Efficient chemical kinetic modeling through neural network maps

Neil Shenvi, J. M. Geremia, and Herschel Rabitz

Citation: The Journal of Chemical Physics **120**, 9942 (2004); doi: 10.1063/1.1718305
View online: http://dx.doi.org/10.1063/1.1718305
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/120/21?ver=pdfcov
Published by the AIP Publishing

Efficient chemical kinetic modeling through neural network maps

Neil Shenvi, J. M. Geremia, and Herschel Rabitz
Department of Chemistry, Princeton University, Princeton, New Jersey 08544

(Received 2 August 2002; accepted 5 March 2004)

An approach to modeling nonlinear chemical kinetics using neural networks is introduced. It is found that neural networks based on a simple multivariate polynomial architecture are useful in approximating a wide variety of chemical kinetic systems. The accuracy and efficiency of these ridge polynomial networks (RPNs) are demonstrated by modeling the kinetics of H_2 bromination, formaldehyde oxidation, and H_2+O_2 combustion. RPN kinetic modeling has a broad range of applications, including kinetic parameter inversion, simulation of reactor dynamics, and atmospheric modeling. © *2004 American Institute of Physics.* [DOI: 10.1063/1.1718305]

I. INTRODUCTION

The reaction dynamics of nonlinear kinetic systems can be modeled by systems of differential and algebraic equations. These models depend on rate constants and time, but may also include a dependence on temperature, control parameters, such as flux, transport coefficients, etc., and spatial coordinates. Although sophisticated models can accurately describe complex chemical reactions, they are often extremely costly to evaluate. As a result, many simplifications are generally made to make model evaluation more tractable. For instance, reduced reaction mechanisms and quasi-steady-state approximations are often used as well as the assumptions of contact with a large heat bath or rapid mixing to remove temperature and spatial dependences.[1–4] However, these and other simplifications can be extremely constraining and are often not applicable to a kinetic system under the conditions of interest.

An alternative to simplifying the kinetics is to employ the full model over all relevant laboratory conditions. This process may be accomplished by constructing a *map*[5] $f(\mathbf{x})$ of the true kinetic model $y(\mathbf{x})$, where $\mathbf{x}$ is an N-dimensional vector of input parameters—e.g., rate constants, temperature, initial concentrations, etc.,—and $y(\mathbf{x})$ is an L-dimensional vector of output values—e.g., concentrations over time. A map is a function $f(\mathbf{x})$ that stands in as $y(\mathbf{x})\approx f(\mathbf{x})$ with $f(\mathbf{x})$ having the following characteristics:

(a) It approximates the real reaction model to a high degree of accuracy.

(b) It is easy to construct; i.e., it can be learned without computing the kinetics an exorbitant number of times.

(c) It can be evaluated efficiently.

Once constructed, a map can provide an accurate and efficient substitute to explicitly integrating the full kinetic model.

Maps are useful tools in a variety of problems. Experiments are often performed to identify the values of unknown parameters in reaction models; however, numerical inversion requires large numbers of model evaluations, each of which can be computationally expensive. By performing numerical inversion using the map rather than the real model, the efficiency of the inversion process can often be improved.[6,7] Maps can also be used to enhance the efficiency of executing larger reaction models.[8–11] For example, atmospheric or reactor dynamics models take spatial dependences into consideration by dividing the domain into discrete, homogeneous cells. Each cell displays reaction dynamics that can be modeled by simple ordinary differential equations. However, these equations must be integrated separately for each cell. Using a map to model the cell kinetics can increase the overall model efficiency. Finally, the map itself can often yield important information regarding the underlying physical and chemical behavior of the chemical kinetic system.

It should be stressed that maps cannot replace full model evaluation in general. Because maps are constructed through repeated evaluation of the full model, there is always a significant overhead in constructing any map. Thus, if the model is to be evaluated only a limited number of times or does not have a well-defined, predetermined parameter domain, mapping techniques will not improve the overall efficiency. Instead, maps are best employed to replace an expensive model function which will be evaluated repeatedly on some predefined parameter domain. In such a case, the computational savings of repeatedly evaluating an inexpensive map function as opposed to an expensive model function makes up for the initial overhead of map construction.

Map construction can be accomplished in many ways. In general, maps are generated by explicitly integrating the reaction model for a discrete collection of representative sets of parameters. Maps are evaluated by interpolating between the sampled points to obtain the kinetic results for other values of the parameters. The simplest approach involves sampling over a regular grid in the N-dimensional parameter space. A map can then be generated by N-dimensional interpolation of the output values. Unfortunately, many important kinetic models require a large number of parameters and direct interpolation is extremely inefficient in high dimensions. An N-dimensional look-up table requires sampling $O(s^N)$ points in the parameter space where s is the number of samples per dimension. The computational cost of interpolation also scales exponentially with dimensions. As a result, this direct scheme is impractical for high-dimensional problems.

To avoid the complications of high-dimensional model-

ing, approximations (often linear in nature) have traditionally been adopted. An example is linear regression, where the kinetic model is approximated as a linear combination of P basis functions, $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x}),\ldots,\phi_P(\mathbf{x})$ (Refs. 12 and 13):

$$y(\mathbf{x}) \approx \sum_i^P c_i \phi_i(\mathbf{x}). \tag{1}$$

The values of the expansion coefficients c_i are obtained by sampling the full kinetic model at R points, $R \geqslant P$, on the parameter space and minimizing the squared magnitude of the residuals:

$$J(\mathbf{c}) = \sum_r^R \left(y(\mathbf{x}_r) - \sum_i^P c_i \phi_i(\mathbf{x}_r) \right)^2. \tag{2}$$

However, there are several difficulties in using the linear least-squares regression approach for high-dimensional models: First, the accuracy of the regression depends on the choice of basis functions. In practice, it is difficult to know what basis functions best approximate the model. Only in the limit of an infinite basis set expansion will Eq. (1) be exact for an arbitrary model function. Second, solving the regression equation becomes very costly in high dimensions because the number of basis functions needed increases rapidly. Finally, to ensure that the regression matrix is nonsingular, the number of points sampled must be greater than the number of basis functions. As the number of basis functions increases rapidly with dimension, so does the number of points which must be sampled.

Recently, a family of high-dimensional model representations (HDMRs) has been introduced for addressing the needs of input→output mapping in high dimensions.[5,14,15] HDMR operates by breaking a high-dimensional problem into a hierarchy of lower-dimensional subspaces whose collective contributions approximate the original function. A number of significant successful applications have been achieved with HDMRs.[6,7,9] The present paper aims to explore a special family of neural networks as an alternative to HDMRs.

In order to overcome the drawbacks of the linear representations in Eq. (1), in this paper, we investigate the utility of multilayer networks (MLNs) as chemical kinetics maps. MLNs can be efficient and highly flexible function approximators that have proven useful in a variety of function approximation and classification problems.[16–22] In addition, a proof of Kolmogorov shows that any multivariate function can be approximated to an arbitrary degree of accuracy by certain types of neural networks.[23] Thus the efficiency and generality of neural networks makes them potentially useful for map construction.

