Nov. 27 - Dec. 1, 1995, pp. 289-294

A Fully Kalman-Trained Radial Basis Function Network for Nonlinear Speech Modeling

Martin Birgmeier

Institut für Nachrichtentechnik und Hochfrequenztechnik, Technische Universität Wien,
Gußhausstraße 25/E389, A-1040 Vienna, Austria

Abstract

This paper presents a radial basis function neural network which is trained to learn the dynamics of nonlinear autonomous systems. Contrary to conventional approaches, not only the output layer weights, but also the other parameters of the RBF network are trained using the extended Kalman filter algorithm. The advantages over conventional methods are that centers and variances of the hidden layer nodes need not be calculated before the optimum output weight matrix is determined, and that on-line training is possible. Due to a suitable factorization of the Riccati difference equation as contained in the Kalman filter, the algorithm can be implemented local to the nodes in the network, and a matrix inversion replaced by simple divisions, thereby significantly reducing the computational complexity. Finally, the network is applied to the task of learning the dynamics of speech signals obtained from sustained vowels, and subsequently used to re-synthesize these vowels autonomously.

1. Introduction

In the last few years, there has been increased interest in the application of neural networks as nonlinear predictors, particularly for the prediction of the time series produced by recursive nonlinear mappings, cf. e.g. Lapedes and Farber (1987), Lapedes and Farber (1988), Ris (1994), and Haykin and Li (1995), who use multilayer perceptrons and radial basis function networks for the prediction of various types of chaotic time series. Kadirkamanathan and Niranjan (1992) and Kadirkamanathan et al. (1992) use the principle of $\mathcal{F}$-projection to arrive at growing Gaussian radial basis function (RBF) networks, where the adaptation step is improved by using the extended Kalman filter in place of steepest descent (LMS) algorithm to estimate either the output weights or additionally the RBF centers.

In this paper, a radial basis function network is presented which has all of its parameters except the number of hidden nodes (i.e. centers, spreads, and output weights) trained via the extended Kalman filter algorithm. The advantages over conventional methods, which separate the learning of the hidden layer parameters from that of the output layer, are that centers and variances of the hidden layer nodes need not be calculated before the optimum output weight matrix is determined, and that on-line training is possible.

In order to test the performance of the training algorithm, and also to investigate the possible application of such networks for speech coding purposes, the network is then trained as a nonlinear predictor, where higher prediction gain than when using conventional linear methods like LPC analysis is expected (Wu et al., 1992).

By feeding predicted samples back into the input delay line which is used in the reconstruction of the attractor during training, the network functions as an autonomous, nonlinear, bounded system, whose attractor re-synthesizes the attractor associated with the training sequence (Casdagli, 1989; Vesin, 1993). For speech signals, Tishby (1990) reports that a neural network trained as a short-term predictor of voiced speech can generate waveforms similar to the original when let run in a closed loop in an autonomous fashion. Such a system may be used for speech compression and modification; for this purpose, Kubin and Kleijn (1994) use a nonparametric method for constructing a predictor codebook directly from the most recent speech segment. This clearly has the disadvantage of requiring relatively large storage and an exhaustive search procedure. Therefore it is desirable to construct a parameterized system which reproduces some desired sound as an attractor in state space.

Thus, the aim of the paper is twofold: First, derive the Kalman-based network training algorithm; second, evaluate the prediction capability of trained networks and use them to synthesize voiced phonemes.

2. The Extended Kalman Filter Learning Algorithm for Radial Basis Function Networks

In this section, the extended Kalman filter learning algorithm for the parameters of an RBF network, centers, covariance matrices, and output weights, is derived.

