1-1-2004

Gaussian filters and filter synthesis using a Hermite/Laguerre neural network

M. Mackenzie
University of Wollongong

K. Tieu
University of Wollongong, ktieu@uow.edu.au

[9] R. H. Paul, *Measure Theory*: Van Nostrand, 1950.
[10] H. Lu, "On stability of nonlinear continuous-time neural networks with delays," *Neural Networks*, vol. 13, no. 10, pp. 1135–1144, 2000.
[11] K. M. Richard, "Asymptotic behavior of nonlinear delayed-differential equations," *J. Differential Equations*, vol. 1, pp. 293–305, 1995.
[12] C. M. Marcus and R. M. Westervelt, "Stability of analog neural networks with delay," *Phys. Rev. A.*, vol. 39, pp. 347–359, 1989.
[13] L. Wang and X. Zou, "Exponential stability of cohen-grossberg neural networks," *Neural Networks*, vol. 15, pp. 415–422, 2002.
[14] H. Ye, A. N. Michel, and K. Wang, "Qualitative analysis of cohen-grossberg neural networks with multiple delays," *Phys. Rev. E*, vol. 51, pp. 2611–2618, 1995.
[15] J. Zhang and X. Jin, "Global stability analysis in delayed hopfield neural models," *Neural Networks*, vol. 13, no. 7, pp. 745–753, 2000.

Gaussian Filters and Filter Synthesis Using a Hermite/Laguerre Neural Network

Mark Mackenzie and Kiet Tieu

***Abstract*—A neural network for calculating the correlation of a signal with a Gaussian function is described. The network behaves as a Gaussian filter and has two outputs: the first approximates the noisy signal and the second represents the filtered signal. The filtered output provides improvement by a factor of ten in the signal-to-noise ratio. A higher order Gaussian filter was synthesized by combining several Hermite functions together.**

I. Introduction

For application in scientific instrumentation and industrial processing, neural networks have the ability to perform nonlinear interpolation on complex physical systems without the need to develop an underlying physical model of the process. In many of these applications, noise is often embedded in the signal, which must be removed. While the sigmoidal neural network can be trained to interpolate noisy signals, the filtering properties are still under development [1], [2]. The objective of this paper is to introduce a neural network filter with clearly defined parameters. This is achieved by implementing a Gaussian moving average filter using an orthonormal neural network. In contrast to the sigmoid neural network, the properties of the Gaussian filter are well developed and are given in Blinchikoff and Zverev [3].

Fig. 1 explains the operation of a moving average filter. The filter integrates only the noisy function $x(\tau)$ enclosed within the window function $k(t-\tau)$ located at t. An average value $y(t)$ of the noisy function at t is obtained. This average value is the correlation of the noisy function with the window function. With a Gaussian window function, this is

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)k(\tau - t)d\tau = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} x(\tau) \exp\left(-\frac{(\tau - t)^2}{2\sigma^2}\right) d\tau \quad (1)$$

Manuscript received August 23, 2001; revised April 12, 2002 and March 17, 2003.

The authors are with the Department of Mechanical Engineering, University of Wollongong, Wollongong, N.S.W. 2522, Australia (e-mail: kiet_tieu@uow.edu.au).

Digital Object Identifier 10.1109/TNN.2003.820558

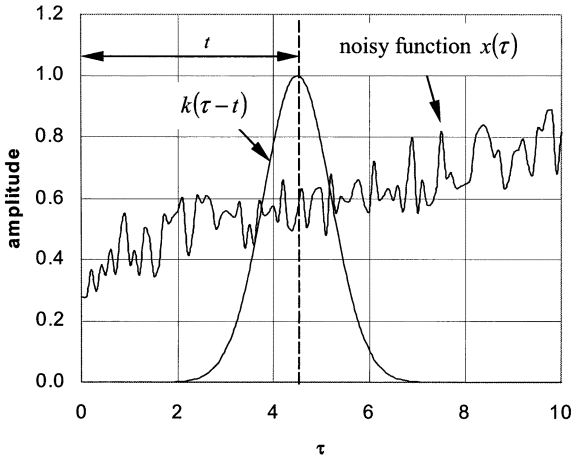


Fig. 1. Gaussian window, moving average filter.

where

$$k(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right). \quad (2)$$

In this paper, we compute the correlation using a neural network with Hermite orthonormal activation functions. By using these particular types of activation functions, a simple expression can be obtained for the correlation with a Gaussian window function. The layout of this paper is as follows. Section II explains how the correlation is obtained. In this section, it is also shown how to synthesize other types of moving average filters based on the Hermite functions. Section III describes how to apply the network to real-life problems by cascading the network. Section IV applies the network to filter noisy signals and Section V applies the network to the detection of signals from an ultrasonic sensor.

II. Theory

The Hermite and Laguerre functions described in this section belong to the general class of orthonormal functions: general definitions, properties, and theorems of orthonormal functions are given in Kreyszig [4]. Other types of orthonormal networks, which so far have been developed, include the Fourier [5], Legendre [6], and (more recently) Tchebychev [7] functions.

A. Gaussian Filter

Consider the approximation of a function by a Hermite network. The network approximates the function by a finite expansion of Hermite functions on an interval $\{-\infty, +\infty\}$

$$x(t) \cong x_h(t) = \sum_{n=0}^{N} a_n h_n(t) \quad \{-\infty \le t \le \infty\} \quad (3)$$

where $\{a_n\}$ is a set of suitably chosen weights and $h_n(t)$ are the Hermite activation functions [8] on the interval $\{-\infty, +\infty\}$. The Hermite activation functions are the product of a Hermite polynomial $H_n(t)$ and Gaussian function, with the coefficients of the polynomials chosen so that the Hermite functions satisfy the orthonormality condition

$$\int_{-\infty}^{+\infty} h_n(t)h_m(t)dt = \begin{cases} 0, & \text{for } m \neq n \\ 1, & \text{for } m = n \end{cases} \quad (4)$$

TABLE I
HERMITE POLYNOMIALS

$H_0(t) = 1$	$H_3(t) = 8t^3 - 12t$
$H_1(t) = 2t$	$H_4(t) = 16t^4 - 48t^2 + 12$
$H_2(t) = 4t^2 - 2$	$H_5(t) = 32t^5 - 160t^3 + 120t$

where

$$h_n(t) = \frac{H_n(t)}{\sqrt{2^n n! \sqrt{\pi}}} \exp\left(\frac{-t^2}{2}\right). \tag{5}$$

