NONLINEAR PREDICTION OF SPEECH SIGNALS USING RADIAL BASIS FUNCTION NETWORKS

Martin Birgmeier

Institut für Nachrichtentechnik und Hochfrequenztechnik, Technische Universität Wien
Gußhausstraße 25/E389, 1040 Vienna, Austria;
phone: (+43 1) 58801x3661; fax: (+43 1) 5870583;
e-mail: Martin.Birgmeier@nt.tuwien.ac.at

Abstract

In this paper, we compare the capabilities of various forms of radial basis function networks as nonlinear short-term predictors for speech signals representing sustained utterances of German vowels. We use RBF and RBF-AR[1] network architectures, trained using a standard algorithm or alternatively the extended Kalman filter (EKF) algorithm, and linear least squares predictors. We also look at cascaded forms of linear/nonlinear predictors. We evaluate both prediction gain and spectral flatness measure of the residual. The results indicate: The RBF-AR structure is the most powerful, EKF training yields better results than standard training for RBF networks, and a non-cascaded RBF-AR predictor produces results superior to cascaded predictors.

1 Introduction

This paper presents a large-scale study of the capabilities of various forms of radial basis function (RBF) neural networks as nonlinear short-term predictors for signals obtained from sustained utterances of German vowels. Nonlinear prediction of speech is interesting for several reasons, amongst them:

- (Voiced) speech, as is the case with any *stable* oscillation, can only be produced by nonlinear systems [5, 9]. Hence, nonlinear prediction comes as a natural means for modeling the resultant time sequences.
- Improved prediction of time sequences of speech carries the potential for a further reduction of the bit rate required in voice transmission systems.

In this work, we compare the following different forms and training algorithms for RBF networks:

- Standard RBF, standard training algorithm [7];
- Standard RBF, extended Kalman filter (EKF) training algorithm [2];
- RBF-AR as in [12], standard training algorithm;
- as a baseline reference, linear least squares prediction.

The signals are taken from a database containing 700 sustained utterances of German vowels, spoken by 89 speakers, 14 of these are female [1]. The signals are sampled at a rate of 48 kHz with 16 bit linear resolution, and have an average duration of 1.5 s. We have chosen this database for two reasons:

- It consists of well-defined stationary speech signal segments such that the inherent non-stationarity of continuous speech does not obscure the comparison of the nonlinear prediction methods.
- We expect that stationary voiced speech segments exhibit the largest margin of nonlinear prediction over linear prediction, which several previous studies have observed as well [10, 11].

We use all samples which are long enough for extraction of training and test sequences (longer than 625 ms), such that about 600 utterances remain for processing. These are decimated by a factor of six to yield signals at an 8 kHz sampling rate.

The study evaluates the capabilities of the aforementioned RBF network architectures with respect to

- number of nodes in the networks' hidden layer: Ten or alternatively fifty, in order to determine the dependence on number of free parameters;
- different test sequences: The test sequence is either identical to the training sequence, or taken from a different segment of the same utterance (with a gap of 500 samples). Using a different test sequence enables us to evaluate the generalization performance of the predictor. This is important for coding applications where the predictor does not operate on the original sequence, e.g., backwards adaptive coding. In forward coding, we are more interested in having equal training and test sequences, and in this case the nonlinear predictor achieves an even higher prediction gain. In any case, both test and training sequences consist of 2000 samples (0.25 s);
- different embedding dimension d and reconstruction delay (lag) τ, i.e., different taps used for the construction of the delay vector which serves as input to the nonlinear predictor;
- direct or cascade form of the nonlinear predictor, where in the cascade form a linear predictor is followed by a nonlinear one.

[1] "AR" stands for auto-regressive in [12].

