Empirical Dynamical System Reduction I: Global Nonlinear Transformations

Michael Kirby and Rick Miranda

ABSTRACT. This investigation introduces the constrained nonlinear Karhunen-Loève transformation (CNLKL) for the data reduction and semi-analytical remodeling of dynamical systems. The unconstrained nonlinear approach is extended in a manner analogous to the linear KL procedure and a general framework for imposing arbitrary optimality criteria for reduction transformations is presented. The nonlinear optimization problem with constraints is formulated in terms of computational neural networks and implemented with a modified backpropagation algorithm. The resulting reduction techniques are applied to the investigation of the Kuramoto-Sivashinsky partial differential equation for several different values of the bifurcation parameter. It is shown by means of these examples that the CNLKL transformation may permit a significant improvement over the linear KL procedure.

CONTENTS

1. Introduction 1
2. Neural Networks and Function Approximation 4
3. The Nonlinear Karhunen-Loève Transformation (NLKL) 9
4. Constrained Nonlinear Karhunen-Loève (CNLKL) 12
5. Reduction and Semi-analytic Remodeling of Dynamical Systems 17
6. Summary and Directions for Future Research 21
References 23

1. Introduction

Over the past decade there has been considerable research activity concerning the reduction of infinite or high-dimensional dynamical models. This work has been driven by two main complementary objectives. The first goal is the development of mathematical techniques to enable numericists and laboratory scientists to represent their data in a low-dimensional setting and thereby facilitate the extraction of salient information from their simulations or experiments. The second objective is to provide algorithms for reducing potentially infinite dimensional model equations to equivalent finite, or low-dimensional, dynamical systems. We will refer to the

1991 *Mathematics Subject Classification.* Primary 34C20, 41A29; Secondary 92B20, 65K10.

latter reduction task as the dynamical reduction, or *remodeling* problem. The existence of an inertial manifold for the system provides the theoretical justification for such a reduction [**11**].

Both aspects of the reduction problem are driven by the hypothesis that the physical phenomenon under investigation has significantly lower dimension than the ambient space in which the model is originated. The inherent redundancy of the coordinate system defining the ambient space results in large systems of equations and high-dimensional data which are often difficult to analyze and process. For example, the search for coherent physical structures is complicated by the fact that they may be hidden in gigabytes of data. Also, the topological structure of phase space can be extremely difficult to identify in high-dimensional settings.

The dimensionality reducing transformation thus has the potential to reveal important features in the data as well as provide a bridge between non-optimal models and dynamical systems which reflect the intrinsic dimensionality of the solutions. The construction of such transformations presents significant challenges due to the fact that essential information may be destroyed with even highly accurate approximations [**18, 41**]. As such, we are primarily interested in transformations which permit perfect reconstruction of the data and its dynamics.

Optimal orthogonal transformations provide one of the main *empirical* (data dependent) techniques for constructing low-dimensional transformations. The primary example of such a transformation is the well-known Karhunen-Loève (KL) procedure, also referred to as principal component analysis [**14**]. It has been applied to the problem of modeling atmospheric dynamics as early as [**27**]. One of the first objective descriptions of coherent structures in turbulent flow is based on this reduction procedure [**28**]. The KL methodology has also been extended to handle high-resolution multivariate data sets [**39, 40**] and has been extensively applied to both the data reduction problem, e.g., [**19, 23**], and the remodeling problem, e.g., [**2, 6, 18, 34**].

The KL procedure is characterized by the fact that it provides an optimal empirical projection onto the subspace of lowest dimension. In the context of the remodeling problem, we refer to this approach (and all methods which require solution data) as *semi-analytical*. Such techniques are based on the empirical transformations which are used to discover appropriate low dimensional coordinate systems for recasting analytical models. This approach is in contrast with the data independent procedures for reduction, such as the nonlinear Galerkin procedure, as described in [**13, 15, 29**].

It is well known that dynamical systems often exhibit solutions which are unevenly distributed in phase space, and which may exist as circles, tori or other manifolds. Thus, we observe that the KL procedure, being a linear reduction, will only find the best encapsulating subspace and may not provide an optimal representation of the set X to which the solutions are attracted [**20**]. Consider, for instance, a periodic solution which traverses a high-dimensional space. This solution is homeomorphic to a circle and a nonlinear transformation exists which will map the curve to the plane. However, the linear KL procedure is incapable of flattening out the curve and provides a high-dimensional basis for its best approximating space.

In view of these observations, it is fruitful to extend the reduction procedure to include nonlinear mappings. Formally, given a data set $X \in \mathbb{R}^N$ we seek a reduction mapping

$$\mathbf{G} : X \to Y$$

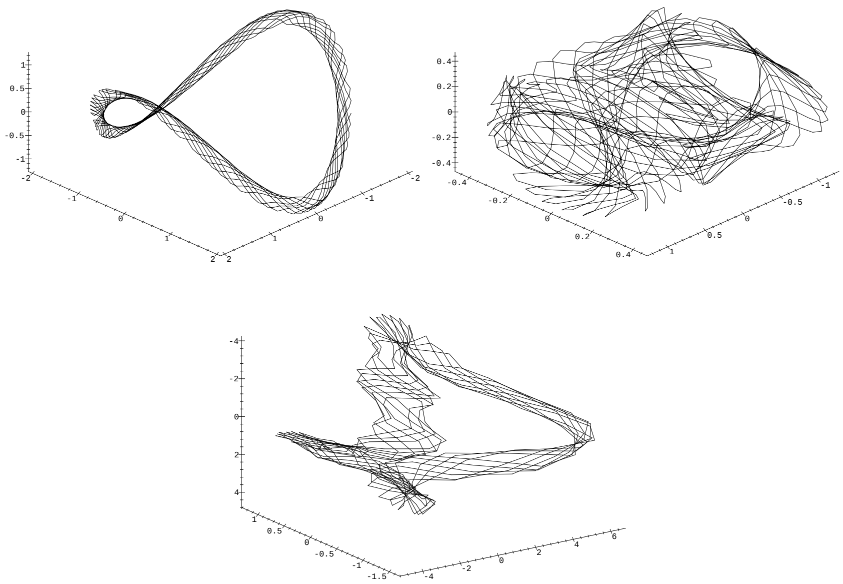

FIGURE 1. A comparison of the linear vs. constrained nonlinear KL reduction of solutions of the Kuramoto-Sivashinsky equation for $\alpha = 87$. Top Left: Projection onto the first 3 KL modes. Top Right: Projection onto the second set of 3 KL modes. Bottom: The nonlinear reduction to the first 3 CNLKL modes.

where the set $Y \in \mathbb{R}^M$ has reduced dimension $M < N$. We view the set Y as a reduction of X. The associated reconstruction mapping reproduces the original data X from the reduced data set Y and is written

$$\mathbf{H} : Y \to X$$

Our interest is in finding a composition of mappings which is equivalent to the identity mapping. Thus, an element $\mathbf{x} \in X$ should be left unchanged by the composition, i.e.,

$$\mathbf{H} \circ \mathbf{G} : \mathbf{x} \to \mathbf{x}$$

Therefore, the reduced dimension M must be taken large enough such that negligible information is lost.

In addition, the reduction mapping may be required to satisfy additional constraints which further characterize the transformation. It will be seen that computational neural networks provide a powerful tool for approximating these functions $\mathbf{H}$ and $\mathbf{G}$.

The Kuramoto-Sivashinsky (KS) partial differential equation

$$\frac{\partial u}{\partial t} + 4\frac{\partial^4 u}{\partial x^4} + \alpha(\frac{\partial^2 u}{\partial x^2} + \frac{1}{2}(\frac{\partial u}{\partial x})^2) = 0. \tag{1}$$

provides many excellent examples of solutions which are not reducible via linear transformations. It has become a benchmark equation for the development of low-dimensional systems, see e.g., [**7, 15, 18, 3, 1**]. It is known to possess an inertial manifold and be mathematically equivalent to a finite dimensional system of ordinary differential equations [**33**]. It will serve as the main application in this investigation.

To further motivate our discussion, we present an example of a high-dimensional torus (corresponding to solutions of the KS equation with $\alpha = 87$) which is reduced via both optimal linear and nonlinear transformations. This traveling oscillatory wave is topologically diffeomorphic to a torus. However, the linear reduction to the best 3-dimensional subspace (shown in the top left of Figure 1), comprised of Fourier modes in this case, leaves a significant residual in the truncated subspace (the projection of the residual onto the next 3 best coordinates is shown at the top right of Figure 1). An optimal nonlinear mapping, on the other hand, may re-embed the manifold into a 3-dimensional subspace with zero residual, as shown at the bottom of Figure 1. The construction of this nonlinear map is achieved by the constrained nonlinear Karhunen-Loève (CNLKL) procedure which is presented in detail in Sections 3 and 4. This example demonstrates that a nonlinear reduction can map a data set to a coordinate system which reflects its intrinsic dimensionality while the linear approach fails.

