The Journal of Chemical Physics

A random-sampling high dimensional model representation neural network for building potential energy surfaces

Cite as: J. Chem. Phys. **125**, 084109 (2006); https://doi.org/10.1063/1.2336223
Submitted: 13 June 2006 . Accepted: 18 July 2006 . Published Online: 24 August 2006

Sergei Manzhos, and Tucker Carrington

View Online   Export Citation

ARTICLES YOU MAY BE INTERESTED IN

Atom-centered symmetry functions for constructing high-dimensional neural network potentials
The Journal of Chemical Physics **134**, 074106 (2011); https://doi.org/10.1063/1.3553717

Perspective: Machine learning potentials for atomistic simulations
The Journal of Chemical Physics **145**, 170901 (2016); https://doi.org/10.1063/1.4966192

Using neural networks to represent potential surfaces as sums of products
The Journal of Chemical Physics **125**, 194105 (2006); https://doi.org/10.1063/1.2387950

A random-sampling high dimensional model representation neural network for building potential energy surfaces

Sergei Manzhos[a)] and Tucker Carrington, Jr.[b)]
Département de chimie, Université de Montréal, Case postale 6128, succursale Centre-ville, Montréal (Québec) H3C 3J7, Canada

(Received 13 June 2006; accepted 18 July 2006; published online 24 August 2006)

We combine the high dimensional model representation (HDMR) idea of Rabitz and co-workers [J. Phys. Chem. **110**, 2474 (2006)] with neural network (NN) fits to obtain an effective means of building multidimensional potentials. We verify that it is possible to determine an accurate many-dimensional potential by doing low dimensional fits. The final potential is a sum of terms each of which depends on a subset of the coordinates. This form facilitates quantum dynamics calculations. We use NNs to represent HDMR component functions that minimize error mode term by mode term. This NN procedure makes it possible to construct high-order component functions which in turn enable us to determine a good potential. It is shown that the number of available potential points determines the order of the HDMR which should be used. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2336223]

I. INTRODUCTION

It is frequently the case in chemical physics, and in science in general, that one knows the value of a function (output) at a set of points in multidimensional space (inputs) and would like to use that information to obtain function values at other points. For example, one does *ab initio* calculations to compute the Born-Oppenheimer potential at a set of points and would like to use those potential values to determine the potential at other points. In this paper, we focus on the potential representation problem, but the ideas we discuss are more general and could be used to solve many input-output problems. Several examples of input-output problems in chemistry are listed in Ref. 1.

In general, representing input-output functions is hard if the dimensionality of the input space is high and the number of input points is large. Potential fitting becomes more arduous as the number of atoms is increased and as the number of *ab initio* points required to obtain a good representation of the potential increases. For a problem with D coordinates $(x_1,x_2,\ldots,x_D)$, it might be thought that one would need n^D points, where n is a representative number of points (probably between 10 and 100) for a single coordinate, to fit the potential. If D is not small, this is clearly an extremely difficult problem. Nonetheless, the problem is often tractable, because potentials are smooth, and because it is not necessary to represent the potential in the full D-dimensional configuration space. A good fitting method is one that uses parameters that are well determined by the known potential values (*ab initio* points). Employing a fitting function whose parameters are well determined makes it possible to reduce the number of input *ab initio* points.

In this article, we combine high dimensional model representation[1–10] (HDMR) and multimode-type[11–15] ideas with a neural network[16–18] to develop a practical HDMR scheme for fitting potentials. Many of the ideas could also be employed for other problems where an HDMR can be used. The multimode representation of the potential and the HDMR approach are based on the simple and powerful idea that potentials can be written as *finite* expansions of mode terms. The potential is represented as a sum of a constant term, terms that depend on only a single coordinate, terms that depend on two coordinates, etc. This representation is useful, because terms that depend on more coordinates are typically smaller than terms that depend on fewer coordinates. The crudest representation of this type is a sum of separable (one-coordinate) terms.[19] Obviously, a separable potential will not permit one to compute accurate vibrational spectra, but for many problems, it has the right general shape. Note that even retaining only terms that depend on a single coordinate yields a potential that correctly includes all the anharmonicity (because it includes cubic, quartic, etc., terms in each of the coordinates).

The multimode potential representation[11,12,15,20] effectively used to accelerate vibrational self-consistent field and vibrational configuration interaction calculations is a particular case of the HDMR approach. Following Rabitz *et al.*[8] and Rabitz and Alis,[9] we denote a function that depends on m coordinates an m-coordinate component function. We call the sum of the m-coordinate component functions the m-mode term and use Mth order HDMR to mean an HDMR that includes no m-mode terms with m larger than M. If m-mode terms with, say, $m>4$ of the HDMR for a D-dimensional potential are negligible, then clearly, it is much better to use HDMR to fit than to use a general approach that ignores the HDMR structure. Ignoring the HDMR structure forces one to do full D-dimensional fits, whereas exploiting the HDMR structure makes it possible to fit the potential while doing only four-dimensional (4D) fits. Fitting to an HDMR is certainly a good way, maybe the only way, of obtaining high

[a)] Electronic mail: sergei.manzhos@umontreal.ca
[b)] Electronic mail: tucker.carrington@umontreal.ca

dimensional potential energy surfaces (PESs). The accuracy and ease of use of an HDMR depend on the choice of the component functions. One option[11–13] is to use what Rabitz *et al.* and Rabitz and Asis call cut HDMR. Another option is a random-sampling high dimensional model representation (RS-HDMR).[3,10] We suggest using neural networks (NNs) to fit component functions that minimize the fitting error. HDMR, NNs, and the RS-HDMR-NN combination are explained in Secs. II A–II C, respectively.

The idea of representing a potential as a sum of terms that depend on one, two, three, etc., coordinates is not new. Using interatomic distance coordinates, it was exploited by Murrell and co-workers,[14,21–26] who used a many-body expansion. Many good potentials were fitted with this approach.[14] The approach of Murrell and co-workers depends on the use of interatomic distance coordinates, but the basic HDMR idea can be used with any set of coordinates. Murrell and co-workers developed good techniques for fitting the interatomic distance component functions. It would be difficult to study dynamics in interatomic coordinates. If some set of more suitable coordinates (e.g., Jacobi coordinates) is used to compute dynamical properties, perhaps a vibrational spectrum, evaluating quadratures of the potential is costly, because the integrals are full-dimensional (the cost of doing quadratures scales exponentially with the number of coordinates in the function being integrated).

An HDMR not only facilitates fitting but is also advantageous because it can be used to make a simple representation of the potential in terms of coordinates suitable for studying dynamics. If an HDMR in suitable coordinates is used, one can directly exploit the fact that full-dimensional quadrature is not necessary. This motivated Carter and Bowman's choice of the multimode representation.[11–13] For the same reason, the potfit algorithm is used to accelerate multiconfiguration time-dependent Hartree-Fock calculations.[27–29] If m-mode terms of an HDMR potential with, for example, $m>4$ are negligible and a product basis is used, potential integrals can be computed by doing only 4D quadratures. When the number of atoms is large, this is clearly much less costly than a full-dimensional quadrature. The HDMR+NN fitting procedure we propose in this paper yields potentials, in coordinates suitable for dynamics calculations, that are written as sums of terms each of which depends on a small number of coordinates and therefore facilitates calculating matrix elements of the potential. It is an important advantage to be able to use lower-dimensional quadratures. We refer to a potential for which matrix elements may be computed with lower-dimensional quadratures as a quadrature-friendly potential. The HDMR+NN procedure could be used either to fit potentials to *ab initio* data or to refit some previously determined (fit or interpolated) potential to obtain a quadrature-friendly potential.

II. HIGH DIMENSIONAL MODEL REPRESENTATION NEURAL NETWORK

A. High dimensional model representation

The idea of representing a function as a sum of mode terms is very attractive. Various forms of this idea have been presented.[1,8,11,12,15,20,30] Mathematicians have also studied this problem. In particular, Sobol worked out a decomposition of a function into "different-dimensional" terms using the Haar series.[31,32] We adopt the notation of Rabitz and co-workers,[1–10,33–39] who have used HDMR to map kinetic equations,[1,4,6,8] to relate atom-atom or atom-diatom interaction potentials to rovibrational spectra and elastic or inelastic cross sections,[34,35,38] to study optimal control,[1,33] to estimate the properties of molecules or substances from their composition,[1,37] to predict compositions with desirable properties,[8] and (in conjunction with the reproducing kernel Hilbert space[40,41] method) to fit a three-dimensional reactive potential surface.[39] A multidimensional function of $\mathbf{x} \equiv (x_1,x_2,\ldots,x_D) \subset K^D = [0,1]^D$ is represented as a sum of mode terms each of which is a sum of component functions,

$$f^{\mathrm{HDMR}}(x_1,x_2,\ldots,x_D) = f_0 + \sum_{i=1}^{D} f_i(x_i) + \sum_{i<j=1}^{D} f_{ij}(x_i,x_j) + \ldots + f_{12\ldots D}(x_1,x_2,\ldots,x_D), \quad (1)$$

where $f_{i_1,i_2,\ldots,i_n}$ are the component functions. The range of each of the coordinates is shifted and scaled to [0, 1]. HDMR is useful, because the m-mode terms become smaller as m is increased.[1,8] The component functions can be constructed in different ways. Rabitz and co-workers discuss two HDMR options that they call cut HDMR[4,7–9,42] and random sampling HDMR (RS-HDMR).[2,3,5,6,8–10] For both, component functions minimize the error functional,[8]

$$\int_{K^D} [f(\mathbf{x}) - f^{\mathrm{HDMR}}(\mathbf{x})]^2 d\mu(\mathbf{x}). \quad (2)$$

The measure $d\mu$ determines the particular form of the error functional and of the component functions. In principle, it would be possible to use Eqs. (1) and (2) to determine simultaneously the best component functions. It is of course much easier to decouple component functions and determine them independently.[2,3,5,6,8–10] If a constraint is imposed, the independently determined and simultaneously determined component functions are identical. Determining the component functions independently is computationally less expensive. It has the disadvantage that independently determined component functions only minimize Eq. (2) if they are subject to a constraint. They are therefore nonoptimal.

