Constraining the topology of neural networks to ensure dynamics with symmetry properties

Luis Antonio Aguirre,[1] Rafael A. M. Lopes,[1] Gleison F. V. Amaral,[1] and Christophe Letellier[2]
[1]*Programa de Pós Graduçao em Engenharia Elétrica, Universidade Federal de Minas Gerais, Avenida Antônio Carlos 6627, 31270-901 Belo Horizonte, Minas Gerais, Brazil*
[2]*CORIA UMR 6614, Université de Rouen, Avenue de l'Université, Boîte Postale 12, F-76801 Saint-Etienne du Rouvray Cedex, France*

(Received 20 June 2003; published 9 February 2004)

This paper addresses the training of network models from data produced by systems with symmetry properties. It is argued that although general networks are global approximators, in practice some properties such as symmetry are very hard to learn from data. In order to guarantee that the final network will be symmetrical, constraints are developed for two types of models, namely, the multilayer perceptron (MLP) network and the radial basis function (RBF) network. In global modeling problems it becomes crucial to impose conditions for symmetry in order to stand a chance of reproducing symmetry-related phenomena. Sufficient conditions are given for MLP and RBF networks to have a set of fixed points that are symmetrical with respect to the origin of the phase space. In the case of MLP networks such conditions reduce to the absence of bias parameters and the requirement of odd activation functions. This turns out to be important from a dynamical point of view since some phenomena are only observed in the context of symmetry, which is not a structurally stable property. The results are illustrated using bench systems that display symmetry, such as the Duffing-Ueda oscillator and the Lorenz system.

DOI: 10.1103/PhysRevE.69.026701 PACS number(s): 07.05.Tp, 07.05.Mh, 05.45.Pq, 07.05.Kf

I. INTRODUCTION

An open question in model building is how to choose the model type. Some examples in nonlinear dynamics include radial basis functions (RBF) [1], nonlinear ordinary differential equations [2–5], nonlinear difference equations [6–9], wave nets, that is, neural networks with wavelet activation functions, [10] and multilayer perceptron (MLP) networks [11,12]. Few papers up to date seem to have compared the performance of different model types, a few exceptions include Refs. [13,14,9] and the last two references consider MLP networks.

The use of MLP networks in global modeling problems applied to nonlinear dynamical systems is not as intense as other representations. One practical difficulty related to MLP networks is that in some cases (systems with symmetry) such models will hardly learn the system symmetry *exactly*. As a consequence, specific features that are not structurally stable, as pitchfork bifurcations, may not be present in the final model. This comes as a consequence of the great flexibility of the network structure and, in a sense, the blessing has become a curse. In particular, it is shown that symmetry can be easily imposed on a MLP network structure. Such constraints will result in a network that is *exactly* symmetrical (in a dynamical sense) and that will be able, in principle, to display pitchfork bifurcations and other symmetry-related phenomena [15]. In addition, networks obtained with such constraints are in general more robust to noise and to overfitting. It is believed that the use of restrictions during network training (one type of which is proposed in this paper) may open a wide range of applications of MLP networks in global modeling of nonlinear dynamics.

This paper is organized as follows: Secs. II and III provide a very brief description of the type of networks being considered, the used nomenclature, and the relation between network topology and fixed points. The main results of the paper are stated in Sec. II C for MLP networks and in Sec. III B for RBF models. Section IV presents some numerical evidences of the results introduced in Secs. II C and III B. Finally, Sec. V discusses the main points of the paper.

II. MULTILAYER PERCEPTRON NETWORKS

A. Preliminaries

The equation of a MLP network can be written as

$$y(k)=f_{\mathrm{o}}\left(b_{\mathrm{o}}+\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j\left(b_j+\sum_{i=1}^{N_i} w_{ji}^{\mathrm{h}} u_i(k)\right)\right), \qquad (1)$$

where $y(k)$ is the network output at time k, $u_i(k)$ is the ith input, w_{ji}^{h} indicates a weight of the hidden layer that connects the ith input (which is the ith output of the previous layer) to the jth neuron of the hidden layer. b is a constant, called bias, and the neuron *activation function* is f. The variables indicated by an ''o'' are related to the output neuron. Finally, N_i is the number of input signals and N_j is the number of neurons in the hidden layer. The function shown on the right-hand side of Eq. (1) is often called *feed forward* because there are no feedback loops *internal* to the network. Common choices for nonlinear activation functions are Gaussian, sigmoidal, and the hyperbolic tangent tanh(x). The weights and bias terms, on the other hand, are determined by optimization algorithms that search to minimize a cost function which usually depends on the difference between the given data and the network output. Much care should be taken in this minimization task to avoid overparametrization problems [16,17].

B. Fixed points

A network in the form of Eq. (1) can be easily written in autoregressive form with n_u lags of exogenous inputs by taking $u_i=u(k-i)$, $i=1,2,\ldots,n_u$, and n_y lags of the output, that is, $u_{n_u+\ell}=y(k-\ell)$, $\ell=1,2,\ldots,n_y$. In this case the network output will become inputs to the network in future iterations and this recursion enables the network to become a discrete dynamical function of order n_y. In steady state, with a constant input, $\overline{u}=u_i=u(k-i)$, $i=1,2,\ldots,n_u$, the network will converge to constant values if it is asymptotically stable. Such values are the steady-state values of the output and are denoted by $\overline{y}=u_{n_u+\ell}=y(k-\ell)$, $\ell=1,2,\ldots,n_y$. Therefore, in steady state the network can be written as

$$\overline{y}=f_{\mathrm{o}}\left(b_{\mathrm{o}}+\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j\left(b_j+\overline{u}\sum_{i=1}^{n_u} w_{ji}^{\mathrm{h}}+\overline{y}\sum_{i=n_u+1}^{N_i} w_{ji}^{\mathrm{h}}\right)\right). \quad (2)$$

The solutions of Eq. (2) for $\overline{u}=0$ are called the fixed points of the network. Note that in this case the input is null.

C. Conditions for symmetry

In order to have fixed points at the origin, an autonomous MLP network, as shown in Eq. (1), should satisfy

$$0=f_{\mathrm{o}}\left(b_{\mathrm{o}}+\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j(b_j)\right)=f_{\mathrm{o}}(b_{\mathrm{o}}+\mathbf{w}^{\mathrm{T}}\mathbf{f}), \quad (3)$$

where vector notation was used. Because the weights and the bias parameters are estimated in such a way as to minimize a cost function that will require a good fit of the network model to the (dynamical) data, it is in general very difficult to choose the activation functions in order to satisfy Eq. (3). Moreover, such functions should be differentiable in order to use general purpose training algorithms.

On the other hand, if the MLP network (1) does not have any of the bias terms Eq. (3) reduces to

$$0=f_{\mathrm{o}}\left(\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j(0)\right)=f_{\mathrm{o}}(\mathbf{w}^{\mathrm{T}}\mathbf{f}_0) \quad (4)$$

and this condition is easily satisfied if $\mathbf{f}_0=0$, that is, $f_j(0)=0$, $j=1,\ldots,N_j$, and $f_{\mathrm{o}}(0)=0$.

Let us denote the static function of an autonomous network without bias terms by $F(\cdot)$. It is desired to derive the conditions for this network to have fixed points that are symmetrical with respect to the origin, e.g., $\overline{y}$ and $-\overline{y}$ are fixed points. Mathematically, the following conditions should be satisfied:

$$\overline{y}-F(\overline{y})=0 \qquad \text{and} \qquad -\overline{y}-F(-\overline{y})=0.$$

In other words the following equations should hold true

$$\overline{y}=f_{\mathrm{o}}\left(\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j\left(\overline{y}\sum_{i=n_u+1}^{N_i} w_{ji}^{\mathrm{h}}\right)\right)$$

$$\overline{y}=-f_{\mathrm{o}}\left(\sum_{j=1}^{N_j} w_j^{\mathrm{o}} f_j\left(-\overline{y}\sum_{i=n_u+1}^{N_i} w_{ji}^{\mathrm{h}}\right)\right).$$

This will happen if and only if all the activation functions f_j and f_{o} are odd and continuous.