The following definitions are used (vector quantities are denoted by boldface letters):

D ..... input dimension (number of input nodes)
K ..... hidden (RBF) layer size
n ..... discrete time index
$\boldsymbol{x}(n)$ ..... input pattern at time n
$d(n)$ ..... desired response (training value) at time n
$y(n)$ ..... actual response at time n
$w_k(n)$ ..... output weight from hidden node k
$\boldsymbol{t}_k(n)$ ..... center of hidden node k
$\boldsymbol{\Sigma}_k(n)$ ..... covariance matrix of hidden node k
$s_k(n)$ ..... output of hidden node k
$\boldsymbol{R}_{\min} = \lambda \boldsymbol{I}$ ... measurement noise covariance matrix (assumed diagonal)
$\boldsymbol{C}(n)$ ..... measurement matrix at time n
$\boldsymbol{G}(n)$ ..... Kalman gain matrix at time n
$\boldsymbol{K}(n)$ ..... (posterior) error covariance matrix at time n
$()^T$ ..... matrix transposition
$\boldsymbol{I}$ ..... identity matrix of compatible size

The extended Kalman algorithm is used to estimate a state vector from measurements (cf. Anderson and Moore (1979)). In the most general case, the state vector consists of all the parameters of the RBF network which are to be learnt. However, this amounts to a state vector length of $l = DK + D^2K + K$, which — even for networks of only moderate size — would yield a prohibitively large state error covariance matrix. Therefore, the algorithm is decomposed in a way that it can be computed in smaller parts. The approach here follows the one described in Birgmeier (1994), which itself is similar to the MEKA method presented by Shah et al. (1992), or the algorithm described by Iiguni et al. (1992), all of which apply to multilayer feedforward neural networks.

Assuming that the optimum set of parameters $\boldsymbol{a}(n)$ is stationary, the Kalman algorithm can be formulated as follows (cf. Haykin (1986)):

$$\boldsymbol{a}(n) = \boldsymbol{a}(n-1) + \boldsymbol{G}(n)\left[d(n) - y(n)\right] \tag{1a}$$

$$\boldsymbol{G}(n) = \boldsymbol{K}(n-1)\boldsymbol{C}^T(n)\left[\boldsymbol{C}(n)\boldsymbol{K}(n-1)\boldsymbol{C}^T(n) + \boldsymbol{R}_{\min}\right]^{-1} \tag{1b}$$

$$\boldsymbol{K}(n) = \boldsymbol{K}(n-1) - \boldsymbol{G}(n)\boldsymbol{C}(n)\boldsymbol{K}(n-1) \tag{1c}$$

Its extension involves the computation of the Jacobian $\boldsymbol{C}(n)$, which is obtained as the linearization about the current value of the nonlinear function $y(\boldsymbol{w}(n), \boldsymbol{t}_1(n) \ldots \boldsymbol{t}_K(n), \boldsymbol{\Sigma}_1^{-1}(n) \ldots \boldsymbol{\Sigma}_K^{-1}(n))$ which defines the relationship between the state and output vectors at time n.

For a radial basis function network with one output only, and using Gaussian kernel functions, the following equation describes this relationship (cf. Haykin (1994)):

$$\begin{aligned} y(n) &= y(\boldsymbol{w}(n), \boldsymbol{t}_1(n) \ldots \boldsymbol{t}_K(n), \boldsymbol{\Sigma}_1^{-1}(n) \ldots \boldsymbol{\Sigma}_K^{-1}(n)) = \\ &= \sum_k w_k \exp\left(-\frac{1}{2}\left(\boldsymbol{x}(n) - \boldsymbol{t}_k(n)\right)^T \boldsymbol{\Sigma}_k^{-1}(n)\left(\boldsymbol{x}(n) - \boldsymbol{t}_k(n)\right)\right) \end{aligned} \tag{2}$$

The computation of the Jacobian $\boldsymbol{C}(n)$ requires the evaluation of the partial derivatives of y vs. the parameters of the RBF network:

$$\frac{\partial y}{\partial w_k} = s_k(\boldsymbol{x}) \tag{3a}$$

$$\frac{\partial y}{\partial \boldsymbol{t}_k} = w_k \frac{\partial}{\partial \boldsymbol{t}_k} s_k(\boldsymbol{x}) = w_k s_k(\boldsymbol{x}) \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{t}_k) \tag{3b}$$

$$\frac{\partial y}{\partial \boldsymbol{\Sigma}_k^{-1}} = w_k \frac{\partial}{\partial \boldsymbol{\Sigma}_k^{-1}} s_k(\boldsymbol{x}) = -\frac{1}{2} w_k s_k(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{t}_k)(\boldsymbol{x} - \boldsymbol{t}_k)^T \tag{3c}$$

The derivatives are evaluated at the current values of their respective arguments. This results in a measurement matrix $\boldsymbol{C}(n)$ of size $(DK + D^2K + K) \times 1$.