Several issues must be addressed regarding the applicability of neural networks to chemical kinetics approximation: First, the size of the network is an issue, and MLNs often require an unfeasibly large number of neurons to successfully model an arbitrary function. Second, the optimal network approximation to an arbitrary function may not be learnable in polynomial time.[24] Although Kolmogorov's theorem implies the existence of an optimal approximation, finding that approximation can be prohibitively difficult. Finally, the problem of overfitting must be considered, as simply fitting a function to the training set is not enough; an effective map must be able to reliably generalize to unsampled regions of the parameter space.

This paper demonstrates that neural networks may be successfully applied to chemical kinetic models and introduces several techniques to increase the performance of neural networks in kinetic problems. Section II gives a brief overview of the theory and architecture of neural networks and compares them to other map generation schemes. The ridge polynomial network (RPN) architecture is discussed and modified for use in kinetic problems. Section III discusses the results of the application of MLNs to several chemical kinetics problems. Conclusions are presented in Sec. IV.

II. NETWORK CONCEPTS

Let $y(\mathbf{x})$ be a model function, where $\mathbf{x}$ is an N-dimensional vector of model parameters. A neural network $f(\mathbf{x})$ can be used to approximate the model function $y(\mathbf{x})$. It will generally be assumed that $y(\mathbf{x})$ is a scalar function (i.e., $L=1$); if $y(\mathbf{x})$ is a vector function, then a separate network can be generated for each component $y_i(\mathbf{x})$ of the output. A simple, two-layer neural network consists of an M-unit hidden layer $\mathbf{z}$ and an one-unit output y'. The hidden layer units, or *neurons*, are connected to the inputs $\mathbf{x}$ through a $M \times N$ connection matrix $\mathbf{W}$. Each component w_{ij} of the connection matrix can be viewed as the strength of the synaptic connection between the jth component of the input and the ith neuron in the hidden layer. The activation of each neuron in the hidden layer is determined by applying an activation function $\sigma_1(\cdot)$ to the sum of its inputs. The activation of a single hidden layer neuron z_i can be expressed as

$$z_i = \sigma_1\left(\sum_j^N w_{ij} x_j \right). \tag{3}$$

The activation of the entire hidden layer $\mathbf{z}$ can be expressed using vector notation as

$$\mathbf{z} = \sigma_1(\mathbf{W} \cdot \mathbf{x}). \tag{4}$$

The output unit y' is connected to the hidden layer neurons through a $1 \times M$ connection matrix $\mathbf{V}$. Each component of the output is determined by applying an activation function $\sigma_2(\cdot)$ to the sum of its inputs. The value of the output y' can be expressed as

$$y' = \sigma_2(\mathbf{V} \cdot \mathbf{z}). \tag{5}$$

Thus a two-layer neural network $f(\mathbf{x})$ can be expressed as

$$f(\mathbf{x}) \equiv f(\mathbf{x}, \mathbf{W}, \mathbf{V}) = \sigma_2(\mathbf{V} \cdot \sigma_1(\mathbf{W} \cdot \mathbf{x})). \tag{6}$$

Figure 1 shows a schematic representation of the two-layer network described above.

Several factors are important to the network's utility as a function approximator. The generality of the network is principally determined by the number of hidden-layer neurons, M. Although Kolmogorov's theorem proves that neural networks are universal function approximators, the proof is only valid in the limit of a prohibitively large number of hidden-

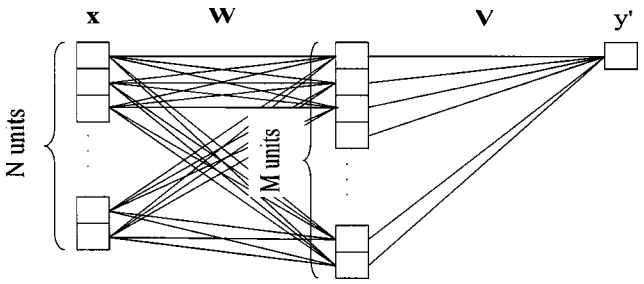

FIG. 1. Schematic representation of a two-layer neural network.

layer neurons. Unfortunately, large values for M improve network accuracy only at the expense of decreased learning and evaluation efficiency. It is also important that the activation function $\sigma_1(\cdot)$ be nonlinear if the network is to be used as a universal function approximator. In most applications, $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ are taken to be sigmoid functions such as the inverse tangent function or the hyperbolic tangent function.

Once the architecture of the network has been fixed, the network must be trained on a set of data composed of input and output pairs. The training set is a collection of R-ordered pairs $(\mathbf{x}_1, y(\mathbf{x}_1))$, $(\mathbf{x}_2, y(\mathbf{x}_2)), \ldots, (\mathbf{x}_R, y(\mathbf{x}_R))$. Training is accomplished by a generalized learning rule that minimizes the magnitude of the residuals between the training set and the network approximation. The best network approximation to the model is found by minimizing the cost function $J(\mathbf{W},\mathbf{V})$,

$$J(\mathbf{W},\mathbf{V}) = \sum_{r=1}^{R} [y(\mathbf{x}_r) - f(\mathbf{x}_r,\mathbf{W},\mathbf{V})]^2, \tag{7}$$

over the components of the connection weight matrices $\mathbf{W}$ and $\mathbf{V}$. This minimization can be accomplished in a number of ways. The most popular method is the error back propagation algorithm, which uses gradient descent to minimize $J(\mathbf{W},\mathbf{V})$ (Ref. 12). The details of the backpropagation algorithm and its implementation can be found in most standard texts on neural networks.[25,26] The back propagation algorithm was used to train all the networks used in this study.

Just as the performance of a linear regression can be improved by the choice of an appropriate basis, the performance of a neural network can be improved by an appropriate choice of the function $\sigma_1(\cdot)$. Traditionally, sigmoidal functions have been chosen as the activation functions, since this type of activation behavior is observed in biological neurons. From the standpoint of map generation, this choice is flexible. If the restriction that each neuron in the hidden layer have an identical, sigmoidal activation function is relaxed, then a neural network with a more general architecture can be constructed. Consider a network of the form

$$f(\mathbf{x}) = c_0 + \sum_{m=1}^{M} f_m(\lambda_m \cdot \mathbf{x}), \tag{8}$$

where c_0 is a constant and $\lambda_1, \lambda_2, \ldots, \lambda_M$ are N-dimensional projection vectors. The ridge functions $f_1(\lambda_1 \cdot \mathbf{x})$, $f_2(\lambda_2 \cdot \mathbf{x}), \ldots, f_M(\lambda_M \cdot \mathbf{x})$ are scalar functions that are each constant along hyperplanes defined by the projection vectors λ_1, $\lambda_2, \ldots, \lambda_M$ (Ref. 27). The network prescribed by Eq. (8) is similar to projection pursuit regression[28,29] (PPR). However, PPR holds the projection axes fixed, whereas networks allow for iterative improvement of the projection axes. This difference allows the network implementation greater flexibility in finding the best projection axes.