In summary, the main objective of this paper is to develop a general framework for incorporating constraints into the nonlinear Karhunen-Loève (NLKL) procedure and to apply it to both the data reduction and semi-analytical remodeling of dynamical systems. The outline of Part I is as follows: in Section 2 the pertinent neural network background including a discussion of the error backpropagation algorithm used for developing our empirical approximations will be presented. In Section 3 it is shown how artificial neural networks may be used to construct the nonlinear reduction mapping, i.e., NLKL. In Section 4 this technique is extended to produce CNLKL, i.e., reduction mappings which satisfy constraints. A nonlinear optimization problem with constraints is proposed and the modified backpropagation equations for its implementation are derived. Section 5 applies these techniques to the data reduction and remodeling problems. The technique is demonstrated on the Kuramoto-Sivashinsky equation simulated at several different dynamical regimes. Finally, in Section 6 directions for future research will be discussed.

In Part II, an alternative local mapping approach is presented which we refer to as the technique of *neural charts*.

2. Neural Networks and Function Approximation

Artificial, or computational, neural networks (ANNs) have become increasingly popular as a tool for the adaptive representation of information. In this investigation, our main interest is in their general ability to approximate nonlinear functions on a compact domain. While much of the nomenclature associated with ANNs has its origins in the modeling of a biological neuron, we will focus on aspects of networks which pertain to the function approximation problem. First, we present the

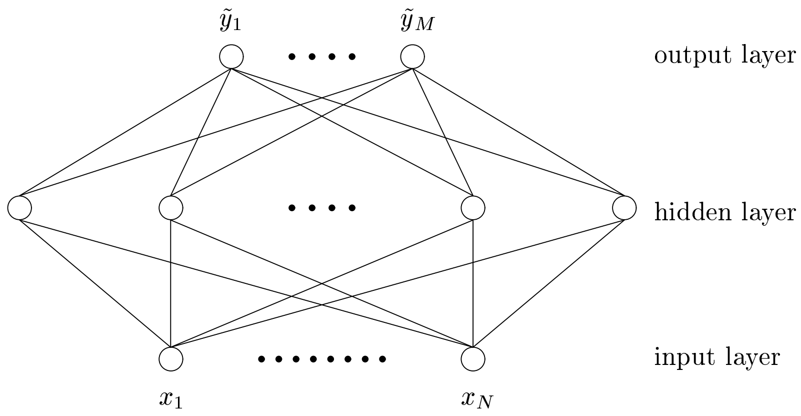

FIGURE 2. The network architecture for approximating the nonlinear reduction mapping $\mathbf{G} : X \to Y$.

background concerning the special case of feedforward networks and outline the basic manner in which they may be used to solve the interpolation problem. Once the background has been established, we will discuss our extensions to the nonlinear reduction problem in terms of the backpropagation algorithm.

2.1. Network Architecture. A feedforward neural network consists of an array of simple computational units, or nodes, divided into layers, as shown in Figure 2. Typically, such a network has one input and one output layer and possibly a number of hidden layers. The networks considered in this paper do not incorporate feedback, i.e., the units feed information forward from the input to the output layer modified by the amplitudes of the connecting weights. These stipulations amount to defining what is referred to as the network architecture.

Notationally, we denote unit k of layer i by $\mathcal{N}_k^{(i)}$ where $i = 0, \ldots, L-1$ and $k = 0, \ldots, N^{(i)}$. Hence the network has L layers (including the input layer) and $N^{(i)} + 1$ nodes in the i'th layer. Associated with each unit $\mathcal{N}_k^{(i)}$ is the state value $S_k^{(i)}$ and the prestate value $P_k^{(i)}$. The state value is computed from the prestate value as

$$S_k^{(i)} = \sigma(P_k^{(i)}) \tag{2}$$

where the function $\sigma(x)$ is referred to as an activation, or transfer function. The prestate value is defined by the state values of the previous layer via

$$P_k^{(i)} = \sum_{j=0}^{N^{(i-1)}} w_{kj}^{(i)} \cdot S_j^{(i-1)} \tag{3}$$

where the weight connecting unit j on layer $i-1$ to unit k on layer i is written $w_{kj}^{(i)}$, as shown in Figure 3.

By fixing the first state value in each layer to have the value $S_0^{(i)} = 1$, the computation of each prestate $P_k^{(i)}$ is an affine transformation of the state values on layer $i-1$, i.e., $P_k^{(i)} = w_{k0} + \sum_{j=1}^{N^{(i-1)}} w_{kj}^{(i)} \cdot S_j^{(i-1)}$. Now the weights $w_{k0}^{(i)}$ play the

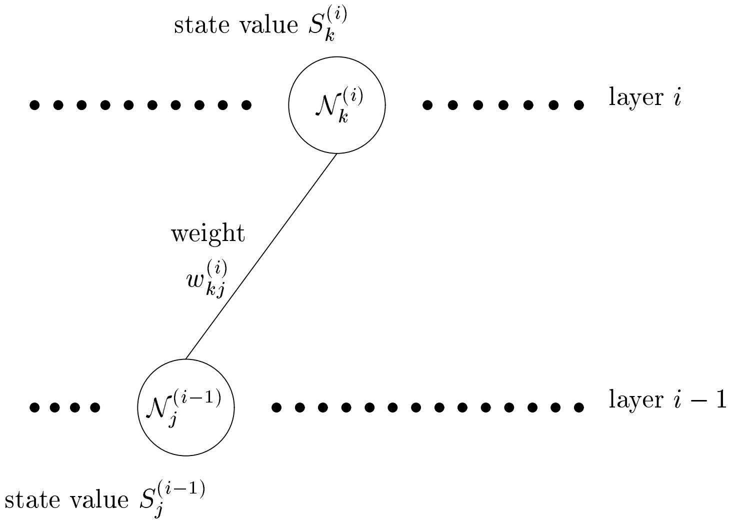

FIGURE 3. An illustration of the functional dependence between the layers.

role of the shift, or unit bias. Note that, in our notation, the $L-1$ weight arrays have indices running from $1, \ldots, L-1$.

The univariate activation function $\sigma(x)$ is generally taken as a sigmoidal, i.e., it is a monotonic function which satisfies the properties

$$\sigma(x) = \begin{cases} A & \text{as } x \to -\infty, \\ B & \text{if } x \to \infty \end{cases} \tag{4}$$

where A, B are constants. A common choice for $\sigma(x)$ is

$$\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{5}$$

Since $-1 < \sigma(x) < 1$ either the sigmoidal or the data may need to be scaled to permit the approximation over the appropriate range of values. We note that the monotonicity of the transfer functions is not required for the universal approximation capability of the network. It is sufficient for $\sigma(x)$ to be bounded and non-constant [**43**]. In general, one hidden layer with nonlinear activation functions is sufficient to approximate continuous functions on a compact domain. Note also that linear mappings may be constructed with linear transfer functions $\sigma(x) = x$. It is often useful to include both linear and nonlinear activation functions in a single network. Even circular transfer functions designed for encoding data which is homeomorphic to a closed curve may be included [**22**].

Recapping, a feedforward neural network is a computational array for approximating functions via a sequence of compositions at each layer of the form $S_k^{(i)}(P_k^{(i)}(S_0^{(i-1)}, \ldots, S_{N^{(i-1)}}^{(i-1)}))$. We write the composite mapping from the input to the output of the ANN as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{f}}(\mathbf{x}; \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L-1)}) \tag{6}$$

where the entries of the $N^{(i)} \times N^{(i-1)}$ matrix $\mathbf{W}^{(i)}$ are the weights $w_{kj}^{(i)}$ between layers i and $i-1$. The tilde notation emphasizes that these functions are approximations, see Section 2.2 for more details. It will also be convenient to define a single vector $\mathbf{w}$ consisting of all the weights in the matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L-1)}$. The vector $\mathbf{w}$ is obtained by concatenating the columns of each matrix $\mathbf{W}^{(i)}$, in order.

The Jacobian of the mapping will be required for the dynamical remodeling problem. It may be expressed compactly for an L layer net in terms of the network parameters as

$$\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{x}} = \mathbf{W}^{(1)^T} \mathbf{\Lambda}^{(1)^T} \cdots \mathbf{W}^{(L-1)^T} \mathbf{\Lambda}^{(L-1)^T} \tag{7}$$

where the diagonal matrix $\mathbf{\Lambda}$ has elements $\mathbf{\Lambda}_{jj}^{(i)} = \sigma'(P_j^{(i)})$.

We now turn to the central issue of how the feedforward ANN described above can be applied to approximate functions.

2.2. Empirical Functions. We assume that the information available concerning the function to be approximated is a collection of P input-output pairs $(\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}) \in X \times Y$, $\mu = 1, \ldots, P$. Further, it is assumed that there exists a function $\mathbf{f}$ which maps the set of input coordinates X to the set of output coordinates Y, i.e.,

$$\mathbf{f} : X \to Y$$

This function $\mathbf{f}$ is to be approximated by $\tilde{\mathbf{f}}$ as defined in equation (6).