In order to make the problem computationally tractable, a localized version of the algorithm is developed. Puskorius and Feldkamp (1991) and Iiguni et al. (1992) have shown that when a feedforward network's weights are ordered by node, the matrix inversion of the Riccati difference equation can be avoided. This result can be generalized to any set of parameters where the Jacobian of the network's outputs with respect to its parameters can be factored to contain a common partial derivative term of the outputs versus a scalar variable. The state vector $\boldsymbol{a}$ is split into $K + 1$ components

$$\boldsymbol{a}_0 = \boldsymbol{w} \tag{4a}$$

$$\boldsymbol{a}_k = \begin{bmatrix} \boldsymbol{t}_k & \boldsymbol{\Sigma}_k^{-1} \end{bmatrix} \qquad 1 \le k \le K. \tag{4b}$$

Additionally the covariance matrices $\boldsymbol{\Sigma}_k$ are constrained to be equal to either $\sigma_k^2 \boldsymbol{I}$ or $\mathrm{diag}(\sigma_{k,1}^2 \ldots \sigma_{k,D}^2)$, such that the size of the $\boldsymbol{a}_k$ is only $D+1$ or $2D$, respectively, and the positiveness of the covariance matrix is guaranteed. The Kalman filter algorithm is then applied separately to each of the sub-problems defined

```
for n
    e(n) ← d(n) − y(n)
    for k from 1 to K
        At hidden node k:
        μ_k(n) ← w_k(n) s_k(n)
        ν_k(n) ← [σ^{-2}_{k,1...D}(x_{1...D}(n) − t_{k,1...D}(n)),   −σ^{-1}_{k,1...D}(x_{1...D}(n) − t_{k,1...D}(n))^2]
        ψ_k(n) ← K_k(n−1) ν_k(n)
        α_k(n) ← ν_k^T(n) ψ_k(n)
        β_k(n) ← μ_k^2
        a_k(n) ← a_k(n−1) + ψ_k(n) μ_k^T(n) 1/(λ + α_k(n)β_k(n)) e(n)
        K_k(n) ← K_k(n−1) − β_k(n)/(λ + α_k(n)β_k(n)) ψ_k(n) ψ_k^T(n)
    endfor
    At output node:
    ξ(n) ← K_0(n−1) s(n)
    G(n) ← ξ(n) 1/(s^T(n) ξ(n) + λ)
    w^T(n) ← w^T(n−1) + G(n) e(n)
    K_0(n) ← K_0(n−1) − G(n) ξ^T(n)
endfor
```