Table I lists the first six Hermite polynomials. The remainder may be determined from the recurrence relation

$$H_{n+1}(t) = 2tH_n(t) - 2nH_{n-1}(t). \tag{6}$$

Some of the Hermite activation functions are shown in Fig. 2(a) to Fig. 2(c). Since the fundamental Hermite activation function $h_0(t)$ is the Gaussian function with $\sigma = 1$, it may be used as the filter function and

$$k(t) = \frac{h_0(t)}{\sqrt{2\sqrt{\pi}}}. \tag{7}$$

The correlation integration of (1) is then approximately equal to the correlation of the Hermite network with the fundamental Hermite function

$$y(t) \cong y_h(t) = \frac{1}{\sqrt{2\sqrt{\pi}}} \int_{-\infty}^{+\infty} \left\{ \sum_{n=0}^{N} a_n h_n(\tau) \right\} h_0(\tau - t) d\tau. \tag{8}$$

This expression can be evaluated from the following relationship for the correlation between the Hermite functions of different order (a derivation can be found in [9]). For $n \geq m$

$$\int_{-\infty}^{+\infty} h_n(\tau) h_m(\tau - t) d\tau = \begin{cases} l_m^{n-m}\left(\frac{t^2}{2}\right), & \text{for } t \geq 0 \\ (-1)^{n+m} l_m^{n-m}\left(\frac{t^2}{2}\right), & \text{for } t < 0 \end{cases} \tag{9}$$

where $l_m^n(t)$ is a orthonormal associated Laguerre function. The case $m < n$ is obtained by reversal of the order of n and m in (9) and multiplication by $(-1)^{n+m}$. The orthonormal associated Laguerre functions are derived from the associated Laguerre polynomials [10] $L_m^n(t)$ by

$$l_m^n(t) = \sqrt{\frac{m}{(n+m)}}\, t^{\frac{n}{2}} e^{-t/2} L_m^n(t). \tag{10}$$

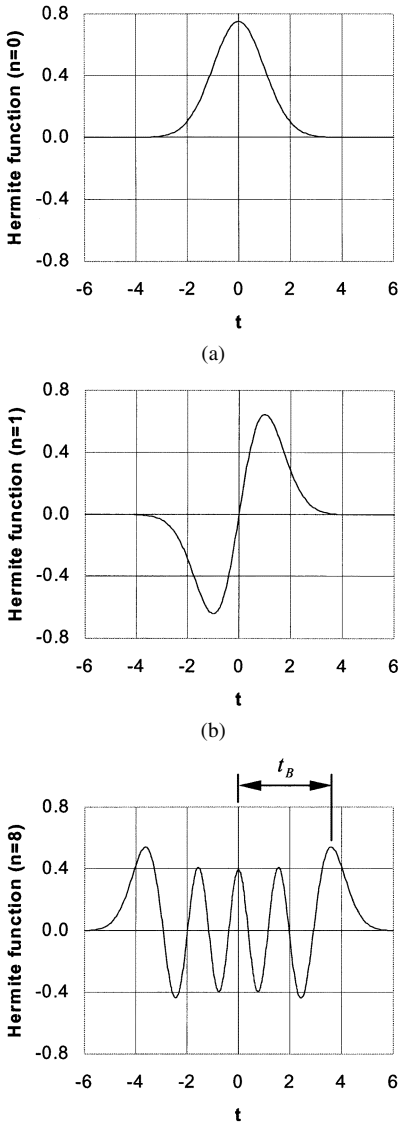
Substituting (9) into (8), we obtain

$$y_h(t) = \begin{cases} \frac{1}{\sqrt{2\sqrt{\pi}}} \sum_{n=0}^{N} a_n l_0^n\left(\frac{t^2}{2}\right), & \text{for } t \geq 0 \\ \frac{1}{\sqrt{2\sqrt{\pi}}} \sum_{n=0}^{N} a_n (-1)^n l_0^n\left(\frac{t^2}{2}\right), & \text{for } t < 0 \end{cases}. \tag{11}$$

The usefulness of this expansion is that by training the Hermite network to the input function, one also immediately obtains the weights of the Laguerre network $\{a_n\}$, which is the correlation of the input function with a Gaussian function.

Having both a subscript and a superscript, the associated Laguerre functions are more complicated than the Hermite functions. However, for the Gaussian filter, only the associated Laguerre functions of order zero are required, thus greatly simplifying the result. The associated Laguerre functions of subscript zero are

$$l_0^n(t) = \frac{t^{\frac{n}{2}} \exp\left(-\frac{t}{2}\right)}{\sqrt{n!}}. \tag{12}$$



(a) (b) (c)

Fig. 2 (a). Hermite function $h_0(t)$. (b) Hermite function $h_1(t)$. (c) Hermite function $h_8(t)$

B. Interpretation as a Gaussian Radial Basis Neural Network (RBNN)

The functions $l_0^n(t^2/2)$ are shown in Fig. 3. For n larger than five, they are quite well approximated by the Gaussian function (Fig. 4, $n = 8$), although the Laguerre function is slightly skewed away from the origin. They slowly shift along the axis and slowly reduce in amplitude as n increases. The location of the central peak t_c and its amplitude A can be determined by differentiation of $l_0^n(t^2/2)$ as

$$t_c = \sqrt{2n} \tag{13}$$

$$A = \frac{n^{\frac{n}{2}}}{\sqrt{n!}} \exp\left(-\frac{n}{2}\right). \tag{14}$$

The variance of the Gaussian approximation remains constant at about $1.5/\sqrt{2}$ for all n.

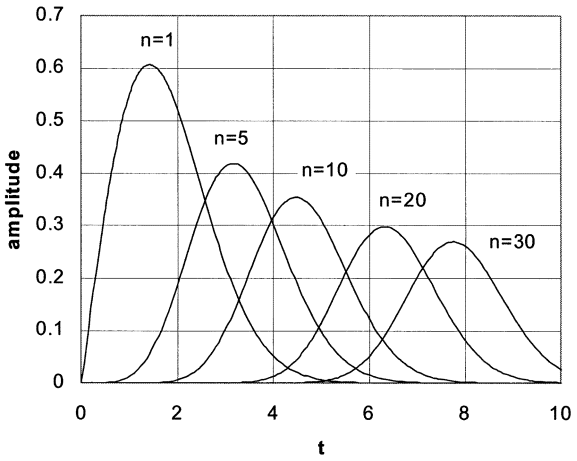


Fig. 3. Associated Laguerre functions $l_0^n(t^2/2)$ of increasing order.

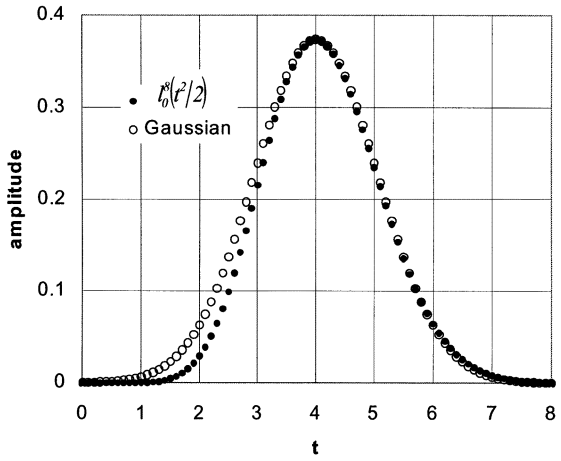


Fig. 4. Comparison of $l_0^8(t^2/2)$ with a Gaussian function.