Although the ridge functions $f_1(\lambda_1 \cdot \mathbf{x})$, $f_2(\lambda_2 \cdot \mathbf{x}), \ldots, f_M(\lambda_M \cdot \mathbf{x})$ can be arbitrary scalar functions, we will consider the case in which they are restricted to Kth-order polynomials:

$$f_m(\mathbf{x}) = \sum_{k=1}^{K} c_{mk} \left(\sum_{i=1}^{N} \lambda_{mi} x_i \right)^k. \tag{9}$$

Combining Eqs. (8) and (9), we obtain

$$f(\mathbf{x}) = c_0 + \sum_{m=1}^{M} \sum_{k=1}^{K} c_{mk} \left(\sum_{i=1}^{N} \lambda_{mi} x_i \right)^k. \tag{10}$$

Thus $f(\mathbf{x})$ is expanded as a Kth-order, N-dimensional polynomial. Many neural network approximations such as the group method of data handling (GMDH) networks[19] and sigma-psi networks[30] are based on multivariate polynomials. The network described by Eq. (10) is known as a Ridge Polynomial Network.[27] RPNs have been shown to provide excellent approximations to functions with high-order correlation between variables. In contrast, HDMR operates under the assumption that a set of input variables already has been identified with only residual low-order correlations present.[5,31]

Although RPNs possess many attractive qualities, they are no more general as function approximators than traditional linear regression techniques, as Eq. (10) may be rewritten to have the form of Eq. (1) with basis functions ϕ_i that are monomials. In fact, a linear regression would find the optimal set of coefficients $\mathbf{c}$ analytically, whereas a network might require many iterations to converge to a solution. Studies made by Turanyi have shown that low-order multivariate polynomials fitted by linear regression can very effectively approximate kinetic models.[10,11] The PRISM method due to Bell *et al.*, which uses localized polynomial maps on hypercubic domains, has also been used to model chemical kinetics.[32] However, linear regression schemes based on multivariate polynomials face several obstacles.

A Kth-order, N-dimensional polynomial has

$$P = \binom{N+K}{K}$$

terms. As a result, linear regression using such a basis set will require the creation of a $P \times P$ regression matrix and the solution of a set of P linear equations. In high dimensions, this matrix algebra can be prohibitively expensive. Second, because model evaluation is costly, we would rather sample as few points as possible. Unfortunately, to ensure that the regression matrix is nonsingular, *at least* P points must be sampled, but more than P points must be sampled to ensure the robustness of the fit. Finally, the coefficients of a polynomial basis set expansion give very little intuitive information about the behavior of the model. Some of these obstacles can be overcome by using clever methods of matrix analysis to approximate the solutions to the regression equa-

tion. Performance can also be improved by setting noncontributing basis function coefficients to zero.[10] However, these methods are approximate and are most effective in relatively low dimensions.

RPNs are attractive arbitrary function approximators for several reasons: First, it can be shown that ridge polynomial networks are general function approximators.[33] Thus, given enough ridge polynomials (i.e., enough neurons), an RPN can approximate an arbitrary function to an arbitrary degree of accuracy. Second, because RPN construction is based on nonlinear regression, an RPN can include a small number of high-order polynomial ridge functions without including all multivariate monomials of the same order. The examples in Sec. III show that a few high-order ridge polynomials will often approximate a model function much more accurately than many low-order polynomials. Third, the ridge functions f_m and the directions along which they are projected λ_m can provide immediate insight into the underlying structure of the model function.

Another advantage of the RPN is the adaptability of its structure. The performance of RPNs can be greatly improved by taking into account knowledge of the model function when constructing the network. This philosophy has long been accepted in other neural network applications, such as map coloring and digit classification.[21] Network architectures are designed to include empirical and theoretical considerations regarding the output. Although these considerations are often educated guesses, in the case of chemical kinetics they are motivated by general knowledge of the chemical system. For instance, RPNs are particularly ill suited to approximate functions which are upper and lower bounded since multivariate polynomials are unbounded. Such bounded functions occur frequently in kinetics models. If the output y is the concentration of a given species, then this output will be non-negative and will be constrained to a range prescribed by mass balance equations.

Instead of using a standard RPN of the form described by Eq. (10), we will use a modified RPN having a nonlinear activation function $\sigma_2(\cdot)$. If the mass balance equations of a system of reactions constrain the concentration of species X to the range $[0, C]$, then we will use the activation function

$$\sigma_2(x) = \frac{C}{1+e^{-x}}, \tag{11}$$

which will constrain the network output to the range $[0, C]$ for x on the domain $[-\infty, \infty]$. Thus we will use a network described by the equation

$$f(\mathbf{x}) = \sigma_2\left(c_0 + \sum_{m=1}^{M}\sum_{k=1}^{K} c_{mk}\left(\sum_{i=1}^{N} \lambda_{mi} x_i\right)^k\right), \tag{12}$$

where the optimal parameters $\mathbf{c}$ and $\lambda_1, \lambda_2, \ldots, \lambda_M$ will be learned by the backpropagation algorithm.

III. ILLUSTRATIONS

An RPN map generation technique was used to approximate the behavior of three chemical kinetic models. In addition to verifying the accuracy and efficiency of the RPN, the number of ridge functions and the order of the ridge polynomials needed to yield a good representation were also investigated.

A. H_2+Br_2 reaction

The mechanism for the formation of HBr from H_2 and Br_2 involves a radical-mediated chain reaction[34] initiated by the dissociation of Br_2:

$$\begin{aligned}
&Br_2 \rightarrow Br\cdot + Br\cdot, \quad k_1 = 1.58\times 10^{-6}\ s^{-1},\\
&Br\cdot + Br\cdot \rightarrow Br_2, \quad k_2 = 1.34\times 10^{9}\ L\,mol^{-1}\,s^{-1},\\
&Br\cdot + H_2 \rightarrow HBr + H\cdot, \quad k_3 = 7.08\times 10^{2}\ L\,mol^{-1}\,s^{-1},\\
&HBr + H\cdot \rightarrow Br\cdot + H_2, \quad k_4 = 7.08\times 10^{9}\ L\,mol^{-1}\,s^{-1},\\
&Br\cdot + HBr \rightarrow Br_2 + H\cdot, \quad k_5 = 2.51\times 10^{10}\ L\,mol^{-1}\,s^{-1},\\
&Br_2 + H\cdot \rightarrow Br\cdot + HBr, \quad k_6 = 2.09\times 10^{-8}\ L\,mol^{-1}\,s^{-1}.
\end{aligned} \tag{13}$$

The values of the rate constants k_3 through k_6 were taken from the literature.[35–37] The value of k_1 is determined by the initiation mechanism and can be varied by changing the initiator concentration (e.g., light intensity, etc.). The value for k_2 is dependent on the reaction medium. Both of these values were chosen based on literature estimates and agreement with experimental data reported in Ref. 34.

