Inversion of Simulated Positron Annihilation Lifetime Spectrum Using a Neural Network

V. C. Viterbo and J. P. Braga*

Departamento de Química - ICEx, Universidade Federal de Minas Gerais, 31270-901 Belo Horizonte, MG, Brasil

A. P. Braga and M. B. de Almeida

Departamento de Engenharia Eletrônica - EEUFMG, Universidade Federal de Minas Gerais, 31270-901, Belo Horizonte, MG, Brasil

Received June 30, 2000

Inversion of positron annihilation lifetime spectroscopy, based on a neural network Hopfield model, is presented in this paper. From a previous reported density function for lysozyme in water a simulated spectrum, without the superposition of statistical fluctuation and spectrometer resolution effects, was generated. These results were taken as the exact results from which the neural network was trained. The precision of the inverted density function was analyzed taking into account the number of neurons and the learning time of the neural network. A fair agreement was obtained when comparing the neural network results with the exact results. For example, the maximum of the density function, with a precision of 0.4% for the percentual relative error, was obtained for 64 neurons.

1. INTRODUCTION

An ill-posed problem is a problem in which one of the three conditions, existence, unicity, or continuity, is not satisfied. This class of problems, although very common in science, has received attention after the pioneering work of Tikhonov, when dealing with inversion in the area of geophysics.[1] A complete list of occurrence of ill posed problems in science can be found in refs 2 and 3.

Inversion of positron annihilation lifetime spectrum[4] is an ill posed problem and this can be seen by considering a simple model. The number of positrons that survive at a given time, $c(t)$, is given by the Laplace transform of the density function, $\lambda\alpha(\lambda)$, where $\alpha(\lambda)$ is the annihilation rate probability and λ the inverse of the half-life of the species under consideration. A model for this problem can be written as a Fredholm integral equation of first order

$$\int_a^b K(t,\lambda) f(\lambda) d\lambda = c(t) \qquad (1)$$

Within a representation the above equation can also be written as $\mathbf{Kf} = \mathbf{c}$. For this kind of problem $\mathbf{K}$ is an ill conditioned matrix, with singular values going very quickly to zero. Consequently, its inverse, $\mathbf{K}^{-1}$, will have very large values compared with the original values of $\mathbf{K}$. Any small error in $\mathbf{c}$ will be sufficient to amplify the solution, giving a totally wrong answer for $f(\lambda)$. This is also sufficient to make the inversion of data in positron annihilation as an ill posed problem.

Among the special techniques to solve an ill posed problem are the Tikhonov regularization[1] and the singular value decomposition.[5] The Tikhonov regularization removes the above singularity by adding some extra information into the original problem, whereas the singular value decomposition solves the problem by finding suitable subspaces to represent the density function and the counting of positron annihilation. Both of these approaches have been used in thermodynamics when inverting second virial data.[6]

A general program to invert the Fredholm integral equation of first order, the computer code being called CONTIN, was developed in 1982 by S. W. Provencher.[7,8] This is a program of general applicability and has been largely used to invert positron annihilation lifetime spectra.[9−12] The CONTIN package has 66 subprograms with 5000 lines and solves the ill posed problem by removing the singularity as in the Tikhonov regularization. The regularization parameter is chosen on the basis of a statistical test.

Another approach to invert positron annihilation lifetime spectra has been proposed by Schrader and Usmar[13] which consists of expressing the solution of (1) as an expansion in the eigenfunctions of the kernel. This approach is conceptualy equivalent to the singular value decomposition method and apparently has not been implemented for inverting positron annihilation data.

The maximum entropy approach was used in refs 14 and 15. A comparasion between application of maximum entropy and the usage of the CONTIN package is discussed in refs 16 and 17.

Methods based on models where a nonlinear fitting procedure is used to search for the model's optimum parameters have also been applied to calculate and invert the positron annihilation data.[18,19] A constrained non-negative least-squares, with a restriction on the second derivative of the positron annihilation, is another method implemented in refs 20 and 21.

* Corresponding author phone: +553134995742; e-mail: jpbraga@oxigenio.qui.ufmg.br.

