Nonlinear Prediction of Speech by Echo State Networks (EURASIP Best Student Paper Award)

Conference Paper · September 2018

3 authors:

Ziyue Zhao
Technische Universität Braunschweig

Huijun Liu

Tim Fingscheidt
Technische Universität Braunschweig

Nonlinear Prediction of Speech by Echo State Networks

Ziyue Zhao, Huijun Liu, Tim Fingscheidt
Institute for Communications Technology
Technische Universität Braunschweig
Schleinitzstr. 22, 38106 Braunschweig, Germany
Email: {ziyue.zhao, h.liu, t.fingscheidt}@tu-bs.de

***Abstract*—Speech prediction plays a key role in many speech signal processing and speech communication methods. While *linear* prediction of speech is well-studied, *nonlinear* speech prediction increasingly receives interest especially with the vast amount of new neural network topologies proposed recently. In this paper, nonlinear speech prediction is conducted by a special kind of recurrent neural network not requiring any training beforehand, the echo state network, which adaptively updates its output layer weights. Simulations show its superior performance compared to other well-known prediction approaches in terms of the prediction gain, exceeding all baselines in all conditions by up to 8 dB.**

I. Introduction

Speech prediction is a means of using some or all past speech samples to predict the present sample or frame under some optimality criterion, often closely related to a model of speech production. Speech prediction is widely used in speech coding approaches [1], employing classical linear predictive coding (LPC) [2], adaptive differential pulse code modulation (ADPCM) [3], or code-excited linear prediction (CELP) [4], [5]. Many of the standard speech codecs are based on the above approaches. LPC is also used in robust speech and audio decoding [6], [7], artificial speech bandwidth extension [8], and model-based noise reduction [9]–[12]. Furthermore, an adaptive speech predictor is also applied in acoustic echo cancellation to whiten the virtual loudspeaker-enclosure-microphone (LEM) system excitation signal [13].

Using either linear combinations or some nonlinear functions of the observations to serve as the prediction input, the prediction approaches are accordingly defined as linear prediction or nonlinear prediction [14]. For linear prediction of speech, a sample-wise or frame-wise prediction can be applied, the latter resulting in fixed predictor weights within an analysis frame, assuming the speech signal to be short-time stationary [3]. The well-known Levinson-Durbin (LD) recursion [15], [16], solving the linear prediction problem with a Toeplitz matrix being involved, is used here to calculate the linear predictive (LP) coefficients. Instead of sharing the same predictor weights within a frame, sample-by-sample linear prediction algorithms adaptively update the predictor weights under some optimality criterion, being a classical form of adaptive filtering [14]. The least-mean-square (LMS) adaptive algorithm updates the filter weights to minimize the mean squared error, while the normalized least-mean-square (NLMS) [17] normalizes the filter weight update to avoid that the gradient depends on the energy of the input. Furthermore, the recursive least-squares (RLS) algorithm achieves a higher convergence speed, which is typically an order of magnitude faster than that of the LMS algorithm, at the expense of increased computational complexity [14].

Nonlinear speech prediction has received increasing attention during the past decades [18]–[20], since the production of the speech signal is actually a nonlinear and nonstationary process [21]. Accordingly, nonlinear adaptive prediction is expected to be more powerful than the aforementioned linear adaptive filtering approaches. Neural networks have been proven to be an effective way to introduce nonlinearity into signal prediction. Feedforward neural networks (NNs) have been applied to the speech prediction task as a non-adaptive nonlinear predictor [22], with the weights of the neural networks being learned from training data by backpropagation and then fixed, which is of course not very suitable for the prediction of nonstationary speech signals. In order to exploit the context of the speech, recurrent neural networks (RNNs) are used for speech prediction [23], where the internal memory is introduced by the recurrent topology. Several RNN topologies have been applied to speech prediction: Pipelined recurrent neural networks [19], [24], recurrent fuzzy neural networks [25], and their combinations [26]. However, these RNNs need to continuously update their neuron weights by using backpropagation through time (BPTT) [27] or real-time recurrent learning (RTRL) [28], which suffers from the gradient vanishing or exploding problem [29]. To solve this problem during training, RNNs with gating techniques, e.g., long short-term memory (LSTM) [29] and gated recurrent units (GRUs) [30], have been introduced.