TABLE 1. *Pseudo-Code for Kalman Training of RBF Networks*

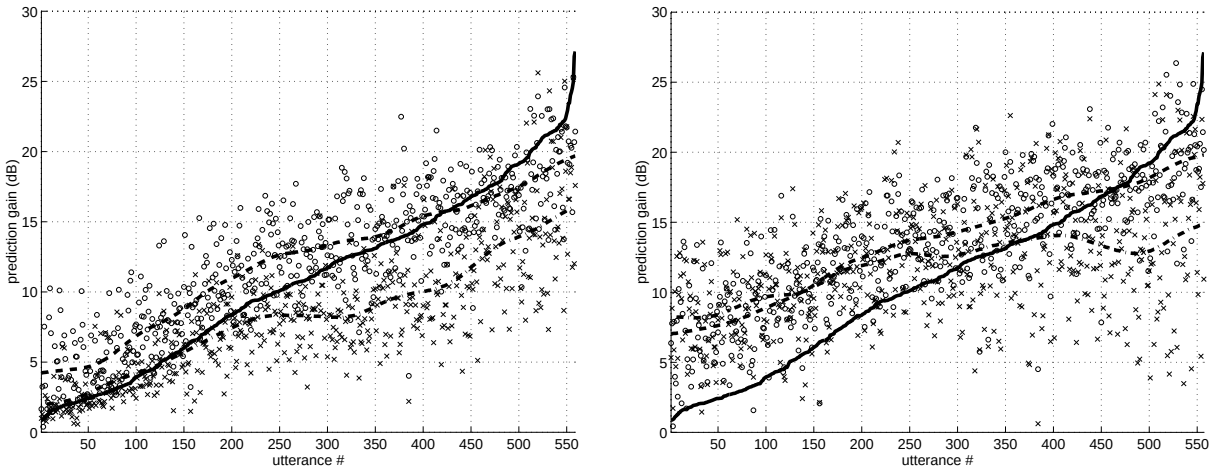

FIGURE 1. *Comparison of prediction results for three training algorithms: Solid line — linear least squares prediction; × — RBF network using standard training algorithm; ∘ — RBF network using extended Kalman filter training algorithm. About 560 German vowels are displayed along the x-axis, sorted according to their LLS prediction gain. Dash-dotted line: Smoothed prediction gain from RBF network using standard algorithm; dashed line: Smoothed prediction gain from RBF network using extended Kalman filter algorithm. (Left) 10 centers, (Right) 50 centers. The high variability in prediction gain results from the nonstationarity of the natural sounds.*

by equation (4), and the scalar variable mentioned above can then be identified as either an output of a hidden node or the global output. This yields the pseudo-code shown in table 1[1]. Note that the algorithm for updating $\boldsymbol{a}_0 = \boldsymbol{w}$ is just the normal Kalman (RLS) filter due to the fact that the output node is a linear combiner.

3. Learning the Dynamics of Nonlinear Autonomous Systems

In order to test the performance of the new learning method, the network is trained as a predictor. The training database consists of around 600 time series of sustained utterances of German vowels. The task is to predict sample $t+3$ from samples $t,\ t-3 \ldots t-21$ at 8 kHz sampling rate. Figure 1 shows results for the prediction performance using three methods: Linear least squares prediction to obtain a baseline reference; standard RBF training using LBG clustering (Gersho and Gray, 1992) for determination of hidden node centers and variances plus least squares solution for the output weights; and the new training

[1]The notation is somewhat similar to Iiguni et al. (1992), so that a comparison of the algorithms is possible.

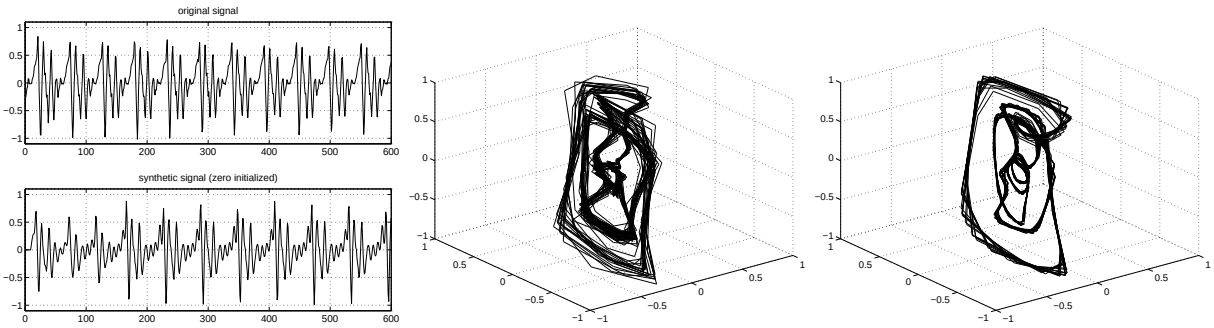

FIGURE 2. *Reconstructed waveforms and attractors for German vowel "a": Top & left — training data, bottom & right — synthesized. The delay taps used in the reconstruction, at 8 kHz are* $\boldsymbol{\tau} = [1 \ldots 9]$. *The first three delay taps are used in the display. The visible difference stems from the different degrees of periodicity of the two systems.*