We considered the model output $y(\mathbf{x})$ to be the concentration of HBr measured at L sample times $t_1, t_2, \ldots, t_L$. Hence $y(\mathbf{x})$ is a vector of length L whose components are each a function of the rate constants $\mathbf{k}$ and the other parameters θ, including the initial concentrations, constant temperature, and sample times. The known parameters θ were treated as implicit parts of the model and were assigned values taken from the reaction conditions used by Bodenstein and Lind.[34] Thus the model function was explicitly dependent only on the rate constants $\mathbf{k}$. Because the rate constants varied over several orders of magnitude, it was useful to scale them logarithmically. We defined the model function and its variables as

$$y(\mathbf{x}), \quad \text{with } \mathbf{x} = \log_{10}(\mathbf{k}). \tag{14}$$

Since the reaction mechanism is nonlinear, $y(\mathbf{x})$ cannot be expressed analytically. Instead, the kinetic model was evaluated by selecting a set of rate constants $\mathbf{k}$ and integrating the system of ordinary differential equations prescribed by Eq. (13) to obtain the concentration of HBr at the sample times $t_1, t_2, \ldots, t_L$. The ordinary differential equations corresponding to the mechanism in Eq. (13) were integrated using the VODE integration package.[38]

Because the model function $y(\mathbf{x})$ was an L-dimensional vector with components corresponding to each observation time, a separate network was constructed for each component of the output vector. The networks were trained using a data set of R input–output ordered pairs $(\mathbf{x}_1, y(\mathbf{x}_1))$, $(\mathbf{x}_2, y(\mathbf{x}_2)), \ldots, (\mathbf{x}_R, y(\mathbf{x}_R))$, which were generated by integrating the kinetic model equations. Network convergence was achieved through the backpropagation learning algorithm.[25,26] It was found that the backpropagation algorithm was usually able to converge to a solution after ap-

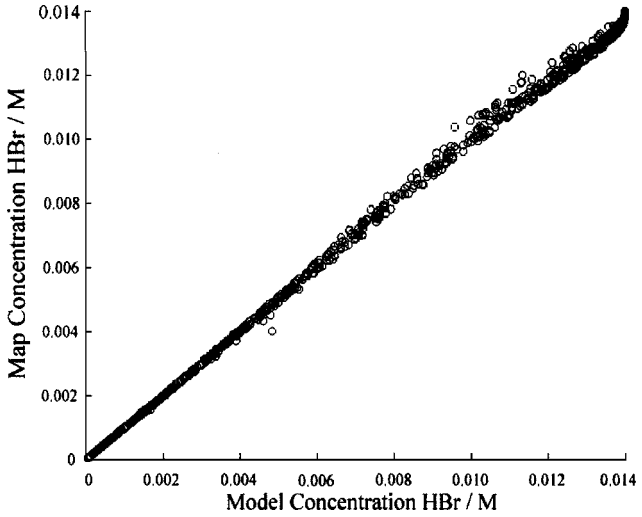

FIG. 2. Map evaluation vs model evaluation of the HBr kinetic system for a 1000-point test set on domain size of {3} at $t=2.4\times10^5$ s. The map function is an RPN with $M=4$ and $K=5$.

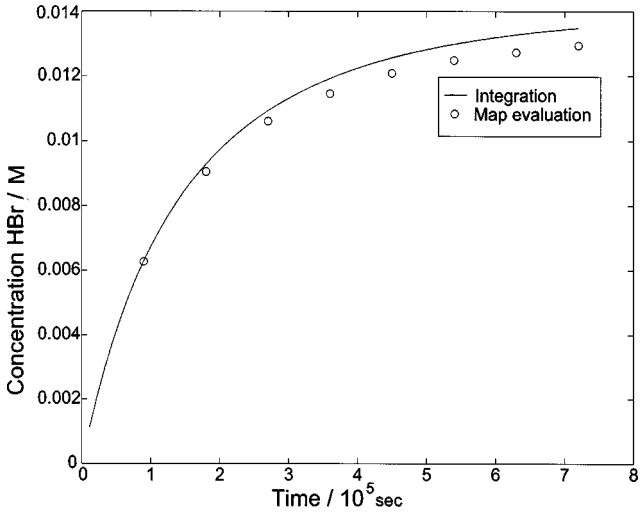

FIG. 3. Comparison of map evaluation with model evaluation for the HBr model function for a point on a domain of size of {3}. The map function is an RPN with $M=4$ and $K=5$.

proximately 10^4 iterations. Given the relatively small sizes of the network and the simplicity of the kinetic equations, the learning process required only a few minutes of computation time. Once the network had converged, a second set of R input–output ordered pairs was generated to test the generalization ability of the network. The accuracy of each network was characterized by the average rms relative error between the model function and the network approximation:

$$J=\sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(\frac{y(\mathbf{x}_r)-f(\mathbf{x}_r)}{y(\mathbf{x}_r)}\right)^2}. \tag{15}$$

The first issue to be considered was the accuracy of the network approximation with respect to domain size. The domain of the parameters $\mathbf{k}$ was characterized by the notation $\{D\}$, where $D=x_i^{\max}-x_i^{\min}$ for $i=1,2,...,N$. For instance, a domain on which k_1 varied between 10^{-8} and $10^{-4}\,\mathrm{s}^{-1}$ would represent a domain size of {4}, since the model parameters $\mathbf{x}$ were scaled logarithmically with respect to the rate constants $\mathbf{k}$. The parameter domains were centered on the literature values for the rate constants given in Eq. (13). The goal of network construction was the generation of a map of the model function on as large a domain as possible without sacrificing accuracy. The size of the network and the order of the approximating polynomial were also considered. Increasing the number of ridge polynomials, M, or the order of the ridge polynomials, K, often increased network accuracy at the expense of increased computational training and a greater possibility of overfitting.

A training set of $R=1000$ points $\mathbf{x}_1$, $\mathbf{x}_2,\ldots,\mathbf{x}_R$ was selected from a uniform distribution over a domain size of {3}. The output values $y(\mathbf{x}_1)$, $y(\mathbf{x}_2),\ldots,y(\mathbf{x}_R)$ were the concentrations of HBr measured at eight sample times between $t=0$ and 2.4×10^5 s. Because the rate constants varied over three orders of magnitude, the reaction dynamics of each model evaluation differed greatly. The range of values of the observable y_8, corresponding to the concentration of HBr at 2.4×10^5 s, varied over almost three orders of magnitude. Some of the systems in the training set had nearly reached completion after 2.4×10^5 s, while others were far from completion.

An RPN using $M=4$ ridge polynomials of order $K=5$ was trained over 10^4 iterations on the training set data. The network converged to a solution which fit the training set data with an average relative error of $<3.0\%$ at each sample time. Figure 2 shows a scatter plot of the network approximation versus the model evaluation for $t=2.4\times10^5$ s. The trained network was able to predict the output of a 10^3 point test set (different from the training set) with an average relative error of $<3.3\%$. Thus the network demonstrated highly accurate predictive ability. Figure 3 shows a comparison of model evaluation and network evaluation for one of the test set systems.