Echo state networks (ESNs) [31], as a special kind of RNN, differ from the above topologies especially in terms of the weights updating. As can be seen in Figure 1, the weights $\mathbf{W}$ of RNN neurons in the so-called *reservoir* of ESNs remain fixed and only the output layer weights $\mathbf{w}_{\text{out}}$ need to be adaptively updated, which is actually only a simple linear regression task [32]. Because of its light computational load for weight updating, an ESN can be used in an adaptive way to predict speech and does not need to be trained beforehand, which is different from the abovementioned RNN topologies. So far, not much research work has been reported about

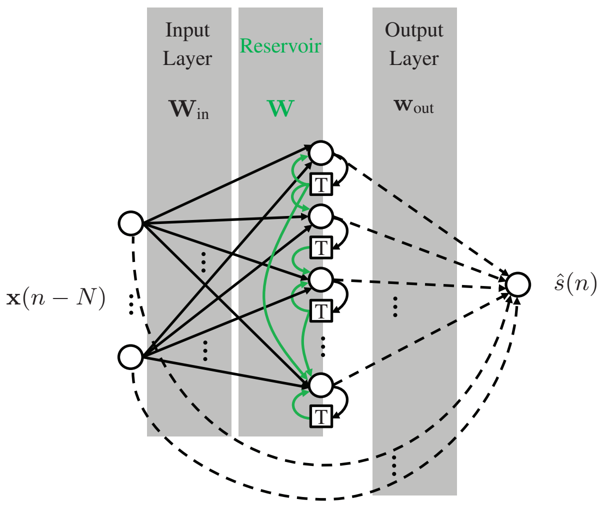

Fig. 1. Topology of the ESN for speech prediction (with direct connections between the input and the output layers). Solid lines and dashed lines denote the fixed random weights and adaptive weights, respectively. For an intuitive viewing the green solid lines are the elements of $\mathbf{W}$, which are randomly and sparsely connected among the neurons inside the reservoir.

the application of ESNs for speech prediction, although they possess suitable properties for this very task. In this paper we accomplish nonlinear adaptive speech prediction by an ESN and compare the prediction performance to various other linear and nonlinear prediction approaches.

This paper is structured as follows: In Section II, two baseline linear adaptive prediction algorithms are briefly reviewed, namely NLMS and RLS. Section III describes the speech prediction by the ESN, with some relation to RLS. Section IV presents the evaluation results and the discussion. Finally, some conclusions are drawn in Section V.

II. Baselines

In this section, two baseline adaptive linear prediction algorithms will be briefly reviewed, serving as baselines later on, and also easing understanding of ESNs in Section III. Concerning notations, $s(n)$ denotes the speech signal, with $n \in \mathbb{N}_0$ being the speech sample index. Then, for an N-step-ahead prediction on the basis of a number of N_p old samples, the input vector is denoted as

$$\mathbf{x}(n{-}N) = [s(n{-}N), s(n{-}N{-}1), \cdots, s(n{-}N{-}N_p{+}1)]^{\mathsf{T}}, \quad (1)$$

with N being the sample index units of the prediction distance, and $[\ ]^{\mathsf{T}}$ being the transpose. Moreover, the weight vector of the predictor is $\mathbf{w}(n) = [w_0(n), w_1(n), \cdots, w_{N_p-1}(n)]^{\mathsf{T}}$, the output sample of the predictor (prediction) is $\hat{s}(n)$, and the present sample to be predicted is $s(n)$.

A. Speech Prediction by NLMS

The cost function of NLMS can be written as

$$J(n) = \big(\hat{s}(n) - s(n)\big)^2 \rightarrow \min, \quad (2)$$

where $J(n)$ is minimized by the instantaneous gradient method [14]. The predictor output is denoted as [14]

$$\hat{s}(n) = \mathbf{w}^{\mathsf{T}}(n)\mathbf{x}(n{-}N), \quad (3)$$

and the weight vector is recursively updated with the normalized input as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\|\mathbf{x}(n{-}N)\|^2 + \Delta} e(n)\mathbf{x}(n{-}N), \quad (4)$$

where μ is the step size, Δ is a regularization parameter, and $e(n) = \hat{s}(n) - s(n)$ is the prediction error. Initialization is done by $\mathbf{w}(0) = \mathbf{0}$, an N_p-element zero vector.

B. Speech Prediction by RLS

Instead of minimizing only the instantaneous squared error $e^2(n)$ as in NLMS (or LMS), the recursive least-squares (RLS) predictor takes all past and current errors into account to form the weighted least squares cost function as [14]

$$J(n) = \sum_{\nu=1}^{n} \lambda^{n-\nu} \big(\hat{s}(\nu) - s(\nu)\big)^2 \rightarrow \min, \quad (5)$$

where $J(n)$ is minimized and the term λ is the forgetting factor putting an exponentially lower weight to the older error contributions. The error is again $e(n) = \hat{s}(n) - s(n)$ with the predictor output

$$\hat{s}(n) = \mathbf{w}^{\mathsf{T}}(n)\mathbf{x}(n - N). \quad (6)$$

The weight vector is recursively updated as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + e(n)\mathbf{g}(n), \quad (7)$$

with $\hat{s}(n)$ from (6) to compute $e(n)$, and the gain vector

$$\mathbf{g}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n{-}N)}{\lambda + \mathbf{x}^{\mathsf{T}}(n{-}N)\mathbf{P}(n-1)\mathbf{x}(n{-}N)}. \quad (8)$$