Using the Gaussian approximation, the output Laguerre network is approximately

$$y_h(t) \cong \sum_{n=0}^{N} \left\{ a_n \frac{n^{\frac{n}{2}} \exp\left(-\frac{n}{2}\right)}{\sqrt{n!}} \right\} \exp\left(-\frac{(t-\sqrt{2n})^2}{2.25} \right). \quad (15)$$

In this form, it may be recognized as a Gaussian RBNN, although there are some differences. Being slightly skewed, Laguerre functions are not radial symmetric and, therefore, the network cannot strictly be defined as an RBNN. However, the real difference between the present method and an RBNN is in the training of the networks. In an RBNN, the centers of the Gaussian radial functions may be freely chosen and the weights are computed to minimize the mean square error between the network and the training function. With the Laguerre output network, the centers of the radial functions cannot be freely chosen and are given by (13). Furthermore, the weights of the Laguerre network do not minimize the error with the training function (they are optimum with respect to the Hermite input network).

C. Equivalence to Kernel Regression

The correlation can also be computed using a Monte-Carlo integration given by

$$y(t) \approx \frac{T}{I\sigma\sqrt{2\pi}} \sum_{i=0}^{I} x(t_i) \exp\left(-\frac{(t-t_i)^2}{\sigma^2} \right) \quad (16)$$

where $\{t_i, x(t_i) : i = 0, 1, 2, \ldots, I\}$ is a discrete data set for the function $x(t)$ across the interval $-T/2 \le t \le T/2$. This summation is the kernel regression in the form given by Priestley and Chao [11]. In simulations with uniformly sampled data [Fig. 9(d)], the Hermite/Laguerre neural network achieved an identical signal-to-noise ratio (SNR) to the kernel regression of (16).

D. Filter Synthesis

The procedure described in Section II-A can be extended to include other types of moving average filters besides the Gaussian filter by combining various Hermite functions together. The Hermite functions have a simple Fourier transform, which allows the filter to be designed in the frequency domain. It can be shown [12] that this Fourier transform (FT) is

$$h_n(t) \xrightarrow{\text{FT}} \sqrt{2\pi}(-j)^n h_n(\omega). \quad (17)$$

This property may be exploited to obtain the time domain coefficients of any filter synthesized in the frequency domain as a summation of Hermite functions. The filtering of a noisy function using the synthesized filter can then be obtained from the correlation of (9).

A simple fourth-order filter can be designed as follows. The Gaussian filter $k(t)$ transforms into the frequency domain as $K(\omega)$, given by

$$\begin{aligned} K(\omega) &= \exp\left(-\frac{\omega^2}{2}\right) \\ &= \frac{1}{1+\frac{\omega^2}{2}+\frac{\omega^4}{8}+\cdots} \\ &= \frac{1}{\left(1+\frac{\omega^2}{2}\right)\left(1+\frac{\omega^4}{8}+\cdots\right)} \end{aligned} \quad (18)$$

where the expansion on the left-hand side was obtained using the Taylor series. The Gaussian filter is predominantly second order with a bandwidth w_B given approximately by

$$w_B = \sqrt{2}. \quad (19)$$

This bandwidth was obtained by neglecting powers greater than two in the Taylor series expansion. The actual bandwidth was 40% lower due to the contribution of the higher order terms in the Taylor expansion. The fourth-order filter is derived from the Gaussian filter by multiplication in the frequency domain by the term $(1+\omega^2/2)$ so that

$$\begin{aligned} K_{4th}(\omega) &= \left(1+\frac{\omega^2}{2}\right)\exp\left(-\frac{\omega^2}{2}\right) \\ &= b_0 h_0(t) + b_2 h_2(t) \\ &\approx \frac{1}{1+\frac{\omega^4}{8}} \end{aligned} \quad (20)$$

where the coefficients b_0 and b_2 of the Hermite functions can be determined by inspection as

$$b_0 = \frac{5\pi^{\frac{1}{4}}}{4} \quad (21)$$

and

$$b_2 = \frac{\sqrt{2}\pi^{\frac{1}{4}}}{4}. \quad (22)$$

The equivalent time domain filter $k_{4\text{th}}(t)$, using (17), is

$$k_{4\text{th}}(t) = \frac{b_0 h_0(t) - b_2 h_2(t)}{\sqrt{2\pi}}. \quad (23)$$

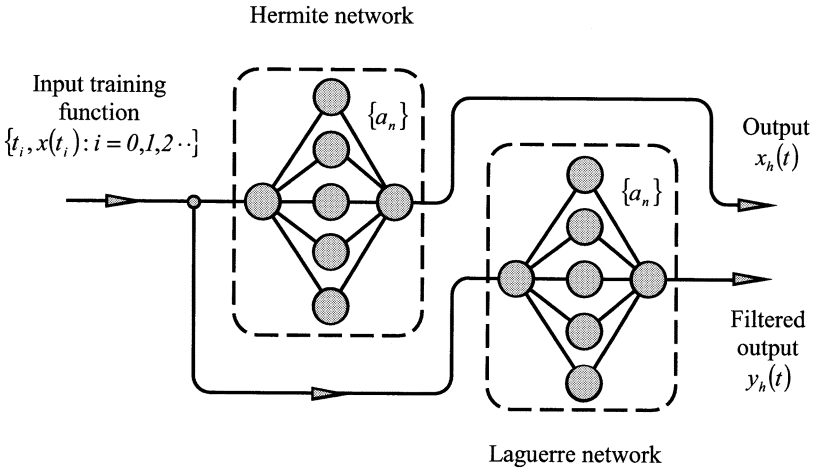


Fig. 5. Diagram of Hermite/Laguerre neural network.

In kernel regression, this filter corresponds to the fourth-order Gaussian kernel function described by Wand and Jones [13].

The filtered output $y_h(t)$ of the noisy function $x(t)$ with the fourth-order filter is given by repeated application of (9) as

$$y_h(t) = \frac{b_0}{\sqrt{2\pi}} \sum_{n=0}^{N} a_n l_0^n \left(\frac{t^2}{2}\right) - \frac{a_0 b_2}{\sqrt{2\pi}} l_0^2 \left(\frac{t^2}{2}\right) + \frac{a_1 b_2}{\sqrt{2\pi}} l_1^1 \left(\frac{t^2}{2}\right) - \frac{b_2}{\sqrt{2\pi}} \sum_{n=2}^{N} a_n l_2^{n-2} \left(\frac{t^2}{2}\right), \qquad (t \geq 0) \tag{24}$$

with a similar equation for $t < 0$. The associated Laguerre functions of higher order in this equation can be found from the recurrence relation given in [10].