In order to investigate the dependence of map accuracy on network size, we studied the behavior of the neural network trained on the last component of the output vector, y_8. It was found that using fewer ridge functions or lower-order polynomials tended to decrease network accuracy. Conversely, using more ridge functions and higher-order polynomials led to increased approximation accuracy. However, when comparing networks with an equal number of total parameters, networks with a small number of high-order polynomial ridge functions often outperformed networks with a large number of low-order polynomial ridge functions. Table I shows the average rms relative error on the test

TABLE I. Average rms relative error of the network approximation to HBr kinetics for a domain size of {3} for a test set consisting of 10^3 points.

		M				
		2	3	4	5	6
K	4	18.10%	3.13%	3.01%	3.02%	2.98%
	5	2.90%	2.89%	2.87%	2.82%	2.96%
	6	2.60%	2.58%	2.52%	2.54%	2.25%
	7	10.74%	2.76%	2.65%	2.57%	2.55%
	8	11.24%	2.83%	2.54%	2.29%	2.38%

set data achieved by various networks after 10^4 iterations. Although the larger, higher-order networks were more accurate, they required increased computation time for each learning iteration.

The accuracy of the network approximation was also dependent on domain size. Training sets consisting of 10^3 points were generated for domain sizes of {2}, {3}, and {4}, and an RPN with $M=4$ and $K=5$ was used to approximate the model function over these domains. Table II shows the accuracy of the RPN on the different domains. As expected, the network was more accurate on smaller domains.

TABLE II. Average rms relative errors for RPN map of HBr kinetics with $M=4$ and $K=5$ for a test set consisting of 10^3 points on various parameter domains.

Domain size	Training set	Test set
{2}	1.34%	1.55%
{3}	2.80%	2.87%
{4}	4.04%	4.37%

B. Oxidation of formaldehyde

Although the formation of HBr displays nonlinear kinetics, it is a relatively small system. Many interesting kinetic systems contain tens or more of species and hundreds of reactions. Therefore, it is useful to test the applicability of RPN map generation to higher-dimensional kinetic systems. The mechanism of the oxidation of formaldehyde by OH includes 15 species and 17 reactions:

$$
\begin{aligned}
&\mathrm{H}+\mathrm{NO_2}\rightarrow\mathrm{OH}+\mathrm{NO}, \quad 1.28\times10^{-10}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{OH}+\mathrm{H_2CO}\rightarrow\mathrm{H_2O}+\mathrm{HCO}, \quad 7.75\times10^{-12}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{HCO}+\mathrm{NO_2}\rightarrow\mathrm{H}+\mathrm{CO_2}+\mathrm{NO}, \quad 4.00\times10^{-11}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{HCO}+\mathrm{O_2}\rightarrow\mathrm{HO_2}+\mathrm{CO}, \quad 5.50\times10^{-12}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{HO_2}+\mathrm{NO}\rightarrow\mathrm{OH}+\mathrm{NO_2}, \quad 8.30\times10^{-12}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{H}+\mathrm{O_2}(+M)\rightarrow\mathrm{HO_2}(+M), \quad 5.44\times10^{-15}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{H}+\mathrm{H_2CO}\rightarrow\mathrm{H_2}+\mathrm{HCO}, \quad 6.69\times10^{-14}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{OH}+\mathrm{HCO}\rightarrow\mathrm{H_2O}+\mathrm{CO}, \quad 5.00\times10^{-11}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{H}+\mathrm{HCO}\rightarrow\mathrm{H_2}+\mathrm{CO}, \quad 5.00\times10^{-11}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{H}\rightarrow\text{destruction}, \quad 10\ \mathrm{s^{-1}},\\
&\mathrm{OH}\rightarrow\text{destruction}, \quad 5\ \mathrm{s^{-1}},\\
&\mathrm{HCO}\rightarrow\text{destruction}, \quad 10\ \mathrm{s^{-1}},\\
&\mathrm{HO_2}\rightarrow\text{destruction}, \quad 10\ \mathrm{s^{-1}},\\
&\mathrm{OH}+\mathrm{OH}\rightarrow\mathrm{H_2O}+\mathrm{O}, \quad 1.99\times10^{-12}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{OH}+\mathrm{HO_2}\rightarrow\mathrm{H_2O}+\mathrm{O_2}, \quad 6.40\times10^{-11}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{H}+\mathrm{HO_2}\rightarrow\mathrm{OH}+\mathrm{OH}, \quad 7.00\times10^{-11}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}},\\
&\mathrm{OH}+\mathrm{H_2CO}\rightarrow\mathrm{HCOOH}+\mathrm{H}, \quad 2.00\times10^{-13}\ \mathrm{cm^3\,molecule^{-1}\,s^{-1}}.
\end{aligned}
\tag{16}
$$

The rate constants for the reactions listed above were taken from a study of the system performed by Yetter *et al.*[39] As in the previous case, the initial concentrations, temperature, and sample times were taken as implicit constant components of the model function. For the purposes of testing the RPN approximation, we examined the network trained on the concentration of OH measured at 0.05 s.

A training set of 10^3 input–output pairs was generated on a domain of size {2}. It should be noted that the output values of the training set spanned almost seven orders of magnitude. Thus the model function exhibited behavior over a large dynamic range. The mass balance equations for this mechanism indicate that the concentration of OH will never exceed its initial concentration of 1.95×10^{12} molecule/cm^3. However, the mass balance equations for this mechanism did not provide tight bounds for the concentration of OH at 0.05 s. For instance, none of the data points in the training set showed output values of more than 10% of the mass balance

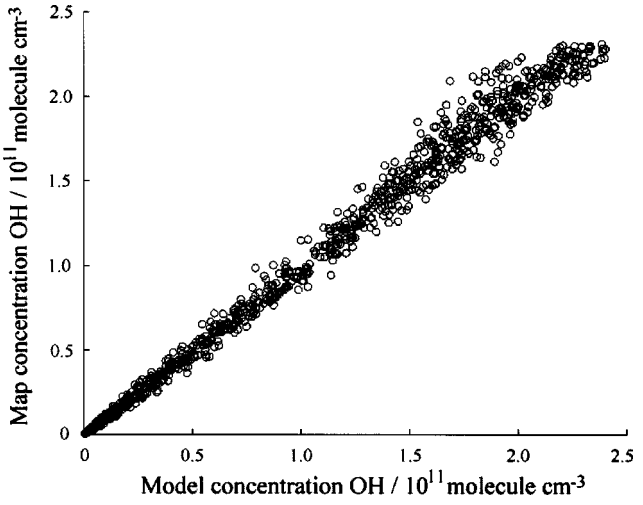

FIG. 4. Map evaluation vs model evaluation of the formaldehyde kinetic system for a test set consisting of 10^3 points on a domain size of {2} at $t=0.05$ s. Map function is an RPN with $M=5$ and $K=5$.

upper limit. This behavior is due in part to the first-order destruction of OH through reaction (11) which causes an exponential decay of OH. Rather than use the mass balance equations to provide an upper bound for the activation function $\sigma_2(\cdot)$, we used the maximum output value of the training set to guide the choice of the activation function. This straightforward approach was found to work well.