In cut HDMR, $d\mu$ is

$$d\mu(x) = \prod_{i=1}^{D} \delta(x_i - y_i)dx_i, \quad (3)$$

where $\mathbf{y}$ is the so-called expansion center. Cut HDMR is equivalent to the multimode procedure of Refs. 11, 12, and 15. All component functions except f_0 are optimized subject to the constraint that a component function must be zero if one of the arguments is equal to its value at the expansion center. The resulting component functions are

$$f_0 = f(\mathbf{y}), \quad f_i(x_i) = f(x_i|\mathbf{y}) - f_0,$$
$$f_{ij}(x_i,x_j) = f(x_i,x_j|\mathbf{y}) - f_i(x_i) - f_j(x_j) - f_0, \quad \ldots, \tag{4}$$

where the notation $f(x_{i_1},x_{i_2},\ldots,x_{i_n}|\mathbf{y})$ means that the components of $\boldsymbol{x}$ other than $x_{i_1},x_{i_2},\ldots,x_{i_n}$ are set equal to those of the expansion center. Bowman and co-workers have used cut HDMR extensively. Cut HDMR is conceptually simple and it has been demonstrated that the associated potential is good enough to enable the calculation of quite accurate vibrational energy levels.[11–13,30,42,43] Nonetheless, it has three important drawbacks. First, the cut HMDR (Refs. 1, 8, and 9) component functions do not minimize error in all of configuration space, but only along slices, planes, etc., passing through $\mathbf{y}$. It is reasonable to expect component functions that minimize error in all space to be in some sense better. Second, the potential is sampled on multidimensional cuts or slices; therefore, for each component function, a separate set of *ab initio* calculations is needed, and the number of sets increases combinatorially with dimension.[15] Third, the quality of a (truncated) cut HMDR depends on the choice of the expansion center. Often, the equilibrium geometry is used as the expansion center.[11–15] However, the choice of the expansion center may influence the quality of a truncated HDMR.[44] A multicut HDMR has been introduced that uses multiple reference points to generalize cut HDMR.[7]

In this paper, we introduce a new potential fitting method based on RS-HDMR.[1,3,10] RS-HDMR component functions have advantages: (1) they minimize error not just along slices, planes, etc., but in all space; and (2) they can be determined from *one set* of points. It is possible not only to minimize error in all space but also to minimize error with respect to a probability distribution chosen to accentuate the quality of the HDMR in certain regions of space (e.g., close to the equilibrium configuration). To obtain a RS-HDMR, one sets $d\mu(\boldsymbol{x})=w(\boldsymbol{x})\Pi_{i=1}^{D}dx_i$ in Eq. (2) When $w(\boldsymbol{x}) =\Pi_{i=1}^{D}w_i(x_i)$, it can be proved[10] that simultaneously and independently determined component functions are identical if

$$\int_{K^1} w_s(x_s)f_{j_1,\ldots,j_k}(x_{j_1},\ldots,x_{j_k})dx_s = 0, \quad s \in \{j_1,j_2,\ldots,j_k\}. \tag{5}$$

One finds the following RS-HDMR component functions:

$$f_0 = \int_{K^D} w(\mathbf{x})f(\mathbf{x})d\mathbf{x},$$

$$f_i(x_i) = \int_{K^{D-1}} \prod_{\substack{k=1\\k\neq i}}^{D} w_k(x_k)f(\mathbf{x})d\mathbf{x}^i - f_0,$$

$$f_{ij}(x_i,x_j) = \int_{K^{D-2}} \prod_{\substack{k=1\\k\neq i,j}}^{D} w_k(x_k)f(\mathbf{x})d\mathbf{x}^{ij} - f_i(x_i) - f_j(x_j) - f_0, \tag{6}$$

where $d\boldsymbol{x}^{i_1,\ldots i_l}$ is $d\boldsymbol{x}$ with $x_{i_1},\ldots,x_{i_l}$ removed. Being a constrained minimization, it is nonoptimal. If w is not factorizable, similar equations can be used, but without a special basis expansion, which in effect changes the definition of the component functions, independently and simultaneously determined component functions are not identical.[3]

It is not easy to use Eqs. (6) because of the many-dimensional integrals. They can be evaluated with Monte Carlo (MC) methods.[2,5] The most obvious approach is to compute each component function on a grid of points. Because different integrals (with different integration variables) are required for different component functions, there are many MC integrals to compute. To do all of these integrals, a huge number of points is necessary. This approach has the additional disadvantage that once component functions are generated on a grid, one must interpolate between the grid values to obtain an HDMR that can be used to determine f^{HDMR} at any point. To resolve these problems, Rabitz and co-workers represent the component functions in terms of basis functions.[2,3,5,10] They used basis functions that are products of one-dimensional (1D) functions and argue that the best 1-D functions are orthogonal polynomials.[5] This approach enables one to obtain all the component functions from a single set of points and yields continuous component functions that are known everywhere, so there is no need for interpolation. It works well and has been employed to solve several problems.[1–3,6,10] Nonetheless, it has deficiencies, and the goal of this paper is to develop a better way of representing and determining the component functions. Some of the disadvantages of the basis set approach are: (1) It might be necessary to use many basis functions to represent the component functions. For potential fitting, the component functions will depend on the choice of coordinates and could be quite complicated. In general, a large number of product basis functions will be required. (2) The MC evaluation of the integrals is tricky, because the integrand is not smooth, and the integrals are harder for higher-order component functions. It would be hard to determine four-coordinate component functions. Applications of HDMR stop at the third order.[1,4,6,8,33–35,37,38] (3) It would be advantageous to work with a weight function that favored important regions of f (e.g., the bottom of the potential), but it is not simple to find polynomials that are orthogonal with respect to a general weight function. The basis set approach is easiest to use if the weight function is factorizable and simple. (4) If the degree of the polynomials is as large as only three, and if the number of fitting points is not large, the polynomial fits have unphysical oscillations unless they are tamed by regularization.[5] (5) Imperfect orthogonality creates problems.[3,5]

We introduce a new scheme for determining RS-HDMR component functions to build higher-order RS-HMDRs of PESs, in as many as seven dimensions and test them by computing energy levels. Rather than representing the component functions with product basis functions, we use NNs.[16–18] NNs are general fitting methods that can be used, in principle, to fit any function. It is well known that full-dimensional potentials can be accurately fitted with NNs.[45–56] In a previous paper, we have shown that energy levels computed on NN fits to PESs of three- and four-atom molecules have errors of less than 1 cm^{-1} and that NNs work

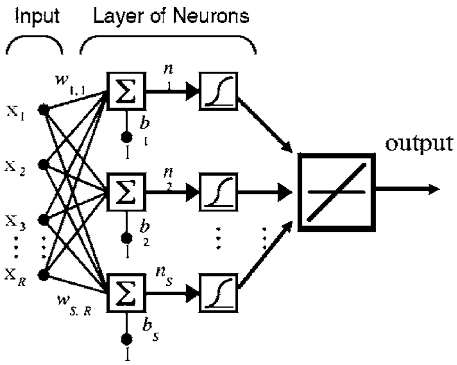

FIG. 1. Classic single-hidden-layer backpropagation neural network used to fit the potential surfaces. The connection weights matrix w is used to form linear combinations of the input variables $\boldsymbol{x}$ via $\boldsymbol{wx}+\boldsymbol{b}$. This is indicated by Σ in the figure. These linear combinations are arguments of the sigmoid node or neurons. A linear combination of the nodes outputs is the final network output.

well on potentials of quite different shapes.[48] Notwithstanding the advantages of NNs, it seems likely that NN fits for larger molecules or large regions of configuration space would be difficult. NNs should, however, be excellent for the purpose of fitting the component functions. If the HDMR idea works, it is not necessary to fit functions of more than a few coordinates. We determine mode terms sequentially and show that our component functions are equivalent to those one would obtain from a simultaneous fit with constraints imposed on the component functions. The constraints are milder than those of Rabitz and co-workers, and we therefore expect, independent of the issue of the quality of NN versus polynomial fits, our component functions to be better. The NN component functions are also known in all space. It can be shown that they are fits to solutions of systems of integral equations (see the Appendix).

B. Neural networks

In principle, any fitting method could be used to construct the component functions. Because we wish to use the same fitting method to determine all of the component functions, and because we want it to work for a wide variety of potential surfaces (bound and unbound systems), we need one that is general: we do not want to tune the fitting method for each component function and for each potential. The general or black-box fitting method we prefer is the NN approach.[16–18,57]

A classic NN is built from univariate nonlinear functions, σ, called nodes, transfer functions, or neurons (see Fig. 1). Each node receives as input a linear combination of values and outputs one number. The outputs are combined and become the input for the next layer. A commonly used[45–56,58] single-hidden-layer NN fitting function is

$$f(x_1,\ldots,x_D)=\sum_{q=1}^{\text{No. of nodes}} c_q\sigma\left(\sum_{p=1}^{D} w_{qp}x_p+b_q\right), \tag{7}$$

where $\sigma(x)=(1+e^x)^{-1}$ is the node function for the first (hidden) layer. It is most advantageous to use node functions of sigmoid type.[59,60] The parameters c_q, w_{qp}, and b_q are adjusted to obtain a good fit. There are theorems proving that with the right choice of the parameters, it is possible to fit any function to arbitrary accuracy with a NN with one hidden layer of neurons.[59,61–63] We are therefore certain that as we increase the number of fitting points and neurons, the quality of our fit should improve. To demonstrate that a NN can be used to fit any function, one uses results from the field of functional representation theory[64,65] and, in particular, theorems of Kolmogorov[66] and Sprecher.[67,68] According to these theorems, it is always possible to represent a multidimensional continuous function of $\boldsymbol{x}$ in terms of a nonlinear function of a linear combination of continuous one-dimensional functions of new variables obtained from the $\boldsymbol{x}$. Furthermore, mathematicians argue that the number of NN parameters required to obtain a good fit scales well as dimensionality is increased.[64,69–73] An important advantage of NNs is the fact that although they make use of univariate functions to fit, they do not fit with sums of products of functions. They should therefore scale less badly with dimension than many fitting methods.