III. Method

A diagram of the network, which has a single input and two outputs, is given in Fig. 5. It consists of two networks: a Hermite network and a Laguerre network. The first output is from the Hermite network and the second, from the Laguerre network, computes the correlation of the noisy function with the filter function. The network is trained by minimizing the error between the Hermite network and the training data from the noisy function. The weights of the trained Hermite network are passed directly to the Laguerre network.

The purpose of the output Laguerre network is only to provide the correlation. It behaves in the same way as a moving average filter.

A. Training

Since the weights of the Laguerre network are passed to it from the Hermite network, only the latter needs to be trained. A criteria often applied for defining the error of neural networks is the root mean square error (rmse), given by

$$\text{rmse} = \sqrt{\int_{-\infty}^{+\infty} \left(x(t) - x_h(t)\right)^2 dt}. \tag{25}$$

The advantage of using orthonormal functions as activation functions compared to sigmoidal functions is the ease and speed with which the optimum weights can be calculated. It may be shown [14] that for orthonormal functions this error is a minimum when the weights are given by the integration

$$a_m = \int_{-\infty}^{+\infty} x(t) h_m(t) dt. \tag{26}$$

To train the Hermite orthonormal network, this integral was approximated numerically by a summation of the training data $\{x(t_i) : i = 0, 1, 2, \ldots, I\}$ on the interval $\frac{-T}{2} \leq t \leq \frac{T}{2}$

$$a_m \cong \frac{T}{I} \sum_{i=0}^{I} x(t_i) h_m(t_i) \tag{27}$$

where I is the total number of training data. The optimum weights were then obtained by applying the gradient descent algorithm on the error $e(t_i)$ between the training data and neural network interpolation

$$a_m(t_{i+1}) = a_m(t_i) - \lambda \frac{\partial e(t_i)}{\partial a_m} \tag{28}$$

where λ is the learning rate coefficient and the error $e(t_i)$ is

$$e(t_i) = \left(x(t_i) - \sum_{n=0}^{N} a_n h_n(t_i)\right)^2. \tag{29}$$

For the Hermite neural network, the gradient descent algorithm takes the particular form

$$a_m(t_{i+1}) = a_m(t_i) + 2\lambda \left(x(t_i) - \sum_{n=0}^{N} a_n h_n(t_i)\right) h_m(t_i). \tag{30}$$

Although there are more sophisticated methods of numerical integration, the advantage of the summation of (27) is that it is also applicable to randomly distributed data; in this form, it computes the Monte-Carlo integration.

B. Cascaded Neural Network

Both the Hermite and Laguerre functions occur in quantum physics, where they are particularly important. Many of their useful results for function approximation can be found from the application in quantum physics. Results in this section concerning the bandwidth were obtained from the quantum physics solution of the harmonic oscillator [15].

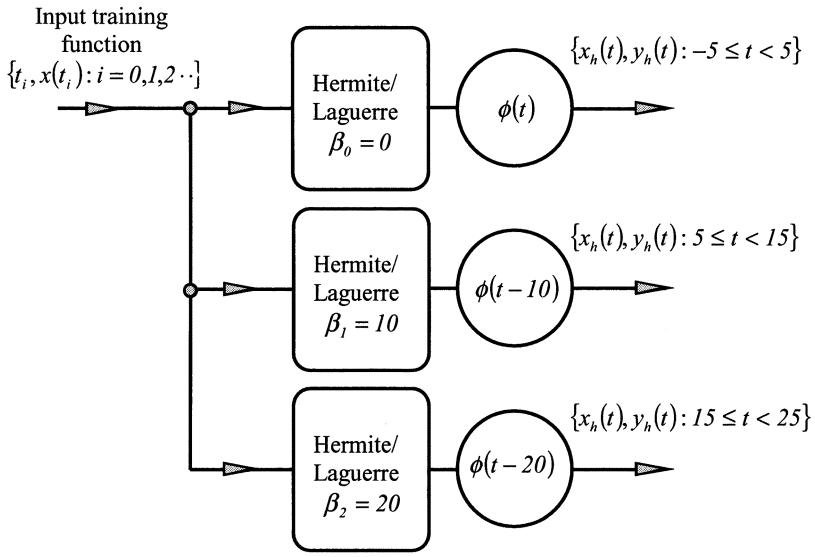


Fig. 6. Cascaded Hermite/Laguerre neural network.

In order to approximate a function, the size of the network needed must be estimated and the training function scaled into the appropriate range of the network. These factors are determined from the spatial and frequency bandwidth of the network. Efficient function approximation involves matching the spatial and frequency bandwidth of the network as closely as possible to that of the function.

Due to the exponential decay of the Hermite functions, the effective range of the functions is considerably less than the infinite interval of the orthonormal condition (4) would suggest. Fig. 2(c) shows a Hermite function of order {8} that illustrates the important properties of the series. The function closely resembles a cosine signal of gradually increasing amplitude and decreasing frequency as it moves further away from the origin (Hermite functions of odd order resemble a sine wave). Beyond a certain range, the Gaussian part of the Hermite function dominates and it rapidly drops to zero. For interpolation purposes, only the central portion of the Hermite function, which resembles a cosine, is effective. Therefore, the spatial bandwidth $2t_B$ [Fig. 2(c)] of the network is the central region of the highest order Hermite function in the network. This is [15]

$$t_B = \pm\sqrt{2n+1} \tag{31}$$

where n is the order of the Hermite function.

To account for any high-frequency components that a function might have, the bandwidth in the frequency domain must also be adequate. In Section II-D, it was noted that the Fourier transform of a Hermite function is also a Hermite function and it follows that the frequency bandwidth ω_B, is

$$\omega_B \le \sqrt{2n+1}. \tag{32}$$

The weakness of the Hermite/Laguerre network is that the Gaussian window of the correlation is relatively large compared to the spatial bandwidth of the Hermite network. For example, with 50 Hermite activation functions the effective spatial bandwidth of the network is only ± 10, whereas the width of the Gaussian window function is ± 3.