An RPN using $M=5$ ridge polynomials of order $K=5$ was trained over 10^4 iterations on the data set. The trained network was able to approximate the model function with an average rms relative error of 6.88%. Figure 4 shows a plot of model evaluation versus network prediction for the 10^3 training set points. The network also displayed good generalization ability by producing 8.88% relative error for a test set of 10^3 points. As in the HBr system, increasing the size of the network was found to reduce the error in learning the training set. Table III shows the accuracy of networks of various sizes. Although larger networks tended to outperform smaller networks, the difference in performance was often small.

C. H_2+O_2 combustion

The final kinetic system was the combustion of H_2 and O_2 to form H_2O. A recent study by Clifford *et al.* used multivariate polynomial regression to generate a map approximation to this system.[8] Following the procedure of Clifford *et al.*, we used the Leeds methane oxidation mechanism[40] which involves 9 species and 46 reactions. The model function used for this system related the values of the state variables $\mathbf{x}(t)$ at time t to the values of the state variables at a later time, $t+\Delta t$. We can write this model function as

$$\mathbf{x}(t+\Delta t)=y(\mathbf{x}(t)). \tag{17}$$

The state variables $\mathbf{x}$ can be any set of system variables such as the temperature, species concentrations, or pressure. We selected the complete set of species concentrations as the state variables $\mathbf{x}$ because the previous study made by Clifford *et al.* was unable to model the behavior of these variables using multivariate polynomial regression due to numerical instability. Thus, according to Eq. (17), the RPN map acted as an integrator propagating the system $\mathbf{x}(t)\rightarrow\mathbf{x}(t+\Delta t)$ for $\Delta t=10^{-6}$ s. Repeated iteration of the map (i.e., the output of the map at $t+\Delta t$ comes the input for the map at $t+2\Delta t$, etc.) produces a fast integrator, replacing the original differential equations. A map application of this type was developed for atmospheric ozone chemistry with HDMR (Ref. 9).

The training set for the network was generated as follows: the initial pressure and temperature of the system were 2.1 atm and 1673 K, respectively. The initial reaction mixture contained a mole fraction of 0.70 Ar. The remainder of the initial reaction mixture was made up of H_2 and O_2. The ratio of H_2 to O_2 in the initial reaction mixture was selected randomly from a range between 1.0 and 4.0. The temperature and volume of the reaction mixture were held constant over the course of the reaction. We defined the measure of completion β to be the ratio of H_2O present to the stoichiometric amount of H_2O present at equilibrium:

$$\beta=\frac{[H_2O]}{[H_2O]_{\max}}. \tag{18}$$

Each system was integrated until it reached $\beta=0.96$ and data points were recorded every 10^{-6} s. Because the species concentrations varied over a dynamic range of almost three orders of magnitude during the reaction, we used the logarithms of the concentrations as the model variables. Following the procedure of Turanyi *et al.*, data points were neglected during the initial stage of the reaction ($\beta<0.04$) and the final stage of reaction ($\beta>0.96$). This process was repeated for 100 different initial reaction mixtures, generating over 4000 data points, 1000 of which were randomly selected to be used as the training set.

It should be noted that this data set represents a sampling of the relatively low-dimensional manifold in concentration space described by the H_2/O_2 combustion. Thus much of the concentration space remains unexplored by our sample data. As a result, our neural network map will not be able to replace the integrator code for any randomly selected point in the concentration space. However, it often happens that in a physically motivated simulation—i.e., a simulation of reactor dynamics—the subspace of interest is only a low-dimensional manifold of the concentration space (see, for instance, Ref. 32). Hence, while use of a network map as an implicit integration routine will not likely replace an integrator code in the general case, it may provide a significant speed-up over explicit integration when the subspace of interest is highly parametrized.

TABLE III. Average rms relative error of the network approximation to formaldehyde kinetics on a domain size of {2} for a test set 10^3 points.

		M				
		3	6	9	12	15
K	3	17.59%	14.55%	15.04%	14.90%	14.90%
	4	15.82%	10.67%	10.52%	9.71%	11.33%
	5	21.51%	8.88%	9.15%	8.16%	8.42%
	6	14.11%	8.69%	8.56%	6.28%	7.87%
	7	15.97%	8.08%	10.09%	6.34%	7.07%

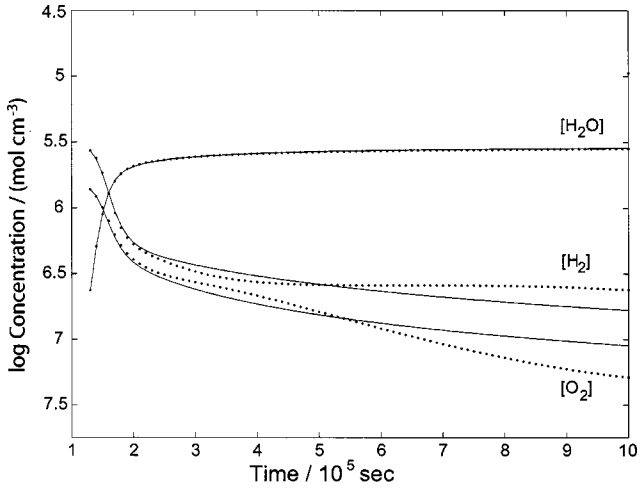

FIG. 5. H_2, O_2, and H_2O concentrations vs time for the H_2+O_2 kinetic system calculated from the original kinetic model (solid lines) and from the RPN maps (dots) with step size 10^{-6} s.

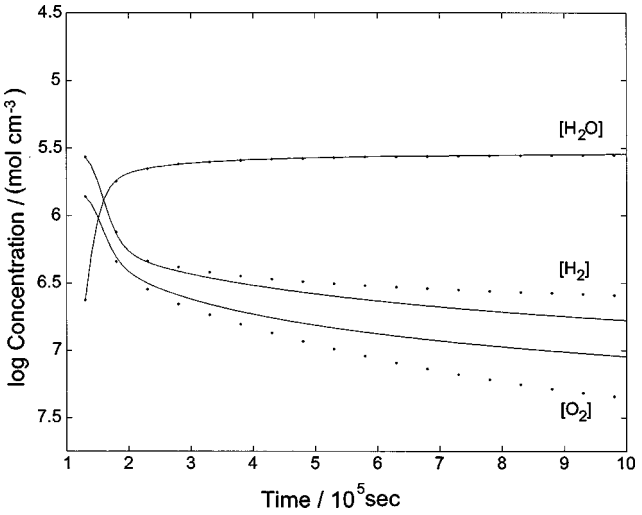

FIG. 7. H_2, O_2, and H_2O concentrations vs time for the H_2+O_2 kinetic system calculated from the original kinetic model (solid lines) and from the RPN maps (dots) with step size 5×10^{-6} s.