An important consequence of all the activation functions being odd functions is that if the network has fixed points other than the trivial at $\overline{y}=0$, then such fixed point will be symmetrical with respect to $\overline{y}=0$.

The main results of this section can be stated as follows. For an autonomous MLP network to have a set of fixed points that are symmetrical with respect to the trivial solution $\overline{y}=0$, it is sufficient that all the activation functions be odd and that all bias terms be zero. If the network has only one fixed point, it will be the trivial one $\overline{y}=0$. The above conditions are not *necessary*, mathematically speaking, because a multilayer perceptron with at least one hidden layer is a universal approximator. An ideal training algorithm operating under ideal conditions would learn symmetry if this feature were correctly represented in the data. In such a case, for instance, the lack of oddness of an activation function could be compensated by a nonzero bias parameter. In practice, however, training is limited by a number of factors and symmetry is lost in most cases. Because of this, different training algorithms will perform differently. However, as symmetry is a structurally unstable property, the lack of ideal conditions will destroy symmetry irrespective of the particular training algorithm used. On the other hand, it should be appreciated that the conditions derived in this section guarantee symmetry and not necessarily good dynamical performance.

It should be noticed that the above result holds for any number of hidden layers and any activation functions as long as all of them are odd. Also, the result will still hold in the case when the network has different activation functions.

III. RADIAL BASIS FUNCTION NETWORKS

A. Definition

A RBF model is a nonlinear map, acting on a d_{e}-dimensional embedding space, of the form

$$f(\mathbf{y})=\omega_0+\sum_i \omega_i\,\phi(\|\mathbf{y}-\mathbf{c}_i\|), \quad (5)$$

where $\mathbf{y}\in\mathbb{R}^{d_{\mathrm{e}}}$, $\|\cdot\|$ is the Euclidean norm, $\omega_i\in\mathbb{R}$ are the weights, $\mathbf{c}_i\in\mathbb{R}^{d_{\mathrm{e}}}$ are the so-called centers, and $\phi(\cdot):\mathbb{R}^+\to\mathbb{R}$ is a radial basis function that is usually chosen *a priori*. Such a function serves as the activation function in RBF models and is symmetrical with respect to the origin. If the function $\phi(\cdot):\mathbb{R}^+\to\mathbb{R}$ and the centers $\mathbf{c}_i$ are selected beforehand, the weights ω_i can be estimated using standard least-squares techniques [1].

In many problems of obtaining nonlinear dynamical models from data, it has been shown that it is useful to include a linear part in the RBF network with autoregressive terms and, if the system is nonautonomous, with exogenous terms such as

$$y(k)=\omega_0+\sum_i^{N_{\mathrm{c}}} \omega_i\,\phi[\|\mathbf{y}(k-1)-\mathbf{c}_i\|]+\sum_{i=1}^{n_y} a_i y(k-i)$$
$$+\sum_{i=1}^{n_u} b_i u(k-i)+e(k), \quad (6)$$

where $\mathbf{y}(k-1)=[y(k-1)\cdots y(k-n_y)u(k-1)\cdots u(k-n_u)]^{\mathrm{T}}$, N_{c} is the number of centers, and $e(k)$ is the error. Model (6) is sometimes referred to as an *affine plus* RBF [18].

B. Conditions for symmetry

To begin the discussion on how to impose symmetry during RBF training, let us consider the problem of selecting the centers $\mathbf{c}_i$ in Eq. (5) which is one of the crucial points in RBF modeling. In one of the first papers that addressed modeling nonlinear (chaotic) dynamics using RBF models, the centers—that need not be observed data—were simply chosen uniformly over the input domain [19]. Smith has briefly commented on four different ways of selecting the centers [20]. A procedure that has proven to produce parsimonious RBF models was described in detail in Ref. [21]. In that method the candidate centers are taken from the whole set of training data. Subsequently, the centers that when included in the model will maximize the increment to the explained variance of the data are effectively used.

For the sake of clarity, consider a one-dimensional RBF model composed of only two basis functions. After training (parameter estimation) the resulting static nonlinear function can be represented as in Fig. 1(a). For the same reasons as discussed in the context of MLP networks (see Sec. II) it is necessary that such a function be odd to guarantee dynamical symmetry properties. In order to constrain the static nonlinear function to be closer to an odd function, two steps are taken and will be discussed below.

1. Symmetrically chosen centers

A key point in obtaining an odd static nonlinear function based on a RBF model is to have a set of symmetrical centers. This has been observed to be useful in global modeling problems even with other model representations [22]. To this effect, the centers are chosen using any procedure, as for instance the method detailed in Ref. [21]. For each center chosen based on such a method, a "mirror" basis function with its respective center, which is symmetrical with respect to the origin to the one previously chosen, is added to the model. The mirror center will usually *not* be an observed data point in the embedding space, but that is not a problem in RBF modeling [19,20].

A pictorial representation of this is shown in Fig. 1(b). Although the resulting static function is not yet exactly odd, there is usually an enormous improvement when compared to the models for which the centers and respective basis functions are not symmetrical at all.

2. Constrained parameter estimation

From Fig. 1 it becomes clear that in order to have a perfectly odd static function, it is not sufficient to take symmetrical centers. In fact, it is necessary that the weights of two given symmetrical centers $\mathbf{c}_j=-\mathbf{c}_i$ satisfy the condition $\omega_j=-\omega_i$. This is illustrated in Fig. 1(c).

Therefore, the problem is to find a model of the form (6) such that we have the following.

(1) $\omega_0=0$.

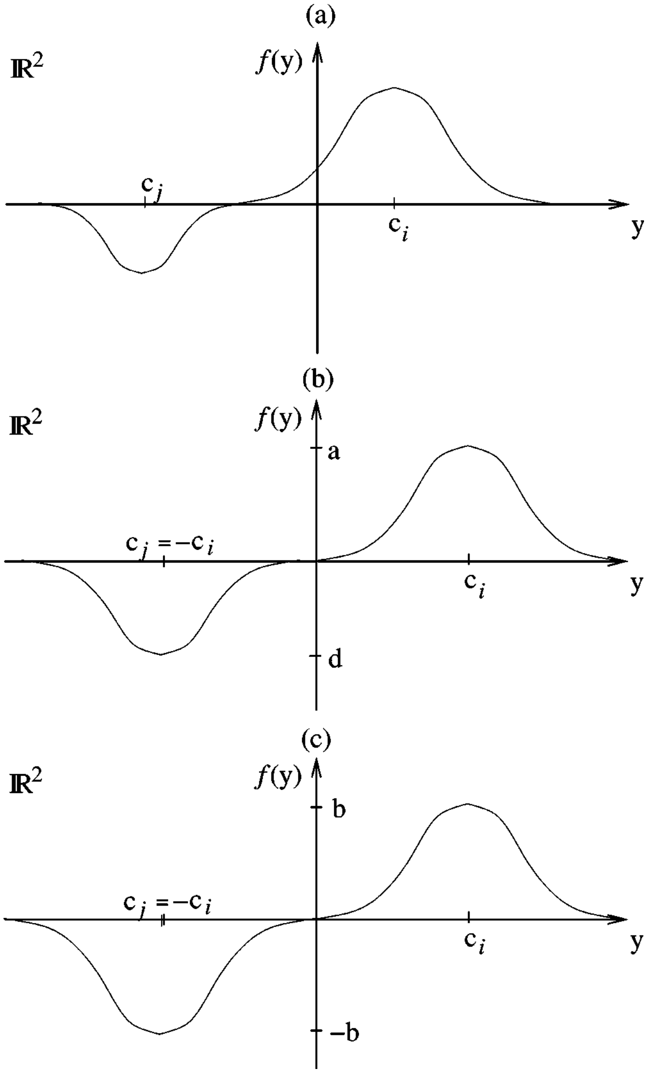

FIG. 1. Schematic illustration of how symmetry can be imposed during training of a RBF network. In (a) no restrictions are imposed, and in (b) the centers are taken to be symmetrical with respect to the embedding space but no restrictions on the weights are used. Finally, in (c) besides taking symmetrically spaced centers, the weights of each pair of symmetrical centers are constrained to be such that $\omega_i=-\omega_j$. Clearly, only in (c) is the function $f(\mathbf{y})$ odd.