A method to bypass this weakness (increasing the spatial resolution) is to cascade a series of $m = 0, 1, 2, \ldots, M$ identical neural networks units along the data range by shifting the origin of each network (Fig. 6). In this way, an acceptably small window can be achieved. Each individual Hermite/Laguerre neural network unit now requires a bias β_m and an output-processing element-transfer function $\phi(t)$, given by

$$\beta_m = 10\ m \tag{33}$$

and

$$\phi(t) = \begin{cases} 1, & \text{for } -5.0 \le t \le 5.0 \\ 0, & \text{for } t > 5.0 \\ 0, & \text{for } t < -5.0. \end{cases} \tag{34}$$

The transfer function of the output-processing element restricts the output signal of each individual Hermite/Laguerre neural network unit so that it does not interfere with the adjacent units. The Hermite and Laguerre cascaded neural network outputs are now given as, respectively

$$x_h(t) = \sum_{m=0}^{M}\sum_{n=0}^{N} a_{mn}\phi(t-\beta_m)h_n(t-\beta_m) \tag{35}$$

and

$$y_h(t) = \begin{cases} \sum_{m=0}^{M}\sum_{n=0}^{N} a_{mn}\phi(t-\beta_m)l_0^n(t-\beta_m) & \text{for } t \ge \beta_m \\ \sum_{m=0}^{M}\sum_{n=0}^{N} a_{mn}\phi(t-\beta_m)(-1)^n l_0^n(t-\beta_m) & \text{for } t < \beta_m. \end{cases} \tag{36}$$

IV. Application to Filtering of Noisy Signals

In this section, we demonstrate the application of the network to the filtering of noisy sine waves with simulated noise. The sine wave signal used for testing was of unit amplitude and of period $T = 40$ seconds (s), and was sampled at a rate of $R = 10$ samples/s. Fig. 7(a) to (c) demonstrates the operation of the network. This figure shows the two outputs: the Hermite [Fig. 7(b)] and the filtered Laguerre network [Fig. 7(c)]. The cascaded network consisted of five identical Hermite/Laguerre networks, each having 50 Hermite and 50 Laguerre activation functions.

In order to apply the network to practical problems, it is useful to know the number of activation functions and the number of training cycles required. Fig. 8(a) shows the rmse of the Hermite and Laguerre networks as a function of the number of activation functions. As the number of activation functions is increased, the error of each output decreases and eventually becomes constant. This figure indicates that 30 activation functions are required to ensure that both the Hermite and Laguerre outputs have reached their optimum root mean square error.

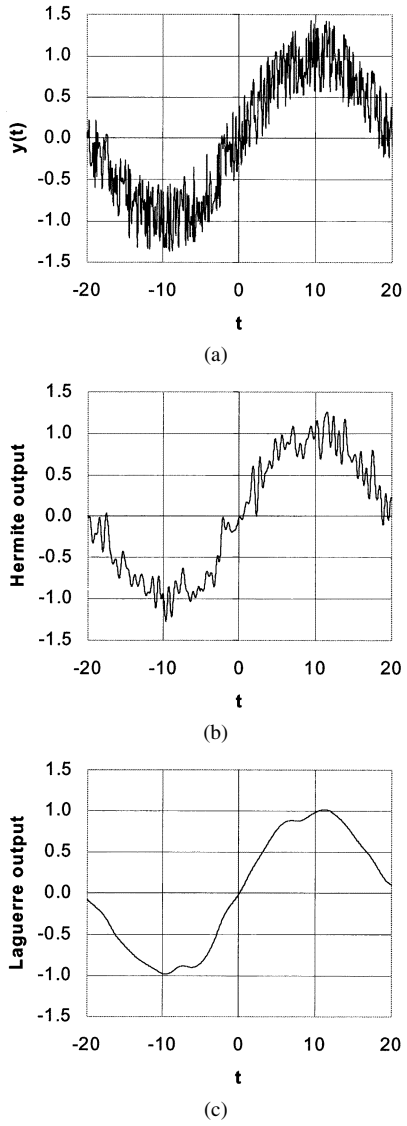


Fig. 7 (a) Noisy sine wave $\mathrm{period} = 40.0$; $\mathrm{SNR} = 7.2$. (b) Hermite output; $\mathrm{SNR} = 22.2$. (c) Laguerre output; $\mathrm{SNR} = 149.7$

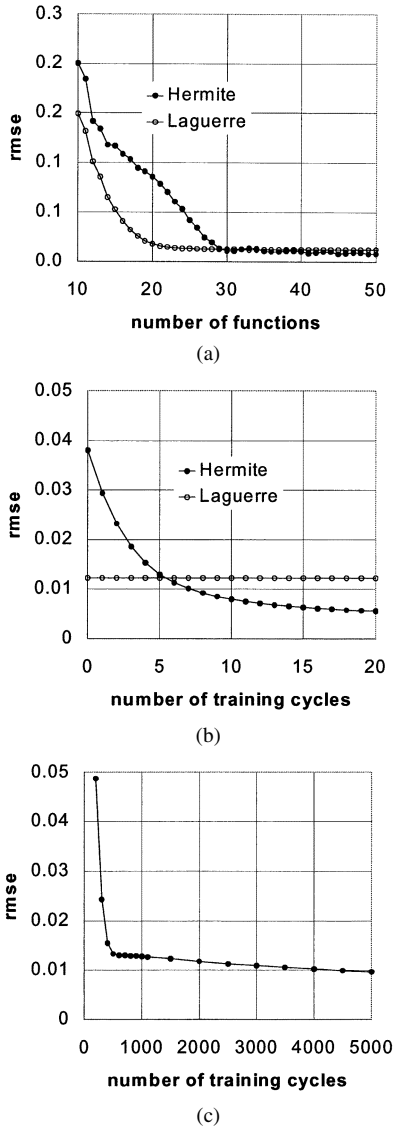


Fig. 8 (a) Root mean square error versus number of activation functions. (b) Hermite/Laguerre neural network training speed. (c) Sigmoid neural network training speed.

Fig. 8(b) shows the root mean square error of the Hermite and Laguerre networks versus the number of training cycles. This figure shows that more than ten training cycles are required to ensure that the weights have reached their optimum value.

The simulated random noise, which was generated artificially from a puesdo-random sequence, was of uniform density with a mean value of zero. To investigate the filtering properties of the Hermite/Laguerre network, the SNR was calculated on the sine wave at the input and output of the network. The input SNR ($\mathrm{SNR}_{\mathrm{in}}$) is defined as the ratio of the signal power without noise to the noise power at the input. The output SNR ($\mathrm{SNR}_{\mathrm{out}}$) is the ratio of the interpolated signal without noise to the noise power at the output.