NNs are good for the purpose of fitting component functions, because the NN parameters are determined using a general procedure that works for any choice of coordinates. The same fitting method can be used for all the component functions and for any potential. It is straightforward to obtain a more accurate potential by simply increasing the size of the network. There are no problems with oscillations, nonorthogonality, etc. It is easy to choose a point distribution that reflects the relative importance of different regions of the potential.

C. Random-sampling high dimensional model representation neural network

We choose to absorb f_0 into the one-mode term and therefore use the HDMR expansion

$$f^{\text{HDMR}}(\mathbf{x})=\sum_{i=1}^{D} f_i(x_i)+\sum_{1\leq i<j\leq D} f_{ij}(x_i,x_j)+\ldots$$
$$+f_{12\ldots D}(x_1,x_2,\ldots,x_D). \tag{8}$$

Component functions are determined by minimizing the integral in Eq. (1) using $d\mu(\boldsymbol{x})=w(\boldsymbol{x})\Pi_{i=1}^{D}dx_i$. In principle, one could determine all of the component functions simultaneously. Although this would give the best possible fit, it is in practice easier to find either component functions or mode terms sequentially or independently. Such component functions and component functions determined simultaneously are identical if a constraint is imposed. Rabitz and co-workers[9,10] determined component functions independently by doing functional derivatives of the integral of Eq. (2) and imposing the constraint in Eq. (5). It is possible to treat mode terms sequentially but to minimize the component functions of a mode term simultaneously. This should yield a better fit. One way to do this is to solve the integral equations presented in the Appendix. Another way is to minimize Eq. (2) indirectly by fitting the mode terms. We fit them with a NN. Either way, minimization is done with respect to a

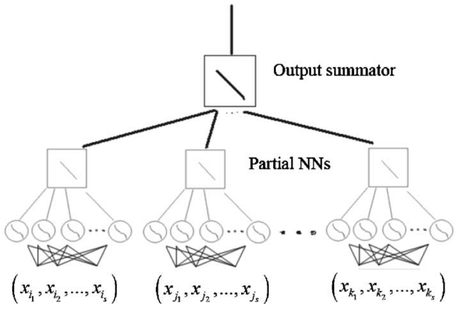
Output summator
Partial NNs
$(x_{i_1}, x_{i_2}, \ldots, x_{i_s})$
$(x_{j_1}, x_{j_2}, \ldots, x_{j_s})$
$(x_{k_1}, x_{k_2}, \ldots, x_{k_s})$

FIG. 2. The structure of the NN for a mode term.

weight function $w(\boldsymbol{x})$ which need not be known explicitly. In the Appendix, we show that a sequential fit of mode terms is equivalent to a simultaneous fit with a constraint. The constraint is weaker than the constraint imposed in Refs. 3 and 8–10 to ensure equivalence of component functions obtained independently and simultaneously. This is essentially due to the fact that we fit all the component functions of a mode term together. A weaker constraint should mean a better fit, because the best fit would be obtained with no constraint. The fitted component functions approximate the solutions of systems of integral equations derived in the Appendix.

Beginning with the one-mode term, the mode terms are determined sequentially. A fit for the s-mode term is built by fitting a NN to the difference of f and the sum of the previously determined mode terms. For example, the three-mode term is obtained by fitting $f(\bar{x}) - (\Sigma_{i=1}^{D} f_i(x_i) + \Sigma_{1 \leqslant i < j \leqslant D} f_{ij}(x_i, x_j))$. The NN with which we fit a mode term is a sum of partial NNs, one for each component function. This NN structure is shown in Fig. 2. The s-mode term can be represented as one NN with C_D^s s-dimensional inputs, where C_D^s is a binomial coefficient. Each s-dimensional input is associated with a different component function and consists of a different set of s coordinates of a point ${}^k\boldsymbol{x}$, $k=1 \cdots N$, where N is the number of sampling points at which the potential is known. Note that all mode terms are fitted using one set of sampling points. There is one output layer and $2C_D^s$ hidden layers (two for each partial NN). For each partial NN, the first hidden layer has a variable number of sigmoid neurons and the second hidden layer has one linear (or saturated linear[74]) neuron. The output layer is a single linear neuron. This NN, built from C_D^s partial NNs for the C_D^s component functions, is trained as a single NN by adjusting parameters so that the NN output obtained by summing the partial outputs nearly coincides with the difference between f and the sum of the previously determined mode terms.All partial NNs for a given mode term have the same number of neurons, but different mode terms may have different numbers of neurons. The fitting is done with the Levenberg-Marquardt (LM) algorithm with Bayesian regularization.[74] The regularization effectively removes unnecessary neurons, which means that we are free to begin the fit with too many. This is useful because it allows us to begin with the same number of neurons for each component function. Our RS-HDMR-NN is programed in MATLAB® using neural network toolbox.[74,75]

1. Lumping mode terms

In the previous section, we outline a scheme that enables one to determine optimal mode sterms sequentially. This must be better than determining optimal component functions sequentially. For similar reasons, it should be better to determine several mode terms simultaneously. Lumping mode-terms in this fashion has two advantages. First, in the approach of the previous section, the total number of partial NNs (a measure of the number of fitting parameters) is $\Sigma_s C_D^s$. If D is large, there is a huge number of partial NNs. If we lump, say, the one-mode, two-mode, and three-mode terms together, we need only C_D^3 partial NNs. Second, fitting mode terms together rather than separately removes constraints and therefore enables us to obtain a better fit. If all the mode terms are lumped together, the optimization is unconstrained, and this would obviously give the best fit. One might worry that lumping the one-mode, two-mode, and three-mode terms together will make doing quadratures more complicated, because there will be no terms in the potential with only one or two coordinates. That, however, is unimportant, because if the three-mode term is included, it will be necessary to do three-dimensional (3D) quadratures anyway.

There are circumstances in which one might not want to lump terms. In general, it is not known *a priori* what order of HDMR will be needed for a particular application. To be certain that the order one uses is not larger than necessary, one should fit mode terms one at a time, because doing so gives the best fit at each order. Fitting term by term also has the advantage that it might sometimes facilitate fits of the higher-order terms. For example, when terms are fitted one by one, the three-mode term is fitted to a difference potential obtained by subtracting one-mode and two-mode fits from the full potential. If this difference potential is smooth, nearly flat, or otherwise well behaved, it should be easier to fit than the lumped 1-2-3 mode term. In previous work, it has been established that fitting a difference potential can improve the quality of the final fitted function outside and close to the limits of the range of the fitting points.[48,76] If one fits term by term, it is also possible to use different subsets of the fitting points for different mode terms. This might be useful.

2. Grouping parameters

In the approach of Sec. II C, there is one partial NN per component function. Unfortunately, the number of component functions and, therefore, the number of fitting parameters increase combinatorially with dimension. In practice, this means that if the number of coordinates is larger than about six, the HDMR+NN fitting procedure is costly. It is possible to speed up the fit by varying the parameters in groups during the fit. The idea is general. If there are K fit parameters $\{p_i\}$, $i=1,2,\ldots,$, K, one can proceed as follows: fix the values of all parameters except $\{p_j\}$, $j=j_1, j_2, \ldots, j_l$, $\{j_1, j_2, \ldots, j_l\} \subset \{1, 2, \ldots, K\}$ and vary $\{p_j\}$ to reduce the error, then fix all except $\{p_m\}$, $m=m_1, m_2, \ldots, m_k$, $\{m_1, m_2, \ldots, m_k\} \subset \{1, 2, \ldots, K\}$ and vary $\{p_m\}$, etc. When all the parameters have been varied, one repeats the procedure, i.e., varies $\{p_j\}$ again while keeping all other parameters fixed, etc. This obviously works best if the groups of parameters are nearly uncorrelated; if parameters are strongly correlated, many cycles are required, but it is still possible to get a good fit. In effect, we are imposing a jagged form on the optimization path in the *full* parameter space. If the parameter jumps are small enough, this should work well. Note that this is not the same as fitting the component functions sequentially,[8,9] because during the course of the fit all of the parameters for a mode term are varied. The idea is more closely related to parameter-by-parameter and layer-by-layer NN learning approaches,[77–79] in which weights and biases are found by linearizing neurons. The linearization causes stalling and has other serious drawbacks. In contrast, we use a second-order fitting algorithms (LM).[80] Over and above the choice of the method used to determine the parameters, one expects grouping the parameters of a component function to work better than grouping subsets of parameters of a full-dimensional NN. The NN parameters are in general strongly correlated, but if a component function is small, its parameters and those of other component functions will be only weakly correlated. If we group parameters, then partial NNs can be fitted independently, and this reduces the memory cost. For the PES fits of Sec. III, grouping parameters also decreases the CPU cost. Because the number of parameters in a group is much less than the total number of parameters, grouping reduces the cost of a LM step. We have verified that, for a comparable number of LM steps, the error achieved using parameter grouping is similar to the global-fit error. To fit, we therefore (1) fix the parameters of all partial NNs except one, (2) allow LM to take several (about $e=10$) steps adjusting the nonfixed parameters to reduce the error, (3) change the identity of the partial NN for which the parameters are not fixed and repeat, and (4) repeat in cycles nc times. Ideally, one would choose $e=1$ so as to cause the smallest possible deviation from the minimum-gradient path. This, however, would necessitate doing many cycles so in practice, we choose the number of LM steps e so that it is much smaller than what is needed to converge a partial NN. With the computer we use $e \cong 12$ is approximately optimal. The total number of LM steps is therefore $e \times \mathrm{nc}$ and is about 1000 in our fits.

III. TESTS AND DISCUSSION

A. How well does a truncated HDMR fit the hydrogen peroxide potential?

We used the RS-HDMR-NN fitting procedure outlined in Sec. II to refit the H_2O_2 PES of Kuhn *et al.*[81] It is standard to test new fitting methods by fitting a known function.[76,82–90] This enables one to directly assess the value of the fitting method without dealing with errors in the potential points themselves. Levels computed on the Kuhn surface agree well with experiment.[91–97] We know[48] that it is possible using NNs, but not an HDMR, to fit the Kuhn surface well enough that levels calculated with the fitted surface differ from levels of the Kuhn surface by <1 cm^{-1}. We wish now to demonstrate that a good fit can be obtained using a truncated HDMR and NNs. The difficulty of a pure NN fit increases significantly as the number of coordinates increases. A pure NN fit for H_2O_2 is possible, but pure NN fits for larger molecules would be very costly or perhaps impossible. Demonstrating that truncated HDMRs can be used to fit potentials is therefore very important.

