to calculate: nearest neighbors can be found very efficiently, for example using k-d tree methods. Moreover, there are no parameters to estimate.
3. The above method finds the optimum embedding dimensions for one-step ahead predictions. Having the best one-step predictions need not imply that one gets the best multi-step predictions as well, so if one wants to fit a predictive model to make multi-step or free-run predictions, it is sometimes useful to re-define E as follows:

$$E(d_1, d_2, \ldots, d_M) = \frac{1}{N - J_0 + 1} \sum_{n=J_0}^{N-T} \max_{1 \le k \le T} |x_{1,n+k} - x_{1,\eta(n)+k}|, \tag{7}$$

where T is the number of prediction steps to be used. Then we find the optimum embedding dimensions from (6) as before. One can also define E as

$$E(d_1, d_2, \ldots, d_M) = \frac{1}{N - J_0 + 1} \sum_{n=J_0}^{N-T} |x_{1,n+T} - x_{1,\eta(n)+T}|. \tag{8}$$

Eq. (7) is used in iterative predictions (where we obtain a k-step predictor by iterating a one-step predictor k times), and Eq. (8) is used in direct predictions (where we obtain a k-step predictor by fitting a model directly to the k-step-ahead data).

4. As mentioned earlier, we cannot usually perform an exhaustive search in practice. We set some maximum dimension $D_{\max}$ in advance, and instead of (6) we consider

$$(d_{1_e}, \ldots, d_{M_e}) = \operatorname{argmin} \left\{ E(d_1, \ldots, d_M) : \ 0 \le d_i \le D_{\max} \ \forall i, \ \sum_{i=1}^{M} d_i \ne 0 \right\}. \tag{9}$$

5. By setting $M = 1$, one can use the above method to find the minimum embedding dimension from scalar time series. We have tested several artificial time series from well-known dynamical systems such as the Lorenz equations, the Hénon map, the Ikeda map and the Mackey–Glass equation; all of our results matched the results published in the literature. We also tested many artificial multivariate time series, i.e., $M > 1$, and got very satisfactory results as will be seen.
6. There are obvious connections between the one-dimensional version of our approach and the method of false nearest neighbors [1]. It is possible that the method of false nearest neighbors could be extended directly to the multivariate case, but we have found the method described here to be suitable for predictions, which is our main application. It is argued elsewhere [11] that one cannot really separate the issue of embedding from the use to which the embedding is to be put, and since our use is modeling, we have stayed as close to the end-result as possible.

4. Predictions

In this section we provide an example, and show that good predictions are difficult to obtain from a scalar time series, but can be obtained from a two-dimensional time series. This is an example of a case where, at least given the limitations of modeling methods, a single time series may not be sufficient to recover the dynamics of the system under investigation.

We obtain embeddings using the locally constant method discussed above. For prediction, there are many sophisticated techniques available but we simply use local linear prediction. The local linear method is simple but is extensively used in scalar chaotic time series predictions in the published literature, including the first paper on chaotic time series prediction by Farmer and Sidorowich [6]. Our motivation for using this method is to provide a straightforward comparison between multivariate time series predictions and univariate time series predictions, and to see how this method performs on multivariate data.

4.1. Kicked double rotor map

The kicked double rotor map is four-dimensional and describes the time-evolution of a mechanical system known as the kicked double rotor [9]. The system is an extension of the kicked single rotor, a two-dimensional system that is described by the well-known dissipative standard map [24].

The double rotor is composed of two thin, massless rods connected as shown in Fig. 1. The first rod, of length l_1, pivots about P_1 (which is fixed), and the second rod, of length $2l_2$, pivots about P_2 (which moves). The angles $\theta_1(t), \theta_2(t)$ specify the orientations at time t of the first and second rods, respectively. A mass m_1 is attached at P_2, and masses $m_2/2$ are attached to each end of the second rod (P_3 and P_4). Friction at P_1 with coefficient v_1 slows the first rod; friction at P_2 with coefficient v_2 slows the second rod and simultaneously accelerates the first rod. The end of the second rod marked P_3 receives periodic impulse kicks at times $t = T, 2T, \ldots$, always from the same direction and with constant strength f_0. See [19] for more details.