The mapping $\tilde{\mathbf{f}}(\mathbf{x}, \mathbf{w})$ is fully determined by $\mathbf{w}$. Hence, these weights are chosen, or trained, such that the input points are mapped to the output points as accurately as possible. For this reason we will refer to the set of input-output pairs $X \times Y$ as the training set. The weights are computed such that an error, or performance function $J(\mathbf{x}^{(\mu)}, \mathbf{y}^{(\mu)}, \mathbf{w})$ is optimized. In terms of our previous notation, the input to the ANN is initialized as $S_j^{(0)}(\mu) = x_j^{(\mu)}$, $j = 1, \ldots, N$ (recall $S_0^{(0)} = 1$) and the corresponding output of the network obtained by the forward propagation of the state values is $S_j^{(L-1)}(\mu) = \tilde{y}_j^{(\mu)}$, $j = 1, \ldots, M$. Unless there is the potential for confusion, we will drop the explicit dependence of the state value on the data point μ. Hence determining the optimal weights is a nonlinear optimization problem and any of a number of techniques, such as gradient descent [**10**], may be used once an appropriate cost, or performance, function has been defined.

One of the most commonly used performance functions is the l_2 error of the approximation, averaged over the data set, i.e.,

$$J = \frac{1}{2} \langle \sum_{j=0}^{M} (y_j - \tilde{y}_j^{(\mu)})^2 \rangle \tag{8}$$

where the angle brackets denote ensemble average and the factor of $1/2$ is added to cancel the 2 when the error is differentiated. Alternatively, if the data is not all available simultaneously, the error due to individual data points may be used to compute a single step improvement of the weights.

In particular, the weights may be updated at discrete times $t = 0, 1, 2, 3, \ldots$ according to the gradient descent algorithm

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \epsilon \mathbf{d}(t) \tag{9}$$

where the parameter ϵ is referred as the learning rate and is chosen to be small while $\mathbf{w}(0)$ is initialized with small random numbers. The steepest descent algorithm chooses the update in the direction of fastest (short term) improvement, i.e.,

$$\mathbf{d}(t) = -\sum_{\mu=1}^{P} \nabla_{\mathbf{w}} E(\mathbf{x}^{(\mu)}, \mathbf{w})$$

The steepest descent algorithm is adequate for many problems but may become unmanageably slow for approximation problems with a large number of weights. Many alternative algorithms exist and conjugate gradient was chosen for this particular investigation [**10**].

2.3. The Error Backpropagation Algorithm. Now we briefly describe the well-known error backpropagation algorithm. It is an efficient recursive method for computing the gradient of the error function for the specific case of the layered network. In Section 4 we show how the constraint equations may directly incorporated into this algorithm.

We begin with the observation that the state value at unit $\mathcal{N}_k^{(i)}$ depends on the prestate value which in turn depends on the weight $w_{kj}^{(i)}$. Thus, the chain rule may be used to compute the gradient of the performance function J w.r.t the weight $w_{kj}^{(i)}$ as

$$\frac{\partial J}{\partial w_{kj}^{(i)}} = \frac{\partial J}{\partial S_k^{(i)}} \frac{\partial S_k^{(i)}}{\partial P_k^{(i)}} \frac{\partial P_k^{(i)}}{\partial w_{kj}^{(i)}} \tag{10}$$

The second and third terms may be evaluated directly by differentiating equations (2) and (3), respectively. The first term is determined recursively as

$$\frac{\partial J}{\partial S_k^{(i)}} = \sum_{m=1}^{N^{(i+1)}} \frac{\partial J}{\partial S_m^{(i+1)}} \frac{\partial S_m^{(i+1)}}{\partial P_m^{(i+1)}} \frac{\partial P_m^{(i+1)}}{\partial S_k^{(i)}} \tag{11}$$

Defining the common term as

$$\Delta_k^{(i)} = \frac{\partial J}{\partial S_k^{(i)}} \frac{\partial S_k^{(i)}}{\partial P_k^{(i)}} \tag{12}$$

enables us to rewrite the recursion in the form

$$\Delta_k^{(i)} = \frac{\partial S_k^{(i)}}{\partial P_k^{(i)}} \sum_m \Delta_m^{(i+1)} \frac{\partial P_m^{(i+1)}}{\partial S_k^{(i)}} \tag{13}$$

Substituting equations (2) and (3), we obtain a convenient expression for the gradient which may, for example, be used to implement the steepest descent algorithm as

$$w_{kj}^{(i)}(t+1) = w_{kj}^{(i)}(t) - \epsilon \langle \Delta_k^{(i)} S_j^{(i-1)} \rangle \tag{14}$$

where the recursion, using the error defined in equation (8), is initialized as

$$\Delta_k^{(L-1)} = -(y_k - S_k^{(L-1)}) \sigma'(P_k^{(L-1)})$$

and backpropagated via

$$\Delta_k^{(i)} = \sigma'(P_k^{(i)}) \sum_m \Delta_m^{(i+1)} w_{mk}^{(i+1)}$$

The interested reader may find further details concerning feedforward neural networks and the error backpropagation algorithm, including the history of its development, in the articles [**36, 43**] and the references therein.

3. The Nonlinear Karhunen-Loève Transformation (NLKL)

In this Section we set up the formalism of the reduction mapping and then demonstrate how the feedforward neural networks described above can be a powerful general tool for constructing reduction mappings.

Given an ensemble of mean subtracted sample vectors $\{\mathbf{x}^{(\mu)}\} \in X \subset \mathbb{R}^N$, where μ represents the discrete time index the pattern, we seek a reduction mapping $\mathbf{G} : X \to Y$ which provides a reduced representation $\mathbf{y}$ of $\mathbf{x}$ written as

$$\mathbf{y} = \mathbf{G}(\mathbf{x})$$

where $\mathbf{y} \in Y \subset \mathbb{R}^M$ and $M < N$. The associated reconstruction mapping is then given as $\mathbf{H} : Y \to X$ and the reconstructed vector is written

$$\mathbf{x} = \mathbf{H}(\mathbf{y})$$

Neural networks may be employed to approximate both $\mathbf{H}, \mathbf{G}$ by $\tilde{\mathbf{H}}, \tilde{\mathbf{G}}$. This is achieved by determining the weights such that the mean square error is as small as possible, i.e.,

$$\langle \|\mathbf{x} - (\tilde{\mathbf{H}} \circ \tilde{\mathbf{G}})(\mathbf{x})\|^2 \rangle = \text{minimum} \tag{15}$$

where the norm $\|\cdot\|$ is to be specified. Note that this error should be essentially zero for a perfect reconstruction transformation.

In Section 4 we will require that the reduced set Y satisfy certain side constraints which for the moment we write collectively as

$$c_i(Y) = 0 \tag{16}$$

where $c_i : Y \to \mathbb{R}$. Taken together, equations (15) and (16) form a nonlinear optimization problem with equality constraints. Later we will consider additional types of constraints.

In the setting above it is assumed that $\mathbf{G}$ and $\mathbf{H}$ may be nonlinear. If we restrict ourselves to the case of orthogonal transformations, then $\mathbf{G}$ is a matrix whose rows are the eigenvectors of the ensemble averaged covariance matrix. The associated reconstruction matrix is $\mathbf{H} = \mathbf{G}^T$. In other words, the mappings are equivalent to the KL transformation.

Recall, for purposes of comparison, the main properties of the linear KL transformation. It is an *orthogonal projection* which simultaneously possesses several important optimality properties, including

- minimal mean-square truncation error (mse)
- maximal data variance
- decorrelation of the coordinate axes
- maximal information compression

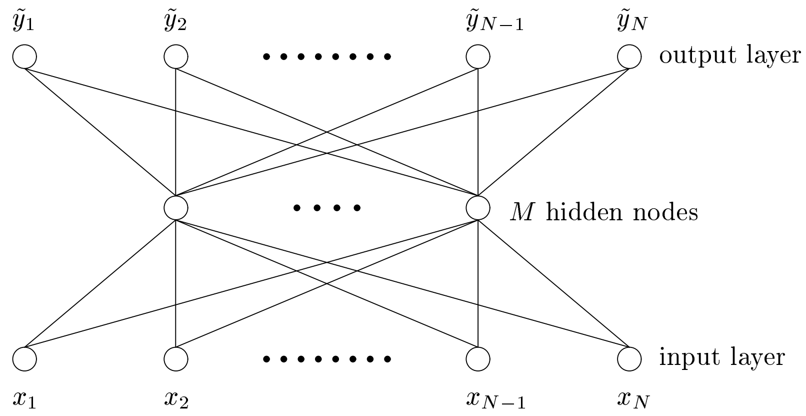

FIGURE 4. The linear projection network has linear activation functions. If the composite network maps the identity, i.e., the input equals the output, then the network computes a projection onto an M-dimensional subspace.

It is a remarkable fact that if the class of transformations is restricted to be orthogonal, then all of the above are equivalent [**12**].

In what follows, it is shown how neural networks can be used to construct general nonlinear reduction mappings with constraints. However, before turning to the general case, it is instructive to consider how linear networks may be used to implement the KL transform. Here we will develop the ideas which will lead us directly to the nonlinear extensions with constraints.

Background: Neural Linear Reduction. There are actually many neural based methods which have been proposed for computing KL expansions, as described in [**4**]. We limit our attention to how a linear feedforward neural network can be used to compute KL subspaces and then extend this architecture to the nonlinear case.