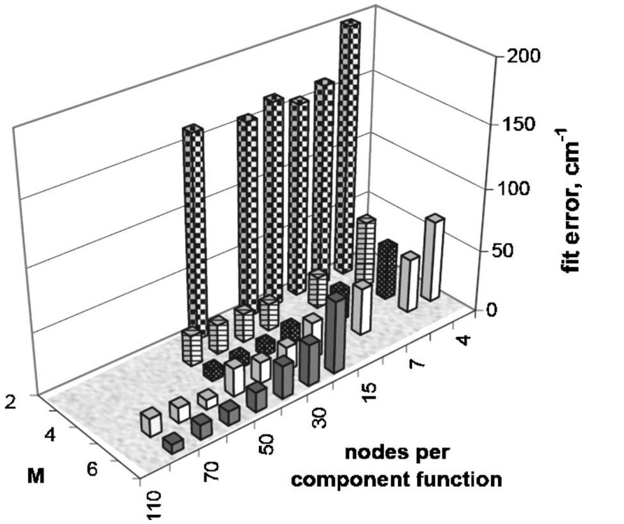

FIG. 3. H_2O_2 test-set rmse for different orders of RS-HDMR-NN with different numbers of nodes per component function. If the number of nodes is large enough, then due to overfitting the error may increase as the number of nodes is increased. This is particularly visible for the fifth order HDMR error. Orders of HDMR are chessboard pattern - second, stripes - third, black with white dots - fourth, white - fifth, and gray -sixth.

The set of fitting points we use is very similar to the set we used in Ref. 48. There are 5000 symmetry unique points with potential values as large as 10 000 cm^{-1}, distributed according to a probability distribution function proportional to an inverted separable potential.[98] See Ref. 48 for details. Symmetrically equivalent points (obtained by exchanging the OH units) and points along equilibrium bond coordinate slices up to 18 000 cm^{-1} are included in the fitting set. This set of points favors the bottom of the well. As in Ref. 48, the fit is done in bond angle and bond length coordinates. Each NN is trained to minimize a mean square error for the fitting points. The quality of all fits is assessed by computing the root mean square error (rmse) for a set of test points of equal size. We aim to attain a surface with a rmse of about 10 cm^{-1}. A surface of this quality reproduces energy levels to within about 1 cm^{-1}.[48] The number of fitting points and the rmse we achieve are similar to those of other H_2O_2 fits.[48,76] In Sec. II C. (1), we discuss the advantages of lumping together mode terms. For H_2O_2, we use one NN for the one-mode term and lump other mode terms together into another NN. The maximum number of coupled coordinates is denoted M, i.e., we lump the two-mode, three-mode,..., M-mode terms together.

In Fig. 3, we plot the rmse for the test points as a function of M and the number of nodes per component function. For a fixed value of M, the rmse generally decreases as the number of nodes is increased, but when the number of nodes is large enough, it attains a constant value. This is due to the finite number of fitting points and the fact that M is not large enough. By increasing the number of nodes and M, we can always improve the fit at the fitting points,[59,61–63] but the number of fitting points limits the extent to which we can decrease the rmse at the test points. In Fig. 3, the attained

TABLE I. H_2O_2 fit quality with different orders of RS-HDMR-NN.

Order	Nodes per component function	No. of component functions	Test-set rmse (cm^{-1})
2	20	15	156
3	30	20	23.4
4	40	15	8.9
5	60	6	7.7
6	90	1	7.4

TABLE II. Mean absolute errors of the first 48 vibrational levels of A_g symmetry of H_2O_2 on top of the ZPE with different orders of HDMR-NN.

Order	Nodes per component function	No. of component functions	Level error (cm^{-1})
2	20	15	10.35
3	30	20	1.14
4	40	15	0.44
5	60	6	0.37
6	90	1	0.35

rmse decreases as M is increased. In Table I, we list, for several M, the best attainable test-set rmse and the number of nodes per partial NN required to achieve this rmse. The final rmse's of all of the fits in this section slightly depend on the initial values of the NN weights and biases.[48,58] The values in the tables and figures are representative. rmse's for fits with $M>1$ are insensitive to the choice of the initial conditions.[48] For $M=1$, it is sometimes necessary to do a few fits to obtain a good rmse. With $M=4$, we obtain a rmse of less than 10 cm^{-1}. This error is close to the error of *ab initio* methods commonly used to produce fitting points,[99–101] and we therefore consider this a good fit. The fourth-order fit in Table I required about a day of CPU time on one UltraSPARC -III 900 MHz processor of a Sun Fire 880 computer. In general, we expect computing the *ab initio* points to take much longer than fitting the surface.

For both the reference and the HDMR-NN fitted potential, we compute energy levels and subtract the appropriate zero point energy (ZPE). The magnitude of the error of these differences is plotted in Fig. 4. It is important to assess the quality of a potential fit on the basis of the accuracy of observables computed using it. The energy levels are computed in diatom-diatom Jacobi coordinates with a product basis Lanczos method.[91,102–104] The basis we use has six potential optimized discrete variable representation (PO-DVR) (Refs. 105 and 106) functions for r(OO), and three PO-DVR functions for both the r(OH), $l_{max}=15$ and $m_{max}=14$. This small basis is used, because it enables us to do quick calculations. It is not large enough to converge the vibrational levels, but comparison of computed levels, for the reference and fitted surfaces, enables us to assess how different their shapes are and how good the fit is. (We have done calculations with a basis large enough to converge the first 48 levels to within 0.1 cm^{-1} and find that the large-basis and small-basis mean absolute level errors agree to within 0.1 cm^{-1}.) Figure 4 shows that the error decreases with both the NN size and M, and that at each M, it reaches a limiting value as the number of nodes is increased. With $M=4$, we attain an error of about 0.5 cm^{-1}. This is extremely important because it validates the motivation of the HDMR expansion. For H_2O_2, we are able to obtain a PES, each of whose terms depends on at most four coordinates, that is about as good as the complicated reference surface. This is significant for two reasons. First, it means that it should be possible to use HDMR-NN to refit surfaces obtained by other means so as to obtain quadrature-friendly surfaces. Second, if $M\approx 4$ is good enough for larger molecules, it implies that HDMR offers a tractable means of obtaining surfaces for larger molecules (for which a full-dimensional fit would not be possible). The difference between the number of coordinates and $M=4$ increases rapidly with the size of the molecule or reacting system. We list the errors achieved with different M in Table II.

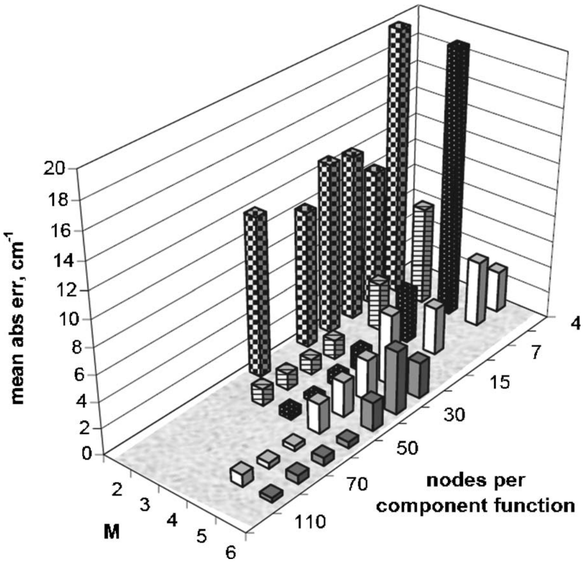

FIG. 4. The mean absolute error in the first $48A_g$ vibrational levels of H_2O_2 calculated on RS-HDMR-NN potential surfaces of different orders and with different numbers of nodes in the partial NNs. The pattern scheme is as in Fig. 3.

B. The number of points determines the HDMR order

It is obvious that the number of fitting points determines the quality of a fitted surface, but it is not clear how to obtain the best possible surface from a limited number of points. One way to do it is to use an HDMR and to keep only mode terms with which a good fit can be obtained. Figure 5 demonstrates that increasing M will improve the fit only if the number of points is sufficiently large. For various fitting set sizes, fits were done with different values of M. For each value of M, we used the best NN size from Tables I and II. According to Fig. 5, for a given M, the mean absolute error decreases to attain a more or less constant value, as the number of fitting points is increased. Increasing the number of fitting points does not improve the fit, because M is fixed. The magnitude of this plateau value is not determined by the number of nodes (for each M, the number of nodes is large enough to converge fits for which the number of fitting points is large). For larger M, the attained constant value is smaller. The ratio of the error at the test points to the error at the fit points decreases as the number of points is increased.

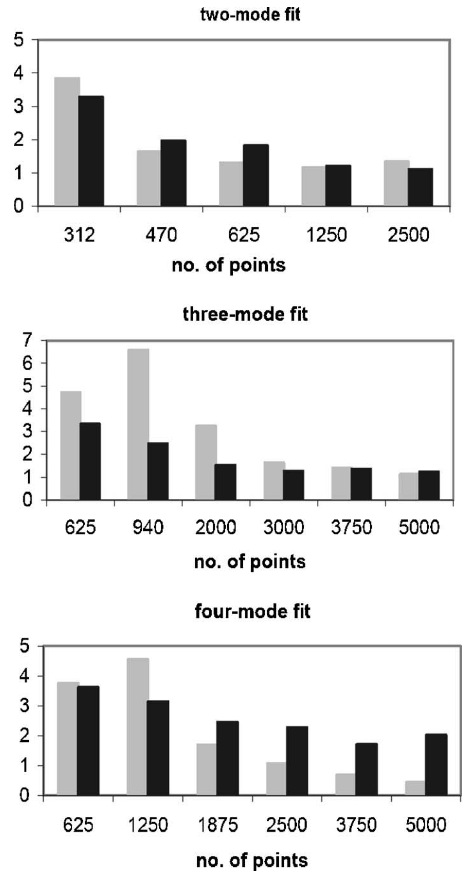

FIG. 5. For different orders of HDMR, the error in vibrational levels of H_2O_2 (gray) and the ratio of the test-set and fitting set errors (black) as a function of the number of (symmetry unique) fitting points.