(2) N_{c} is even. Moreover, $N_{\mathrm{c}}/2$ centers are chosen by some criterion and the other $N_{\mathrm{c}}/2$ centers are taken to be symmetrically placed with respect to the origin. Such centers are referred to as mirror centers.

(3) The $N_{\mathrm{c}}+n_y+n_u$ weights of the model should be estimated in such a way that the n_y+n_u weights of the linear part are constraint free but the N_{c} weights of the basis functions are constrained in pairs to satisfy $\omega_j=-\omega_i$, where i and j indicate the indices of a given center and its mirror. Apart from the constraint $\omega_j=-\omega_i$, the basis function weights are free parameters and together with the n_y+n_u weights of the linear part can be estimated to minimize a given cost function.

In what concerns the static nonlinear function, as far as the property of being odd is concerned, the presence of the linear part (which by definition yields a static odd function) will not alter the above requirements. Therefore, the aforementioned items (1) to (3) apply to both RBF and affine plus RBF models. Therefore, the following model is sought

$$y(k)=\sum_{i}^{N_c}\hat{\omega}_i\,\phi[\|\mathbf{y}(k-1)-\mathbf{c}_i\|]+\sum_{i=1}^{n_y}\hat{a}_i y(k-i)$$
$$+\sum_{i=1}^{n_u}\hat{b}_i u(k-i)+\xi(k), \qquad (7)$$

where the hats stand for estimated parameters or weights and $\xi(k)$ is the model residual at time k. Taking Eq. (7) over a window of data, the resulting set of N equations can be written in matrix form as

$$\mathbf{y}=\Psi\hat{\boldsymbol{\theta}}+\boldsymbol{\xi}, \qquad (8)$$

where $\hat{\boldsymbol{\theta}}\in\mathbb{R}^{N_c+n_y+n_u}$ is the vector of weights to be estimated and the regressors matrix $\Psi\in\mathbb{R}^{N\times(N_c+n_y+n_u)}$ is known.

It is clear that the set of constraints detailed in item (3) above can be written in the following form: $\mathbf{0}=S\hat{\boldsymbol{\theta}}$, where $\mathbf{0}$ is an $N_c/2$-dimensional vector of zeros and $S\in\mathbb{Z}^{(N_c/2)\times N_c+n_y+n_u}$ is a matrix with elements 0 or 1. For example, suppose the RBF model is composed of $N_c=4$ centers, $n_y=2$ autoregressive terms, and $n_u=1$ exogenous input, then the set of constraints

$$\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}1&1&0&0&0&0&0\\0&0&1&1&0&0&0\end{bmatrix}\begin{bmatrix}\hat{\omega}_1\\\hat{\omega}_2\\\hat{\omega}_3\\\hat{\omega}_4\\\hat{a}_1\\\hat{a}_2\\\hat{b}_1\end{bmatrix}, \qquad (9)$$

which is obviously in the form $\mathbf{0}=S\hat{\boldsymbol{\theta}}$, implies $\omega_1=-\omega_2$ and $\omega_3=-\omega_4$. Moreover, there are no constraints on the parameters of the linear part of the model. A cost function commonly used in modeling problems is the sum of squared residuals $\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}$. Therefore, the solution to the problem of finding a vector $\boldsymbol{\theta}$ that minimizes $\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}$ and satisfies the set of constraints $\mathbf{0}=S\hat{\boldsymbol{\theta}}$, that is,

$$\hat{\boldsymbol{\theta}}_{\mathrm{CLS}}=\arg\min[\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{\xi}],$$
$$\boldsymbol{\theta}{:}\mathbf{0}=S\hat{\boldsymbol{\theta}}, \qquad (10)$$

is given by Ref. [23],

$$\hat{\boldsymbol{\theta}}_{\mathrm{CLS}}=(\Psi^{\mathrm{T}}\Psi)^{-1}\Psi^{\mathrm{T}}\mathbf{y}-(\Psi^{\mathrm{T}}\Psi)^{-1}S^{\mathrm{T}}$$
$$\times[S(\Psi^{\mathrm{T}}\Psi)^{-1}S^{\mathrm{T}}]^{-1}(S\hat{\boldsymbol{\theta}}_{\mathrm{LS}}-\mathbf{0}), \qquad (11)$$

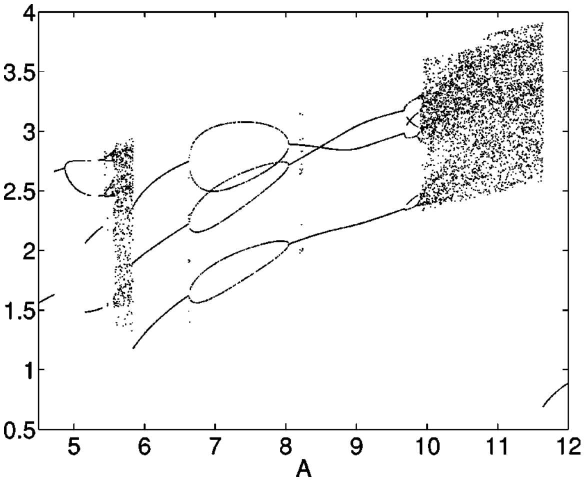

FIG. 2. Bifurcation diagram for system (12). At $A\approx6.6$ and $A\approx8.0$ a period-3 limit-cycle undergoes supercritical and subcritical pitchfork bifurcations, respectively. This is a signature of symmetry in the system.

where CLS stands for *constrained least squares* and $\hat{\boldsymbol{\theta}}_{\mathrm{LS}}$ is the standard least-squares solution which is given by the first term on the right-hand side of Eq. (11).

In closing this section an important remark is made. The training data for the RBFs when the aforementioned constraints are imposed *remain untouched*. In other words, the constraints are related to the network structure and not to the data.

IV. NUMERICAL EXAMPLES

A. Symmetrically constrained MLP networks

This section will consider the Duffing-Ueda oscillator given by Ref. [24]

$$\ddot{y}+k\dot{y}+y^3=u(t) \qquad (12)$$

with $k=0.1$ and $u(t)=A\cos(\omega t)$. In this work $\omega=1$ rad/s and the input amplitude is varied in the range $4.5\leqslant A\leqslant12$ as a bifurcation parameter. Within this range of values this system displays a rich variety of bifurcations as can be seen in Fig. 2. This figure was obtained by Poincaré sampling in the space $([y\ \dot{y}],\omega)=\mathbb{R}^2\times\mathbb{S}^1$ after discarding many cycles in order to avoid any transients. It should be noticed that in this system the input and output will always be phase synchronized. In other systems, however, where phase synchronization does not always occur, the sinusoidal amplitude is better and more rigorously treated as an initial condition rather than a bifurcation parameter [25].