The matrix $\mathbf{P}(n)$ is updated as

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n{-}1) - \lambda^{-1}\mathbf{g}(n)\mathbf{x}^{\mathsf{T}}(n{-}N)\cdot\mathbf{P}(n{-}1), \quad (9)$$

where $\mathbf{P}(n)$ is initialized with $\mathbf{P}(0) = \Delta^{-1}\mathbf{I}$ and Δ is the regularization parameter, $\mathbf{I}$ is the identity matrix. Initialization of the weight vector is done by $\mathbf{w}(0) = \mathbf{0}$, an N_p-element zero vector.

III. New Speech Prediction by Echo State Networks

A. ESN Topology

It can be seen in Figure 1 that the ESN in the form that we employ for speech prediction contains basically three parts: An input layer with N_p neurons, a reservoir with M neurons and an output layer with a single neuron. The input layer is linearly connected to the reservoir with an $M \times N_p$ input weight matrix $\mathbf{W}_{\text{in}}$. In the reservoir, many neurons (M in number) are randomly and sparsely connected via a delay unit with themselves and/or with each other, which forms a random sparse reservoir weight matrix $\mathbf{W}$ with the dimension of $M \times M$. The internal reservoir state $\mathbf{y}(n)$ is defined as the output vector of the reservoir neurons. It is computed from the weighted previous reservoir state, mixed with the weighted inputs according to [32]

$$\mathbf{y}(n) = \mathbf{f}\left(\mathbf{W}_{\text{in}}\mathbf{x}(n-N) + \mathbf{W}\mathbf{y}(n-1)\right), \tag{10}$$

where $\mathbf{f} = [f_1, f_2, \cdots, f_M]^\mathsf{T}$ is the set of activation functions for all reservoir neurons. Then, the output of the ESN, i.e., the predicted speech sample $\hat{s}(n)$, can be obtained as

$$\hat{s}(n) = f_{\text{out}}\left(\mathbf{w}_{\text{out}}^\mathsf{T}(n)\bar{\mathbf{y}}(n)\right), \tag{11}$$

where f_{out} is the activation function in the output layer and $\mathbf{w}_{\text{out}}$ is the output weight vector. The term $\bar{\mathbf{y}}(n)$ could either be the state vector $\bar{\mathbf{y}}(n) = \mathbf{y}(n)$, or a concatenated vector of the state vector and the input vector [32], which is denoted as $\bar{\mathbf{y}}(n) = \left[\mathbf{x}(n-N)^\mathsf{T}, \mathbf{y}(n)^\mathsf{T}\right]^\mathsf{T}$ with $N_p + M$ elements. In the latter case, a direct linear connection between the input and the output layer is available, and the output weight vector $\mathbf{w}_{\text{out}}$ has $N_p + M$ weights instead of M.

In order to adaptively predict the speech signal, the weights of the ESN need to be updated each sample instant n, as with the sample-by-sample approaches in Section II. However, only the output weight vector $\mathbf{w}_{\text{out}}$ is updated every time index, while the input weight matrix $\mathbf{W}_{\text{in}}$ and the reservoir weight matrix $\mathbf{W}$ always remain unchanged after they have been initialized. It is just because of this unique setting that the ESN can easily update its output weights using the linear adaptive algorithm as presented in the following, and at the same time, introduces a nonlinearity $\mathbf{f}$ (and potentially f_{out}) during the signal prediction.

B. ESN Weights Updating

A kind of extended RLS algorithm is used for the output weight vector $\mathbf{w}_{\text{out}}$ updating [33] in this paper. Therefore, the cost function is the same as in (5) and the error here can be written as $e(n) = \hat{s}(n) - s(n)$ with $\hat{s}(n)$ from (11). To recursively minimize the cost function (5) the weights vector is updated as [34]

$$\mathbf{w}_{\text{out}}(n+1) = \mathbf{A}\mathbf{w}_{\text{out}}(n) + e(n)\mathbf{g}_{\text{ex}}(n), \tag{12}$$