Once the level error has reached a plateau value and the ratio has reached unity, the best possible fit has been achieved. It is important that this best achievable fit is attained at about 1000 points for $M=2$, at about 3000 points for $M=3$, and at that for $M=4$, 5000 points are probably not quite enough. As M is increased, the number of points required to reach the best possible fit increases. *This means that the maximum value of M, for which one can hope to achieve the best possible fit, is determined by the number of fitting points*. For a given number of fitting points, one should not attempt to use an M that is too large. We know that about 5000 points are enough to determine a fourth-order HDMR for H_2O_2, and that it is quite good. This is good news because it implies that it should be possible to fit potentials with low-order HDMRs and with only thousands of points.

To confirm that the number of points required to get a good fit increases with the order of the mode terms, we have done similar tests on an artificial synthetic potential. We use

$$V = D_1\sum_i y_i^2 + D_2\sum_{i>j} y_i y_j \ldots \equiv V_1 + V_2 + \ldots + V_T, \qquad (9)$$

where $y_i \equiv 1-e^{-\alpha(x_i-x_e)}$. Potentials of this kind have been used for understanding overtone spectra.[107,108] Each of the one-mode component functions is a Morse fit of the potential curve of Pradhan *et al.* for HCl^+.[109] We fit this potential to a function of the x coordinates, not the y. The D_i are chosen so that the variance of each V_i term is $d=3$ times smaller than that of the previous term. The results are not sensitive to d. We present results for a seven-dimensional potential. Seven dimensions are enough to make the fitting difficult without being too time consuming. The potential is sampled according to the inverted potential probability distribution function[98] up to half the dissociation energy and then scaled so that the range of the sampled points is 10 000 cm^{-1}, the range of the random points used to fit H_2O_2. The scaling facilitates comparing fitting errors for this synthetic potential and the H_2O_2 potential. We fit the potential of Eq. (9) with a T-mode term. With enough NN parameters and fitting points, it is possible to fit to arbitrary accuracy. For the results depicted in Fig. 6, the number of nodes is optimized to minimize the test-set error. We have done calculations for different numbers of points. In each case, the quality of the fit is limited by the number of fitting points. From Fig. 6 it is clear that, for a given T, increasing the number of points decreases the fit error. For example, 3000 points are enough to fit the three-mode potential with a three-mode term to within 10 cm^{-1}. This fit takes about 4 h with the same CPU used for the H_2O_2 fits. Figure 6 demonstrates that if higher-order mode terms of the true potential are negligible, then a good fit can be obtained from a small number of points. If the largest non-negligible mode term is the three-mode term, then even for a seven-dimensional (7D) surface, it is possible to get a good fit with only about 3000 points. Many more points would obviously be required to make a general 7D fitted function.

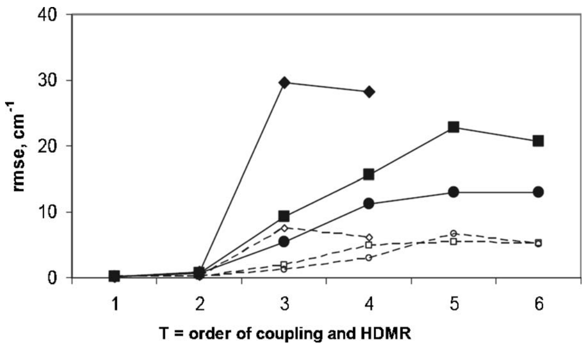

FIG. 6. The test (filled shapes and solid lines) and fitting (open shapes and dashed lines) set errors of a fit to the seven dimensional PES of Eq. (9). The numbers of points used in the fit are diamond - 2000, squares - 3000, and circles - 5000. The connecting lines are to guide the eye.

We have demonstrated that for H_2O_2 and for the synthetic Morse-type potential, more points are required to achieve a good fit if M is larger. This implies that for a given number of points, there is a maximum value of M for which a good fit can be achieved, i.e., if a higher-order fit is attempted, the coupling terms of order higher than M cannot be extracted. By considering uncertainties of the parameters, one can show that it is also true that the NN parameters of a mode term are uncertain unless M is small enough. The uncertainty of the kth parameter is V_{kk}, where V is the variance-covariance matrix.[80,110] The diamonds in Fig. 7 are average relative uncertainties for all the parameters of a fit of the kind described in Sec. III A. We define the average relative uncer-

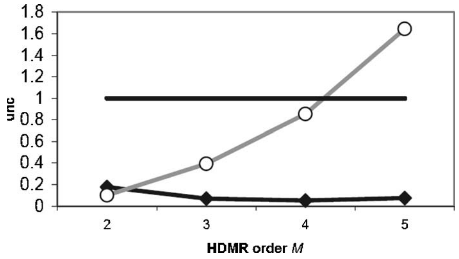

FIG. 7. The uncertainties of H_2O_2 NN parameters. Black diamonds: uncertainties of all parameters of an Mth order HDMR. Open circles: uncertainties of parameters of the $(M+1)$th mode term. The connecting lines are to guide the eye. The black line at unc=1 is a "certainty limit" standard.

tainty as $unc=3\times \text{mean}|V_{kk}\times \text{pinv}(\boldsymbol{X})|$, where $\boldsymbol{X}$ is a column vector of all NN parameters, and $\text{pinv}(\boldsymbol{X})=\boldsymbol{X}/(\boldsymbol{X}^T\boldsymbol{X})$ is the pseudoinverse. If we did not use the pseudoinverse, small NN parameters would make *unc* artificially large. The pseudoinverse effectively circumvents this problem and enables us to deal with the large range that NN parameters have. The open circles are average relative uncertainties of the NN parameters for the $(M+1)$th mode term when we fit including the M-mode term and the $(M+1)$-mode term. In this case, the sum we use to compute *unc* is over only the NN parameters for the $(M+1)$th mode term. Note that as M is increased, the uncertainty of the NN parameters for the $(M+1)$th mode term increases. This confirms that, for a fixed number of points, as M is increased, it becomes increasingly harder to fit the NN parameters of the M-mode term. For all fits of Fig. 7, the number of nodes is the optimal value of Tables I and II.

IV. CONCLUSION

The HDMR idea is very enticing for two reasons. First, it seems clear that it should be possible to obtain a good surface by including only low-order mode terms. For example, even retaining only the one-mode term gives a surface that has qualitatively the right shape (and accounts for all of the anharmonicity in the chosen coordinates). It is obviously easier to fit lower-dimensional mode terms than a full-dimensional PES. Second, due to the fact that low-dimensional quadratures are much less costly than high-dimensional quadratures, an HDMR facilitates quantum dynamics calculations. In this paper, we propose a new scheme for defining and computing HDMR component functions and demonstrate that, using the new component functions, it is possible to fit potentials well. Our H_2O_2 surface, fitted with a fourth-order HDMR, has a rmse, computed on a set of points not used to do the fitting, of less than 10 cm^{-1} and levels computed on the surface are good to within about 1 cm^{-1}. There is nothing special about H_2O_2 or the coordinates we use to do the fit (bond coordinates), and we therefore expect that a low-order HDMR should be sufficient for many potentials. This is very encouraging.

The fourth-order HDMR H_2O_2 fit is made possible by our NN scheme for defining and computing component functions. Previous users of the general HDMR idea have used either cut HDMR, which does not minimize error in all space, or a RS-HDMR with component functions represented as sums of basis functions with coefficients determined by doing large dimensional integrals by Monte Carlo. A RS-HDMR uses component functions that minimize error in all space. Our NN scheme for making component functions is simple and general and can be used to obtain high-order HDMR mode terms. Using the RS-HDMR-NN procedure we propose, all the component functions are constructed from one set of *ab initio* points. The *ab initio* points may be distributed according to any probability distribution function (we use one that puts more points at the bottom of the potential). The component functions of a given mode term are determined simultaneously and minimize error at each order. Rather than imposing constraints to make it possible to determine component functions independently,[1,8–10] we use NN fits. The NNs represent functions that are solutions of coupled integral equations obtained by imposing constraints weaker than those of Refs. 3 and 8–10. We therefore expect that the HDMR built from the component functions determined with NNs should provide a better fit.

The principle conceptual difference between the established approach for computing component functions and the one we propose is the use of a fitting method. In the approach of Refs. 1 and 8–10, equations for the component functions are found by functional differentiation. It is possible to derive equations for the component functions we use (see the Appendix), but rather than solving them, we use a sequential fitting method. We use fits to find the component functions, because we choose to simultaneously fit all the component functions of a mode term. It might be possible to use different fitting methods, but we favor NNs because of their generality and robustness. Note that, due to the use of fitting, we do not need equations for the component functions. Because we do not determine component functions independently, but by fitting, we can lump together mode terms, i.e., skip lower-order mode terms and do a NN fit only for the largest order that is retained. This improves the quality of the final representation and reduces the cost of the fits. The cost reduction will be especially important for large molecules or reacting systems, because the number of component functions increases rapidly with the number of atoms. The cost of the fits can be further reduced by exploiting the fact that NN parameters of different component functions are weakly correlated (see Sec. II C 2) and can therefore be fitted in groups.

No matter how clever the fitting algorithm is, it will always be true that computing the *ab initio* points required to make a good fit will be costly and that a better surface can be obtained from more points. It is therefore natural to ask: given a fixed number of points how does one extract the best possible surface? One of the important results of this paper is the idea that the number of *ab initio* points determines the maximum HDMR order one should use. If the maximum order is M and one attempts to use an $M+1$ order HDMR, one finds that the additional parameters are poorly determined and the fit quality is not improved. It is clearly best to use a representation whose parameters are well determined.

A great advantage of the RS-HDMR-NN approach is its generality. To use it, one does not have to choose special coordinates. The HDMR can be built in a black-box, molecule-independent way. We expect RS-HDMR-NN to be a very effective potential fitting tool.

ACKNOWLEDGMENTS

The authors thank Xiao-Gang Wang for the code for the vibrational calculations and Richard Dawes for useful discussions. This research was funded by the Natural Sciences and Engineering Research Council of Canada.