This system and the respective bifurcation diagram have been previously considered in the context of NARMAX polynomials [6] and of wavelet networks and MLPs [10]. In the latter reference, it was reported that no MLP network was found that would satisfactorily reproduce the system dynamics. A wavelet network—a network with a structure close to a RBF but with wavelet-type basis functions—was obtained for which a bifurcation diagram quite similar to the one in

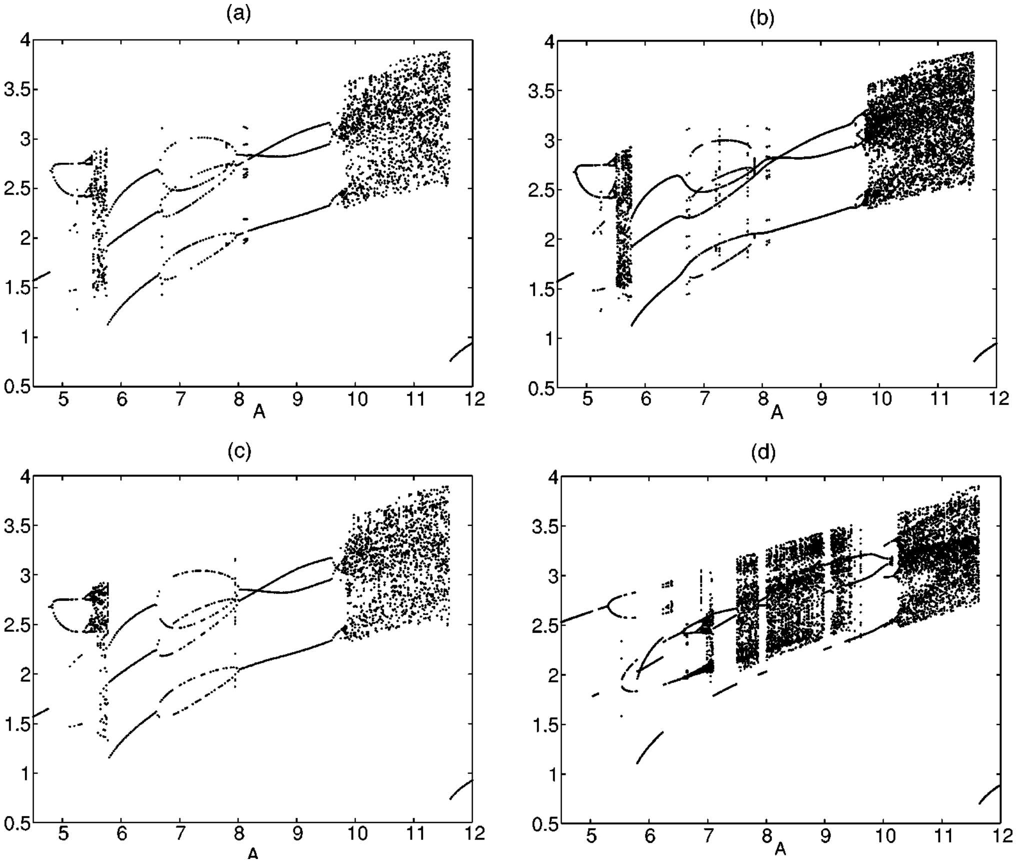

FIG. 3. Bifurcation diagrams of networks (a) HLL, (b) HLLB, (c) HHL, and (d) HHLB. Cases (a) and (c) correspond to networks without any bias terms and cases (b) and (d) to networks with bias terms.

Fig. 2 was shown [10]. In spite of the clear similarity, saddle-node bifurcations appear instead of the pitchfork bifurcations. This happens as a consequence of lack of symmetry in the network. In the remainder of this section, this assertion will not only be illustrated but also, using the main result described in Sec. II C, symmetry will be easily imposed on the MLP network during training thus enabling the final model to undergo a pitchfork bifurcation.

The following overall network topology was found to be very competitive: $u_1(k)=u(k-1)$, $u_2(k)=u(k-2)$, $u_3(k)=y(k-1)$, $u_4(k)=y(k-2)$, and $u_5(k)=y(k-3)$; just two hidden neurons, i.e., $N_j=2$, one of which was always $f_1(\cdot)=\tanh(\cdot)$, and the other was either $f_2(\cdot)=\tanh(\cdot)$ or $f_2(\cdot)=\mathrm{lin}(\cdot)$, where "lin" indicates that the output of this neuron is a linear combination of the inputs. The output neuron in both cases is linear, that is, for both $f_o(\cdot)=\mathrm{lin}(\cdot)$. These two networks are referred to as HHL and HLL, respectively, and for them the bias parameters were omitted from the topology. Other two similar networks were trained, but in such cases the bias parameters were maintained in the models. Such networks are referred to as HHLB and HLLB.

Training was performed using a Levenberg-Marquardt algorithm with ten noise terms to reduce noise influence [26]. Training was halted whenever the network error achieved a minimum of 10^{-3}. The input-output data were the same as used in the context of polynomial models as discussed in Ref. [27]. Such a data set was generated by simulation and had no noise added. Because training starts with a random choice of initial values for the weights, 100 independent simulations were carried out for each network. Figure 3 shows typical bifurcation diagrams of one MLP network of each category. As can be seen, the presence of bias parameters precludes the network to undergo pitchfork bifurcations. Also, it must be pointed out that the bifurcation seen in Fig. 3(b) at $A\approx 6.6$ is *not* a pitchfork but rather a saddle node, as illustrated in Fig. 4. Pitchfork bifurcations were only observed for HLL and HHL networks. No unconstrained network presented pitchfork bifurcations irrespective of the stopping criterion.

A second set of experiments was carried out. This time the

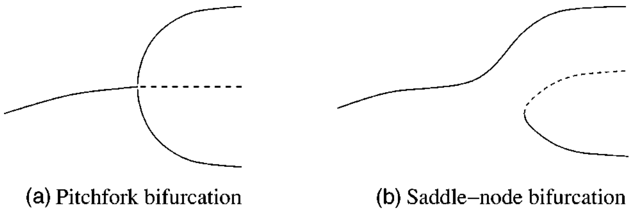

FIG. 4. By adding a simple term, the symmetry of the system is broken. In (a) the original branch loses stability giving birth to two stable branches via a pitchfork bifurcation. In (b) the original branch does not bifurcate at all. The second stable branch appears through a saddle-node bifurcation. Such a scenario is observed in Fig. 3(b) at A≈6.6.-

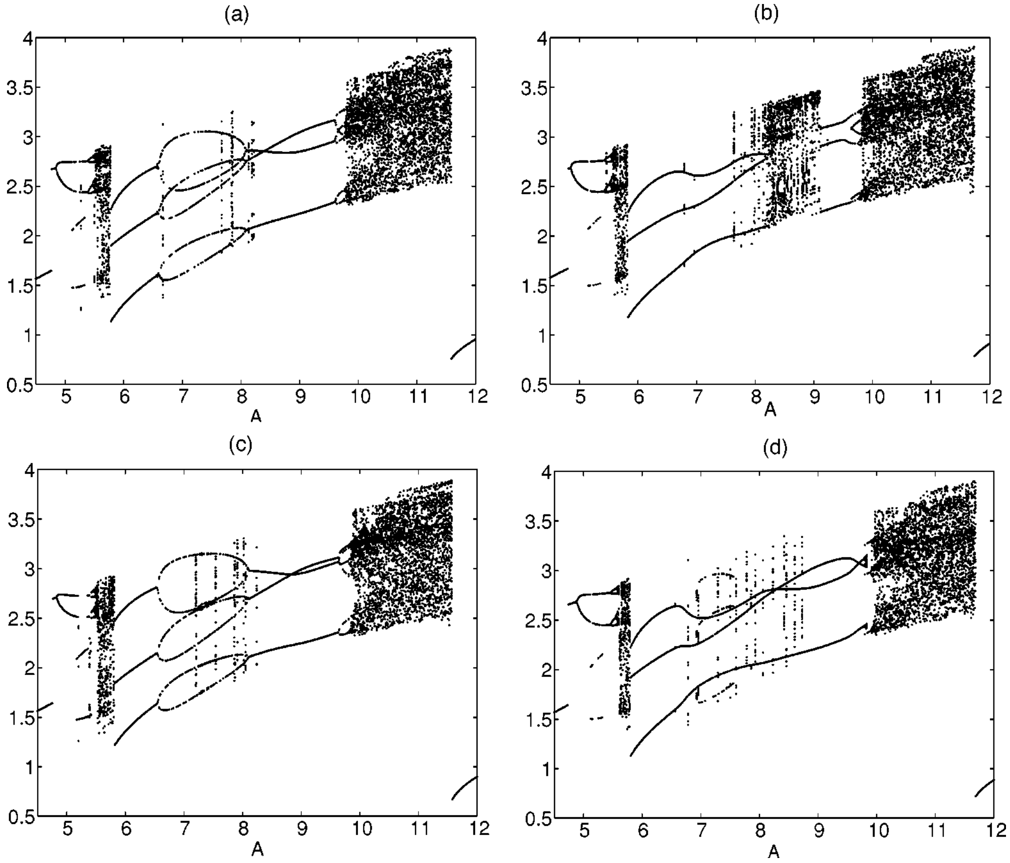

FIG. 5. Bifurcation diagrams of networks (a) HHHLL, (b) HHHLLB, (c) HHHHL, and (d) HHHHLB. Cases (a) and (c) correspond to networks without any bias terms and cases (b) and (d) to networks with bias terms.