The map is

$$\mathbf{X}_{n+1} = \mathbf{M}\mathbf{Y}_n + \mathbf{X}_n, \qquad \mathbf{Y}_{n+1} = \mathbf{L}\mathbf{Y}_n + \mathbf{G}(\mathbf{X}_{n+1}), \tag{10}$$

where $\mathbf{X} = (x_1, x_2)^{\mathrm{T}} \in S^1 \times S^1$, $\mathbf{Y} = (y_1, y_2)^{\mathrm{T}} \in \mathbb{R} \times \mathbb{R}$, and $\mathbf{G}(\mathbf{X}) = (c_1 \sin(x_1), c_2 \sin(x_2))^{\mathrm{T}}$. x_1, x_2 are the angular positions of the rods at the instant of the nth kick, $x_j = \theta_j(nT)$, while y_1, y_2 are the angular velocities of

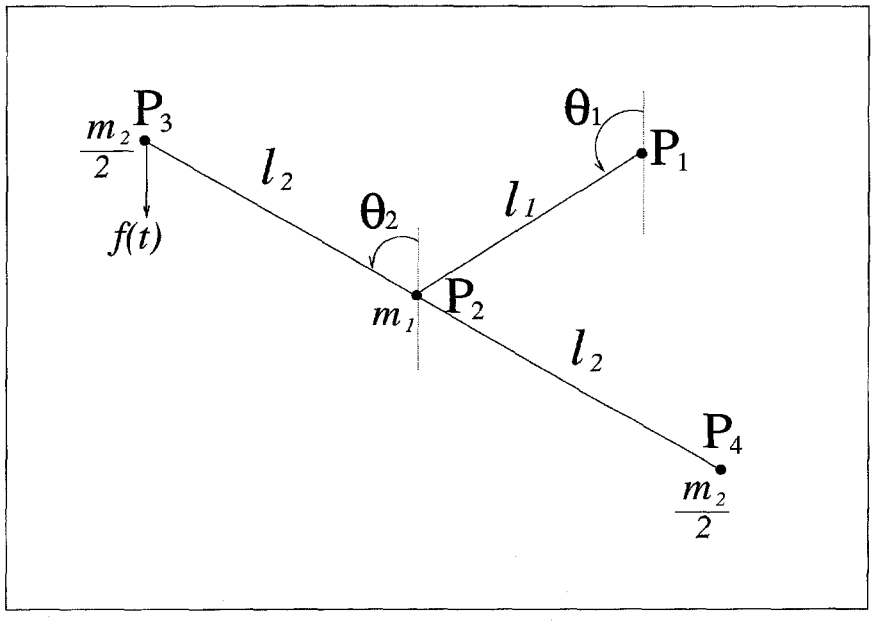

Fig. 1. The kicked double rotor [19].

the rods immediately after the nth kick, $y_j = \dot{\theta}_j(nT^+)$. S^1 is the circle $R(\mathrm{mod}\ 2\pi)$. $\mathbf{L}$ and $\mathbf{M}$ are constant 2×2 matrices defined by

$$\mathbf{L} = \sum_{j=1}^{2} \mathbf{W}_j e^{\lambda_j T}, \quad \mathbf{M} = \sum_{j=1}^{2} \mathbf{W}_j (e^{\lambda_j T} - 1)/\lambda_j,$$

$$\mathbf{W}_1 = \begin{pmatrix} a & b \\ b & d \end{pmatrix}, \quad \mathbf{W}_2 = \begin{pmatrix} d & -b \\ -b & a \end{pmatrix},$$

where $a = (1 + v_1/\Delta)/2$, $d = (1 - v_1/\Delta)/2$, $b = -v_2/\Delta$, $\lambda_{1,2} = -(v_1 + 2v_2 \pm \Delta)/2$, $\Delta = (v_1^2 + 4v_2^2)^{1/2}$. c_1 and c_2 are given by $c_j = f_0 l_j / I$, $j = 1, 2$, where $I = (m_1 + m_2)l_1^2 = m_2 l_2^2$. From now on we assume that $v_1 = v_2 \equiv v$. Then we have

$$\lambda_{1,2} = -v(3 \pm \sqrt{5})/2, \qquad a, d = (1 \pm \sqrt{5}/5)/2, \qquad b = -\sqrt{5}/5.$$

We set the parameters v, T, I, m_1, m_2, l_2 to 1, l_1 to $1/\sqrt{2}$, and the forcing f_0 to 9. The map (10) has chaotic behavior. We record 50 000 time series data of the velocities y_1, y_2 after all transients have died away by iterating the map (10), so we have a two-dimensional time series $(y_{1,1}, y_{2,1}), (y_{1,2}, y_{2,2}), \ldots, (y_{1,50\,000}, y_{2,50\,000})$.

4.2. *Predictions from a scalar time series*

If we pick a one-dimensional time series, either $y_{1,1}, y_{1,2}, \ldots, y_{1,50\,000}$ or $y_{2,1}, y_{2,2}, \ldots, y_{2,50\,000}$, and predict either y_1 or y_2 purely from this one-dimensional time series, it is very hard to get good predictions even one step ahead, especially for the y_1 data series. The results are as follows.