Such an architecture was proposed in [**35**] for constructing linear dimensionality reducing transformations and referred to as the linear *autoencoder*. This feedforward network consists of a layer of N input nodes, M hidden nodes and N output nodes, connected as shown in Figure 4. The state values are computed from the prestate values using a linear activation function $\sigma(x) = x$. When the identity mapping is trained (e.g., via gradient descent) to minimize the mean square l_2 error

$$J = \langle \sum_{j=1}^{N} (x_j - \tilde{y}_j^{(\mu)})^2 \rangle \tag{17}$$

the network computes the optimal KL subspace of dimension M.

This network utilizes the input data as the desired output data, i.e., the input-output pairs have the form $(\mathbf{x}^{(\mu)}, \mathbf{x}^{(\mu)}) \in X \times X$ and the composition of mappings is dimensionality preserving. The network funnels the data through an M-dimensional hidden layer. This hidden layer behaves as a bottleneck and each state value encodes a coordinate value of the M-dimensional projection of the data onto the weight

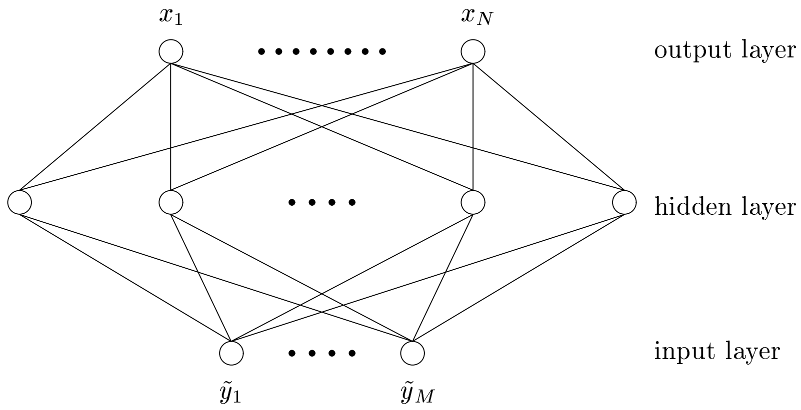

FIGURE 5. The network architecture for approximating the nonlinear reconstruction mapping $\mathbf{H} : Y \to X$.

vector. It is clear, however, that the mapping provided by the network in Figure 4 is not unique, since the network does not restrict the resulting basis to be orthogonal. Even if we restrict the weights such that $\mathbf{W}^{(1)} = {\mathbf{W}^{(2)}}^T$, the solution is still not unique (consider rotations). In fact, the linear projection network suggested is only capable of determining a basis which spans the same optimal subspace as the KL projection of the same dimension.

To obtain the full sequence of N KL eigenvectors, the architecture in Figure 4 must be employed N times with $M = 1$. The first training session is on the mean subtracted data and produces the eigenvector corresponding to the largest (unique) eigenvalue. The second training session is on the ensemble of residuals $\{\mathbf{x}'\}$ computed by subtracting the 1-dimensional reconstructions of the data points

$$\mathbf{x}' = \mathbf{x} - S_0^{(1)} \mathbf{w}^{(1)}$$

It is interesting to note that with $M = 1$, the architecture in Figure 4 has the property that the relation $\mathbf{w}^{(1)} = \mathbf{w}^{(0)}$ is automatically satisfied once the network has converged to the first KL eigenvector. Note that the interpretation of the sequential training procedure is somewhat different if the eigenvalues are not simple.

It is worth remarking that if all the data required to compute the covariance matrix is available, then the direct computation of the KL eigenvectors using standard computer packages is superior to the sequential approach described above. In general, if the data is being accumulated continuously, then neural networks do offer some advantages since they adapt more slowly to the solution. The approach proposed in Section 4 may be used for the simultaneous (as opposed to sequential) calculation of KL eigenvectors with continuously available data; see [**31, 37**] for comparison. The theory of the linear projection network is discussed in detail in [**4, 5**].

Neural Nonlinear Reduction. The basic nonlinear reduction network for implementing NLKL has been proposed by many authors as an extension to the linear approach for computing *nonlinear* principal components, see, e.g., [**8, 25, 26, 30, 38**]. There have been a number of applications of this technique for nonlinearly reducing data produced by dynamical systems, see e.g., [**8, 9, 20, 21, 32**] and initial applications specific to the remodeling problem have been very promising [**20, 21**]. In this Section we show how feedforward neural network architectures used to train the identity mapping may be used to compute nonlinear empirical reductions, i.e., data dependent dimensionality reducing mappings. Later, this basic framework will be extended to permit reduction mappings with constraints.

We saw above how neural networks with linear units and 1 hidden, or bottleneck layer, could be used to compute the linear KL transformation. In view of the fact that a neural network with one hidden layer and a nonlinear transfer function is capable of approximating arbitrary continuous functions (and their derivatives) on a compact domain, the linear autoencoder may be extended to produce a nonlinear dimensionality reducing mapping, i.e., NLKL.

Recalling that $X \subset \mathbb{R}^N, Y \subset \mathbb{R}^M$ where $M < N$, the nonlinear reduction mapping $\mathbf{G} : X \to Y$ is approximated by $\tilde{\mathbf{G}}$ with the architecture shown in Figure 2 while the associated reconstruction mapping $\mathbf{H} : Y \to X$ is approximated by $\tilde{\mathbf{H}}$ via the architecture shown in Figure 5. The single network resulting from connecting these networks, shown in Figure 6, approximates the composition of functions $\mathbf{H} \circ \mathbf{G}$ by

$$\tilde{\mathbf{x}} = (\tilde{\mathbf{H}} \circ \tilde{\mathbf{G}})(\mathbf{x})$$

If this composite network is trained to map the identity function in a fashion similar to the linear network described above then $\tilde{\mathbf{x}} \approx \mathbf{x}$ where the quality of the approximation is norm dependent. Since the dimensionality reduction mapping is now nonlinear, the method is referred to as the nonlinear KL transform (NLKL).

The architecture of the bottleneck reduction network is characterized by the number of nodes in each layer and we will designate it as $N^{(0)} - N^{(1)} - N^{(2)} - N^{(3)} - N^{(4)}$. Since the input and the output are identical $N^{(0)} = N^{(4)} = N$. Also, in all of the simulations in this paper we take $l = 2$ as the bottleneck layer, so $N^{(2)} = M$. An additional hidden layer may be used to approximate the mapping function $\mathbf{G}$ to permit an arbitrary nonlinear mapping to the layer immediately preceding the bottleneck layer.

One of the difficulties with the bottleneck network for implementing nonlinear KL is that the training problem is ill-posed in the sense that there are an infinite number of solutions which minimize the mean square reconstruction error. A subset of these solutions consist of networks which satisfy statistical or dynamical constraints at units or layers of the network.

4. Constrained Nonlinear Karhunen-Loève (CNLKL)

The standard NLKL procedure, as described in the references above, requires only that the input data be mapped to itself, or that the distance between the network output and input be minimized according to some norm (typically the mean-square error). Since the NLKL procedure is nonlinear, the mathematical structure (equivalent optimality criteria) of the linear KL transformation is lost. For NLKL minimizing the mean-square error does not, for instance, ensure uncorrelated data in the reduced coordinates (even the sequential algorithm does not overcome this).

One of the main contributions of this paper is a procedure to simultaneously satisfy a set of (nonconflicting) optimality criteria, similar to the orthogonal case. Hence, in analogy to the linear KL transformation, we seek to construct a nonlinear

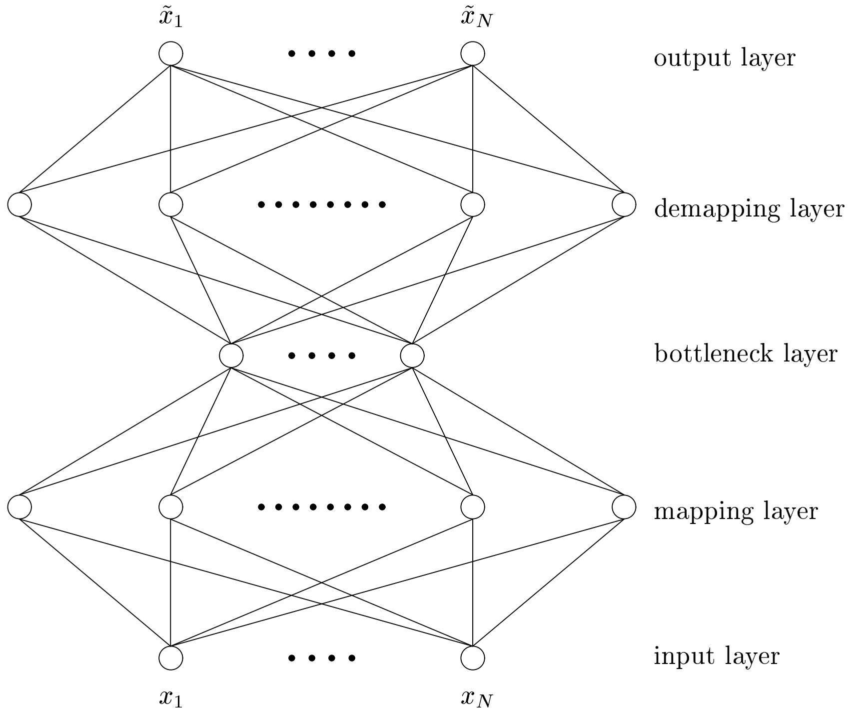

FIGURE 6. The bottleneck network architecture.

reduction mapping such that the reduced data set Y possesses additional properties, such as uncorrelatedness, and minimal entropy, which may be prescribed based on the application. Our technique incorporates user defined constraints into existing bottleneck network architectures. This is achieved by including penalty terms in the cost function at the units where the constraints are to be imposed.