networks had the number of neurons increased. The results are summarized in Fig. 5, where the notation follows from the previous discussion. Since the data used to train this (see Fig. 5) second set of networks are the same as for the networks corresponding to Fig. 3, it becomes clear that imposing symmetry is still effective. Moreover, in some cases [compare Fig. 3(b) with Figs. 5(b) and Fig. 5(d)] the lack of constraints renders the networks less robust to changes in the topology.

B. Symmetrically constrained RBF networks

This example uses the well-known Lorenz system

$$\dot{x}=\sigma(y-x),$$

$$\dot{y}=\rho x-y-xz,$$

$$\dot{z}=xy-\beta z. \tag{13}$$

Choosing $\sigma=10,\beta=8/3$, and $\rho=28$, the solution of Eqs. (13) settles to the well-known Lorenz attractor shown in Fig. 6.

If a homogeneous window of data on the attractor shown in Fig. 6 is taken to train RBF models, it is not very difficult to find good models with approximately symmetrical properties even without any constraints. However, inhomogeneous data are a reality in nonlinear data analysis and modeling [20]. In order to address such a situation, the data shown in Fig. 7(a) were used to train RBF models *without* any symmetry constraints. One of the best models found performed as shown in Fig. 7(b). Taking the mirror for each chosen center, a family of RBF models was obtained and the reconstructed attractor of one of the best models in that fam-

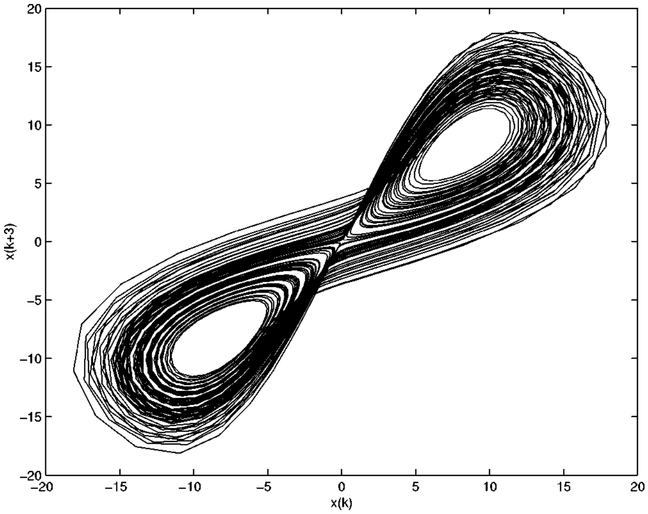

FIG. 6. Bidimensional delay reconstruction of the Lorenz attractor from the *x* variable. This reconstructed attractor has an inversion symmetry.

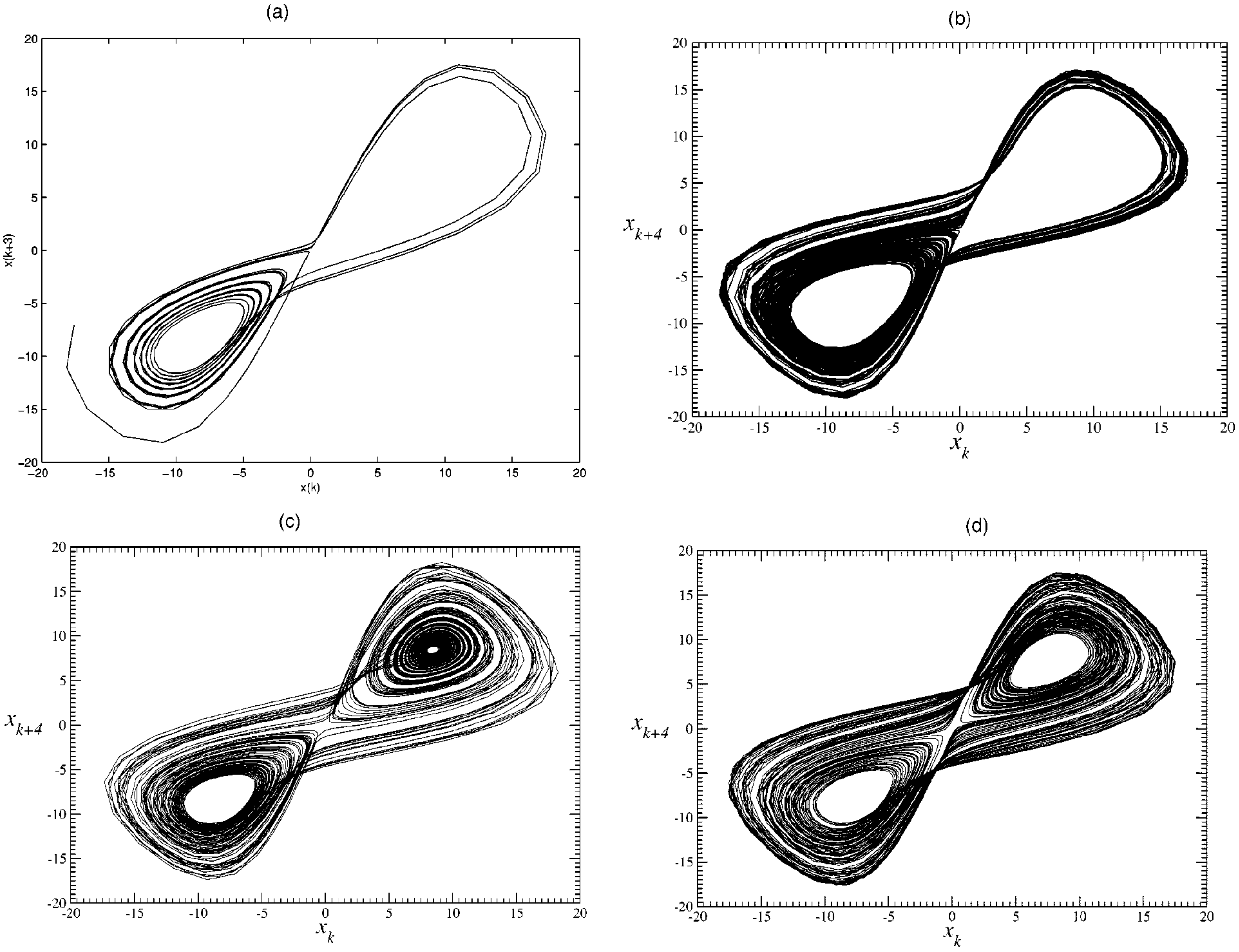

FIG. 7. (a) Inhomogeneous window of data produced by the Lorenz system. Data produced by a RBF model trained (b) without any constraints. Model with $N_c=50$, $d_e=n_y=5$; (c) with pairs of symmetrical centers but no constraints on the weights. Model with $N_c=60$, $d_e=n_y=9$; and (d) with pairs of symmetrical centers and constrained weights. Model with $N_c=80$, $d_e=n_y=6$.

ily is shown in Fig. 7(c). Although symmetry has somewhat improved, the resulting reconstructed attractor is clearly unsymmetrical. Finally, taking centers as before but imposing symmetry constraints during parameter estimation, as detailed in Sec. III B 2, a family of models was obtained. The attractor of one such model is shown in Fig. 7(d), which is very much symmetrical, as it would be expected.