We first calculate the embedding dimension using the method proposed in Section 3 taking $M = 1$ because we have a one-dimensional time series. As the time series is generated from a map, 1 is the best choice of the time delay τ. We found the minimum embedding dimension for each of the time series to be 3. It seems that 3 is too small because the original system's dimension is 4. Actually, in our numerical experiments we found that neither coordinate y_1 nor y_2 could reconstruct the original system's dynamics at all, no matter how large an embedding dimension we used. For example, a set of time-delay values of y_1 cannot determine the values of x_1 coordinate of the system. But, if y_1 could recover the dynamics of the original system, a set of time-delayed values of y_1 should be able to determine the values of the x_1 coordinate because the reconstructed system would be equivalent to the original system.

Now we use local linear method to test predictions of these two scalar time series. We use the first 49 900 data points to fit the local linear predictor and test 1-step ahead predictions on the last 100 data points. The prediction results are shown in Figs. 2(a) and (b), and are obviously poor, especially on the y_1 data series. The root-mean-square errors (RMSE) of predictions are 0.7135 and 0.5160 for the y_1 and the y_2 series, respectively, where RMSE is defined by

$$\mathrm{RMSE} = \left(\sum_{i=1}^{L} (\hat{x}_{K+i} - x_{K+i})^2 \right)^{1/2} \Big/ \left(\sum_{i=1}^{L} (\overline{x} - x_{K+i})^2 \right)^{1/2}, \tag{11}$$

x is the time series, K is the number of fitting data points, L is the number of testing data points, $\hat{x}$ is the predicted value of x, and $\overline{x}$ is the mean value of time series x.

We also tried to use embedding dimensions as high as 9, but the prediction results were even worse than those we obtained using embedding dimension 3.