A theoretical SNR can be derived from the bandwidth of the Hermite/Laguerre neural network. Since the simulated random noise is a form of white noise, it is uniformly distributed across the frequency domain. The noise power at the input and output is then proportional to the bandwidth at the input and output, respectively, [16] and it follows that

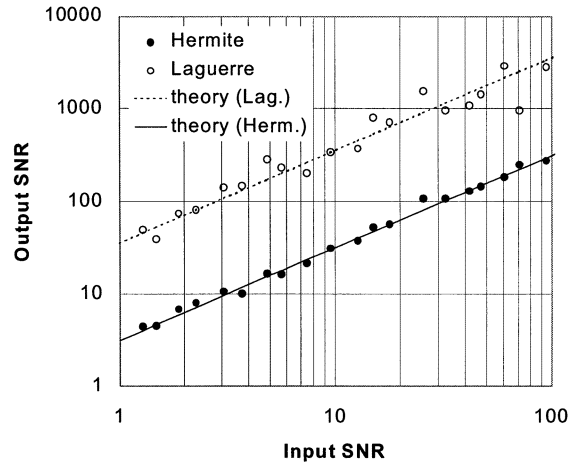
$$\frac{\mathrm{SNR}_{\mathrm{out}}}{\mathrm{SNR}_{\mathrm{in}}} = \frac{\omega_{\mathrm{in}}}{\omega_{\mathrm{out}}}. \tag{37}$$

At the input, the effective bandwidth of the noise is

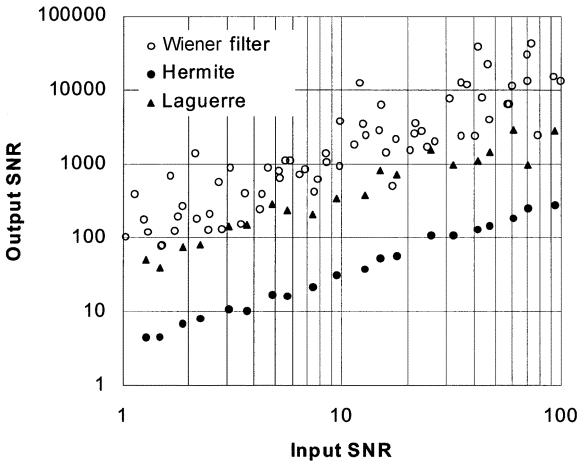
$$\omega_{\mathrm{in}} = \pi R. \tag{38}$$

At the output, the bandwidth is equal to that of the filter. Since the Laguerre output approximates a Gaussian filter, it has the same bandwidth as that of the Gaussian filter given by (19). The Hermite output has the bandwidth given by (32). Fig. 9(a) compares the SNR of the Laguerre and Hermite outputs for different input noise power. A reasonable fit to the theoretical SNR of (37) is achieved in both cases. The Hermite output deviates from a straight line to a lesser extent than the Laguerre output, probably due to the sharper cutoff of the Hermite network (Fig. 10). An improvement in SNR by a factor of ten is achieved with the Laguerre output.

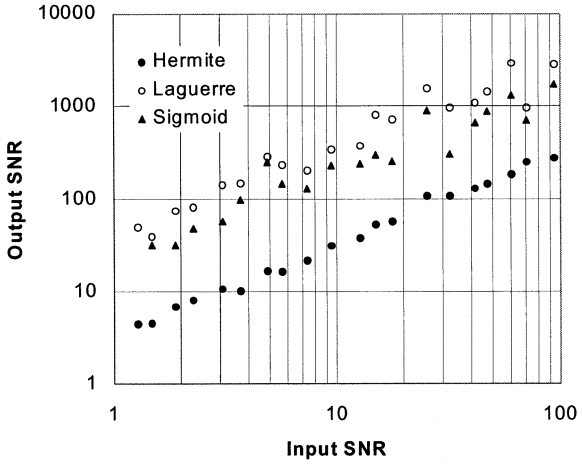
An upper limit to the filtering performance of the Laguerre output can be obtained by comparison with a Wiener filter [Fig. 9(b)]. The Wiener filter [17] is the theoretical optimum linear filter for a sine wave, which is achieved when the filter function is itself a sine wave. The Wiener filter shown in Fig. 9(b) consisted of a sine wave of the



(a)



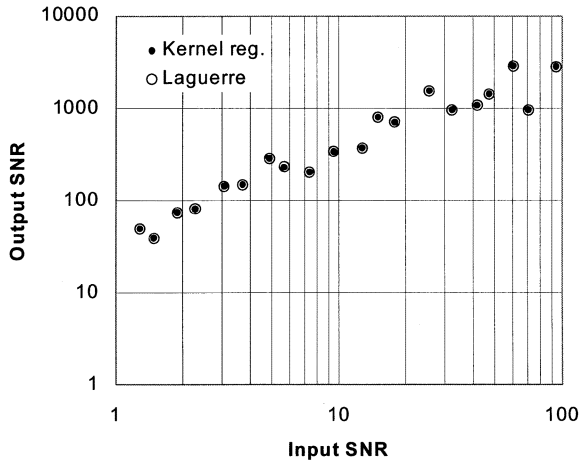
(b)



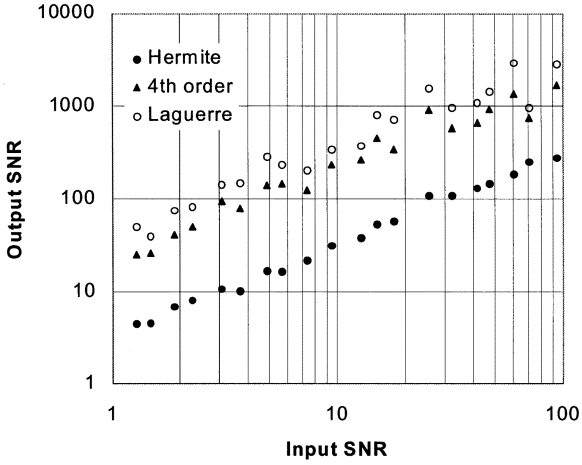
(c)

Fig. 9 (a) Laguerre and Hermite output SNR compared with theory. (b) Laguerre and Hermite output SNR compared with matched filter. (c) SNR of sigmoid neural network.

same frequency as the noisy sine wave, but with a phase and amplitude determined by numerical integration. The noise present is due to phase noise, amplitude noise, and numerical computational errors associated with the integration. Although the Wiener filter has a higher SNR than the Laguerre output, it requires the signal to be known prior to filtering. The results of Fig. 9(a) and (b) provide a quantitative measure of the SNR performance that can be expected using the Hermite/Laguerre neural network.



(d)



(e)

Fig. 9 *(Continued).* (d) SNR of kernel regression. (e) SNR of fourth-order Laguerre filter.

Fig. 9(c) shows the SNR obtained with a sigmoidal neural network compared to that of the Hermite/Laguerre network. The sigmoidal neural network consisted of a single layer of five sigmoid activation functions trained using the gradient descent algorithm. The Laguerre output achieved an improvement in SNR by a factor of 1.8, compared to the sigmoid neural network. Although the sigmoid neural network consisted of only five elements, it was considerably slower in training due to the larger number of training cycles required to achieve a satisfactory fit to the data. Fig. 8(c) shows the number of training cycles required by the sigmoidal neural network, which is of the order of 1000 cycles.