In all cases the centers were chosen using the error reduction ratio criterion detailed in Ref. [21]. In order to get a broad picture in model space, the number of centers N_c and the dynamical order of the model n_y were varied over a wide range of values. No attempt was made in order to optimize the values obtained for such variables. In practice, in order to aid choosing such modeling parameters, the Schwartz criterion can be used to choose the number of basis functions [1] and a modified false-neighbor approach can be used to estimate n_y [28].

It is instructive to point out that when parameter constraints are imposed during the training of the RBF network, there are only $N_c/2$ "free" basis functions. Therefore, the model that produced the data shown in Fig. 7(d) (that has 80 centers) would be equivalent to a RBF with 40 centers and no constraints in what concerns number of free parameters. This modest increase in network size can be thought of as the price to be paid in order to guarantee symmetry.

Some results on network modeling of the Lorenz dynamics were recently published [29–31]. The latter reference is very informative as far as symmetry properties are concerned. Observing the generous amount of plots in the paper, it seems fair to conclude that both measurement and dynamical noise preclude the network to learn *exactly* the underlying symmetry. In fact, dynamical noise is even more harmful to the symmetry than measurement noise. The obvious lack of symmetry in many of the reported attractors can be clearly understood in the light of the results discussed in Sec. II C given that the referred authors use networks with bias terms and activation functions that are not odd.

Topological analysis

This section aims to provide more rigorous evidences, based on topological analysis, that the constrained RBF network produces, in fact, a symmetrical attractor. The RBF model obtained without any constraint [Fig. 7(b)] is topologically equivalent to a modified Lorenz system proposed by Rössler. The modified Lorenz system

$$\dot{x} = x - y - xz,$$

$$\dot{y} = bx - cy + d,$$

$$\dot{z} = x^2 - az \qquad (14)$$

has symmetry properties like the Lorenz system as long as $d = 0$. When $d \neq 0$, the system cannot display symmetry since it no longer obeys the relation $\mathbf{f}(\Gamma \cdot \mathbf{x}) = \Gamma \cdot \mathbf{f}(\mathbf{x})$, where $\mathbf{x} = (x,y,z)$ and $\Gamma = \mathrm{diag}[-1\ -1\ 1]$ which defines the rotation symmetry around the z axis. When $d = 0$, the system has three fixed points as the Lorenz system. One is a saddle located at the origin of the phase space and the two others are symmetric with respect to the origin. When $d \neq 0$, one of the symmetric fixed points disappears and the attractor is not symmetric anymore (Fig. 8). This attractor is topologically equivalent to the attractor solution of the RBF model estimated without any constraints [Fig. 7(b)]. Such an equivalence may be exhibited in a refined way by computing a first-return map to a Poincaré section defined by the fixed point located in the left wing [Figs. 9(a) and 9(b)]. Two increasing monotonic branches are observed. This is a very particular map quite rarely observed and peculiar to nonsymmetrical systems. This confirms that the unconstrained RBF model is not symmetrical.

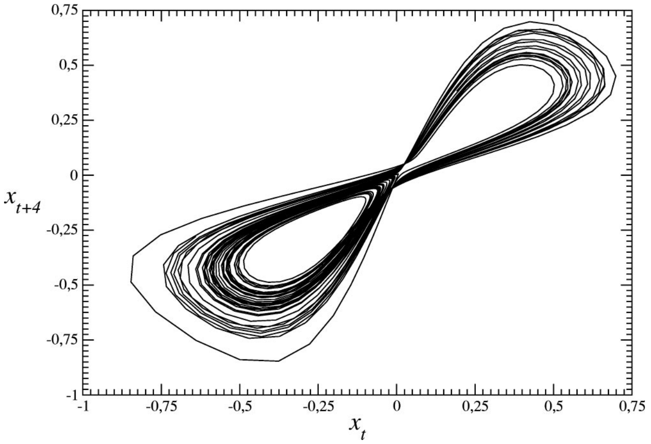

FIG. 8. Avatar of the Lorenz system for which the symmetry is broken. Parameters: $(a,b,c,d) = (0.1, 0.07, 0.38, 0.0015)$.

When the RBF model is estimated using pairs of symmet-

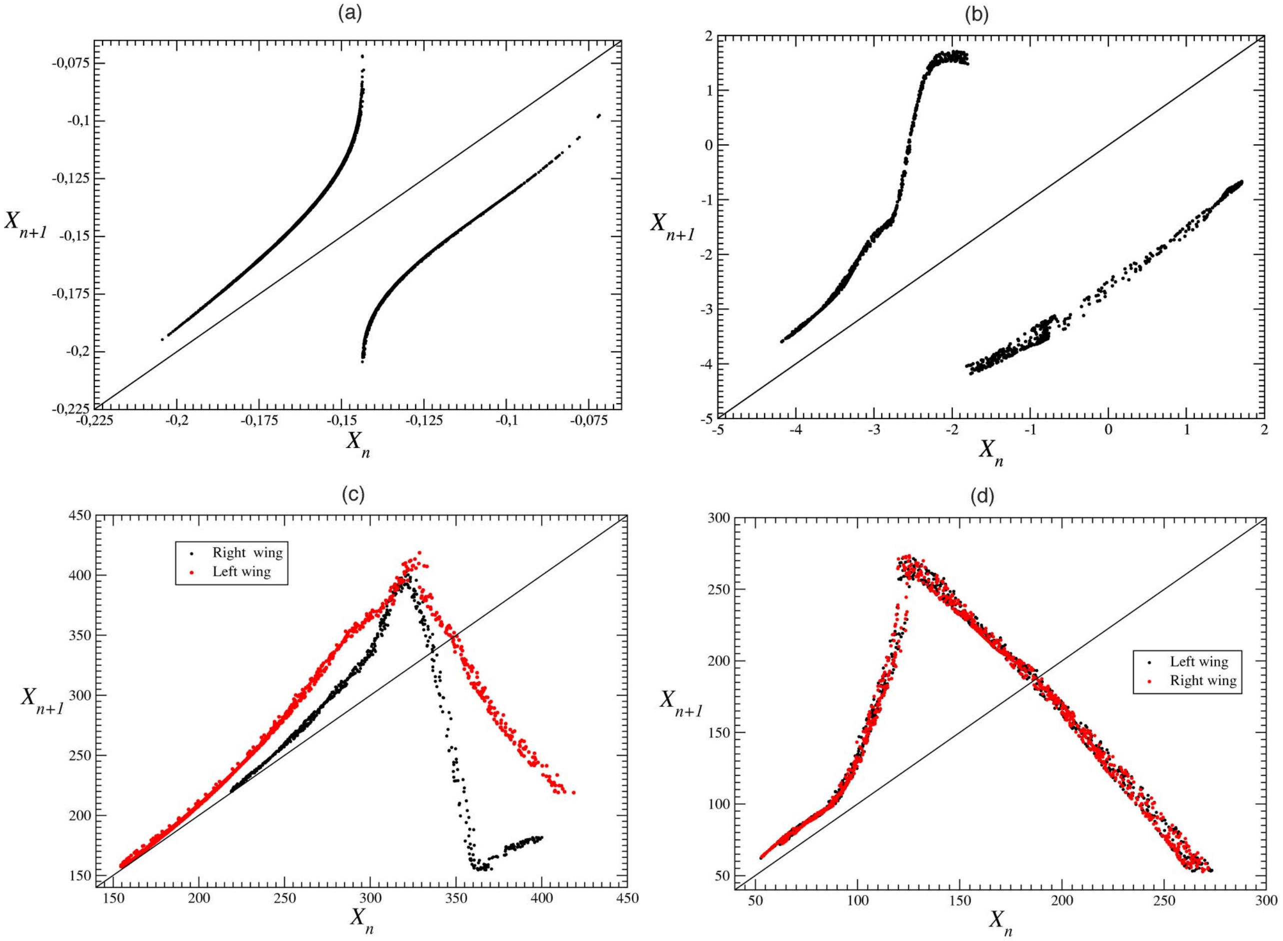

FIG. 9. First-return maps to a Poincaré section for (a) the asymmetric avatar of the Lorenz system and (b)–(d) for the different RBF models obtained. (b) Map computed for a RBF model trained without any constraints, (c) with pairs of symmetrical centers but no constraints on the weights, and (d) with pairs of symmetrical centers and constrained weights.