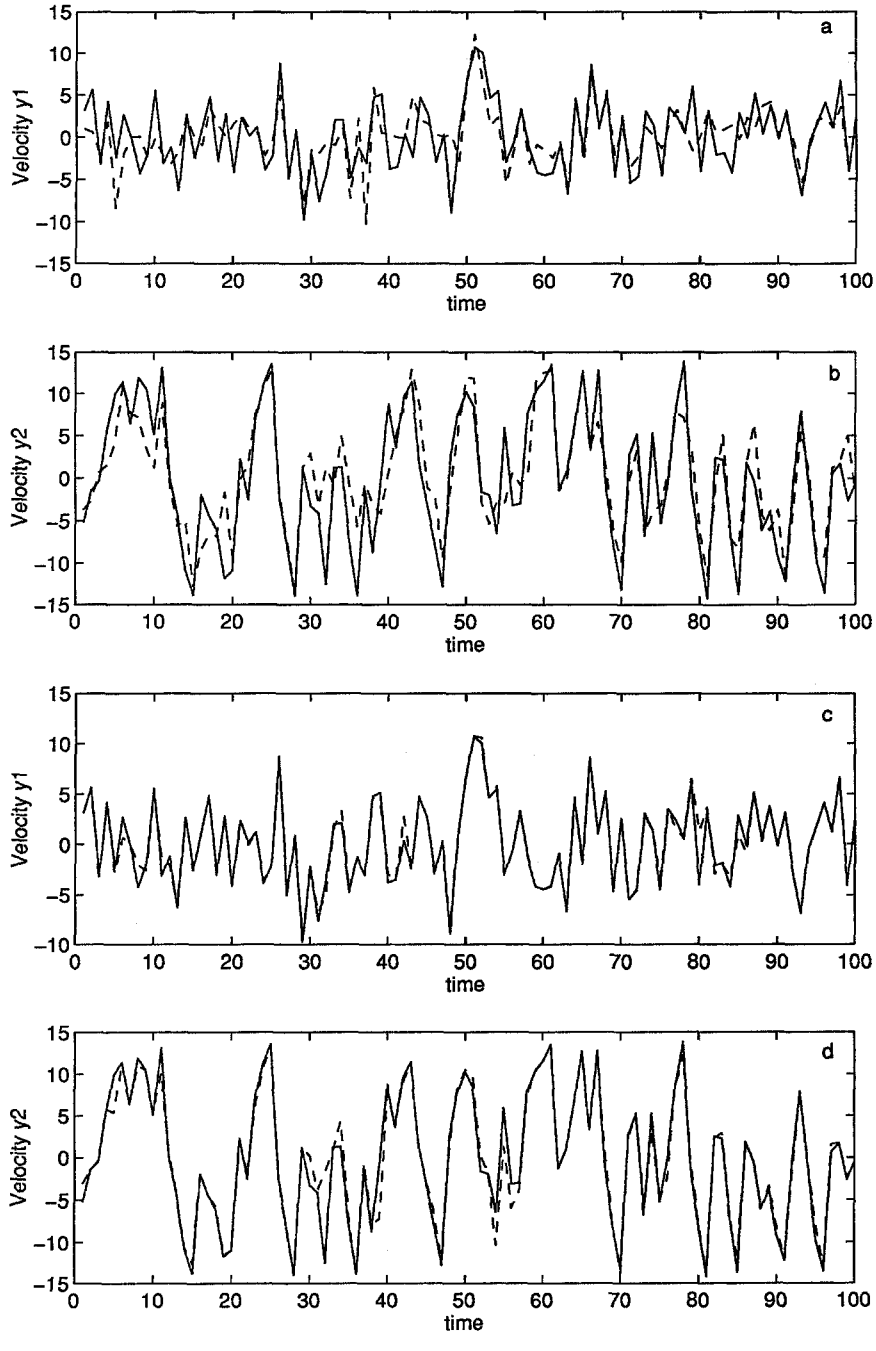

Fig. 2. One-step-ahead predictions of velocities of kicked double rotor map. Solid line is actual data points and dashed line is predicted data points. (a) Predictive model fitted based on only scalar time series of velocity y_1. (b) Predictive model fitted based on only the scalar time series of the velocity y_2. (c) Predictive model fitted based on both velocity time series, y_1 and y_2 (i.e., a two-dimensional time series). (d) Same as (c) but predictions made on y_2.

4.3. Predictions from multivariate time series

Next we assume both of the time series are available to us, so we have a two-dimensional time series. We first use the method of Section 3 to calculate the embedding dimensions $d_{1_1}, d_{2_1}, d_{1_2}, d_{2_2}$ needed to build the following predictive models:

$$\begin{aligned} y_{1,n+1} &= F_1(y_{1,n}, y_{1,n-1}, \ldots, y_{1,n-(d_{1_1}-1)}, y_{2,n}, y_{2,n-1}, \ldots, y_{2,n-(d_{2_1}-1)}), \\ y_{2,n+1} &= F_2(y_{1,n}, y_{1,n-1}, \ldots, y_{1,n-(d_{1_2}-1)}, y_{2,n}, y_{2,n-1}, \ldots, y_{2,n-(d_{2_2}-1)}). \end{aligned} \tag{12}$$

Minimizing E of Eq. (9), we find the embedding dimensions are $d_{1_1} = d_{2_1} = d_{1_2} = d_{2_2} = 2$.. Now we fit a local linear model to the functions F_1, and F_2, respectively, using the first 49 900 data points, and then test 1-step ahead predictions on the remaining 100 data points. The prediction results are shown in Figs. 2(c) and (d). The prediction results here are much better than those obtained based on scalar time series: the RMSE prediction errors are 0.1885 and 0.2259, respectively for the y_1 and the y_2 series.

5. Relationships between variables

Apart from the prediction problem, it is interesting in multivariate time series analysis to identify relations among different variables. Such relations have applications in experimental and natural systems analysis. For example, suppose we have an important variable which is very difficult to measure; in this case, if we can find the functional relation between this variable and other variables which are easy to measure, then we do not need to measure this variable in future because its values can be estimated by the relation between this variable and other variables to be measured easily. (Precisely this problem occurred in a chemical engineering problem we have met.) Another application is in data compression and transmission: given a way to express one variable in terms of others, it is not necessary to store or transmit the values of that variable. Another possible application is in control engineering, where one might want to minimize the number of control variables: if there exists a relation between one variable and others, then this variable can be eliminated from the set of control variables.

5.1. Non-predictive relations

We still suppose we have a M-dimensional time series, $\{(x_{1,n}, x_{2,n}, \ldots, x_{M,n}),\ n = 1, \ldots, N\}$ available. Without loss of generality, consider the problem of identifying relationships between $x_{l,n}$ and $x_{k_1,n}, x_{k_2,n}, \ldots, x_{k_J,n}$, where for $j = 1, 2, \ldots, J$ $(J < M)$ $1 \leq k_j \leq M$ and $k_j \neq l$. A *non-predictive* relation is

$$\begin{aligned} x_{l,n} = F(&x_{k_1,n-t_{1_1}}, x_{k_1,n-t_{1_2}}, \ldots, x_{k_1,n-t_{1_{d_1}}}, x_{k_2,n-t_{2_1}}, x_{k_2,n-t_{2_2}}, \\ &\ldots, x_{k_2,n-t_{2_{d_2}}}, \ldots, x_{k_J,n-t_{J_1}}, x_{k_J,n-t_{J_2}}, \ldots, x_{k_J,n-t_{J_{d_J}}}), \end{aligned} \tag{13}$$

where t_{j_i} can be negative or non-negative. If, for example, $t_{1_1} < 0$, say $t_{1_1} = -1$, then the current value of $x_{l,n}$ at the time n is determined by the value of $x_{k_1,n+1}$ at the *future* time $n + 1$ and other values. This is why we call this relation as non-predictive relation.

Such a relation exists if Takens' embedding theorem is applicable. For example, if $x_{l,n}$ can recover the dynamics of the original system, the embedding theorem implies that $x_{l,n}$ can also reconstruct any other vari-

ables which come from the original system; therefore, functional relations between $x_{l,n}$ and other variables will exist.