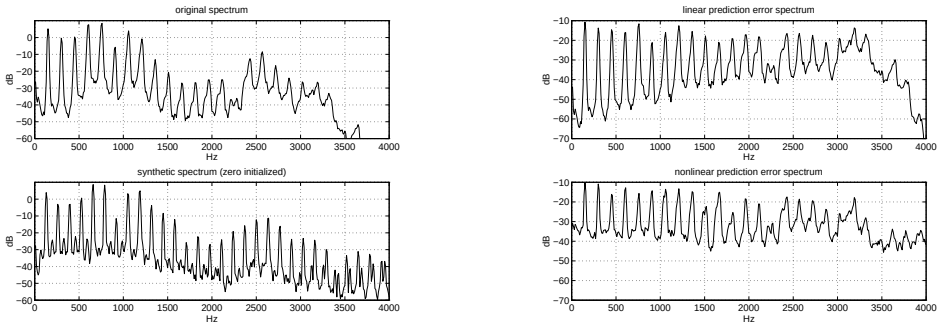

FIGURE 3. *(Left) Smoothed spectra of original and synthesized German vowel "a": Upper — training data, lower — synthesized. The synthesized version has the same formants, but a different pitch. (Right) Smoothed spectra of linear and nonlinear prediction errors.*

algorithm. It can be seen that the new algorithm produces results superior to either of the other two algorithms for low numbers of hidden nodes, and results comparable to the standard algorithm when there are many hidden nodes. In both cases, the Kalman-based algorithm is faster than the standard RBF training algorithm; the reason is that the clustering algorithm repeatedly needs to compute the activations of the hidden nodes, and that the least-squares solution of the output weights is more complex than the corresponding Kalman filter solution.

By Taken's theorem, an attractor may be reconstructed from a time-series by combining a sufficient number d of delayed samples of the observations into a d-dimensional embedding of the dynamics of the nonlinear system (Parker and Chua, 1987; Farmer and Sidorowich, 1988). Then, by feeding predicted values back into the input delay line, the resulting nonlinear autonomous dynamical system is expected to re-synthesize the original attractor.

Fig. 2 shows the original and the re-synthesized waveforms and attractors for an utterance of the German vowel "a". The data is sampled at a frequency of 8 kHz and scaled to the range $[-1.0 \ldots 1.0]$. The input dimension is $d = 9$, with the delay parameters $\tau_i = i/8$ kHz (equivalent to 125 μs). 20 centers are initialized in the hypercube $[-0.2 \ldots 0.2]^9$, with initial standard deviation set to 10.0. The training set consists of 2000 samples. For synthesis, the delay line is initialized with zeros (i.e. the origin belongs to the basin of attraction for this attractor). Note the very fast convergence of the synthesized vowel to its attractor, as seen in the lower left plot of fig. 2, where the initial zero sequence is visible.

The left side of fig. 3 shows a comparison of the (smoothed) spectra of the original and re-synthesized vowels. Note the appearance of subharmonics in the synthesized versions, indicating that the system (nearly) operates in the chaotic regime. The synthesized vowel has maxima in its spectrum corresponding to the formants of an "a", which is supported by the high perceptual quality when listening to it. However, the pitch period is somewhat larger in the synthesized version; this may be explained by the fact that the oscillator adds some additional "loops" to the attractor.

When used as a predictor, the network achieves about the same prediction gain as a global linear least squares approximation over a test sequence taken from the same utterance. (-14.5 dB for the nonlinear case vs. -14.6 dB for the linear). However, as also noted by Tishby (1990), the spectrum of the residual is "whiter" in the nonlinear case, as can be seen in the right part of fig. 3. In terms of the flatness measure

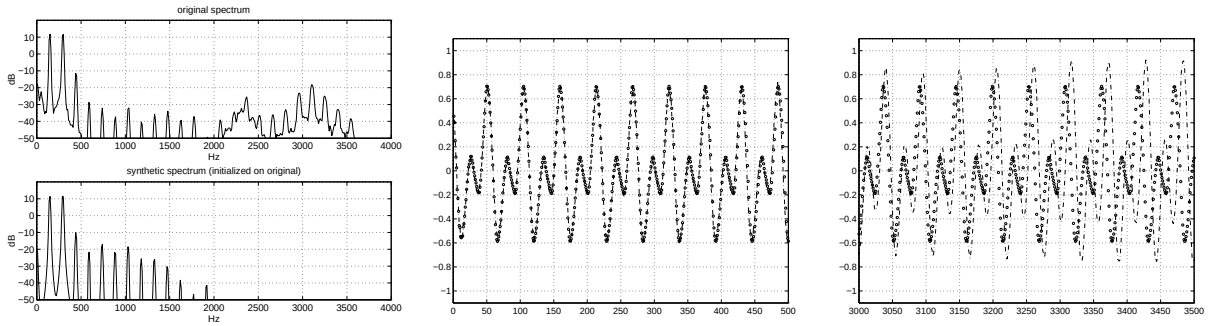

FIGURE 4. *German vowel "i": (Left) Smoothed spectra: Top — training data, bottom — synthesized. (Middle) Original (dashes) and synthesized (initialized from 9 samples of the original, circles) sequences, first 500 samples. (Right) Original and synthesized, samples 3000–3500.*