APPENDIX: EQUATIONS FOR COMPONENT FUNCTIONS

Here, we derive equations that could, in principle, be solved to obtain the component functions that minimize Eq. (2) mode term by mode term. We then compare these equations to equations of Refs. 3, 9, and 10 for component functions obtained independently. Rather than solving the equations derived here, we represent the component functions with NNs and choose the NN parameters to minimize, mode term by mode term, the sum of squared deviations. The resulting NN component functions will approximate the solutions of equations derived here. The equations in this appendix are important, because comparing them with the equations of Rabitz and co-workers enables one to understand better how the constraints that make sequential and simultaneous mode-term minimization equivalent compare to the constraints that make independent and simultaneous component function minimization equivalent.

To derive equations for the component functions and the constraints one must impose to ensure that simultaneous and sequential minimization are equivalent, we use the following procedure. We first choose f_0 to minimize

$$I_0 = \int_{K^D} [f(\mathbf{x}) - f_0]^2 w(\mathbf{x}) d\mathbf{x}. \tag{A1}$$

We then minimize

$$I_1 = \int_{K^D} \left[f(\mathbf{x}) - f_0 - \sum_{i=1}^{D} f_i(x_i) \right]^2 w(\mathbf{x}) d\mathbf{x}, \tag{A2}$$

with respect to f_0 and f_i and determine the constraints necessary to make f_0 obtained from Eq. (A1) identical to f_0 obtained from (A2). We then minimize

$$I_2 = \int_{K^D} \left[f(\mathbf{x}) - f_0 - \sum_{i=1}^{D} f_i(x_i) - \sum_{\substack{i,j=1 \\ i>j}}^{D} f_{ij}(x_i,x_j) \right]^2 w(\mathbf{x}) d\mathbf{x}, \tag{A3}$$

with respect to f_0, f_i, and f_{jk} and determine the constraints necessary to make f_0 obtained from (A2) identical to f_0 obtained from (A3) and f_i obtained from (A2) identical to f_i obtained from (A3). Finally, we generalize to obtain equations for the constraints. The probability density function (pdf) $w(\mathbf{x})$ is arbitrary, and in our calculations we exploit this generality by choosing it to improve the quality of the fit in what we consider important regions of the potential. The derivation is similar to that in Ref. 9, except that we use a nonfactorizable pdf $w(\mathbf{x})$ (cf. Ref. 10). Different equations for a nonfactorizable pdf are derived in Ref. 3 by expanding component functions in terms of polynomials orthonormal with respect to the weight function.

f_0 is easily found by differentiating Eq. (A1) with respect to f_0,

$$f_0 = \int_{K^D} f(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}. \tag{A4}$$

Next, we differentiate Eq. (A2) with respect to f_0 and f_i. We find

$$\frac{\partial I_1}{\partial f_0} = -2 \int_{K^D} \left[f(\mathbf{x}) - f_0 - \sum_{j=1}^{D} f_j(x_j) \right] w(\mathbf{x}) d\mathbf{x} = 0, \tag{A5}$$

$$f_0 = \int_{K^D} f(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} - \int_{K^D} \left(\sum_{i=1}^{D} f_i(x_i) \right) w(\mathbf{x}) d\mathbf{x}, \tag{A6}$$

and

$$\frac{\delta I_1}{\delta f_i(x_i)} = -2 \int_{K^D} \left[f(\mathbf{x}') - f_0 - \sum_{j=1}^{D} f_j(x_j') \right] \times \delta(x_i - x_i') w(\mathbf{x}') d\mathbf{x}' = 0, \tag{A7}$$

$$\int_{K^{D-1}} f(\mathbf{x}|x_i) w(\mathbf{x}|x_i) d\mathbf{x}^i = (f_i(x_i) + f_0) \int_{K^{D-1}} w(\mathbf{x}|x_i) d\mathbf{x}^i + \sum_{i \neq j=1}^{D} \int_{K^{D-1}} f_j(x_j) w(\mathbf{x}|x_i) d\mathbf{x}^i. \tag{A8}$$

To ensure that f_0 in (A6) and (A4) are identical, we must impose the constraint

$$\int_{K^D} \left(\sum_{i=1}^{D} f_i(x_i) \right) w(\mathbf{x}) d\mathbf{x} = 0. \tag{A9}$$

The $f_i(x_i)$ component functions are therefore solutions of a system of integral equations (A8). We have again used the notation $d\boldsymbol{x}^i \equiv \Pi_{i\neq j=1}^{D} dx_j$ and used $f(\boldsymbol{x}|x_i)$ to mean that the variable x_i is special. Written out explicitly, the RS-HDMR one-mode component functions are

$$f_i(x_i) = \frac{\displaystyle\int_{K^{D-1}} f(\boldsymbol{x}|x_i)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i - \sum_{i\neq j=1}^{D}\int_{K^{D-1}} f_j(x_j)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i}{\displaystyle\int_{K^{D-1}} w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i} - f_0. \tag{A10}$$

Next we differentiate Eq. (A3) with respect to f_0, f_i, and f_{jk}. We find

$$\frac{\partial I_2}{\partial f_0} = -2\int_{K^D}\left[f(\boldsymbol{x}) - f_0 - \sum_{j=1}^{D} f_j(x_j) - \sum_{\substack{i,j=1\\ i>j}}^{D} f_{kj}(x_k,x_j)\right]w(\boldsymbol{x})d\boldsymbol{x} = 0, \tag{A11}$$

$$f_0 = \int_{K^D} f(\boldsymbol{x})w(\boldsymbol{x})d\boldsymbol{x} - \int_{K^D}\left(\sum_{i=1}^{D} f_i(x_i)\right)w(\boldsymbol{x})d\boldsymbol{x} - \int_{K^D}\left(\sum_{\substack{k,l=1\\ k>l}}^{D} f_{kl}(x_k,x_l)\right)w(\boldsymbol{x})d\boldsymbol{x}, \tag{A12}$$

and

$$\frac{\delta I_2}{\delta f_i(x_i)} = -2\int_{K^D}\left[f(\boldsymbol{x}') - f_0 - \sum_{j=1}^{D} f_j(x_j') - \sum_{\substack{k,j=1\\ k>j}}^{D} f_{kj}(x_k',x_j')\right]^2 w(\boldsymbol{x}')\,\delta(x_i - x_i')d\boldsymbol{x}' = 0 \tag{A13}$$

and

$$\frac{\delta I_2}{\delta f_{ij}(x_i,x_j)} = -2\int_{K^D}\left[f(\boldsymbol{x}') - f_0 - \sum_{k=1}^{D} f_k(x_k') - \sum_{\substack{k,l=1\\ k>l}}^{D} f_{kl}(x_k',x_l')\right]^2 w(\boldsymbol{x}')\,\delta(x_i - x_i')\,\delta(x_j - x_j')d\boldsymbol{x}' = 0. \tag{A14}$$

The second term of the right-hand side of (A12) is zero by virtue of the constraint (A9). To ensure that f_0 in (A4) and (A12) are identical, we must impose the constraint

$$\int_{K^D}\left(\sum_{\substack{k,l=1\\ k>l}}^{D} f_{kl}(x_k,x_l)\right)w(\boldsymbol{x})d\boldsymbol{x} = 0. \tag{A15}$$

To ensure that f_i derived from (A7) and (A13) are identical, we must impose the constraint

$$\sum_{\substack{k,j=1\\ k>j}}^{D}\int_{K^{D-1}} f_{kj}(x_k,x_j)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i = 0. \tag{A16}$$

(A14) is equivalent to

$$-\int_{K^{D-2}} f(\boldsymbol{x}|x_i,x_j)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} + (f_0 + f_i(x_i) + f_j(x_j) + f_{ij}(x_i,x_j))\int_{K^{D-2}} w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} + \sum_{\substack{k=1\\ k\neq i,j}}^{D}\int_{K^{D-2}} f_k(x_k)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij}$$
$$+\sum_{\substack{k>l=1\\ k,l\neq i,j}}^{D}\int_{K^{D-2}} f_{kl}(x_k,x_l)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} = 0, \tag{A17}$$

from which one obtains

$$f_0+f_i(x_i)+f_j(x_j)+f_{ij}(x_i,x_j) = \frac{\left(\int_{K^{D-2}} f(\boldsymbol{x}|x_i,x_j)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - \sum_{\substack{k=1\\ i\neq k\neq j}}^{D}\int_{K^{D-2}} f_k(x_k)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - \sum_{\substack{k>l=1\\ k,l\neq i,j}}^{D}\int_{K^{D-2}} f_{kl}(x_k,x_l)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij}\right)}{\int_{K^{D-2}} w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij}}. \tag{A18}$$

To summarize, we see that consecutive determination of the component functions is equivalent to imposing constraints (A9) and (A16). Note that (A15) follows from (A16). To generalize these results, we now minimize

$$I_3 = \int_{K^D}\left[f(\boldsymbol{x}) - f_0 - \sum_{i=1}^{D} f_i(x_i) - \sum_{\substack{i,j=1\\ i>j}}^{D} f_{ij}(x_i,x_j) - \sum_{\substack{k,l,m=1\\ k>l>m}}^{D} f_{klm}(x_k,x_l,x_m)\right]^2 w(\boldsymbol{x})d\boldsymbol{x}. \tag{A19}$$

For f_0, we set

$$\frac{\partial I_3}{\partial f_0} = -2\int_{K^D}[\cdots]w(\boldsymbol{x})d\boldsymbol{x} = 0 \tag{A20}$$

and find

$$f_0 = \int_{K^D} f(\boldsymbol{x})w(\boldsymbol{x})d\boldsymbol{x} - \int_{K^D}\left(\sum_{i=1}^{D} f_i(x_i)\right)w(\boldsymbol{x})d\boldsymbol{x} - \int_{K^D}\left(\sum_{\substack{k,l=1\\ k>l}}^{D} f_{kl}(x_k,x_l)\right)w(\boldsymbol{x})d\boldsymbol{x} - \int_{K^D}\left(\sum_{\substack{k,l,m=1\\ k>l>m}}^{D} f_{klm}(x_k,x_l,x_m)\right)w(\boldsymbol{x})d\boldsymbol{x}. \tag{A21}$$