Abarbanel et al. [2] have studied several examples, including the Lorenz equations and various circuits, to show that non-predictive relations exist between two different variables measured from those systems. In this section, we present an example – the Kicked Double Rotor Map, where two variables are needed to reconstruct another variable. The map was given in Section 4.1. Here we use the values of angles x_1, x_2 to reconstruct the values of velocities y_1, y_2.

As in Section 4.1, we record 50 000 time series data of the angles x_1, x_2 and the velocities y_1, y_2 after all transients have diminished by iterating the map (10). Let us denote the four time series by $x_{1,1}, x_{1,2}, \ldots, x_{1,50\,000}$; $x_{2,1}, x_{2,2}, \ldots, x_{2,50\,000}$; $y_{1,1}, y_{1,2}, \ldots, y_{1,50\,000}$; and $y_{2,1}, y_{2,2}, \ldots, y_{2,50\,000}$, respectively. Then we aim to build the following relations between the velocities and the angles:

$$y_{1,n} = F_1(\mathbf{V}_{x_1}(n), \mathbf{V}_{x_2}(n)), \qquad y_{2,n} = F_2(\mathbf{U}_{x_1}(n), \mathbf{U}_{x_2}(n)), \tag{14}$$

where $\mathbf{V}_{x_i}(n)$, $\mathbf{U}_{x_i}(n)$ are time-delay vectors built from x_i time series, $i = 1, 2$; and they consist of both past and future delays. For example, if d is the embedding dimension of $\mathbf{V}_{x_i}(n)$, then

$$\begin{aligned} \mathbf{V}_{x_i}(n) &= (x_{i,n-D}, x_{i,n-D+1}, \ldots, x_{i,n}, \ldots, x_{i,n+D-1}) \quad \text{if } d = 2D, \\ \mathbf{V}_{x_i}(n) &= (x_{i,n-D}, x_{i,n-D+1}, \ldots, x_{i,n}, \ldots, x_{i,n+D}) \qquad \text{if } d = 2D+1. \end{aligned} \tag{15}$$

We use the method proposed in Section 3 to determine the embedding dimension of (14), except that we change the form of the time-delay vectors consist of both past and future values. We find that the optimum embedding dimensions are such that (14) becomes

$$\begin{aligned} y_{1,n} &= F_1(x_{1,n-1}, x_{1,n}, x_{1,n+1}, x_{2,n-1}, x_{2,n}), \\ y_{2,n} &= F_2(x_{1,n-1}, x_{1,n}, x_{2,n-1}, x_{2,n}, x_{2,n+1}). \end{aligned} \tag{16}$$

Now we use the local linear method to fit the functions F_1 and F_2 of (16), respectively, using the first 49 900 data points, and then test reconstruction of the velocities on the remaining 100 data points. The results are shown in Figs. 3(a) and (b). One can see that the reconstructions are nearly perfect. We also calculated the RMSE errors between the reconstructed and the actual values, which are 0.0134 and 0.0090, respectively, for the y_1 and the y_2 series.

5.2. Predictive relations

For the non-predictive relation shown in (13), we allowed t_{j_i} to be negative or non-negative. For a *predictive relation* we require all the t_{j_i} to be positive, so such a relation can be used to predict the future values of one variable based only on the past values of the other variables. This type of relationship is important if the variable we need to reconstruct is an indicator required to make decisions about the future; in this case, it is unreasonable to assume that the future values of other variables have already measured.

Although in principle building a predictive relation is equivalent to building a non-predictive relation, provided Takens' embedding theorem is applicable, in practice the former relation is more difficult to build than the latter one. For example, we cannot build a predictive relation between the velocity and the two angles in the kicked double rotor even though we built a non-predictive relation in Section 5.1.

We take Rössler's equations [20] to be our example in this section. The equations read

$$\dot{x} = -y - z, \qquad \dot{y} = x + 0.2y, \qquad \dot{z} = 0.2 + xz - 5.7z. \tag{17}$$