as defined in Markel and Gray (1976), linear prediction is inferior with a value of $6.8 \cdot 10^{-4}$ compared to nonlinear prediction with $3.3 \cdot 10^{-3}$ (the original signal yields a value of $1.6 \cdot 10^{-7}$).

An interesting case is the synthesis of the German vowel "i". The same sampling frequency, decimation factor, input dimension, initial centers, and number of training samples as before are used. On the other hand, the delay taps used are $\tau_i = (3i-2)/8$ kHz, which is chosen to correspond to the low first formant frequency of this vowel. From the left part of fig. 4, one can see that the pitch and low-frequency formant of this vowel give a perfect match between original and synthesized version. However, the second and third formants are missing. This can be explained by the network not capturing the fine structure of the attractor which is caused by these formants, since they have an energy more than 30 dB below that of the first formant.

On the other hand, looking at the middle and right parts of fig. 4, one can see that the original and free-running synthesized versions of the vowel exhibit *zero phase difference* for many periods of the signal[2]. In fact, only when the *amplitude* of the original changes (starting at about sample 3000), the speaker at the same time lowers his pitch frequency, which then results in a divergence between the two waveforms. This means that the long-term prediction error is small over many iterations of the system, which implies that the highest Lyapunov exponent of the underlying system must be very close to zero. This in turn indicates that the system is not chaotic, but rather exhibiting a stable limit cycle. The conclusion is that in this case, the classical model of voiced speech, namely spectrally shaping a pulse train, together with suitable generation of amplitude and pitch frequency, is adequate[3]. The nonlinear system only offers the advantage of producing a stable limit cycle of the desired shape, with a large basin of attraction, in an autonomous fashion.

One of the advantages of using the Kalman algorithm to determine the parameters of an RBF network is the smaller number of free parameters. Nonetheless, their initial settings still influence the approximation capability of the network, and best results are obtained when using the following values:

- RBF centers closely spaced around the origin, relative to the diameter of the attractor.
- RBF variances large relative to the diameter of the attractor.
- λ (corresponding to estimated measurement noise) about the diameter of the attractor.
- Initial state error covariances large (diagonal).

A network using these values works very well for time-series prediction, yielding results comparable to those found in the literature, even for small embedding dimension and a small number of hidden nodes. When running as an oscillator, hand-tuning of the initial values of the parameters is necessary to achieve optimal results.

4. CONCLUSIONS

This paper has shown that it is possible to train all the parameters of a radial basis function network using an efficient form of the extended Kalman filter algorithm. This obviates the need for determining centers and variances of the hidden layer nodes a priori, opening the possibility of training the network on-line.

The learning algorithm has been applied to the task of predicting the dynamics of nonlinear systems and shown to yield results better than or comparable to those found in the literature. The trained

[2] In this case, in order to achieve initial synchronization, the delay line is initialized on the original signal.

[3] Fricatives are best modeled as spectrally shaped random noise anyhow, cf. Kubin et al. (1993).

networks have been used to re-synthesize the original attractors by feeding their output back to the input delay line; however, this mode of operation, in contrast to simple nonlinear prediction, requires careful hand-tuning of the parameters of the training algorithm and network architecture in order to achieve satisfactory results.

It is argued that — at least for highly periodic utterances like the sustained vowel observed above — the underlying model is not chaotic since the long-term prediction error remains very small over many periods of the signal. On the other hand, the Kalman-trained RBF network is able to exhibit chaotic behavior, thereby reproducing the characteristics of natural voiced speech utterances. In both cases, the network converges to the attractor very fast (within less then three periods of the signal), providing an advantage over the conventional method of spectrally shaping a periodic excitation source.

Acknowledgments

The author wishes to thank Gernot Kubin for the many helpful and enlightening discussions with him during the work which lead to the presentation of this paper.