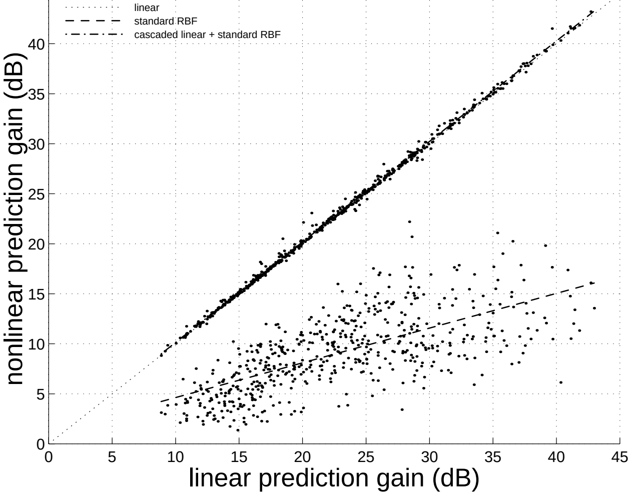

Figure 1: *Prediction gain for RBF network trained using standard training algorithm, without and with cascaded linear predictor. $d = 24$, $\tau = 1$, 10 centers, test seq. $\neq$ training seq. Dots are used to indicated actual linear/nonlinear prediction gain pairs, straight lines are fitted using linear regression.*

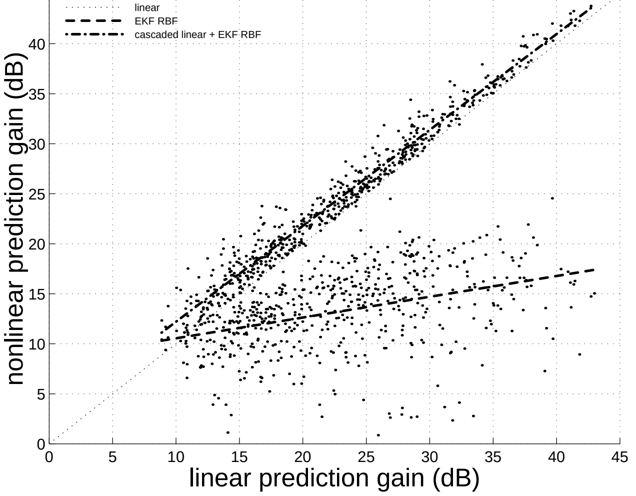

Figure 2: *Prediction gain for RBF network trained using EKF training algorithm, without and with cascaded linear predictor. Other parameters as in fig. 1.*

2 Nonlinear Prediction by RBF Networks

We use two different types of RBF networks in our simulations; the standard one is described by [7]

$$o(\boldsymbol{i}) = \sum_{k=1}^{K} w_k \exp\left(-\frac{\|(\boldsymbol{i} - \boldsymbol{t}_k)\|^2}{2\sigma_k^2}\right) \tag{1}$$

and implements a *local constant* mapping of the input vector $\boldsymbol{i}$ to the scalar output o. The RBF-AR network is described by [12]

$$o(\boldsymbol{i}) = \sum_{d=0}^{D} w_{0,d} i_d + \sum_{k=1}^{K} \sum_{d=0}^{D} w_{k,d} i_d \exp\left(-\frac{\|(\boldsymbol{i} - \boldsymbol{t}_k)\|^2}{2\sigma_k^2}\right) \tag{2}$$

and implements a *local affine* mapping by defining $i_0 \equiv 1$ ($i_1 \dots i_k$ are the components of the input vector).

We train both structures using the standard two-step algorithm: We first find the hidden nodes' centers $\boldsymbol{t}_k$ using the generalized Lloyd clustering algorithm [3, pp. 362 ff.]; from this, we get an average distortion for each center, which we scale by an empirically found constant factor to yield the hidden nodes' covariances σ_k; finally, we solve for the output weights w_k or $w_{k,d}$ in a least squares sense.

In addition, for the RBF network, we also use the extended Kalman filter (EKF) to simultaneously train the hidden nodes' centers and variances, plus output weights [2].

We use a rather large predictor memory which covers 24 samples of the signal, both for linear and nonlinear predictors. We do this to ensure that especially the linear predictor used in the cascaded form is able to exploit linear relationships over a longer time window, or alternatively, to model more than the usually assumed 3–5 poles of an AR model. Note that there is a difference between predictor memory and predictor order: Only if successive samples are used to construct the input vector, both are the same. Using samples spaced apart by some longer delay, we may still have the same memory, but a lower predictor order.