To ensure that f_0 in (A21) equals f_0 in (A4), we must impose

$$\int_{K^D}\left(\sum_{\substack{k,l,m=1\\ k>l>m}}^{D} f_{klm}(x_k,x_l,x_m)\right)w(\boldsymbol{x})d\boldsymbol{x} = 0. \tag{A22}$$

In general, the required constraint is

$$\int_{K^D}\left(\sum_{i_1\ldots i_l} f_{i_1\ldots i_l}(x_{i_1},\ldots,x_{i_l})\right)w(\boldsymbol{x})d\boldsymbol{x} = 0. \tag{A23}$$

The univariate functions satisfy

$$\frac{\delta I_3}{\delta f_i(x_i)} = -2\int_{K^D}[\cdots]'\,\delta(x_i - x_i')w(\boldsymbol{x}')d\boldsymbol{x}' = 0, \tag{A24}$$

$$\int_{K^{D-1}} f(\boldsymbol{x})w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i - (f_0 + f_i(x_i))\int_{K^{D-1}} w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i - \int_{K^{D-1}}\sum_{i\neq j=1}^{D} f_j(x_j)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i - \int_{K^{D-1}}\sum_{\substack{k,l=1\\ k>l}}^{D} f_{kl}(x_k,x_l)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i - \int_{K^{D-1}}\sum_{\substack{k,l,m=1\\ k>l>m}}^{D} f_{klm}(x_k,x_l,x_m)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i = 0. \tag{A25}$$

To ensure that f_i in (A25) equal those in (A10), we must impose

$$\int_{K^{D-1}}\sum_{\substack{k,l,m=1\\ k>l>m}}^{D} f_{klm}(x_k,x_l,x_m)w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i = 0, \tag{A26}$$

which generalizes as

$$\int_{K^{D-1}}\sum_{i_1\ldots i_l} f_{i_1\ldots i_l}(x_{i_1},\ldots,x_{i_l})w(\boldsymbol{x}|x_i)d\boldsymbol{x}^i = 0. \tag{A27}$$

Note that (A23) follows from (A27). For the bivariate functions,

$$\frac{\delta I_3}{\delta f_{ij}(x_i,x_j)} = -2\int_{K^D}[\cdots]'\,\delta(x_i - x_i')\,\delta(x_j - x_j')w(\boldsymbol{x}')d\boldsymbol{x}', \tag{A28}$$

$$\int_{K^{D-2}} f(\boldsymbol{x})w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - (f_0 + f_i(x_i) + f_j(x_j) + f_{ij}(x_i,x_j))\int_{K^{D-2}} w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - \int_{K^{D-2}}\sum_{k\neq i,j}^{D} f_k(x_k)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - \sum_{\substack{k>l=1\\ k,l\neq i,j}}^{D}\int_{K^{D-2}} f_{kl}(x_k,x_l)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} - \int_{K^{D-2}}\sum_{klm} f_{klm}(x_k,x_l,x_m)w(\boldsymbol{x}|x_i,x_j)d\boldsymbol{x}^{ij} = 0. \tag{A29}$$

To ensure that f_{kl} in (A29) equal those in (A18), we impose

$$\int_{K^{D-2}} \sum_{klm} f_{klm}(x_k,x_l,x_m) w(\boldsymbol{x}|x_i,x_j) d\boldsymbol{x}^{ij} = 0, \tag{A30}$$

which generalizes as

$$\int_{K^{D-l+1}} \sum_{i_1\ldots i_l} f_{i_1\ldots i_l}(x_{i_1},\ldots,x_{i_l}) w(\boldsymbol{x}|x_{k_1},\ldots,x_{k_{l-1}}) d\boldsymbol{x}^{k_1\ldots k_{l-1}} = 0, \tag{A31}$$

of which (A23) and (A27), are particular cases. Therefore, sequential minimization of the error functional mode term by mode term is equivalent to its simultaneous minimization under the constraint (A31). This constraint is milder than that of Refs. 3, 9, and 10, as it is imposed on a sum and not on each term in the sum, and we expect therefore a better minimization of the error functional.

If severer constraints are imposed, the HDMR component functions used in previous papers[3,9,10] are obtained from those derived from (A8) and (A18). If $w(\boldsymbol{x})=\Pi_{i=1}^{D} w_i(x_i)$, where the partial pdfs $w_i(x_i)$ are normalized, then (A8) becomes

$$\begin{aligned} w_i(x_i)\int_{K^{D-1}} f(\boldsymbol{x}|x_i) \prod_{i\neq j=1}^{D} w_j(x_j)dx_j &= w_i(x_i)(f_i(x_i)+f_0) \\ &\quad + w_i(x_i)\sum_{i\neq j=1}^{D} \int_{K^{D-1}} f_j(x_j) \prod_{i\neq k=1}^{D} w_k(x_k)dx_k, \end{aligned} \tag{A32}$$

$$\begin{aligned} \int_{K^{D-1}} f(\boldsymbol{x}|x_i) \prod_{i\neq j=1}^{D} w_j(x_j)dx_j &= f_0 + f_i(x_i) + \sum_{i\neq j=1}^{D} \int_{K^1} f_j(x_j)w_j(x_j)dx_j. \end{aligned} \tag{A33}$$

If we impose the condition [equivalent to Eq. (6) of Ref. 10 with $l=1$] $\int_{K^1} f_j(x_j)w_j(x_j)dx_j=0$, we obtain simply

$$f_i(x_i) = \int_{K^{D-1}} f(\boldsymbol{x}|x_i) \prod_{i\neq j=1}^{D} w_j(x_j)dx_j - f_0, \tag{A34}$$

which is the result of Ref. 10. If w is not factorizable and one imposes the condition

$$\sum_{i\neq j=1}^{D} \int_{K^{D-1}} f_j(x_j) w(\boldsymbol{x}|x_i) d\boldsymbol{x}^i = 0, \quad \forall\, i, \tag{A35}$$

then from (A18) we obtain

$$f_i(x_i) = \frac{\int_{K^{D-1}} f(\boldsymbol{x}|x_i) w(\boldsymbol{x}|x_i) d\boldsymbol{x}^i}{\int_{K^{D-1}} w(\boldsymbol{x}|x_i) d\boldsymbol{x}^i} - f_0, \tag{A36}$$

which is the result of Ref. 3. To avoid coupled equations for the higher-order component functions, more complicated constraints are necessary. If constraints are imposed to zero out the last two terms in (A17), one obtains a formula for bivariate component functions

$$\begin{aligned} f_0 + f_i(x_i) + f_j(x_j) + f_{ij}(x_i,x_j) &= \frac{\int_{K^{D-2}} f(\boldsymbol{x}|x_i,x_j) w(\boldsymbol{x}|x_i,x_j) d\boldsymbol{x}^{ij}}{\int_{K^{D-2}} w(\boldsymbol{x}|x_i,x_j) d\boldsymbol{x}^{ij}}, \end{aligned} \tag{A37}$$

which is equivalent to Eq. (5) of Ref. 3.

[1] G. Li, C. Rosenthal, and H. Rabitz, J. Phys. Chem. A **105**, 7765 (2001).
[2] O. F. Alis and H. Rabitz, J. Math. Chem. **29**, 127 (2001).
[3] G. Li, J. Hu, S.-W. Wang, P. G. Georgopoulos, J. Schoendorf, and H. Rabitz, J. Phys. Chem. A **110**, 2474 (2006).
[4] G. Li, S.-W. Wang, C. Rosenthal, and H. Rabitz, J. Math. Chem. **30**, 1 (2001).
[5] G. Li, S.-W. Wang, and H. Rabitz, J. Phys. Chem. A **106**, 8721 (2002).
[6] G. Li, M. Artamonov, H. Rabitz, S.-W. Wang, P. G. Georgopoulos, and M. Demiralp, J. Comput. Chem. **24**, 647 (2003).
[7] G. Li, J. Schoendorf, T.-S. Ho, and H. Rabitz, J. Comput. Chem. **25**, 1149 (2004).
[8] H. Rabitz, O. F. Alis, J. Shorter, and K. Shim, Comput. Phys. Commun. **117**, 11 (1999).
[9] H. Rabitz and O. F. Alis, J. Math. Chem. **25**, 197 (1999).
[10] S.-W. Wang, P. G. Georgopoulos, G. Li, and H. Rabitz, J. Phys. Chem. A **107**, 4707 (2003).
[11] S. Carter, J. M. Bowman, and L. B. Harding, Spectrochim. Acta, Part A **53**, 1179 (1997).
[12] S. Carter, S. J. Culik, and J. M. Bowman, J. Chem. Phys. **107**, 10458 (1997).
[13] S. Carter and J. M. Bowman, J. Phys. Chem. A **104**, 2355 (2000).
[14] J. N. Murrell, S. Carter, S. Frantos, P. Huxley, and A. J. C. Varandas, *Molecular Potential Energy Functions* (Wiley, Toronto 1984).
[15] S. Carter and N. C. Handy, Chem. Phys. Lett. **352**, 1 (2002).
[16] B. G. Sumpter, C. Getino, and D. W. Noid, Annu. Rev. Phys. Chem. **45**, 439 (1994).
[17] B. Widrow and M. A. Lehr, Proc. IEEE **78**, 1415 (1990).
[18] J. Zupan and J. Gasteiger, Anal. Chim. Acta **248**, 1 (1991).
[19] C. J. Stone, Ann. Stat. **13**, 689 (1985).
[20] G. Rauhut, J. Chem. Phys. **121**, 9313 (2004).
[21] G. C. Schatz and H. Elgersma, Chem. Phys. Lett. **73**, 21 (1980).
[22] J. N. Murrell and S. Carter, J. Phys. Chem. **88**, 4887 (1984).
[23] G. C. Schatz, M. S. Fitzcharles, and L. B. Harding, Faraday Discuss. Chem. Soc. **84**, 359 (1987).
[24] X. Liu and J. N. Murrell, J. Chem. Soc., Faraday Trans. **87**, 435 (1991).
[25] J. N. Murrell, S. Carter, I. M. Mills, and M. F. Guest, Mol. Phys. **42**, 605 (1981).
[26] S. Carter, I. M. Mills, J. N. Murrell, and A. J. C. Varandas, Mol. Phys. **45**, 1053 (1982).
[27] A. Jaeckle and H. D. Meyer, J. Chem. Phys. **104**, 7974 (1996).
[28] A. Jaeckle and H. D. Meyer, J. Chem. Phys. **109**, 3772 (1998).
[29] A. Jaeckle and H. D. Meyer, J. Chem. Phys. **102**, 5605 (1995).
[30] S. Carter, J. M. Bowman, and N. C. Handy, Theor. Chem. Acc. **100**, 191 (1998).
[31] I. M. Sobol, Sov. Math. Dokl. **8**, 810 (1967).
[32] I. M. Sobol, Sov. Math. Dokl. **1**, 655 (1960).
[33] J. M. Geremia, E. Weiss, and H. Rabitz, Chem. Phys. **267**, 209 (2001).
[34] J. M. Geremia and H. Rabitz, J. Chem. Phys. **115**, 8899 (2001).
[35] J. M. Geremia, H. Rabitz, and C. Rosenthal, J. Chem. Phys. **114**, 9325 (2001).
[36] N. Shenvi, J. M. Geremia, and H. Rabitz, J. Chem. Phys. **120**, 9942 (2004).
[37] M. Y. Hayes, B. Li, and H. Rabitz, J. Phys. Chem. A **110**, 264 (2006).
[38] J. M. Geremia and H. Rabitz, Phys. Rev. A **64**, 022710 (2001).
[39] T.-S. Ho and H. Rabitz, J. Chem. Phys. **119**, 6433 (2003).
[40] T. Hollebeek, T.-S. Ho, and H. Rabitz, Annu. Rev. Phys. Chem. **50**, 537 (1999).
[41] T.-S. Ho and H. Rabitz, J. Chem. Phys. **104**, 2584 (1996).
[42] A. Chakraborty, D. G. Truhlar, J. M. Bowman, and S. Carter, J. Chem. Phys. **121**, 2071 (2004).
[43] J. Wu, X. Huang, S. Carter, and J. M. Bowman, Chem. Phys. Lett. **426**, 285 (2006).
[44] I. M. Sobol, Reliab. Eng. Syst. Saf. **79**, 187 (2003).
[45] L. M. Raff, M. Malshe, M. T. Hagan, D. I. Doughan, M. G. Rockey, and R. Komanduri, J. Chem. Phys. **122**, 084104 (2005).
[46] P. A. Agrawal, L. M. Raff, M. T. Hagan, and R. Komanduri, J. Chem.