References

Anderson, B. D. and J. B. Moore, 1979. *Optimal Filtering*. Prentice-Hall.

Birgmeier, M., 1994. A Neural Network Trained with the Extended Kalman Algorithm Used for the Equalization of a Binary Communication Channel. In J. Vlontzos, J.-N. Hwang, and E. Wilson, eds., *Proc. 1994 IEEE NNSP IV*, pp. 527–534.

Casdagli, M., 1989. Nonlinear Prediction of Chaotic Time Series. *Physica D*, 35, 335–356.

Farmer, J. D. and J. J. Sidorowich, 1988. Exploiting Chaos to Predict the Future and Reduce Noise. Tech. Rep. LA-UR-88-901, Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545.

Gersho, A. and R. M. Gray, 1992. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers.

Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company.

Haykin, S. and X. B. Li, 1995. Detection of Signals in Chaos. *Proc. IEEE*, 83, no. 1, 95–122.

Haykin, S. S., 1986. *Adaptive Filter Theory*. Prentice-Hall.

Iiguni, Y., H. Sakai, and H. Tokumaru, 1992. A Real-Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter. *IEEE Tr. SP*, pp. 959–966.

Kadirkamanathan, V. and M. Niranjan, 1992. A Function Estimation Approach to Sequential Learning with Neural Networks. Tech. Rep. CUED/F-INFENG/TR.111, Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, England.

Kadirkamanathan, V., M. Niranjan, and F. Fallside, 1992. Models of Dynamic Complexity for Time-Series Prediction. In *ICASSP'92*, vol. 2, pp. 269–272.

Kubin, G., B. S. Atal, and W. B. Kleijn, 1993. Performance of Noise Excitation for Unvoiced Speech. In *Proceedings of the IEEE Workshop on Speech Coding for Telecommunications*, pp. 1–2, St. Jovite/Québec (Canada).

Kubin, G. and W. B. Kleijn, 1994. Time-Scale Modification of Speech Based on a Nonlinear Oscillator Model. In *ICASSP'94*, vol. I, pp. 453–456, Adelaide (Australia).

Lapedes, A. and R. Farber, 1987. Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling. Tech. Rep. LA-UR-87-2662, Los Alamos National Laboratory, Los Alamaos, New Mexico 87545.

Lapedes, A. and R. Farber, 1988. How Neural Nets Work. In Y. C. Lee, ed., *Evolution, Learning and Cognition*, pp. 331–346. World Scientific, Singapore; Los Alamos, NM 87545.

Markel, J. D. and A. H. Gray, 1976. *Linear Prediction of Speech*. Springer-Verlag, Berlin.

Parker, T. S. and L. O. Chua, 1987. Chaos: A Tutorial for Engineers. *Proc. IEEE*, 75, no. 8, 982–1008.

Puskorius, G. V. and L. A. Feldkamp, 1991. Decoupled Extended Kalman Filter Training of Feedforward Layered Networks. In *IJCNN'91*, vol. I, pp. 771–777.

Ris, C., 1994. Order Estimation and Nonlinear Prediction with Radial Basis Functions. In *EUSIPCO'94*, vol. III, pp. 1492–1495. European Association for Signal Processing.

Shah, S., F. Palmieri, and M. Datum, 1992. Optimal Filtering Algorithms for Fast Learning in Feedforward Neural Networks. *Neur. Netw.*, 5, 779–787.

Tishby, N., 1990. A Dynamical Systems Approach to Speech Processing. In *ICASSP'90*, pp. 365–368, Albuquerque, NM.

Vesin, J. M., 1993. Local Models for Nonlinear Signal Processing. In D. Docampo and A. R. Figueiras, eds., *Adaptive methods and emergent techniques for signal processing and communications*, pp. 384–390. Universidad de Vigo, Vigo, Spain. Based on the proceedings of COST 229 action WG 1 and 2 workshop.

Wu, L., M. Niranjan, and F. Fallside, 1992. Fully Vector-Quantized Neural Network-Based Code-excited Nonlinear Predictive Speech Coding. Tech. Rep. CUED/F-INFENG/TR.94, Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, U.K. submitted to IEEE Trans. SAP.