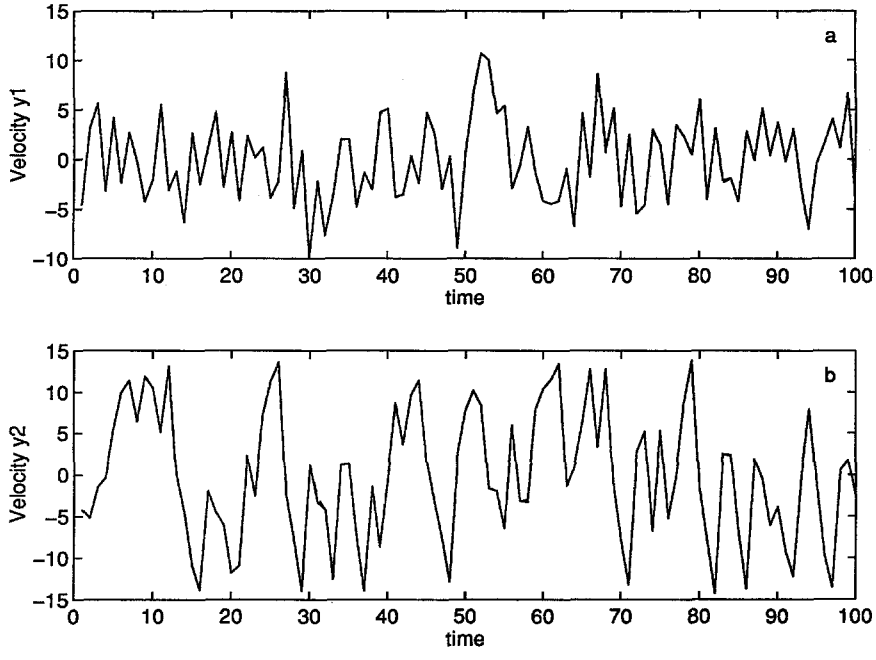

Fig. 3. Reconstruction of non-predictive relations between the angles and the velocities of the kicked double rotor map. Solid line is the actual data points and the dashed line is the reconstructed data points. Both lines are fitted perfectly. (a) Using the angles to reconstruct the velocity y_1. (b) Using the angles to reconstruct the velocity y_2.

We numerically integrate the equations with integration step 0.01, and record 5000 time series data of x and z coordinates from the numerical solution with sampling time 0.1 after transients have diminished.

Then we build the following predictive relations between the x and the z coordinates:

$$x_{n+1} = F_1(z_n, z_{n-\tau_z}, \ldots, z_{n-(d_z-1)\tau_z}), \qquad z_{n+1} = F_2(x_n, x_{n-\tau_x}, \ldots, x_{n-(d_x-1)\tau_x}), \tag{18}$$

where d_z, τ_z and d_x, τ_x are the embedding dimensions and time-delays for the z and the x coordinates, respectively. First we use the method of mutual information to find the time delays for time series x and z, respectively, giving $\tau_x = 12$ and $\tau_z = 4$. Then we use the method of Section 3 to choose the optimum embedding dimensions and we find that $d_x = 4$ and $d_z = 3$.

Now we use the local linear method to fit the functions F_1 and F_2 of (18), respectively, using the first 4500 data points and then test the models on the remaining 500 data points. The results are shown in Figs. 4(a) and (b). The reconstructions are nearly perfect. We also calculated the RMSE errors between the reconstructed and the actual values, which are 0.0070 and 0.0092, respectively for the x and the z series.

6. Synchronization

In our numerical experiments we also found an interesting phenomenon: synchronization can occur between our reconstructed system and the original system. We illustrate with Rössler's equations; essentially the same results are obtained with the Lorenz equations [14].

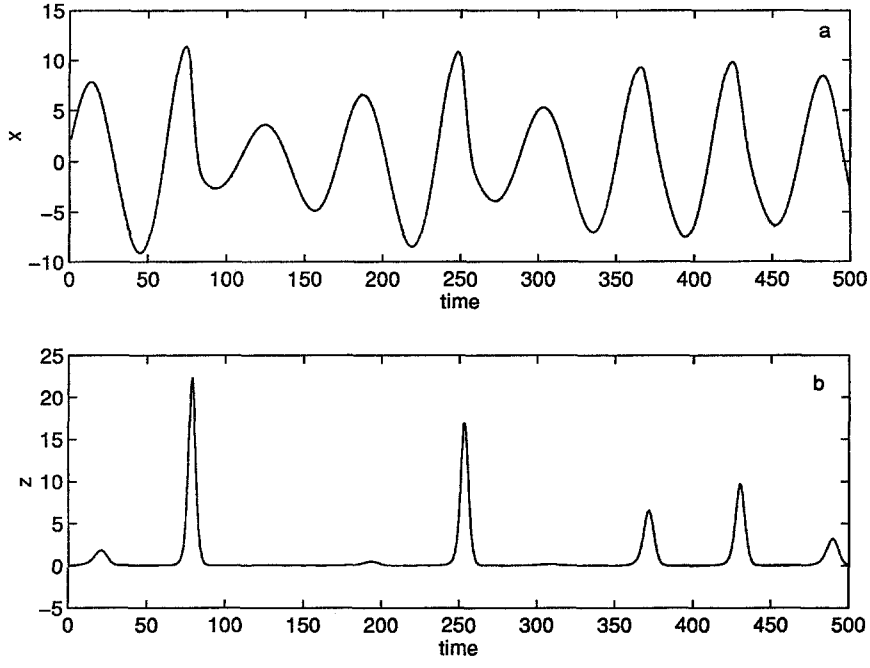

Fig. 4. Reconstruction of predictive relations between the x and the z coordinates of the Rössler equations. Solid line is the actual data points and the dashed line is the reconstructed data points. Both lines are fitted perfectly. (a) Using the z coordinate to reconstruct the x coordinate. (b) Using the x coordinate to reconstruct the z coordinate.