For all simulations, we evaluate both the prediction gain and residual spectral flatness measure (SFM) [6]. Regarding the latter, we know that already linear prediction (using infinite memory) produces a white residual, indicated by a SFM of one (0 dB). For a short-time memory, this is only approximated, with lower values for the SFM. Nonlinear least-squares prediction using infinite memory produces a martingale difference (MD) residual, which may be seen as an intermediate step between a white process and a sequence of independent variables (strictly white process) [8]. Clearly, the SFM does not differentiate between white and MD residuals; however, we expect that higher values for the SFM of the residual obtained by nonlinear prediction, using only a short-time memory, provides another measure for better performance of nonlinear over linear prediction.

3 Simulation Results

Figs. 1 to 3 show the prediction gains obtained using the three training methods described in the previous section, each without and with a cascaded linear predictor. Combined results for the SFMs obtained using these six methods are displayed in fig. 4. Linear regression is used to display average dependencies. The individual results are indicated by dots and exhibit large variations around the mean values. From this, we immediately see that there is no simple relationship between linear and nonlinear prediction gains. (The same holds true between the original signal's and the residual's SFM, although

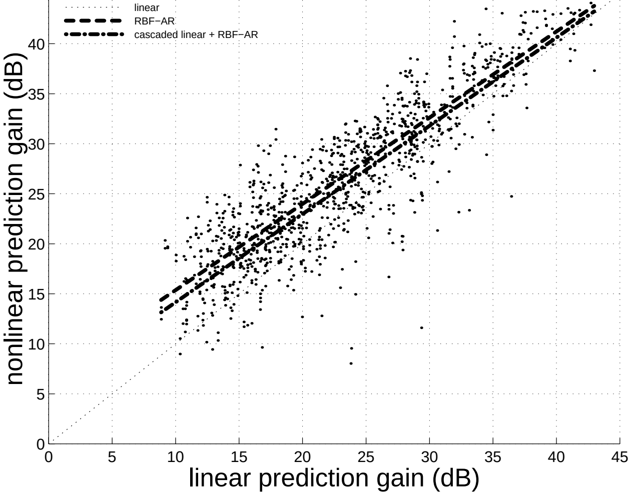

FIGURE 3: *Prediction gain for RBF-AR network trained using standard training algorithm, without and with cascaded linear predictor. Other parameters as in fig. 1.*

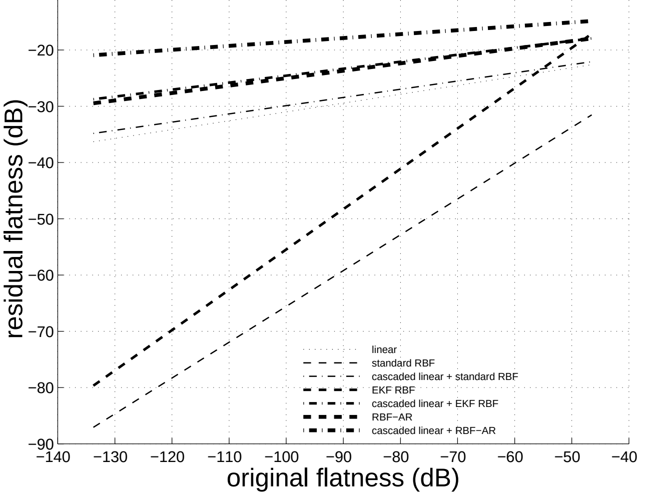

FIGURE 4: *Spectral flatness measure for six different methods. $d = 24$, $\tau = 1$, 10 centers, test seq. $\neq$ training seq.*

individual values are not shown in fig. 4.)

A linear predictor of order 24, operating on consecutive samples, provides a baseline reference for "optimum" short-term prediction. In the non-cascaded forms, only the RBF-AR model outperforms linear prediction with respect to prediction gain and SFM. We explain this by the fact that there is a large linear component in the mapping which cannot be modeled adequately by standard sum-of-Gaussians approaches with only a few centers, but which is well captured by the affine mapping output layer of the RBF-AR network. Hence, we only expect the standard RBF network to provide any improvement if it operates as the second stage of a cascaded predictor.