We remark that neural networks with constraints have been proposed in different contexts including network pruning via weight decay [**42**], and the temporal and configurational constraints developed in [**16**].

Implementing the Constraint Equations. In this section we show how to add general constraints to the nonlinear reduction problem. The constraints may explicitly involve the weights, or the state values and may be either local or global in nature.

To begin, consider the application of a constraint at unit $\mathcal{N}_n^{(m)}$. The constraint is imposed as a penalty term indexed in the same manner as the node, $J_n^{(m)}$. The total cost function J for the network is then defined as the sum of the individual penalties over the entire network

$$J = \sum_{m,n} J_n^{(m)} \tag{18}$$

In general, we will consider penalties which involve either one- or two-point statistics evaluated over an ensemble of data $X \times Y$, (or $X \times X$ for mapping the identity). It will often be useful to require that a unit constraint be applied layer-wise, i.e., simultaneously to all the units in a layer. This may be accomplished simply by summing the constraints over all the units in the layer.

As usual, to implement a gradient based algorithm for this general cost function J we require the derivatives $\frac{\partial J}{\partial w_{kj}^{(i)}}$ for all relevant i,j, and k. The partial derivative of the total cost function w.r.t. the weights is

$$\frac{\partial J}{\partial w_{kj}^{(i)}} = \sum_m \sum_n \frac{\partial J_n^{(m)}}{\partial w_{kj}^{(i)}}. \tag{19}$$

Again, given the hierarchy of dependencies at unit $\mathcal{N}_k^{(i)}$ used to derive equation (10), we apply the chain rule to differentiate $J_n^{(m)}$ w.r.t the weight $w_{kj}^{(i)}$ as

$$\frac{\partial J_n^{(m)}}{\partial w_{kj}^{(i)}} = \frac{\partial J_n^{(m)}}{\partial S_k^{(i)}} \frac{\partial S_k^{(i)}}{\partial P_k^{(i)}} \frac{\partial P_k^{(i)}}{\partial w_{kj}^{(i)}} \tag{20}$$

Substituting the above into equation (19) and factoring appropriately, we obtain

$$\frac{\partial J}{\partial w_{kj}^{(i)}} = \left(\sum_{mn} \frac{\partial J_n^{(m)}}{\partial S_k^{(i)}} \right) \frac{\partial S_k^{(i)}}{\partial P_k^{(i)}} \frac{\partial P_k^{(i)}}{\partial w_{kj}^{(i)}} \tag{21}$$

Comparing this with equation (10), backpropagation with constraints has the new term

$$\frac{\partial J}{\partial S_k^{(i)}} = \sum_{m,n} \frac{\partial J_n^{(m)}}{\partial S_k^{(i)}} \tag{22}$$

Noting that if $i = m$ in the above sum, then the term $\frac{\partial J_n^{(m)}}{\partial S_k^{(i)}}$ is computed directly from the definition of the constraint. If $i < m$, then the term is computed using recursion as usual.

The constraint at any given node may consist of the superposition of several constraints, i.e.,

$$J_n^{(m)} = \sum_\beta \kappa_n^{(m)}(\beta) J_n^{(m)}(\beta) \tag{23}$$

where the sum is over the constraint index β. The relative importance of each penalty is determined by the scalars $\kappa_n^{(m)}(\beta)$. In this study the constants κ were taken to be small relative to the l_2 penalty coefficient which was taken to be 1. Thus the total performance index may be viewed as a perturbation of the standard cost function defined in equation (17). With multiple constraints at a given node the modified term given by equation (22) is now also summed over β.

Now, in the modified backpropagation algorithm, penalties contributing to the cost function J are backpropagated from the unit where they are applied. In other words, at each unit where a penalty term is applied there is an additional contribution to the gradient which is also backpropagated. Thus the only new terms that need be computed to extend the backpropagation algorithm to accommodate additional state value constraints are given by $\frac{\partial J_n^{(m)}}{\partial S_k^{(i)}}$. We now compute this quantity for a variety of interesting constraints.

Goal-Target Errors. With the general approach to defining the performance function advocated in this paper, we may view the standard l_2 error either as the main performance objective, or as a constraint. With this in mind, we now write the l_2 error obtained by attempting to map the discrete input set X to the target set Y

$$J^{(L-1)}(1) = \frac{1}{2}\sum_{\mu n}(y_n(\mu) - S_n^{(L-1)}(\mu))^2$$

For construction of the identity mapping $Y = X$, i.e., we substitute $x_n = y_n$ above. Hence, at the start of the recursion $\frac{\partial J^{(L-1)}(1)}{\partial S_k^{(L-1)}(\mu)} = S_k^{(L-1)}(\mu) - x_k(\mu)$. Note that unlike the following cases, this error is applied only at output units. It is also applied across an entire (output) layer so the cost term is independent of the node index. In what follows, we will not explicitly write the constraint label β.

Zero Node Mean. In many modeling applications it is desirable to represent the data in terms of a centered coordinate system, i.e., one which is has its origin at the centroid of the data set. This may be accomplished in the current framework by imposing a zero unit mean side constraint. The ensemble averaged state of a selected unit n in layer m may be forced to zero by penalizing the quantity

$$J_n^{(m)} = \frac{1}{2}\langle S_n^{(m)}\rangle^2 \tag{24}$$

Differentiating we obtain the expression for initiating the backpropagation at the node $\mathcal{N}_n^{(m)}$

$$\frac{\partial J_n^{(m)}}{\partial S_n^{(m)}} = \langle S_n^{(m)}\rangle / P \tag{25}$$

where, again, P is the number of training points in $X \times Y$.

Similarly, all of the units on a given layer m may be required to simultaneously have zero mean by applying the penalty term

$$J^{(m)} = \frac{1}{2}\sum_{n=1}^{N^{(m)}}\langle S_n^{(m)}\rangle^2 \tag{26}$$

be a minimum.

Minimum Node Correlation. Minimizing the mean square error in the bottleneck architecture will not produce uncorrelated variables on the bottleneck. This may be done, however, by explicitly requiring zero correlation as a side constraint. In addition, it is possible to eliminate correlation between any nodes values in the network. In this paper we employed the penalty term

$$J^{(m)} = \frac{1}{2}\frac{1}{N^{(m)}(N^{(m)}-1)}\sum_{\substack{j=1 \\ j\neq n}}^{N^{(m)}}\sum_{n=1}^{N^{(m)}}\langle S_j^{(m)} S_n^{(m)}\rangle^2 \tag{27}$$

which converges to zero when the state values on layer m are uncorrelated.

The new term in the backpropagation is then

$$\frac{\partial J^{(i)}}{\partial S_k^{(i)}(\mu)} = \frac{2}{PN^{(i)}(N^{(i)}-1)} \sum_{\substack{j=1 \\ j\neq k}}^{N^{(i)}} \langle S_k^{(i)} S_j^{(i)} \rangle S_j^{(i)}(\mu). \tag{28}$$

where the index μ indicates that the derivative and the state value are evaluated at the data point $\mathbf{x}^{(\mu)}$.

Regularization Constraint. This constraint assumes that the data is smooth and time dependent. To smooth, or *regularize* the state value of the unit $\mathcal{N}_n^{(m)}$ we require that sequential patterns be close to each other. This is achieved by penalizing sequential state values with large differences more than sequential state values which are close together. This constraint may be imposed by minimizing the quantity

$$J_n^{(m)} = \frac{1}{2} \sum_{\mu} (S_n^{(m)}(\mu) - S_n^{(m)}(\mu - 1))^2 \tag{29}$$

Now we differentiate w.r.t to state value of the μ'th pattern to give

$$\frac{\partial J_k^{(i)}}{\partial S_k^{(i)}(\mu)} = 2S_k^{(i)}(\mu) - S_k^{(i)}(\mu-1) - S_k^{(i)}(\mu+1) \tag{30}$$

Unlike the previous correlation and mean related constraints, we do not expect this penalty term to go to zero.