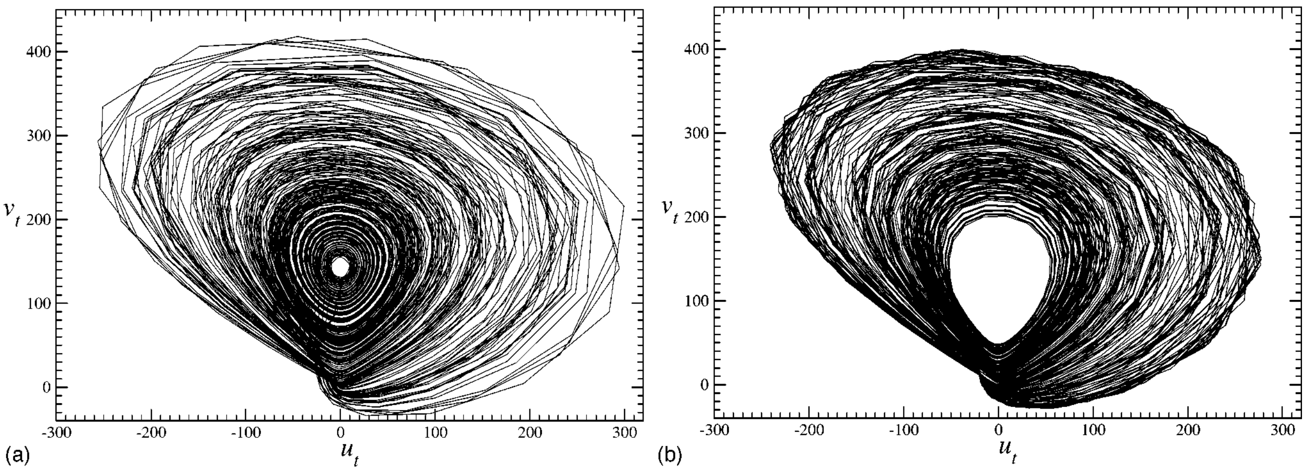

FIG. 10. Image of the attractors of the RBF model trained (a) with pairs of symmetrical centers but no constraints on the weights and (b) with pairs of symmetrical centers and constrained weights.

ric centers, both wings of the attractors are visited [Fig. 9(c)]. To the eye the attractor looks symmetric. In order to check that carefully, an analysis in the image of this attractor must be performed. An image of a symmetric attractor is one of its representations without any residual symmetry [32]. When the symmetry is exact as for the Lorenz system, a Poincaré section of the image of the Lorenz system provides the usual Lorenz map. When the symmetry is not exact, the first-return map exhibits a layered structure as observed for data recorded on an electronic Chua's circuit [33]. Thus, computing a first-return map in the image of a symmetric attractor helps to accurately check the quality of the symmetry of the dynamics.

An easy way for constructing the image of the attracting solution of the RBF model estimated from pairs of symmetric centers [Fig. 7(c)] is to apply the coordinate transformation

$$u_t = x_k^2 - x_{k+\tau}^2,$$

$$v_t = 2 x_k x_{k+\tau},$$

$$w_t = x_{k+2\tau}^2, \qquad (15)$$

which defines a local diffeomorphism from the reconstructed phase space $\mathbb{R}^3(x_k, x_{k+\tau}, x_{k+2\tau})$ into its image space $\mathbb{R}^3(u_t, v_t, w_t)$ [32]. The image attractor corresponding to the RBF model with pairs of symmetric center is shown in Fig. 10(a). The Poincaré section is then computed in a standard way. The image first-return map exhibits clearly a layered structure. The departure from the symmetry is still quite significant in this model. The contributions of each wings are obviously organized in two different ways. In particular, the right wing has three monotonic branches; the small increasing one is not present in the original dynamics [Fig. 9(c)].

When the RBF model is estimated using pairs of symmetrical centers with constrained weights, the model symmetry and dynamics are significantly improved. In particular, the image attractor [Fig. 10(b)] presents a hole as expected from the original dynamics. The attractor [Fig. 7(d)] presents a perfect symmetry, well confirmed by the image first-return map [Fig. 9(d)] since there is no longer a layered structure.

Finally, when *symmetrical* data are used in the training a very good model can be estimated *without* constraints, as can be seen in Fig. 11. The first-return map displayed in Fig. 11(b) reveals that the Lorenz cusp is better represented by the network trained from symmetrical data when compared to the networks trained with unsymmetrical data and symmetry constraints. It should be pointed out, however, that the network that produced the results in Fig. 11 is not exactly (i.e., mathematically) symmetrical.

V. DISCUSSION AND CONCLUSIONS

Symmetry is important in a number of situations, for instance, in order to reproduce some particular nonlinear phenomena such as pitchfork bifurcations and some chaos producing mechanisms [34]. In order to illustrate the application of the symmetry constraints for MLP networks, the well-known Duffing-Ueda oscillator was used as a bench test. Many simulation results have confirmed that imposing symmetry on the network increases significantly the number of networks that reproduce the desired dynamics. In a sense, symmetry of the flow can be thought of as a *dynamical* consistency hint analogous to statistical hints proposed by other authors [35]. When such constraints were not imposed the outcome was that *no* identified network was able to reproduce the pitchfork bifurcation, although some networks had saddle-node bifurcations instead, which actually had a similar appearance [compare Figs. 3(a) and 3(b)]. The need to use odd activation functions in MLP networks for the sake of symmetry has been previously observed by Bagarinao and co-workers [11,36]. On the other hand, in order to illustrate the use of the symmetry constraints for the RBF networks, the Lorenz system was used. In particular, a set of data in which the symmetry was not well represented was used for network training.

In a recent paper that considered the Duffing-Ueda oscillator [10], it has been reported that the authors were unable to find a MLP network with sigmoidal activation function