We record 5000 time series data of x and z coordinates in the same way as in Section 5.2. Then we build a predictive model as shown in (3), that is,

$$z_{n+1} = F(x_n, x_{n-\tau_x}, \ldots, x_{n-(d_x-1)\tau_x}, z_n, z_{n-\tau_z}, \ldots, z_{n-(d_z-1)\tau_z}), \tag{19}$$

where $\tau_x = 12$ and $\tau_z = 4$ as we chose in Section 5.2, d_x and d_z are the embedding dimensions. Using our method, we found that $d_x = d_z = 1$. So the model (19) becomes

$$z_{n+1} = F(x_n, z_n). \tag{20}$$

Now we use the first 4000 data points to fit this model with local linear method and then test synchronization on the remaining 1000 data points between z generated by this model and z generated by the Rössler equations (17) with the x coordinate of (17) as the driving signal.

We randomly choose an initial value for z_{4000} in (20); and then make *free-run predictions* on z_n ($n = 4001, \ldots, 5000$), where x is a driving signal, and its values are known at any time and are generated by (17). We have shown in Fig. 5(a) the results with four different initial values of z_{4000}; and in Fig. 5(b) the corresponding errors between the actual z values and the reconstructed z values by (20). One can see that synchronization happens very quickly.

We also tried to build the model with z as a driving signal, that is,

$$x_{n+1} = G(x_n, x_{n-\tau_x}, \ldots, x_{n-(d_x-1)\tau_x}, z_n, z_{n-\tau_z}, \ldots, z_{n-(d_z-1)\tau_z}). \tag{21}$$

In this case, synchronization did not occur between the x by this model and the x by the (17) with the z of (17) as a driving signal. Actually one can check directly from the equations that synchronization does not happen in the Rössler equations if taking the z coordinate as a driving signal.