Maximum Variance. Although it might be anticipated that maximum variance approximations will result if the mean square error is minimized, this may be rigorously enforced as a side constraint. The variance at unit n in layer m is given by

$$J_n^{(m)} = \rho_n^{(m)} = \frac{1}{2} \langle (S_n^{(m)} - \langle S_n^{(m)} \rangle)^2 \rangle \tag{31}$$

Hence

$$\frac{\partial \rho_k^{(i)}}{\partial S_k^{(i)}(\mu)} = \frac{1}{P}(S_k^{(i)}(\mu) - \langle S_k^{(i)} \rangle) \tag{32}$$

Minimum Entropy. Shannon's entropy is a measure of the dispersion of information as determined by the variances (normalized to be probabilities) of the state values. Using the definition for variance in equation (31) the total variance in layer i is $\rho_T^{(i)} = \sum_{j=1}^{N^{(i)}} \rho_j^{(i)}$. The normalized variance is then defined as $\tilde{\rho}_j^{(i)} = \rho_j^{(i)}/\rho_T^{(i)}$. Shannon's entropy for layer i is then given by (letting $J^{(i)} = H^{(i)}$)

$$H^{(i)} = -\sum_{j=1}^{N^{(i)}} \tilde{\rho}_j^{(i)} \ln \tilde{\rho}_j^{(i)} \tag{33}$$

Differentiating we obtain

$$\frac{\partial H^{(i)}}{\partial S_k^{(i)}(\tau)} = \frac{\partial \rho_k^{(i)}}{\partial S_k^{(i)}(\tau)} [\sum_{j=1}^{N^{(i)}} (\rho_j^{(i)}/(\rho_T^{(i)})^2)(1 + \ln \tilde{\rho}_j^{(i)}) - (1 + \ln \tilde{\rho}_k^{(i)})/\rho_T^{(i)}]$$

Orthogonal Mappings. This weight based constraint concerns the mapping from the state values of layer $i-1$ to the prestate values of layer i. Recall that the matrix of weights $\mathbf{W}^{(i)}$ defines the mapping between these 2 layers. We may constrain that the rows of this matrix be orthogonal. First define the inner product

$$C^{(i)}_{kk'} = \sum_{j=0}^{N^{(i)}} w^{(i)}_{kj} w^{(i)}_{k'j}$$

Now, assuming $N^{(i)} \leq N^{(i-1)}$, we require that the the projection be orthogonal in the sense that the penalty

$$J^{(i)} = \sum_{k=0}^{N^{(i)}} \sum_{k'>k} (C^{(i)}_{kk'})^2$$

is reduced to zero. The derivative is then

$$\frac{\partial J^{(i)}}{\partial w^{(i)}_{kj}} = \sum_{\substack{k=0 \\ k' \neq k}}^{N^{(i)}} C^{(i)}_{kk'} w^{(i)}_{k'j}.$$

5. Reduction and Semi-analytic Remodeling of Dynamical Systems

The dynamical remodeling problem is concerned with the reduction of systems of differential equations such as the Navier-Stokes and other phenomenological models which may possess low-dimensional representations, see e.g. [**39**]. The starting point of these investigations is often a partial differential equation representing macroscopic variables derived via conservation laws, or its equivalent infinite dimensional system of ordinary differential equations. It is assumed that the dynamics are asymptotically attracted to a finite dimensional phase space which reflects the global dimension of the system. The goal of the reduction is to transform the original model (and/or data) to a coordinate system which has a number of energetic modes corresponding to the intrinsic dimensionality of the model.

In this Section we investigate, by means of several examples, the utility of the CNLKL procedure for both the reduction and remodeling applications. The data for the examples is generated by integrating the KS equation (1) for $\alpha = 17.75, 84.25, 87, 91$. The dynamics for these parameter values correspond to a homoclinic cycle, a limit cycle, a traveling beating wave and chaos.

Sample solution curves are obtained by numerically integrating the KS equation using a Fourier-Galerkin approximation as described in [**18**]. In this study all the data was collected from the integration of 10 complex ordinary differential equations. As an illustration, the imaginary and real parts of the complex Fourier modes for the high-dimensional limit cycle at $\alpha = 84.25$ are shown in Figure 11; for purposes of comparison, the same variables are shown for the chaotic trajectories corresponding to $\alpha = 91$ in Figure 8.

5.1. Data Reduction. The CNLKL procedure provides a powerful means to visualize the solution sets in an optimally low-dimensional setting and search for topological structure in a manner analogous to the search for coherent physical structures. As we demonstrate, the nonlinearity of the reduction may permit a revealing picture of the topological structure of the phase-space, including a look at the fixed points and their interconnections.

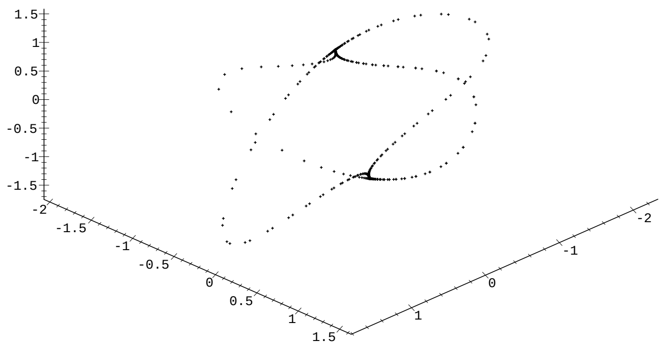

FIGURE 7. The nonlinear reduction of the Kuramoto-Sivashinsky equation for $\alpha = 17.75$ to the first 3 CNLKL modes.

Homoclinic Cycle. As a striking example of this we consider the KS equation with parameter value $\alpha = 17.75$, a case which has been carefully examined in [**7, 17, 18**]. The dynamics consist of a wave pulsing between 2 bimodal states which are $\pi/2$ translations of each other, i..e., a homoclinic loop. Each steady state has a single unstable direction connecting it to the other. A picture of the nonlinearly reduced dynamics, shown in Figure 7, reveals a 3-dimensional saddle at each of the equilibria. The same picture of the qualitative dynamics, found via local analysis, is presented in [**17**].

Our nonlinear reduction from the trajectories on the full 20-dimensional systems into the 3-dimensional space shown was accomplished with a $20-30-3-30-20$ bottleneck architecture with the zero correlation and zero mean constraints. The training error was negligible ($< O(10^{-6})$) and the constraints were satisfied to the same order of accuracy after a few thousand iterations of conjugate gradient descent.

Chaos. One of the main objectives of the reduction of dimensionality is to aid in the analysis of complex dynamics produced by chaotic systems. In Figure 8 the result of integrating chaotic solutions of the KS equation with $\alpha = 91$ is shown. These trajectories are highly correlated and based on analytical dimensionality estimates [**33**] are an excellent candidate for reduction.

The result of nonlinearly mapping the chaotic trajectory shown in Figure 8 to 5 dimensions (the conjectured intrinsic dimensionality) with $O(10^{-4})$ mean-square error is shown in Figure 9. This mapping was constrained such that the compressed dynamics were uncorrelated and had zero mean.

Our dynamical reduction for this case, as developed below, is more difficult to evaluate given nearby trajectories diverge exponentially fast. Hence small errors in the approximations lead to substantially different solution curves, as expected.

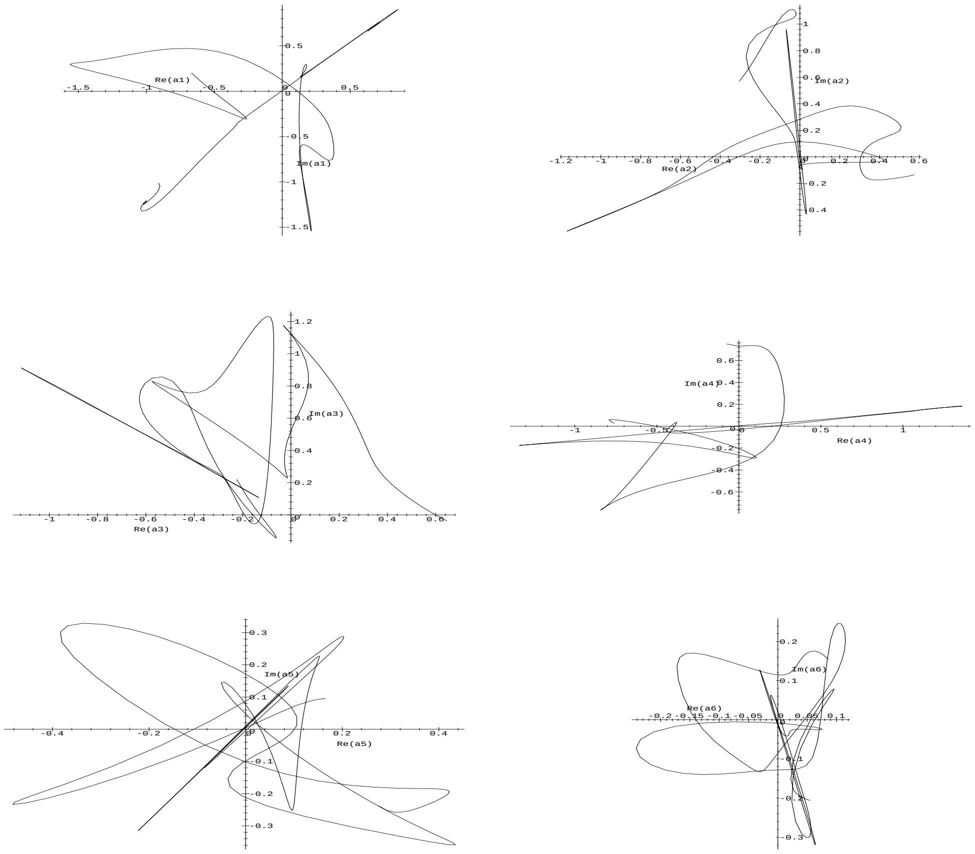

FIGURE 8. From left-to-right then top-to-bottom figures correspond to the $\mathrm{Re}(a_i(t))$ versus $\mathrm{Im}(a_i(t))$ for first six complex Fourier modes where $\alpha = 91.0$.