10.1021/ci0000833 CCC: $20.00
Published on Web 02/24/2001

Table 1. Parameters for the Exact Density Function, Eq 2

variables	values	variables	values
a_1	0.42250	α_2	12.0688
a_2	1.57270	σ_1	−0.57710
α_1	11.6951	σ_2	0.93530

In this paper a Hopfield neural network will be applied to invert the positron annihilation data. Once the weights and external impulse to the neurons are calculated, this neural network will evolve to an equilibrium state. Each state of the neuron will then represent a component of the solution for the inverse problem.

2. ILL POSED PROBLEMS AND NEURAL NETWORK

The ill posed nature of the problem can also be circumvented by constructing a multilayer neural network that learns about the input and output of the problem. The neural network in this case will be able to predict the inverse problem, avoiding the direct calculation of the $\mathbf{K}^{-1}$. This approach has been used recently in thermodynamics[22] and for the inversion of positron lifetime annihilation lifetime spectra.[23] A large number of systems have to be used in this approach and this may represent an inconvenience.

Multilayer neural network architecture to handle a single system has also been devised. Kulkarni[24] has proposed a multilayer percepton architecture to invert data. Initial guesses for the hidden layer were done using the singular value method. Another architure for the multilayer has been sugested in ref 25 and used to invert an ill posed image reconstruction.

Farhat et al.[26] has treated the ill posed problem for image reconstruction using a Hopfield neural network. The minimum of the neural network function, with a regularized term, as in ref 27, was found using a recurrence relation. In another similar approach, the time dependent Hopfield model, the minimum is used to find the minimum of the energy function.[28] This approach will be adopted here and will be discussed below.

3. THE MODEL SYSTEM AND THE DISCRETIZATION PROCEDURE

A model system was chosen to study the inversion of positronium annihilation data by using a Hopfield neural network. The data for the protein lysozyme in water[9] was taken as a reference for the inversion procedure. A function of the form

$$f(\lambda) = a_1 e^{-\alpha_1(\ln(\lambda)-\sigma_1)^2} + a_2 e^{-\alpha_2(\ln(\lambda)-\sigma_2)^2} \qquad (2)$$

was fitted to the positronium annihilation rate spectrum published in the above reference. The constants a_i, α_i, σ_i, ($i = 1, 2$), determined by the nonlinear fitting procedure, are given in Table 1. Although function (2) gives a reasonable fit to the published spectrum, residual error is equal to 0.3, the important aspect to emphasize here is that eq 2, with the parameters in Table 1, was taken as an exact result for the present work. The function (2) is presented in Figures 3−5 as a solid line.

Since there is no analytical result for the Laplace tranform of the function (2), the integral (1) was calculated numerically. An automatic adaptive routine, based on the eight-panel Newton-Cotes rule,[29] with $a = 10^{-10}$ ns, $b = 5$ ns, and a tolerance of 10^{-10} ns, was used to generate the theoretical data for the positron annihilation data

$$c(t) = \int_0^{t_{max}} (a_1 e^{-\alpha_1(\ln(\lambda)-\sigma_1)^2} + a_2 e^{-\alpha_2(\ln(\lambda)-\sigma_2)^2}) e^{-\lambda t}\, d\lambda \qquad (3)$$

The error in the integration procedure was not taken into consideration for the inversion procedure since a total of eight to nine significant figures was achieved.

The Hopfield neural network needs the above results in a representation. Defining the quantities

$$\mathbf{K} = \begin{bmatrix} w_1K(t_1, \lambda) & w_2K(t_1, \lambda_2) & \cdots & w_nK(t_1, \lambda_n) \\ w_1K(t_2, \lambda_1) & w_2K(t_2, \lambda_2) & \cdots & w_nK(t_2, \lambda_n) \\ \vdots & \vdots & \ddots & \vdots \\ w_1K(t_m, \lambda_1) & w_2K(t_m, \lambda_2) & \cdots & w_nK(t_m, \lambda_n) \end{bmatrix} \qquad (4)$$

$$\mathbf{c} = [c(t_1)\ c(t_2)\ \ldots\ c(t_m)]^T \qquad (5)$$

and

$$\mathbf{f} = [f(\lambda_1)\ f(\lambda_2) \ldots\ f(\lambda_n)]^T \qquad (6)$$

where (w_1, w_2, ..., w_n) are appropriate weights, m is the training set size and n is the basis size. In this paper a rectangular basis was used, and thus $w_j = \lambda_j - \lambda_{j-1}$, $f_j = f(\bar{\lambda}_j)$, $K_{ij} = K(t_i,\bar{\lambda}_j)$, $c_i = c(t_i)$, and $\bar{\lambda}_j = (\lambda_{j-1} + \lambda_j)/2$.