RPN maps were generated using $M=6$ ridge polynomials of order $K=6$. A separate map was trained over 10^4 iterations for each component of the nine-dimensional output, $y(\mathbf{x})$. All of the maps displayed average rms relative errors of $<2\%$ for a single time step [i.e., $\mathbf{x}(t)\rightarrow\mathbf{x}(t+\Delta t)$]. The collection of maps was then used in place of the integrator to generate the trajectories of all the species for a test set of initial conditions on which the network was not trained. The trajectories of the species are shown in Figs. 5 and 6. The map approximation shows good accuracy in reproducing the results of numerical integration, except for later times in the reaction for species in small concentrations.

To test the capabilities of the RPN maps for modeling with larger step sizes, a second training set was created by recording data points at a larger interval of $\Delta t=5\times10^{-6}$ s. Using a set of 10^2 random initial conditions, 938 data points were generated, all of which were used as the training set. As before, nine RPN maps were generated using $M=6$ ridge polynomials of order $K=6$. All of the maps again displayed average rms errors of less than 2% for a single step, $\mathbf{x}(t)\rightarrow\mathbf{x}(t+\Delta t)$. This new collection of maps was used in place of the integrator for the same initial conditions as the previous trial. The trajectories obtained are shown in Figs. 7 and 8. Although the step size for this set of maps was five times larger than the first set of maps in Figs. 5 and 6, the new maps displayed almost identical accuracy.

Finally, we considered the efficiency of RPN evaluation relative to the evaluation of the model function. To ensure that the comparisons were valid, the VODE integrator was run using the largest possible error tolerance per step for which the result showed comparable accuracy to the map solutions. Even when the integrator code was pushed to the limit of stability, map evaluation was still significantly faster. Table IV shows the computation time required for the evaluation of an RPN relative to model evaluation for the various kinetic systems studied.

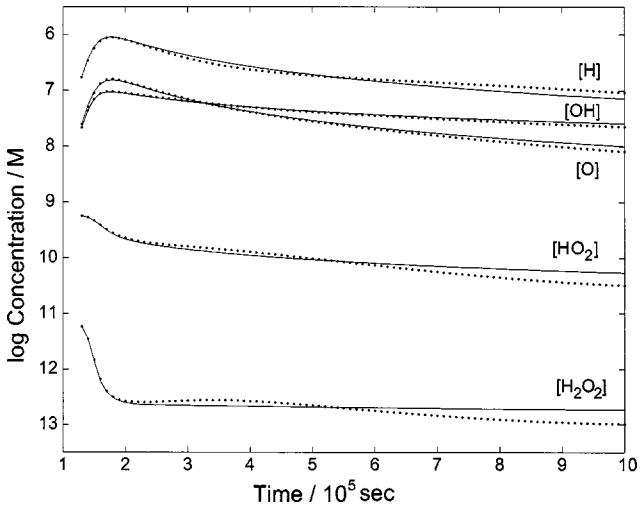

FIG. 6. O, OH, H, HO_2, and H_2O_2 concentrations vs time for the H_2+O_2 kinetic system calculated from the original kinetic model (solid lines) and from the RPN maps (dots) with step size 10^{-6} s.

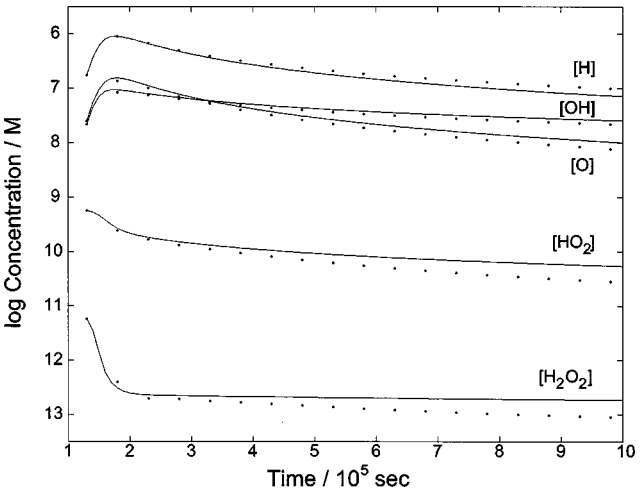

FIG. 8. O, OH, H, HO_2, and H_2O_2 concentrations vs time for the H_2+O_2 kinetic system calculated from the original kinetic model (solid lines) and from the RPN maps (dots) with step size 5×10^{-6} s.

TABLE IV. Efficiency and accuracy of model evaluation vs network evaluation. In the case of the H_2O kinetic maps, the map error refers to the error of a single integration step.

	Model	RPN	Increase in speed	Map error
HBr	9.0×10^{-3} s	9.2×10^{-5} s	98	<4%
Formaldehyde	1.7×10^{-3} s	1.3×10^{-5} s	131	<9%
H_2O, $\Delta t=10^{-6}$	4.1×10^{-4} s	1.7×10^{-4} s	2.5	<2%
H_2O, $\Delta t=5\times10^{-6}$	1.7×10^{-3} s	1.7×10^{-4} s	10	<2%

Computation time was defined as the time required to evaluate the entire model function using the RPN approximation or explicitly integrating the chemical kinetics. For instance, for the HBr system, a single evaluation consisted of computing the concentration of HBr at the eight sample times for a given set of rate constants. For the formaldehyde system, a single evaluation consisted of computing the concentration of OH at 0.05 s for a given set of rate constants. For the H_2/O_2 system, a single evaluation consisted of computing the concentrations of all nine species at time $t+\Delta t$ given the concentrations at time t.

For this last system, special care was taken to make sure that the efficiency comparisons were valid. Because the VODE integrator has a large overhead at the beginning of the computation and then becomes more efficient as the integration progresses, we used the cost of a single Δt time step averaged over an entire trajectory from t_0 to t_f. However, in some applications, certain parameters such as the rate constants, temperature, etc., will change after each time step and necessitate recalculation of the system Jacobian. Thus averaging over a long trajectory represents a scenario in which the overhead of numerical integration is minimized.

In all our test cases, the neural network maps were more efficient than direct model evaluation. The computation times listed in Table IV *do not* include the overhead of map generation, which will generally require an initial evaluation of the model at a large number of points on the parameter domain of interest. However, if an application requires extensive evaluation of a model over a known parameter domain (as do many atmospheric and reactor dynamics simulations), then the initial cost of map construction can be negligible relative to the savings in computation time due to substitution of the map function for the model function in subsequent evaluations.

IV. CONCLUSIONS

RPN networks were found to provide excellent approximations to complex kinetic models for the three systems studied, despite significant differences in the chemistry and overall dynamics of the various models. The cases studied included those where the parameters were rate constants and initial conditions. Network evaluation was one to two orders of magnitude faster than model evaluation. These efficient network maps could play important roles in improving the speed of existing kinetics and nonlinear parameter inversion codes.