This regime is currently being investigated further, both using the global nonlinear approach presented in this paper, as well as the method of neural charts introduced in Part II.

5.2. Dynamical Remodeling. We now review the general nonlinear reduction technique applied to dynamical systems [**20, 21**]. The empirical mappings $\mathbf{G}$ and $\mathbf{H}$ will be approximated by the CNLKL procedure.

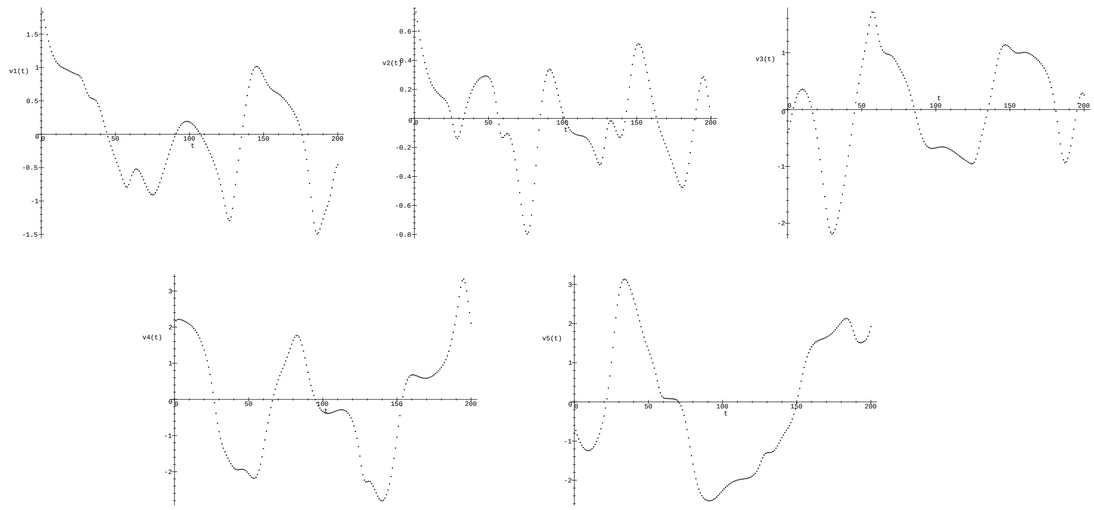

FIGURE 9. Representation of uncorrelated chaotic dynamics on the bottleneck manifold.

Consider a system of ordinary differential equations

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}) \tag{34}$$

which has solutions that asymptotically reside on an attractor $X \subset \mathbb{R}^N$. If the attractor X can be differentiably embedded as a manifold $Y \subset \mathbb{R}^M$ in a lower dimensional space, then the vector field on X may be transported to Y via the embedding.

Let $\mathbf{G}$ be a nonlinear embedding which takes an element of the attractor in X to an element in Y. Specifically $\mathbf{G} : X \to Y$ is given by $y_i(t) = G_i(\mathbf{x}(t))$. The assumption that $\mathbf{G}$ is a differentiable embedding implies that the induced vector field

$$\dot{y}_j(t) = \sum_{i=1}^{n} \frac{\partial G_j}{\partial x_i} F_i(\mathbf{x}(t)) \tag{35}$$

on Y is topologically equivalent to the original dynamical system on X. If the original vector field $\mathbf{F}$ and the embedding function $\mathbf{G}$ are known then equation (35) gives explicit equations for the dynamical system on the low-dimensional manifold Y. Hence it is the construction of such an embedding $\mathbf{G}$ which offers a direct route to the dimensionality reduction for the original system.

Suppose further that we have a mapping $\mathbf{H} : Y \to Z \subset \mathbb{R}^N$ defined by $z_i(t) = H_i(\mathbf{y}(t))$. If $X = Z$ then $\mathbf{H}$ is an inverse of $\mathbf{G}$. In practice, the elements of Z will closely approximate the elements of X, as trajectories, $x_i(t) \approx z_i(t)$. The nonlinearly transformed dynamical system on the reduction Y is then given by

$$\frac{dy_j}{dt}|_{\mathbf{y}(t)} = \sum_{i=1}^{N} \frac{\partial G_j}{\partial x_i} F_i(\mathbf{H}(\mathbf{y}(t))) \tag{36}$$

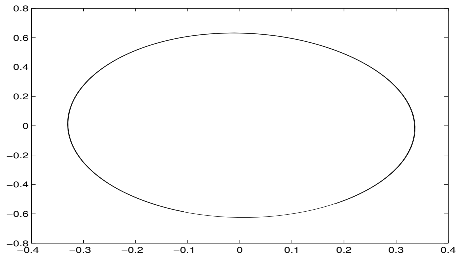

FIGURE 10. Integrated values $v_1(t)$ versus $v_2(t)$ on the bottleneck $\alpha = 84.25$.

where $j = 1 \ldots M$. The Jacobian $\frac{\partial G_j}{\partial x_i}$ is readily computed via equation (7). Hence this formulation has reduced the original system of N equations to a smaller system of $M < N$ equations on Y.

To illustrate the utility of the *nonlinear* reduction we consider the application of the CNLKL transformation to a high-dimensional closed curve corresponding to solutions of the KS equation for $\alpha = 84.25$. The mappings $\mathbf{G}$ and $\mathbf{H}$ were approximated as described in Section 4, using data sampled from the high-dimensional Fourier-Galerkin simulation. It was found that only 2 nodes on the bottleneck layer were required to reproduce the identity mapping to high accuracy. The periodic solution obtained as a result of directly integrating equation (36) is shown in Figure 10. In this case, 2 equations are integrated on the zero-mean uncorrelated bottleneck manifold. If the reconstruction mapping $\mathbf{H}$ is applied to the ellipse in Figure 10, then the high-dimensional trajectories in Figure 11 are reproduced. Similarly, if the reduction mapping $\mathbf{G}$ is applied to the data in Figure 11, then the ellipse in Figure 10 is produced. This nonlinear reduction provides a substantial improvement over previous results with the KL procedure [**18**].

6. Summary and Directions for Future Research

In this paper we develop the CNLKL technique for constructing global nonlinear dimensionality reducing mappings with constraints to model data and dynamical systems in terms of their intrinsic dimensionalities. The CNLKL procedure was applied to the reduction of the Kuramoto-Sivashinsky partial differential equation for several different values of the bifurcation parameter. The initial model consisted of a 20 dimensional dynamical system resulting from a standard Fourier-Galerkin projection of the Kuramoto-Sivashinsky equation. Our examples demonstrated that the reduction of this system by the CNLKL procedure led to low-dimensional coordinate representations which clearly revealed the topological structure of the solution sets. For instance, a high-dimensional limit cycle was reduced and remodeled in the optimal 2D space using the procedure; linear methods fail to provide this minimal reparametrization.

The nonlinear optimization with constraints is implemented as a perturbation of the unconstrained optimization problem. The conjugate gradient algorithm was

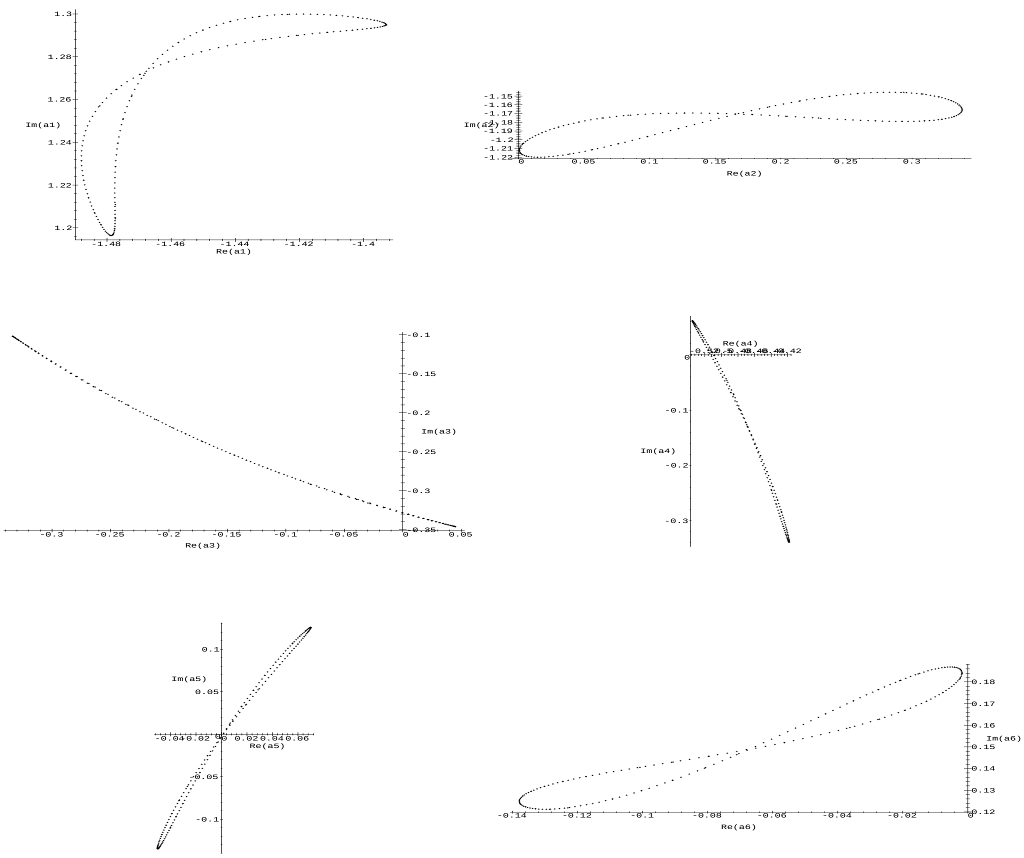

FIGURE 11. From left-to-right figures correspond to the $\mathrm{Re}(a_i(t))$ versus $\mathrm{Im}(a_i(t))$ for the first 6 complex Fourier modes where $\alpha = 84.25$.

employed to determine appropriate weights for approximating the required reduction mappings. While its performance was satisfactory for the problems considered in this paper, other optimization algorithms may be required for larger problems.