Fig. 9(d) shows the SNR obtained using the kernel regression of Section II-C compared to that of the Hermite/Laguerre network. The identical noise performance to the Laguerre output confirms the equivalence between each network.

Fig. 10 shows the frequency response of the Hermite output, the Laguerre output, and the fourth-order Laguerre filter of Section II-D. These curves were obtained by inputting a cosine wave of unit amplitude into the network and recording the output amplitude. The frequency bandwidth (3 dB) of the Hermite output was in agreement with the theoretical bandwidth given by (32). The bandwidth of the fourth-order Laguerre filter is almost double that of the Laguerre output and has a sharper cutoff. However, there is a reduction in the SNR [Fig. 9(e)] compared to the Laguerre output due to the wider bandwidth.

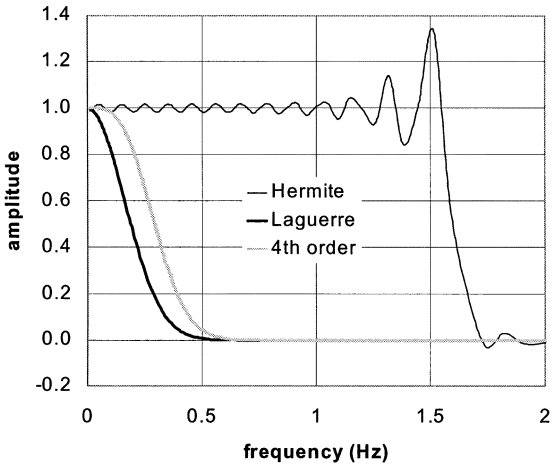


Fig. 10. Frequency response of Hermite/Laguerre neural network.

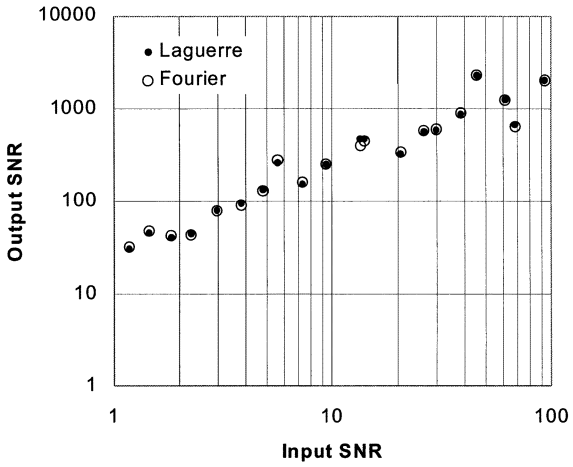


Fig. 12. SNR of Fourier correlator and Hermite/Laguerre correlator.

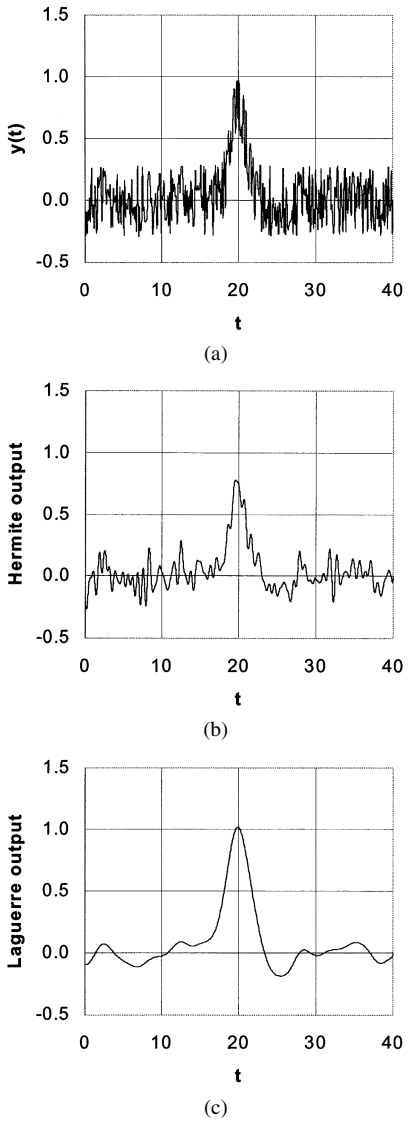


Fig. 11 (a) Noisy returned Gaussian pulse; SNR $= 0.92$. (b) Hermite output (fitted to noisy returned signal); SNR $= 2.8$. (c) Laguerre output (correlation of Hermite output and Gaussian function); SNR $= 11.8$.

V. Application as a Correlator Detector

In addition to functioning as a filter, the network can also be configured as a Gaussian correlator detector for application in range finding. Range finding is typically used in robotics where an ultrasonic transmitter and receiver are employed. The sensor operates in the same way as sonar; that is, the range of an object is calculated from the time of flight between a transmitted pulse and a received pulse. In this section, we apply the Hermite/Laguerre network to simulated noisy signals from an ultrasonic sensor.

Optimum SNR occurs when the noisy returned signal is cross correlated with the clean transmitted signal [18]. Assuming a Gaussian pulse shape for the transmitted signal, the Hermite/Laguerre network may be used as a correlator at the receiver of the sensor.

For application of the Hermite/Laguerre network as a correlator, the clean transmitted signal is represented by the window function $k(t) = \pi^{1/4} h_0(t)$; the noisy returned signal represents the input training function to the network $x(t)$ and the Laguerre output $y_h(t)$ represents the correlation of the transmitted and returned signal.

Fig. 11 shows the simulation of the Hermite/Laguerre correlator for an input SNR of 0.92 and signal duration of 40 s. The cascaded network consisted of four identical Hermite/Laguerre networks, each having 50 Hermite and 50 Laguerre activation functions. Fig. 11(a) is the returned pulse buried in noise, Fig. 11(b) is the Hermite output fitted to the noisy returned signal, and Fig. 11(c) is the Laguerre network output.

In order to assess the performance of the Hermite/Laguerre correlator, it was compared in simulation to a standard Fourier transform correlator [19]. The Fourier transform correlator consisted of 50 cosine and 50 sine functions with a fundamental period equal to 40 s. Fig. 12 shows the output SNR of the Hermite/Laguerre correlator compared to the Fourier correlator as a function of the input SNR. The Hermite/Laguerre correlator achieves an identical SNR to that of the Fourier correlator.

VI. Conclusion

A neural network has been developed for correlation with a Gaussian function using the Hermite orthonormal functions. The correlation of a Hermite function with a Gaussian function results in an associated Laguerre function. Consequently, by training a Hermite network to an input function, the correlation output is also immediately available via the Laguerre network. The spatial and frequency bandwidths of the network have been defined and a sufficient spatial bandwidth was achieved

by cascading the network. The network has been shown to have equivalent properties to kernel regression and may be interpreted as a specialized type of RBNN with training by Hermite orthonormal functions.