Phys. **124**, 134306 (2006).

[47] D. I. Doughan, L. M. Raff, M. G. Rockey, M. T. Hagan, P. A. Agrawal, and R. Komanduri, J. Chem. Phys. **124**, 054321 (2006).

[48] S. Manzhos, X.-G. Wang, R. Dawes, and T. Carrington, Jr., J. Phys. Chem. A **110**, 5295 (2006).

[49] S. Lorez, A. Gross, and M. Scheffler, Chem. Phys. Lett. **395**, 210 (2004).

[50] F. V. Prudente and J. J. Soares Neto, Chem. Phys. Lett. **287**, 585 (1998).

[51] F. V. Prudente, P. H. Acioli, and J. J. Soares Neto, J. Chem. Phys. **109**, 8801 (1998).

[52] H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson, J. Phys. Chem. A **102**, 4596 (1998).

[53] T. S. Blank and S. D. Brown, J. Chemom. **8**, 391 (1994).

[54] D. F. R. Brown, M. N. Gibbs, and D. C. Clary, J. Chem. Phys. **105**, 7597 (1996).

[55] T. S. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, J. Chem. Phys. **103**, 4129 (1995).

[56] B. G. Sumpter and D. W. Noid, Chem. Phys. Lett. **192**, 455 (1992).

[57] M. H. Hassoun, *Fundamentals of Artificial Neural Networks* (MIT, Cambridge, MA, 1995).

[58] J. B. Witkoskie and D. J. Doren, J. Chem. Theory Comput. **1**, 14 (2005).

[59] K. Funahashi, Neural Networks **2**, 183 (1989).

[60] V. Maiorov and A. Pinkus, Neurocomputing **25**, 81 (1999).

[61] K. Hornik, M. Stinchcombe, and H. White, Neural Networks **3**, 551 (1990).

[62] K. Hornik, Neural Networks **2**, 359 (1989).

[63] K. Hornik, Neural Networks **4**, 251 (1991).

[64] F. Scarselli and A. C. Tsoi, Neural Networks **11**, 15 (1998).

[65] M. A. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations* (Cambridge University Press, Cambridge, 1999).

[66] A. N. Kolmogorov, Dokl. Akad. Nauk SSSR **114**, 953 (1957).

[67] D. A. Sprecher, Proc. Am. Math. Soc. **16**, 200 (1965).

[68] D. A. Sprecher, Trans. Am. Math. Soc. **115**, 340 (1965).

[69] V. Kurkova, Neural Networks **5**, 501 (1992).

[70] A. Barron, IEEE Trans. Inf. Theory **39**, 930 (1993).

[71] C. Chui and X. Li, J. Approx. Theory **70**, 131 (1992).

[72] H. Mhaskar and C. Miccheli, Adv. Appl. Math. **13**, 350 (1992).

[73] H. Mhaskar and C. Miccheli, IBM J. Res. Dev. **38**, 277 (1994).

[74] H. Demuth and M. Beale, *Neural Network Toolbox User's Guide* (The MathWorks, Inc., Natick, MA, 2004).

[75] MATLAB 7.0 Release 14, (The MathWorks, Inc., Natick, MA, 2004).

[76] A. Kawano, Y. Guo, D. L. Thompson, A. F. Wagner, and M. Minkoff, J. Chem. Phys. **120**, 6414 (2004).

[77] S. Ergezinger and E. Thomsen IEEE Trans. Neural Netw. **6**, 31 (1995).

[78] G.-J. Wang and C.-C. Chen IEEE Trans. Neural Netw. **7**, 768 (1996).

[79] Y. Li, D. Zhang, and K. Wang Neural Processing Lett. **23**, 229 (2006).

[80] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1992).

[81] B. Kuhn, T. R. Rizzo, D. Luckhaus, M. Quack, and M. A. Suhm, J. Chem. Phys. **111**, 2565 (1999).

[82] G. G. Maisuradze, A. Kawano, D. L. Thompson, A. F. Wagner, and M. Minkoff, J. Chem. Phys. **121**, 10329 (2004).

[83] G. G. Maisuradze, D. L. Thompson, A. F. Wagner, and M. Minkoff, J. Chem. Phys. **119**, 10002 (2003).

[84] G. G. Maisuradze and D. L. Thompson, J. Phys. Chem. A **107**, 7118 (2003).

[85] Y. Guo, A. Kawano, D. L. Thompson, A. F. Wagner, and M. Minkoff, J. Chem. Phys. **121**, 5091 (2004).

[86] K. C. Thompson and M. A. Collins, J. Chem. Soc., Faraday Trans. **93**, 871 (1997).

[87] M. A. Collins and D. H. Zhang, J. Chem. Phys. **111**, 9924 (1999).

[88] D. L. Crittenden, K. C. Thompson, M. Chebib, and M. J. T. Jordan, J. Chem. Phys. **121**, 9844 (2004).

[89] D. K. Hoffman, A. Frishman, and D. J. Kouri Chem. Phys. Lett. **262**, 393 (1996).

[90] V. Szalay, J. Chem. Phys. **119**, 8804 (1999).

[91] R. Chen, G. Ma, and H. Guo, J. Chem. Phys. **114**, 4763 (2001).

[92] C. Camy-Peyret, J.-M. Flaud, J. W. C. Johns, and M. Noel, J. Mol. Spectrosc. **155**, 84 (1992).

[93] J.-M. Flaud, C. Camy-Peyret, J. W. C. Johns, and B. Carli, J. Chem. Phys. **91**, 1504 (1989).

[94] A. Perrin, A. Valentin, J.-M. Flaud, C. Camy-Peyret, L. Schriver, A. Schriver, and P. Arcas, J. Mol. Spectrosc. **171**, 358 (1995).

[95] J. J. Hillman, D. E. Jennings, W. B. Olson, and A. Goldman, J. Mol. Spectrosc. **117**, 46 (1986).

[96] P. Helminger, W. C. Bowman, and F. C. DeLucia, J. Mol. Spectrosc. **85**, 120 (1981).

[97] W. B. Cook, R. H. Hunt, W. N. Shelton, and F. A. Flaherty, J. Mol. Spectrosc. **171**, 91 (1995).

[98] S. Garashchuk and J. C. Light, J. Chem. Phys. **114**, 3929 (2001).

[99] R. A. Friesner, Proc. Natl. Acad. Sci. U.S.A. **102**, 6648 (2005).

[100] M. L. Abrams and C. D. Sherrill, J. Phys. Chem. A **107**, 5611 (2003).

[101] A. Dutta and C. D. Sherrill, J. Chem. Phys. **118**, 1610 (2003).

[102] 2. M. J. Bramley and T. Carrington, Jr., J. Chem. Phys. **99**, 8519 (1993).

[103] M. J. Bramley, J. W. Tromp, T. Carrington, Jr., and G. C. Corey, J. Chem. Phys. **100**, 6175 (1994).

[104] C. Leforestier, J. Chem. Phys. **101**, 7357 (1994).

[105] H. Wei and T. Carrington, Jr., J. Chem. Phys. **97**, 3029 (1992).

[106] J. Echave and D. C. Clary, Chem. Phys. Lett. **190**, 225 (1992).

[107] T. Carrington, Jr., J. Chem. Phys. **86**, 2207 (1987).

[108] M. Quack, Annu. Rev. Phys. Chem. **41**, 839 (1990).

[109] A. D. Pradhan, K. P. Kirby, and A. Dalgarno, J. Chem. Phys. **95**, 9009 (1991).

[110] R. J. LeRoy, Chemical Physics Research Report No. CP-628 (University of Waterloo, Ontario, Canada, 1997).