The constraints presented here, as well as others, have only begun to be investigated. Further experimenting and analysis is required to assess their relative importance and utility to the reduction and remodeling problem.

In addition to continuing the development of the constrained nonlinear reduction transformation for the analysis of dynamical systems, we plan to investigate the application of the CNLKL procedure for image analysis applications, such as the computation of nonlinear principal images, extending work in [**40, 23, 24**].

Acknowledgements The authors would like to thank Doug Hundley for carefully reading the manuscript and providing useful suggestions. They would also like to thank Katie Coughlin for her efforts in organizing a very stimulating Workshop. The research is supported in part by the NSF under grant numbers ECS-9312092 and DMS-9505863.

References

[1] N. Aubry, R. Guyonnet, and R. Lima. Spatio-temporal symmetries and bifurcations via bi-orthogonal decomposition. *Journal of Nonlinear Science*, 2:183–215, 1992.

[2] N. Aubry, P. Holmes, J.L. Lumley, and E. Stone. Models for coherent structures in the wall layer. *J. Fluid Mech.*, 192:115–172, 1988.

[3] Nadine Aubry, Régis Guyonnet, and Ricardo Lima. Spatio-temporal and statistical symmetries. *Journal of Statistical Physics*, 81(3/4), 1995.

[4] Pierre Baldi, Yves Chauvin, and Kurt Hornik. Back-propagation and unsupervised learning in neural networks. In Yves Chauvin and David E. Rumelhart, editors, *Backpropagation: Theory, Architectures and Applications*, Developments in Connections Theory, chapter 4, pages 99–135. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1995.

[5] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[6] Gal Berkooz, Philip Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flow. *Annu. Rev. Fluid Mech.*, 25:539–575, 1993.

[7] D. Armbruster, J. Guckenheimer, and P. Holmes. Kuramoto-Sivashinsky dynamics on the center-unstable manifold. *SIAM J. Appl. Math.*, 49:676, 1989.

[8] David Demers and Garrison Cottrell. Nonlinear dimensionality reduction. In Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, number 5, pages 580–587, San Mateo, CA, 1993. Morgan Kaufmann Publishers.

[9] Kenji Doya and Allen I. Selverston. Dimension reduction of biological neuron models by artificial neural networks. *Neural Computation*, 6:696–717, 1994.

[10] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, second edition, 1987.

[11] Ciprian Foias, George R. Sell, and Roger Temam. Inertial manifolds for nonlinear evolutionary equations. *Journal of Differential Equations*, 73:309–353, 1988.

[12] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Boston, MA, 1990.

[13] F. Jauberteau, C. Rosier, and R. Temam. The nonlinear Galerkin method in computational fluid dynamics. *Applied Numerical Mathematics*, 6:361–370, 1989/90.

[14] I.T. Jolliffe. *Principal Component Analysis.* Springer, New York, 1986.

[15] M.S. Jolly, I.G. Kevrekidis, and E.S. Titi. Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: Analysis and computations. *Physica D*, 44, 1990.

[16] Michael I. Jordan. Constrained supervised learning. *Journal of Mathematical Psychology*, 36:396–425, 1992.

[17] I.G. Kevrikidis, B. Nicolaenko, and C. Scovel. Back in the saddle again: a computer assisted study of the Kuramoto-Sivashinsky equation. *Siam J. Appl. Math.*, 50:760–790, 1990.

[18] M. Kirby and D. Armbruster. Reconstructing phase-space from PDE simulations. *Z. angew. Math. Phys.*, 43:999–1022, 1992.

[19] M. Kirby, J.P. Boris, and L. Sirovich. An eigenfunction analysis of axisymmetric jet flow. *J. of Comp. Phys.*, 90(1):98–122, 1990.

[20] M. Kirby and R. Miranda. Nonlinear reduction of high-dimensional dynamical systems via neural networks. *Phys. Rev. Letters*, 72(12):1822–1825, 1994.

[21] M. Kirby and R. Miranda. The remodeling of chaotic dynamical systems,. In S.H. Dagli, B.R. Fernandez, J. Ghosh, and R.T. Soundar Kumara, editors, *Intelligent Engineering Through Artificial Neural Networks*, number 4, pages 831–836, New York, 1994. The American Society of Mechanical Engineers.

[22] M. Kirby and R. Miranda. Circular nodes in neural networks. *Neural Computation*, 8(2):390–402, 1996.

[23] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE trans. PAMI*, 12(1):103–108, 1990.
[24] M. Kirby, Frank Weisser, and G. Dangelmayr. Speaking with images: a model problem in the representation of still and moving images. *Pattern Recognition*, 26(1):63–73, 1993.
[25] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.
[26] Mark A. Kramer. Autoassociative neural networks. *Computers chem. Engng*, 16(4):313–328, 1992.
[27] E. Lorenz. Empirical orthogonal eigenfunctions and statistical weather prediction. Science Report No. 1, Statistical Forecasting Project 1, M.I.T., Cambridge, MA, 1956.
[28] J.L. Lumley. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, Nauka, Moscow, 1967. edited by A.M. Yaglom and V.I. Tatarski.
[29] M. Marion and R. Temam. Nonlinear Galerkin methods. *SIAM J. of Numer. Anal.*, 26:1139–1157, 1990.
[30] E. Oja. Data compression, feature extraction, and autoassociation in feedforward neural networks. In T. Kohonen, K. Mäkisara., O. Simula, and J. Kangas, editors, *"Artificial Neural Networks"*, pages 737–745, NY, 1991. Elsevier Science Publishers.
[31] E. Oja. Principal components, minor components and linear neural networks. *Neural Networks*, 5:927–935, 1992.
[32] R. Rico-Martinez and I.G. Kevrekidis. Continuous time modeling of nonlinear systems: A neural network based approach. In *Proc. IEEE Int. Conf. Neural Networks*, volume III, pages 1522–1525, 1993.
[33] James C. Robinson. Inertial manifolds for the Kuramoto-Sivshinsky equation. *Physics Letters A*, 184:190–193, 1994.
[34] J.D. Rodriguez and L. Sirovich. Low-dimensional dynamics for the complex Ginzburg-Landau equation. *Physica D*, 43:77, 1990.
[35] D. E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representation by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition and Perception*, volume 1 of *Computational Models of Cognition and Perception*, chapter 8, pages 318–362. MIT Press, Cambridge, MA, 1986.
[36] David E. Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. In Yves Chauvin and David E. Rumelhart, editors, *Backpropagation: Theory, Architectures and Applications*, Developments in Connections Theory, chapter 1, pages 1–34. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1995.
[37] Terence David Sanger. Optimal unsupervised learning in a single layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989.
[38] Eric Saund. Dimensionality-reduction using connectionist networks. *IEEE Transactions on Pattern Analysis and Machine Intellignence*, 11(3):304–314, 1989.
[39] L. Sirovich. Turbulence and the dynamics of coherent structures, Part I: Coherent structures. *Quarterly of Applied Mathematics*, XLV(3):561–571, 1987.
[40] L. Sirovich and M. Kirby. A low-dimensional procedure for the characterization of human faces. *J. of the Optical Society of America A*, 4:529–524, 1987.
[41] E. Stone and M. Kirby. Dependence of bifurcation structures on the approximation of $O(2)$ symmetry in minimal ODEs. In E.L. Allgower, K. Georg, and R. Miranda, editors, *Exploiting Symmetry in Applied and Numerical Analysis*, volume 29 of *Lectures in Applied Mathematics*, pages 389–404, Providence, R.I., 1993. American Mathematical Society.
[42] A. Weigend, B. Huberman, and D.E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, 1(3):193–209, 1990.
[43] Halbert White. *Artificial Neural Networks: Approximation and Learning Theory*. Blackwell Publishers, Cambridge, MA, 1992.

Department of Mathematics,Colorado State University,Ft. Collins, CO 80523
E-mail address: `kirby@math.colostate.edu`