On the other hand, for the RBF-AR network, the cascaded form performs worse than the direct form (cf. fig. 3). Here, the linear predictor produces a residual whose attractor is essentially concentrated around the origin; whereas the original attractor is "unfolded" in state space, this is not true for the residual. In [11], this results in an apparent increase in the dimensionality of the residual. The net result is that we cannot fully exploit the local affine fitting characteristic of the RBF-AR network any more. Interestingly, from fig. 4 we see that the cascaded RBF-AR predictor has a higher SFM than the non-cascaded form. We currently do not have an explanation for this fact.

We also see that EKF-trained RBF networks have higher prediction gain and SFM than those trained by the standard method. Regarding SFM, the cascaded EKF-trained RBF predictor achieves the same spectral flattening as the non-cascaded RBF-AR network, despite its much smaller number of parameters; it covers about half of the SFM difference between a linear predictor and a cascaded RBF-AR predictor.

In general, for different test and training sequences (even though they are taken from the same utterance), we have an additional prediction gain over linear prediction of about 3–6 dB. This compares quite well with results reported in [10, 11, 4], where average improvements due to nonlinear prediction are reported to be around 3 dB. The results in [10, 11] are obtained using a nonlinear predictor not specifically adapted to a single vowel, hence the lower gain improvement.

For the case where training and test sequences are identical, we achieve an improvement in prediction gain of between 10–15 dB over linear prediction when using the RBF-AR network (cf. fig. 5). This may indicate that the network is already overtrained; in fact, it has $(24 + 1) \times (10 + 1) = 275$ output weights, not counting the hidden nodes' centers and covariances, compared to only 2000 training samples. However, when increasing the number of training and test samples to 4000, we find that on average, the prediction gain is reduced evenly by only 1.6 dB compared to the values given in fig. 5 for the RBF-AR network. Hence, the major cause of degradation for differing training and test sequences lies in the instationary nature of the signals. This indicates that nonlinear prediction may be especially suitable for forward adaptive coding purposes, where large gains are possible since both coder and decoder operate on the same data.

For an embedding dimension of only eight with a lag of $\tau = 3$, i.e., the same total length of memory but larger lag, the nonlinear predictors incur a smaller loss in prediction gain than the linear predictor (cf. fig. 6). In this case, the nonlinear component apparently becomes more pronounced, thereby impairing the effectiveness of linear prediction. This is explained by the low dimensionality of the signal: By Takens' embedding theorem, we only need a reconstruction dimension which is larger than twice the topological dimension of the attractor;

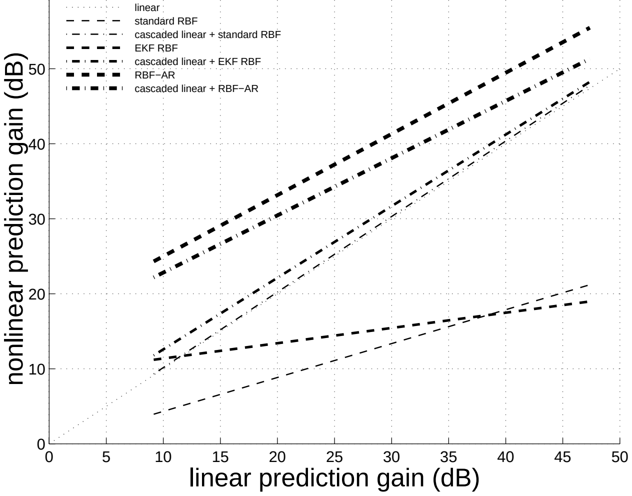

FIGURE 5: *Prediction gains for six different methods, test seq. = training seq. $d = 24$, $\tau = 1$, 10 centers.*

hence, eight taps suffice for an attractor dimension of less than four.

Using fifty instead of ten centers improves prediction gains by an additional 0.5–5 dB. However, for a high number of centers coupled with a large embedding dimension $d = 24$, the RBF-AR network cannot be used, both since its number of parameters exceeds the number of training samples, and since their computation becomes infeasible.