4. THE HOPFIELD NETWORK

The publication of the seminal paper describing the Hopfield neural network in 1982[30] was one of the key factors responsible for the revival of the neural networks area. The ideas behind the Hopfield network, such as recurrency and creation of stable states (atracttors), were in fact explored by other authors[31] in the 1970s, but only Hopfield was able to formally describe them in the neural networks framework. The success of his model was certainly due to the neat and concise description of a neural network topology, its learning rule, and, furthermore, the proof that such a neural network stabilizes onto stable states by minimizing the network *energy*. Hopfield also associated network binary states $f_i(t) = \pm 1$, where index i refers to the output value of neuron i, with spins of an Ising model,[30] that enabled the direct application of statistical mechanics theory to analyze network behavior.

The Hopfield network is formed by a single recurrent layer of fully connected threshold logic units, as shown in Figure 1. The state of a neuron i in the original Hopfield model is a function of its linear output u_i, which is obtained by the weighted sum of all its inputs, which are connected to the outputs of all other neurons (but not to its own output): $f_i(t+1) = \phi(u_i(t))$, where $u_i(t) = \sum_{j,j\neq i} T_{ji} f_j(t)$, T_{ji} is the weight of the connection between nodes i and j, and $\phi(\cdot)$ is the node's activation function. The model was first applied to the solution of the associative memory problem, where pixels of the image correspond to the binary states of network neurons. Stored patterns (images) corresponded to fixed points, or attractors, in the network state space. The convergence properties of attractors allow a mechanism for correcting errors, since distorted versions of the image may

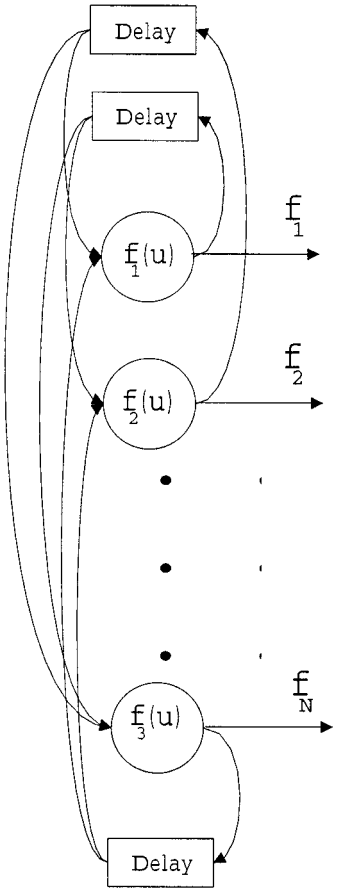

Figure 1. A schematic view of a Hopfield network.

correspond to states that are within the basin of attraction of the target pattern. Small deviations from the fixed point are corrected throughout the network trajectory in the state space, which ends up in the fixed point.

The most important result of the original Hopfield work was certainly the proof that the network minimizes energy. In the associative memory problem, for example, stored patterns (attractors) correspond to energy minima, and distorted versions of these patterns have higher energy that is minimized by the network. Hopfield quickly noticed that his network could also be applied to solve general optimization problems, as long as the function to be optimized could be specified in terms of the network energy function. Following his original 1982 paper, he generalized his ideas and published another work in 1985[32] presenting a solution of the classical travelling salesman problem with a Hopfield network.

In this work, the Hopfield network is applied to the solution of an inverse ill posed problem, which is described in terms of an optimization problem. The general matrix form of the Fredholm integral equation of first order, $\mathbf{Kf} = \mathbf{c}$, can now be written in terms of network energy function[28]

$$E(f_1, f_2, ..., f_n) = \frac{1}{2}(\sum_{n}^{j=1} K_{1j}f_j - c_1)^2 + \frac{1}{2}(\sum_{n}^{j=1} K_{2j}f_j - c_2)^2 + ... + \frac{1}{2}(\sum_{n}^{j=1} K_{nj}f_j - c_n)^2 \quad (7)$$

where $f_j = \phi(u_j(t))$.