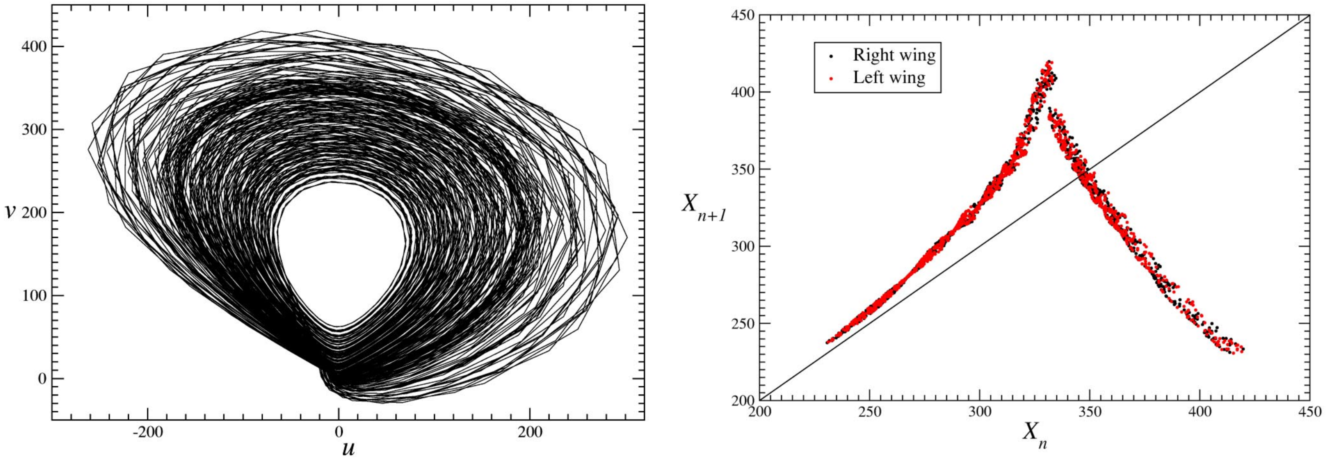

FIG. 11. Results obtained with a RBF network trained from symmetrical data but with no symmetry constraints whatsoever. (a) Image of the attractor, (b) first-return map on which the data on the left and right wings are very difficult to distinguish.

that would display the same sequence of bifurcations as the original system. A possible explanation for the reported failure is that the authors used sigmoidal activation functions which are not odd, as required by the results in Sec. II C. In such a case, even if the bias terms were omitted, symmetry would not be guaranteed. The relevance of the results in Sec. II C is highlighted by the fact that the sigmoidal and related activation functions are the most frequently used [37].

It is important to notice that, mathematically, the removal of the bias terms and the choice of odd activation functions is not necessary since a MLP network with one hidden layer is a universal approximator. What is necessary is to satisfy Eq. (3) exactly. In practice, however, this is very difficult to achieve because training is nonideal. A simple practical solution to this problem seems to be the removal of bias terms in addition to the choice of odd activation functions. If only odd functions are used, there is still no guarantee that the resulting networks will have exactly symmetrical fixed points. In fact, even using odd functions pitchfork bifurcations could not be reproduced and have been classified as being more difficult to reconstruct when compared to other bifurcations [36].

The same reasoning of the preceding paragraph applies to RBF networks which are also universal approximators. The constraints that have been developed and proposed in this paper do not add to the network extra abilities (if this were the case it would be contradictory to the fact that such networks are global approximators) but rather they restrict the network in such a way that learning becomes easier and usually more successful in especially hard problems. Without imposing symmetry the resulting networks were not perfectly symmetrical but otherwise produced attractors that to the eye seemed accurate.

The greater potential for the techniques presented in this paper is for those cases when prior knowledge of the system being symmetrical is available and such a feature is not well represented in the set of data used for training. It is worth pointing out that in some cases the measured data can be transformed in such a way as to be symmetrical. If this were the case, the techniques developed in this paper could be used on the transformed data.

It is believed that the relative lack of knowledge about how network models relate to the dynamics could partially explain their limited use in modeling nonlinear dynamics when symmetry-related issues are important. This paper has shown sufficient conditions that if satisfied by the topology of a MLP network will guarantee symmetry of fixed points. Similarly, a procedure for ensuring symmetry in RBF networks has been proposed. Such results should prove helpful to the wide body of network users who intend to use this type of models to reproduce specific nonlinear phenomena with symmetry properties.

ACKNOWLEDGMENTS

The authors are grateful to CNPq, FAPEMIG (Brazil), and CNRS (France) for financial support.

[1] K. Judd and A. Mees, Physica D **82**, 426 (1995).

[2] G. Gouesbet and C. Letellier, Phys. Rev. E **49**, 4955 (1994).

[3] B.P. Bezruchko and D.A. Smirnov, Phys. Rev. E **63**, 016207 (2000).

[4] J. Timmer, H. Rust, W. Horbelt, and H.U. Voss, Phys. Lett. A **274**, 123 (2000).

[5] C. Lainscsek, C. Letellier, and I. Gorodnitsky, Phys. Lett. A **314**, 409 (2003).

[6] L.A. Aguirre and S.A. Billings, Int. J. Bifurcation Chaos Appl. Sci. Eng. **5**, 449 (1995).

[7] K.H. Chon, J.K. Kanters, R.J. Cohen, and N.H. Holstein-Rathlou, Physica D **99**, 471 (1997).

[8] E.M.A.M. Mendes and S.A. Billings, Int. J. Bifurcation Chaos Appl. Sci. Eng. **7**, 2593 (1997).

[9] E. Bagarinao, Jr., K. Pakdaman, T. Nomura, and S. Sato, Phys. Rev. E **60**, 1073 (1999).

[10] S.A. Billings and D. Coca, Int. J. Bifurcation Chaos Appl. Sci. Eng. **9**, 1263 (1999).

[11] E. Bagarinao, Jr., K. Pakdaman, T. Nomura, and S. Sato, Physica D **130**, 211 (1999).

[12] R. Bakker, J.C. Schouten, C.L. Giles, F. Takens, and C.M. van den Bleek, Neural Comput. **12**, 2355 (2000).

[13] M. Small, K. Judd, and A. Mees, Phys. Rev. E **65**, 046704 (2002).

[14] A. Çinar, Chemom. Intell. Lab. Syst. **30**, 147 (1995).

[15] B.F. Redmond, V.G. LeBlanc, and A. Longtin, Physica D **166**, 131 (2002).

[16] D.J.C. MacKay, Neural Comput. **4**, 448 (1992).

[17] M. Small and C.K. Tse, Phys. Rev. E **66**, 066701 (2002).

[18] A.I. Mees, M.F. Jackson, and L.O. Chua, IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **39**, 19 (1992).

[19] D.S. Broomhead and D. Lowe, Complex Syst. **2**, 321 (1988).

[20] L.A. Smith, Physica D **58**, 50 (1992).

[21] S. Chen, C.F.N. Cowan, and P.M. Grant, IEEE Trans. Neural Netw. **2**, 302 (1991).

[22] L.A. Aguirre, U.S. Freitas, C. Letellier, and J. Maquet, Physica D **158**, 1 (2001).

[23] N.R. Draper and H. Smith, *Applied Regression Analysis*, 3rd ed. (Wiley, New York, 1998).

[24] Y. Ueda, *The Road to Chaos* (Aerial Press, Santa Cruz, CA, 1993).

[25] O. Ménard, C. Letellier, J. Maquet, L. Le Sceller, and G. Gouesbet, Int. J. Bifurcation Chaos Appl. Sci. Eng. **10**, 1759 (2000).

[26] M. Norgaard, Technical University of Denmark, Technical Report No. 97-E-851, 1997 (unpublished).

[27] L.A. Aguirre and S.A. Billings, Physica D **80**, 26 (1995).

[28] L. Cao, Physica D **110**, 43 (1997).

[29] M.R. Cowper, B. Mulgrew, and C.P. Unsworth, Signal Process. **82**, 775 (2002).

[30] B. Pilgram, K. Judd, and A.I. Mees, Physica D **170**, 103 (2002).

[31] G. Boudjema and B. Cazelles, Chaos, Solitons Fractals **12**, 2051 (2001).

[32] C. Letellier and R. Gilmore, Phys. Rev. E **63**, 016206 (2001).

[33] C. Letellier, G. Gouesbet, and N. Rulkov, Int. J. Bifurcation Chaos Appl. Sci. Eng. **6**, 2531 (1996).

[34] C. Kahlert, Int. J. Bifurcation Chaos Appl. Sci. Eng. **3**, 963 (1993).

[35] Y.S. Abu-Mostafa, IEEE Trans. Neural Netw. **12**, 791 (2001).

[36] E. Bagarinao, Jr., T. Nomura, K. Pakdaman, and S. Sato, Physica D **124**, 258 (1998).

[37] A.C. Tsoi and A. Back, Neurocomputing **15**, 183 (1997).