4 Conclusions

The main result is: With adequate modeling, rather high additional prediction gains compared to those achieved by linear prediction can be obtained by RBF networks. For sustained utterances of vowels, the difference is between 10–15 dB if test and training sequences are identical, 3–6 dB otherwise. Best results are obtained using the RBF-AR network structure, which however entails a much larger set of parameters and therefore higher complexity than standard RBF structures. For the latter, EKF training yields better results than standard training procedures. Also, with the RBF-AR structure, cascading linear and nonlinear predictors produces worse results than using the nonlinear predictor alone. On the other hand, for the standard RBF structure, using a cascaded form is advantageous, and the EKF-trained nonlinear RBF predictor attains about half of the additional prediction gain offered by the much more complex RBF-AR structure.

We are currently investigating the performance of nonlinear long-term predictors. There, the advantages over linear prediction become even more apparent, in that it is possible to exceed the prediction gain of long-memory linear long-term predictors by short-memory nonlinear long-term predictors.

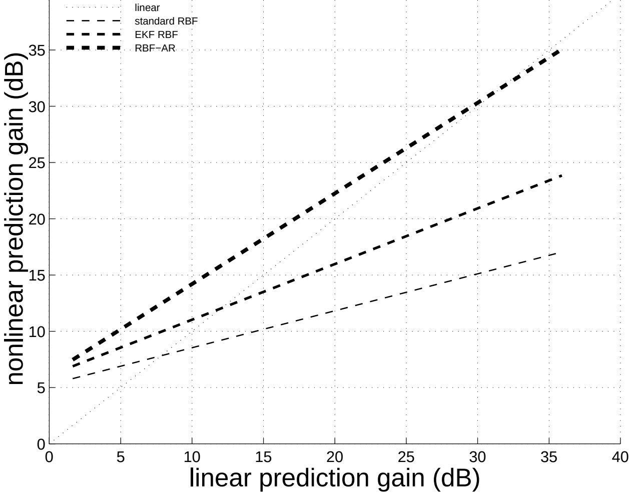

FIGURE 6: *Prediction gain for standard RBF, EKF RBF, and RBF-AR network structures/training methods, changed embedding. $d = 8$, $\tau = 3$, 10 centers, test seq. $\neq$ training seq.*

Acknowledgments

The author wishes to thank Gernot Kubin for his helpful comments during the course of preparing this paper.

References

[1] H.-P. Bernhard. Analyse von Sprachsignalen mit Methoden der Chaostheorie. *Elektrotechnik und Informationstechnik*, 111(12):648–649, December 1994.

[2] M. Birgmeier. A fully Kalman-trained radial basis function network for nonlinear speech modeling. *Proc. IEEE ICNN*, vol. 1, 259–264, 1995.

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

[4] S. Haykin and L. Li. Nonlinear adaptive prediction of nonstationary signals. *IEEE Tr. SP*, 42(2):526–535, February 1995.

[5] G. Kubin. Nonlinear processing of speech. *Speech Coding and Synthesis*, chapter 16. Elsevier Science B. V., 1995.

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

[7] J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neur. Comp.*, 1(2):281–294, 1989.

[8] A. Segall. Stochastic processes in estimation theory. *IEEE Tr. IT*, 22(3):275–286, May 1976.

[9] J. Thyssen, H. Nielsen, and S. D. Hansen. Nonlinearities in speech. *IEEE Workshop on Nonlinear Signal and Image Processing*, vol. II, 662–665, Halkidiki, Greece, 1995.

[10] N. Tishby. A dynamical systems approach to speech processing. *ICASSP'90*, 365–368, Albuquerque, NM, April 1990.

[11] B. Townshend. Nonlinear prediction of speech signals. *Nonlinear Modeling and Forecasting*, vol. XII, 433–453. Addison-Wesley.

[12] J. M. Vesin. Local models for nonlinear signal processing. *Adaptive methods and emergent techniques for signal processing and communications*, pages 384–390. Universidad de Vigo, Spain, June 1993.