In the formulation above, the network is expected to reach a stable state $\mathbf{f} = [\mathbf{f_1 f_2 ... f_n}]$ that minimizes $||\mathbf{Kf} - \mathbf{c}||_2^2$. From the energy function above, a continuous Hopfield model can be described by using the Hamiltonian relation

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial f_i} \quad (8)$$

that results in

$$\frac{du_i}{dt} = -(\sum_{n}^{j=1} K_{1j}f_j - c_1)K_{1i} - (\sum_{n}^{j=1} K_{2j}f_j - c_2)K_{2i} - ... -(\sum_{n}^{j=1} K_{nj}f_j - c_n)K_{ni} \quad (9)$$

It can now be concluded that

$$T_{ij} = -\sum_{n}^{l=1} K_{li}K_{lj} = T_{ji} \quad (10)$$

$$I_i = \sum_{n}^{j=1} K_{ji}c_j \quad (11)$$

Therefore, the evolution in time of the linear outputs u_i of network neurons is given by

$$\frac{du_i}{dt} = \sum_{n}^{j=1} T_{ij}f_j + Ii \quad (12)$$

with $i = 1, 2, ..., n$. The equation above is evolved in time until an equilibrium is reached, which is expected to correspond to a solution for the vector $\mathbf{f} = [\mathbf{f_1 f_2 ... f_N}]$ that minimizes $||\mathbf{Kf} - \mathbf{c}||_2^2$. These equations were integrated by a Runge−Kutta-Fehlberg method as described in ref 29. A tolerance of 10^{-10} was used to propagate the solution for several maximum integration times. A unit of time of 1×10^6 was adopted in integration of the Hopfield equations.

5. RESULTS AND DISCUSSION

An important step in the inversion procedure presented here is the decision about the number of neurons to be used in the Hopfield differential equation. This can be carried out by analyzing the residual error, $||\mathbf{Kf} - \mathbf{c}||_2^2$, for a different number of neurons. This analysis gives $||\mathbf{Kf} - \mathbf{c}||_2^2 = 0.2940$ for 16 neurons, $||\mathbf{Kf} - \mathbf{c}||_2^2 = 2.335 \times 10^{-4}$ for 32 neurons, and $||\mathbf{Kf} - \mathbf{c}||_2^2 = 1.770 \times 10^{-16}$ for 64 neurons. A very precise representation of the integral eq 1 can therefore be obtained with 64 neurons, and this will be the number of neurons to be used here. Due to the oscillating character of the density function, a total of 64 neurons is also appropriate to describe it.

The simulated counting of positronium annihilation, considering 16 measurements, is given in Figure 2. From these measurements the neural network was evolved, and the states of the neurons, each one corresponding to one measurement, is given in Figure 3, together with the exact density function. Even for this modest calculation, which has a residual error not neglible, the results are in reasonable agreement with the expected values. For example the

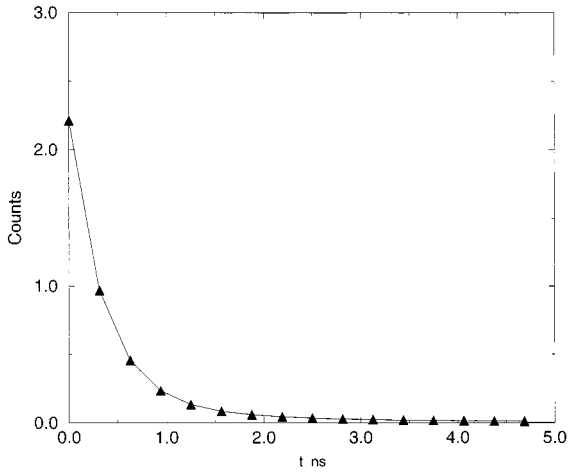

Figure 2. Simulated positron annihilation lifetime spectrum for 16 measurements.