The RPN map generation shares some of the characteristics and limitations of other interpolative map generation methods. First, because RPN modeling is inherently an *interpolative* modeling scheme, it assumes the intrinsic smoothness of the underlying model functions. Thus neural network modeling would not be expected to succeed in systems displaying chaotic behavior or in any systems which are intrinsically nonsmooth. Second, it is unlikely that neural network maps would be successful in applications which demanded *extrapolation* into unsampled regions of the parameter space. RPNs share these limitations with all interpolative modeling schemes. However, unlike most existing map generation methods, RPNs are not based on linear regression and thus can overcome many of the obstacles of high dimensionality.

In the modeling of the HBr, formaldehyde, and H_2O systems, the neural network maps were accurate despite the high dimensionality and the sparseness of the training data. The training sets for all three systems consisted of 10^3 points sampled from a uniform distribution over the parameter domain. Due to the high dimensionality of the model functions, this sampling is exceedingly sparse. For instance, consider the six-dimensional HBr model function and the nine-dimensional H_2O model function. If 10^3 points were sampled over a regular six-dimensional mesh and a regular nine-dimensional mesh, there would be 3.2 and 2.2 points sampled per dimension, respectively. The formaldehyde model function is even more extreme. Here 10^3 points on a 17-dimensional mesh correspond to only 1.5 points per dimension. Thus RPNs combine the desirable sparse data interpolation properties of projection pursuit regression and the nonlinearity of neural networks. These properties are very important when modeling complex kinetic functions. If model evaluation is costly, then the training set should be generated using as few points as possible.

The RPN architecture, although successful, could be improved. In particular, the use of a nonlinear activation function to constrain the network output demonstrated the utility of intelligent network design. For instance, conservation of species relations, equilibrium limits, and time-scale dependences could all be included explicitly in the neural network model. Future studies will address the problem of tailoring network architecture to kinetic modeling.

ACKNOWLEDGMENT

The authors would like to thank Professor Robert Gunning of the Princeton University Department of Mathematics for his assistance.

[1] L. Györgyi, T. Turányi, and R. J. Field, J. Phys. Chem. **94**, 7162 (1990).
[2] A. Rojnuckarin, C. A. Floudas, H. Rabitz, and R. A. Yetter, J. Phys. Chem. **97**, 11689 (1993).
[3] T. Turányi, New J. Chem. **14**, 795 (1990).
[4] T. Turányi, L. Györgyi, and R. J. Field, J. Phys. Chem. **97**, 1931 (1993).
[5] H. Rabitz, Ö. F. Aliş, J. Shorter, and K. Shim, Comput. Phys. Commun. **117**, 11 (1999).
[6] J. M. Geremia, H. Rabitz, and C. Rosenthal, J. Chem. Phys. **114**, 9325 (2001).
[7] N. Shenvi, J. M. Geremia, and H. Rabitz, J. Phys. Chem. A **106**, 12315 (2002).

[8] L. J. Clifford, A. M. Milne, T. Turányi, and D. Boulton, Combust. Flame **113**, 106 (1998).
[9] J. A. Shorter, P. C. Ip, and H. Rabitz, J. Phys. Chem. A **103**, 7192 (1999).
[10] T. Turányi, Comput. Chem. (Oxford) **18**, 45 (1994).
[11] T. Turányi, in *Proceedings of the Twenty-Fifth Symposium (International) on Combustion* (The Combustion Institute, Pittsburgh, 1994), pp. 949–955.
[12] Y. Bard, *Nonlinear Parameter Estimarion* (Academic, New York, 1974).
[13] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences* (McGraw-Hill, New York, 1992).
[14] H. Rabitz and Ö Aliş, J. Math. Chem. **25**, 197 (1999).
[15] H. Rabitz and Ö. Aliş, in *Senistivity Analysis*, edited by A. Saltelli, K. Chan, and M. Scott (Wiley, New York, 2000), pp. 199–223.
[16] B. Cheng and D. Titterington, Stat. Sci. **9**, 2 (1994).
[17] R. Durbin and D. E. Rumelhart, Neural Comput. **1**, 133 (1989).
[18] E. Eisenstein, I. Kanter, D. A. Kessler, and W. Kinzel, Phys. Rev. Lett. **74**, 6 (1994).
[19] G. E. Fulcher and D. E. Brown, IEEE Trans. Neural Netw. **5**, 372 (1994).
[20] C. L. Giles and T. Maxwell, Appl. Opt. **26**, 4972 (1987).
[21] M. Tusar, J. Supan, and J. Gasteigen, J. Chim. Phys. Phys.-Chim. Biol. **89**, 1517 (1992).
[22] S. Ventura, M. Silva, D. Perez-Bendito, and C. Hervas, J. Chem. Inf. Comput. Sci. **37**, 517 (1997).
[23] R. Hecht-Nielsen, in *Proceedings of the IEEE International Conference on Neural Networks* (IEEE, New York, 1989), pp. 593–605.
[24] S. Judd, *Neural Network Design and the Complexity of Learning* (MIT Press, Cambridge, MA, 1990).
[25] J.-K. Wu, *Neural Networks and Simulation Methods* (Dekker, New York, 1994).
[26] J. Zurada, *Introduction to Artificial Neural Systems* (West, St. Paul, 1992).
[27] Y. Shin and J. Ghosh, IEEE Trans. Neural Netw. **6**, 610 (1995).
[28] J. Friedman and W. Stuetzle, J. Am. Stat. Assoc. **70**, 817 (1981).
[29] P. J. Huber, Ann. Stat. **13**, 435 (1985).
[30] J. Ghosh and Y. Shin, Int. J. Neural Syst. **3**, 323 (1992).
[31] H. Rabitz and K. Shim, J. Chem. Phys. **111**, 10640 (1999).
[32] J. B. Bell, N. J. Brown, M. S. Day, M. Frenklach, J. F. Grcar, R. M. Propp, and S. R. Tonse, in Proceedings of the Combustion Institute, 2000, Vol. 28.
[33] C. K. Chui and X. Li, in *Multivariate Approximations: From CAGD to Wavelets*, edited by K. Jetter and F. I. Utreras (World Scientific, Singapore, 1993),
pp. 77–89.
[34] M. Bodenstein and S. C. Lind, Z. Phys. Chem., Neue Folge **57**, 168 (1907).
[35] I. Glassman, *Combustion*, 2nd ed. (Academic, New York, 1987).
[36] *CRC Handbook of Bimolecular and Termolecular Gas Reactions*, edited by J. Kerr (CRC, Boca Raton, 1981).
[37] K. Laidler, *Chemical Kinetics*, 3rd ed. (Harper and Row, New York, 1987).
[38] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **10**, 1038 (1989).
[39] R. A. Yetter, H. Rabitz, F. L. Dryer, R. G. Maki, and R. B. Klemm, J. Chem. Phys. **91**, 4088 (1989).
[40] http://www.chem.leeds.ac.uk/combustion/hydrox.dat, 2001