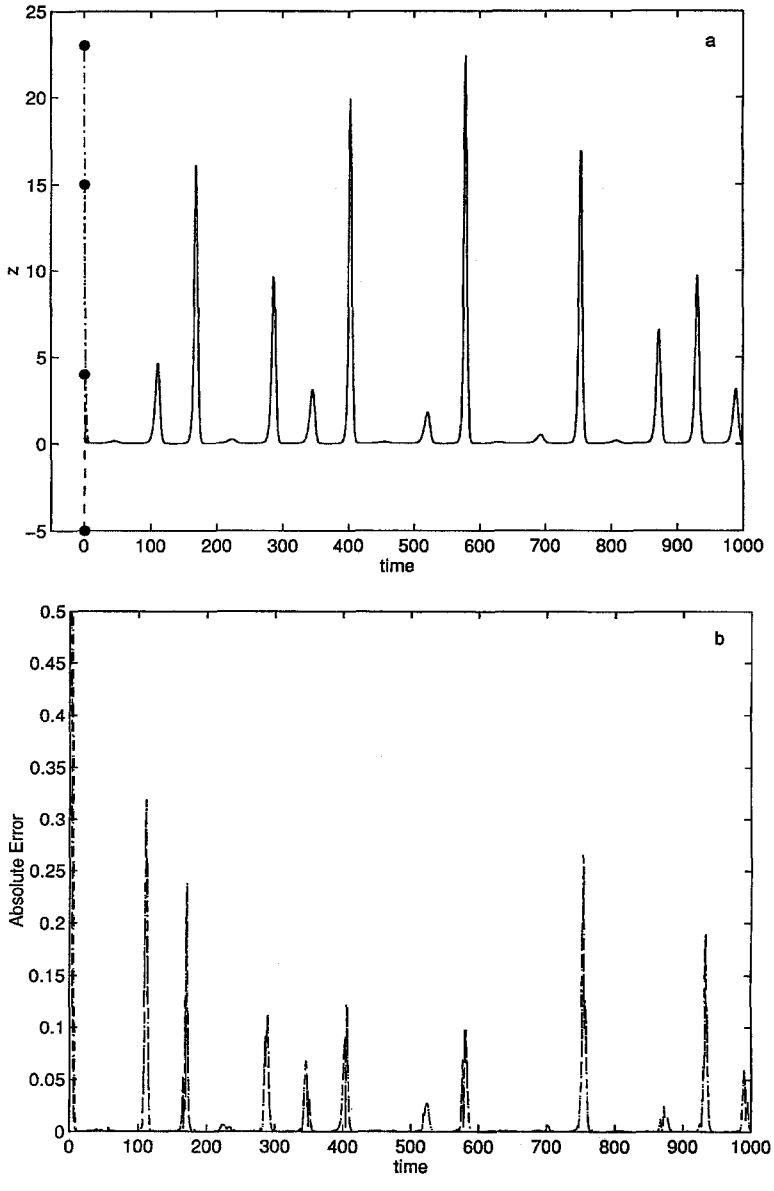

Fig. 5. (a) Synchronization between our reconstructed system and the Rössler system. Solid line is the actual data points and the dashed line or dotted line is the data points generated by our reconstructed system. The big black dots are the different initial values used in our reconstructed system to generate the iterations. The synchronization happens very quickly. (b) The absolute errors between the actual data points and the data points generated by our model for different initial values corresponding to the above picture (Fig. 5(a)).

7. Summary and comments

We have discussed multivariate chaotic time series analysis, and have proposed a simple but effective method for determining the embedding dimensions from multivariate time series. The method performed well in our tests and was not sensitive to the number of data points.

We have shown that predictions using multivariate time series can be significantly better than those using univariate time series. We have also discussed the identification of relationships (both predictive and non-predictive) between variables from multivariate time series data. Finally, we have discovered that synchronization can occur between our reconstructed systems and the original systems.

We have not discussed time series contaminated by noise in this paper. The work is still being investigated and will be described elsewhere.

Acknowledgements

This research was supported by a grant from the Australian Research Council.

References

[1] H.D.I. Abarbanel, Analysis of Observed Chaotic Data. Springer, New York, 1996.
[2] H.D.I. Abarbanel, T.A. Carroll, L.M. Pecora, J.J. Sidorowich, L.S. Tsimring, Predicting physical variables in time-delay embedding, Phys. Rev. E, 49 (1994) 1840.
[3] A.M. Albano, A.I. Mees, G.C. deGuzman, P.E. Rapp, Data requirements for reliable estimation of correlation dimensions, in: H.Degn, A.V. Holden, L.F. Olsen, (Eds.), Chaos in Biological Systems, Plenum, New York, 1987, pp. 207–220.
[4] D.S. Broomhead, G.P. King, Extracting qualitative dynamics from experimental data, Physica D 20 (1985) 217–236.
[5] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D 35 (1989) 335–356.
[6] J.D. Farmer, J.J. Sidorowich, Predicting chaotic time series, Phys. Rev. Letters 59 (1987) 845–848.
[7] A.M. Fraser, H.L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A 33 (1986) 1134–1140.
[8] P. Grassberger, R. Hegger, H.Kantz, C.Schaffrath, T.Schreiber, On noise reduction methods for chaotic data, Chaos 3 (1993) 127.
[9] C. Grebogi, E. Kostelich, E. Ott, J.A. Yorke, Multi-dimensioned intertwined basin boundaries: Basin structure of the kicked double rotor, Physica D 25 (1986) 347–360.
[10] K. Judd, A.I. Mees, On selecting models for nonlinear time series, Physica D 82 (1995) 426–444.
[11] K. Judd, A.I. Mees, Embedding as a modeling problem, Physica D (1998) in press.
[12] J. Kadtke, Classification of highly noisy signals using global dynamical models, Phys. Lett. A 203 (1995) 196.
[13] E.J. Kostelich, T. Schreiber, Noise reduction in chaotic time-series data: A survey of common methods, Phys. Rev. E 48 (1993) 1752.
[14] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (1963) 130–141.
[15] A.I. Mees, Dynamical systems and tesselations: Detecting determinism in data, Int. J. Bifur. Chaos 1 (1991) 777–794.
[16] A.I. Mees, P.E. Rapp, L.S. Jennings, Singular value decomposition and embedding dimension, Phys. Rev. A 36 (1987) 340–346.
[17] E. Ott, C. Grebogi, J. Yorke, Controlling chaos, Phys. Rev. Lett. 64 (1990) 1196–1199.
[18] E. Ott, T. Sauer, J.A. Yorke, Coping With Chaos, Wiley, New York, 1994.
[19] F.J. Romeiras, C. Grebogi, E. Ott, W.P. Dayawansa, Controlling chaotic dynamical systems, Physica D 58 (1992) 165–192.
[20] O.E. Rossler, An equation for continuous chaos, Phys. Lett. 57A (1976) 397–398.
[21] T. Sauer, J.A. Yorke, M. Casdagli, Embedology, J. Stat. Phys. 65 (1992) 579–616.
[22] F. Takens, Detecting strange attractors in turbulence, in: D.A. Rand, L.S. Young, (Eds.), Dynamical Systems and Turbulence, vol. 898, Springer, Berlin, 1981, 365–381.
[23] A.S. Weigend, N.A. Gershenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, New York, 1994.
[24] G.M. Zaslavsky, The simplest case of a strange attractor, Phys. Lett. A 69 (1978) 145.