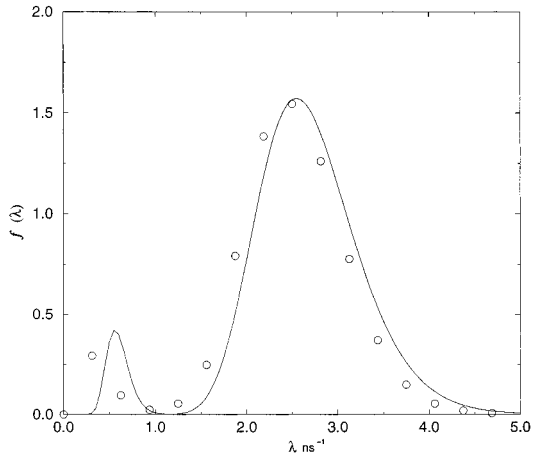

Figure 3. States of the neurons (bullets) and exact density function for $n = 16$ and $t = 5$.

maximum of the inverted density function happens at $\lambda = 2.50$ ns^{-1} with $f(\lambda) = 1.55$ compared with the exact values of $\lambda = 2.54$ ns^{-1} with $f(\lambda) = 1.57$. The above results also show the necessity to increase the number of neurons if a more precise description of the density function is required. With a few neurons, some regions of the density function, such as the maximum of the curve, may not be well described.

To see the effect of adding neurons the number of neurons was changed to 32, keeping the learning time the same as in the previous calculation. The result, presented in Figure 4, is considerably improved. The maximum now happens at $\lambda = 2.55$ ns^{-1} with $f(\lambda) = 1.58$. The neural network tries to find a minimum global error for the problem, and when the number of neurons is increased some points can improve at the expense of some other point. The important issue is to find a minimum for all neurons in the network. This effect can be seen in the present case where the position of the maximum has improved.

There is a guarantee that the energy function will reach a stable low energy region, that can be a minimum (local or global), in the Hopfield model, and therefore a very precise result can be achieved by increasing the learning time and the number of neurons. That is the situation presented in Figure 5 where $n = 64$ and $t = 90$. The inverted result gives precise density function, describing with great accuracy the system under consideration.

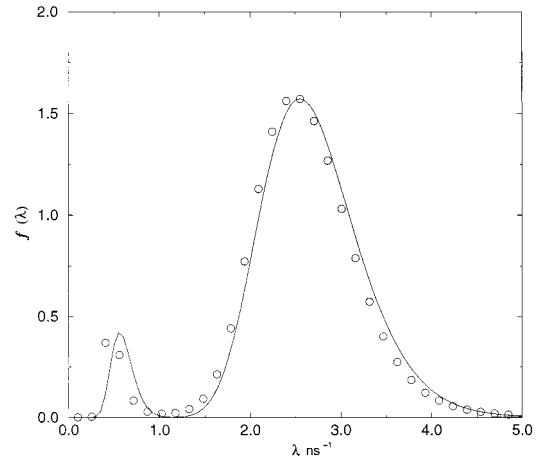

Figure 4. The same as in Figure 2 but for a neural network of 32 neurons.

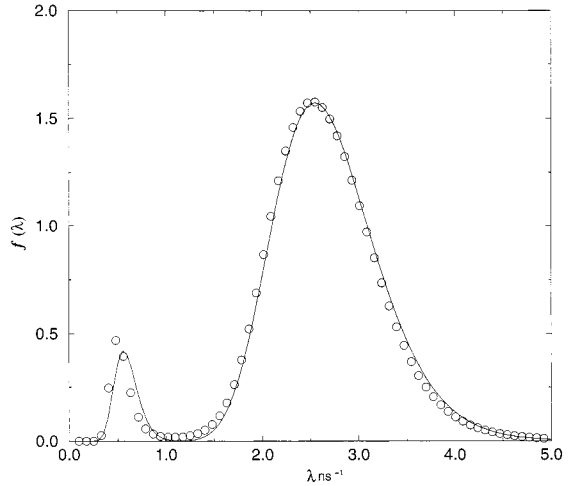

Figure 5. The same as in Figure 4 but for $n = 64$ and $t = 90$.

6. CONCLUSIONS

Inversion of positron annihilation lifetime spectrum, using the Hopfield neural network, was analyzed on simulated data. The efficiency of the method was discussed varying the number of neurons and the learning time in the training process. The results showed an excellent agreement between the Hopfield neural network and the exact data.