The filtering provided by the Laguerre output produces a factor-of-ten improvement in SNR compared to the Hermite output. When compared with a sigmoid neural network as a filter for noisy sine waves, it achieves a SNR that is greater by a factor of 1.8. Besides the Gaussian filter, a fourth-order filter was synthesized using a combination of two Hermite functions. This type of filter offers greater frequency response and a sharper cutoff than the Gaussian filter at the expense of a lower SNR.

Configured as a correlator detector, it has an equivalent SNR to that of a Fourier transform correlator.

REFERENCES

[1] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, 1992.

[2] S. Lawrence and C. L. Giles, "Overfitting and neural networks: conjugate gradient and backpropagation," in *IEEE Int. Conf. Neural Networks*, Como, Italy, 2000, pp. 114–119.

[3] H. J. Blinchikoff and A. I. Zverev, *Filtering in the Time and Frequency Domains*. New York: Wiley, 1976, ch. 3.

[4] E. Kreyszig, *Advanced Engineering Mathematics*, 6th ed. New York: Wiley, 1988, ch. 4, sec. 4.7.

[5] S. Osowski, "Neural network for estimation of harmonic components in a power system," in *Proc. Inst. Elect. Eng.—C*, vol. 139, 1992, pp. 129–135.

[6] S.-S. Yang and C.-S. Tseng, "An orthogonal neural network for function approximation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 779–784, Oct. 1996.

[7] T.-T. Lee and J.-T. Jeng, "The Chebyshev-polynomials-based unified model neural networks for function approximation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 925–935, Oct. 1998.

[8] G. G. Walter, *Wavelets and Other Orthogonal Systems With Applications*. Boca Raton, FL: CRC, 1994, p. 87.

[9] M. R. Mackenzie and A. K. Tieu, "Hermite neural network correlation," *IEEE Trans. Signal Processing*, vol. 51, pp. 3210–3219, Dec. 2003.

[10] G. G. Walter, *Wavelets and Other Orthogonal Systems With Applications*. Boca Raton, FL: CRC, 1994, p. 86.

[11] M. B. Priestley and M. T. Chao, "Non-parametric function fitting," *J. R. Stat. Soc. Series B*, vol. 34, pp. 385–392, 1972.

[12] G. G. Walter, *Wavelets and Other Orthogonal Systems With Applications*. Boca Raton, FL: CRC, 1994, p. 23.

[13] M. P. Wand and M. C. Jones, *Kernel Smoothing*. London, U.K.: Chapman & Hall, 1995, p. 32.

[14] E. Kreyszig, *Advanced Engineering Mathematics*, 6th ed. New York: Wiley, 1988, ch. 4, sec. 4.7, eq. 7.

[15] P. A. Lindsay, *Introduction to Quantum Mechanics for Electrical Engineers*. New York: Mc Graw-Hill, 1967, pp. 60–67.

[16] B. P. Lathi, *Modern Digital and Analog Communication Systems*. New York: Holt-Saunders, 1983, pp. 107–109.

[17] D. C. Champeney, *Fourier Transforms and Their Applications*. London, U.K.: Academic, 1975, pp. 131–135.

[18] ——, *Fourier Transforms and Their Applications*. London, U.K.: Academic, 1975, pp. 135–137.

[19] D. E. Newland, *Random Vibrations and Spectral Analysis*, 2nd ed. White Plains, NY: Longman, 1984, ch. 10.

Coupled Principal Component Analysis

Ralf Möller and Axel Könies

***Abstract*—A framework for a class of coupled principal component learning rules is presented. In coupled rules, eigenvectors and eigenvalues of a covariance matrix are simultaneously estimated in coupled equations. Coupled rules can mitigate the stability-speed problem affecting noncoupled learning rules, since the convergence speed in all eigendirections of the Jacobian becomes widely independent of the eigenvalues of the covariance matrix. A number of coupled learning rule systems for principal component analysis, two of them new, is derived by applying Newton's method to an information criterion. The relations to other systems of this class, the adaptive learning algorithm (ALA), the robust recursive least squares algorithm (RRLSA), and a rule with explicit renormalization of the weight vector length, are established.**

***Index Terms*—Adaptive learning algorithm, minor component analysis (MCA), neural networks, Newton's method, Oja's rule, principal component analysis (PCA), robust recursive least squares learning algorithm, speed-stability tradeoff.**

I. INTRODUCTION

PRINCIPAL component analysis (PCA) of high-dimensional data is an ingredient of many signal processing applications. PCA strives to extract the "principal" directions in the data space where the variance of the data is maximal, thus paving the way for dimension reduction and data compression. Other applications rely on minor component analysis (MCA) which finds the directions of minimal variance. Neural network approaches to PCA (MCA) pursue an "online" approach where an estimate of the principal (minor) directions is updated after each presentation of a data point. Therefore, these methods are especially suited for high-dimensional data, since the computation of the large covariance matrix can be avoided, and for the tracking of nonstationary data, where the covariance matrix slowly changes over time.

Many neural principal component analyzers have been suggested in the literature, with important initial contributions by Oja [1] and Sanger [2]; an overview can be found in [3]. The number of neural networks for minor component analysis is somewhat smaller; a recent review is given in [4]. Although the field has been active for two decades now, the attempts to improve the methods and to suggest new approaches and information criteria are continuing.

For a single PCA (MCA) neuron, the vector of synaptic weights is modified by a learning rule in a way that it converges toward that eigenvector of the covariance matrix of the data distribution which has the largest (smallest) corresponding eigenvalue; this eigenvector has the desired principal (minor) direction. With multiple PCA (MCA) neurons, the weight vectors should converge to the orthogonal set of eigenvectors of the covariance matrix with maximal (minimal) eigenvalues. Since generally only the direction of eigenvectors is defined but not their length, neural PCA/MCA rules often constrain the length of the weight vectors, some by converging toward a defined length (usually unit length [1] or a length related to the eigenvalue [5]), others by keeping the weight vector length constant over time [4].

Manuscript received September 26, 2002; revised April 1, 2003 and May 26, 2003.

R. Möller was with the Cognitive Robotics Group, Max Planck Institute for Psychological Reserach, Munich D-80799, Germany. He is now with the Computer Engineering Group, Faculty of Technology, Bielefeld University, Bielefeld D-33594, Germany (e-mail: moeller@techfak.uni-bielefeld.de).

A. Könies is with the Stellarator Theory Group, Max Planck Institute for Plasma Physics, D-17491 Greifswald, Germany.

Digital Object Identifier 10.1109/TNN.2003.820439