The ill posed nature of the above inversion problem requires special techniques for its numerical treatment. When handling positronium annihilation data the two most often used methods consist of (i) a decomposition of eq 1, after a representation has been found, in an adequate basis and (ii) solving the Fredholm integral equation using a regularization method. In both cases there is a need to find the size of basis and a regularization parameter. One advantage of the present neural network approach over these two methods is that there is no need to choose a free parameter. Nevertheless one has to find the maximum time of the learning process and an adequate number of the neurons. Due to the convergence nature of the Hopfield neural network approach, the integration time can be set as large as possible. What has to be done in fact is to decide about the number of neurons in the network, which is a relatively simple task when compared with the choice of parameters in the approaches (i) and (ii) above.[6]

The computer program to perform the time evolution of neurons state, not counting the integration subroutine, has 60 lines. The discretization of eq 1 has been done with another computer code which has 120 lines. All together the code has 180 lines. The computation time to obtain converged values for Figures 3−5 are respectively 32s, 66s, 2800s in a workstation Alpha/Digital 433 MHz. No information a priori was given to the states of the neurons in the above calculations. The initial state, for all neurons, being $\mathbf{u} = \mathbf{0}$, is not a very good initial guess. Therefore the computational time can be reduced by giving a better initial guess for the state of the neurons.

Noise and apparatus resolution can be included in the present approach. The analysis with 16 neurons actually presents a good example of how noise can be handled within the Hopfield neural network approach. The residual error for this case was 0.2949 and even though the inverted density function is tolerable, if the errors are increased, the neurons will converge to those points which will make the integral (1) valid. Further studies are planned, where the above neural network model will be applied to real laboratory data, taking into account the apparatus resolution.

ACKNOWLEDGMENT

We would like to thank Dr. Welington Ferreira de Magalhães, from the Positron Laboratory (LEAP) at Departamento de Química − UFMG, for the incentive to carry on the present work. This work was supported by CNPq and FAPEMIG, Brazil.

REFERENCES AND NOTES

(1) Tikhonov, A. N.; Arsénine, V. *Solutions of ill-posed problems*; Mir: Moscow, 1974.
(2) Tikhonov, A. N.; Goncharsky A. V. *Ill-Posed Problem in Natural Sciences*; Mir: Moscow, 1987.
(3) Wing, G. M.; Zahrt, J. D. *A primer on integral equation of first order*; SIAM: Philadelphia, 1991.
(4) Mogensen, O. E. *Positron Annihilation in Chemistry*; Springer-Verlag: Heidelberg, 1995.
(5) Golub G. H.; Van Loan, C. F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, 1989.
(6) Lemes, N. H. T.; Braga, J. P.; Belchior, J. C. Spherical potential energy function from second virial coefficient using Tikhonov regularization and truncated singular value decomposition. *Chem. Phys. Lett*. **1998**, *296*, 233−238.
(7) Provencher, S. W. A constrained regularization method for inverting data represented by linear algebraic equation or integral equations. *Comput. Phys. Commun*. **1982**, *27*, 213−227.
(8) Provencher, S. W. Contin: A general purpose constrained regularization program for inverting noisy linear algebraic and integral equations. *Comput. Phys. Commun*. **1982**, *27*, 229−242.
(9) Gregory, R. B.; Procyk, A. In *International Symposium on Positron anihilation studies of fluids*; Sharma, S. C., Ed.; World Scientific: Arligton, 1987; p 524.
(10) Gregory, R. B.; Zhu, Y. Analysis of Positron Annihilation Lifetime Data by Numerical Laplace Inversion with the Program CONTIN. *Nucl. Instr. Methods A* **1990**, *290*, 172−182.
(11) Gregory, R. B. Analysis of Positron Annihilation Lifetime Data by Numerical Inversion: Corrections for Source terms and Zero-Time Shift Errors. *Nucl. Instr. Methods A* **1991**, *302*, 496−507.
(12) Jean, Y. C.; Dai, G. H. Free volume hole distributions of polymers probeb by positron annihilation spectroscopy: experiences of using the CONTIN program. *Nucl. Instr. Methods B* **1993**, *79*, 356−361.
(13) Schrader, D. M.; Usmar, S. G. In *International Symposium on Positron anihilation studies of fluids*; Sharma, S. C., Ed.; World Scientific: Arligton, 1987; p 215.
(14) Shulka, A.; Peter, M.; Hoffmann, L. Analysis of positron lifetime spectra using quantified maximum entropy and a general linear filter. *Nucl. Instr. Methods A* **1993**, *335*, 310−317.
(15) Hoffmann, L.; Shulka, A.; Peter, M.; Barbiellini, B.; Manuel, A. A. Linear and nonlinear approaches to solve the inverse problem: applications to positron annihilation experiments. *Nucl. Instr. Methods A* **1993**, *335*, 276−287.
(16) Wang, C. L.; Maurer, F. H. J. New approach to determine orthopositronium lifetime distribution in polymers: A comparison between maximum entropy and the numerical Laplace inversion. *Macromolecules* **1996**, *29*, 8249−8253.
(17) Dublek, G.; Eichler, S. Do MELT or CONTIN programs accurately reveal the o-Ps lifetime distribution in polymers? Analysis of simulated lifetime spectra. *Phys. Stat. Sol*. **1998**, *168*, 333−350.
(18) Kirkegaard, P.; Eldrup, M.; Mogensen, O. E.; Pedersen, N. J. Program system for analysing positron lifetime spectra and angular correlation curves. *Comput. Phys. Commun*. **1981**, *23*, 307−335.
(19) Kansy, J. Microcomputer program for analysis of positron annihilation lifetime spectra. *Nucl. Instr. and Methods A* **1996**, *384*, 235−244.
(20) Deng, A. H.; Panda, B. K.; Fung, S.; Beling, C. D. Positron Lifetime Analysis from generalized Constrained Least-Squares Technique. *Phys. Status Solidi B* **1997**, *201*, 413−416.
(21) Deng, A. H.; Panda, B. K.; Fung, S.; Beling, C. D.; Schrader, D. M. Positron lifetime analysis using the matrix inverse Laplace transformation method. *Nucl. Instr. Methods B* **1998**, *140*, 439−448.
(22) de Almeida, M. B.; Braga, A. P.; Braga, J. P.; Belchior, J. C.; Yared, G. F. G. Radial basis function networks for obtaining long-range dispersion coefficients from second virial data. *Phys. Chem. Chem. Phys.* **2000**, *2*, 103−107.
(23) Pazsit, I.; Chakarova, R.; Linden, P.; Maurer, F. H. J. Unfolding positron lifetime spectra with neural networks. *Appl. Surf. Sci*. **1999**, *149*, 97−102.
(24) Kulkarni, A. D. Solving ill-posed problems with artificial neural networks. *Neural Networks* **1991**, *4*, 477−484.
(25) Blass, W. E.; Mahan, S. L.; Chin, G. Convolution connection paradigm neural network enables linear system theory-based image enhancement. *Int. J. Imaging System Tech*. **1995**, *6*, 350−357.
(26) Farhat, N. H.; Bai, B. Echo inversion and target shape estimation by Neuromorphic processing. *Neural Networks* **1989**, *2*, 117−125.
(27) Zhou, Y. T.; Chellappa, R.; Vaid, A.; Jenkins, B. K. Image restoration using a neural network. *IEEE T. Acoust. Speech* **1988**, *36*, 1141−1151.
(28) Vemuri, V.; Jang, G. S. Inversion of Fredholm Integral Equation of First Kind with Fully Connected Neural Networks. *J. Franklin Inst.* **1992**, *329*, 241−257.
(29) Forsythe, G. E.; Malcolm, M. A.; Moler, C. B. Computer methods for mathematical computations; Prentice-Hall: NJ, 1977.
(30) Hopfield, J. J. Neural networks and physical systems with collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* **1982**, *79*, 2554−2558.
(31) Little, W. A.; Shaw, G. L. A statistical theory of short and long-term memory. *Behavioral Biol.* **1975**, *14*, 115−133.
(32) Hopfield, J. J.; Tank, D. W. Neural computation of decision in optimation problems. *Biological Cybernetics* **1985**, *52*, 141−152.

CI0000833