where $\mathbf{g}_{\text{ex}}(n)$ is an extended gain vector, and $\mathbf{A} = \alpha\mathbf{I}$ is the transition matrix. Parameter $\alpha \approx 1$ assures the stability of the method and $\mathbf{I}$ is the identity matrix with the dimension of $M \times M$ or $(N_p + M) \times (N_p + M)$ based on how $\bar{\mathbf{y}}(n)$ is defined. The extended gain vector can be expressed as (compare to (8))

$$\mathbf{g}_{\text{ex}}(n) = \frac{\mathbf{A}\mathbf{P}_{\text{ex}}(n-1)\bar{\mathbf{y}}(n)}{\beta + \lambda + \bar{\mathbf{y}}^\mathsf{T}(n)\mathbf{P}_{\text{ex}}(n-1)\bar{\mathbf{y}}(n)}, \tag{13}$$

and $\mathbf{P}_{\text{ex}}(n)$ is recursively updated as

$$\begin{aligned}\mathbf{P}_{\text{ex}}(n) =& \lambda^{-1}\mathbf{A}\mathbf{P}_{\text{ex}}(n-1)\mathbf{A}^\mathsf{T} - \\ & \lambda^{-1}\mathbf{A}\mathbf{g}_{\text{ex}}(n)\bar{\mathbf{y}}^\mathsf{T}(n) \cdot \mathbf{P}_{\text{ex}}(n-1)\mathbf{A}^\mathsf{T} + \beta q\mathbf{I},\end{aligned} \tag{14}$$

where $\mathbf{P}_{\text{ex}}(n)$ is also initialized with $\mathbf{P}_{\text{ex}}(0) = \Delta^{-1}\mathbf{I}$ and Δ is the regularization parameter, β and q are tuning parameters. The weight vector is initialized as $\mathbf{w}_{\text{out}}(0) = \mathbf{0}$, a zero vector with M or $N_p + M$ elements.

Method	NLMS	RLS
Parameter	$\mu = 1.70$, $\Delta = 0.27$	$\lambda = 0.995$, $\Delta = 0.01$

Method	ESN			
Parameter	$\lambda = 0.999$	$\beta = 0.25$	$q = 0.30$	$\Delta = 0.007$

TABLE I
PARAMETER CHOICES FOR $N_p = 10$.

IV. Evaluation and Discussion

A. Simulation Setup

In this section, the ESN-based nonlinear adaptive predictor is investigated for the prediction of speech signals, and some other baseline speech prediction approaches are also simulated for comparison. All approaches are implemented as one-step-ahead prediction (i.e., $N = 1$), except for the LD recursion, which is a one-*frame*-ahead prediction, with the frame shift being the same as the frame length. Prediction performance is evaluated using the prediction gain [3]

$$G_p = 10 \cdot \log_{10} \frac{\mathrm{E}\left\{s^2(n)\right\}}{\mathrm{E}\left\{\left(s(n) - \hat{s}(n)\right)^2\right\}} \,[\mathrm{dB}], \tag{15}$$

where $\mathrm{E}\{\}$ is the expectation operator. American English speech files with 16 kHz sampling rate, 16-bit PCM, from the NTT database [35] are used for the speech prediction simulations, in which 4 female speakers together with 4 male speakers are included, each speaker represented with 12 speech files of about 8s duration. All speech files are normalized to the range $s(n) \in [-1, 1]$.

Concerning the settings of the ESN, the elements in the input weight matrix $\mathbf{W}_{\text{in}}$ are uniformly distributed random values between –1 and 1. In the reservoir, $M = 100$ neurons are used and 10% of them are randomly connected, which forms the sparse reservoir weight matrix $\mathbf{W}$. Furthermore, the spectral radius, which is the maximum of all eigenvalues of the reservoir weight matrix, is set to be 0.5 to ensure the property of asymptotical stability, so that the ESN is uniquely controlled by the input and the effect of the initial states vanishes [36], [37]. The sigmoid function is used for all activation functions f_m, $m \in \{1, 2, \cdots, M\}$, in the reservoir, and a linear function f_{out} is used as the activation function for the output layer. Additionally, the input layer is also directly connected to the output layer, i.e., $\bar{\mathbf{y}}(n) = \left[\mathbf{x}(n-N)^\mathsf{T}, \mathbf{y}(n)^\mathsf{T}\right]^\mathsf{T}$, since this was found to be advantageous. For the parameters being responsible for the ESN weights updating, we choose $\alpha = 1$ and λ, β, q and Δ are selected depending on the number of the input nodes N_p. These hyper-parameters are found separately to optimize the prediction gain (15) on the French and German speech files of the NTT database (development data). To illustrate the result of this optimization, see Table I for more details in the case of $N_p = 10$. Please note that, since the ESN is used in an online fashion to predict the speech signal, there is no need to train the actual ESN beforehand.

The settings of the baseline prediction approaches are as

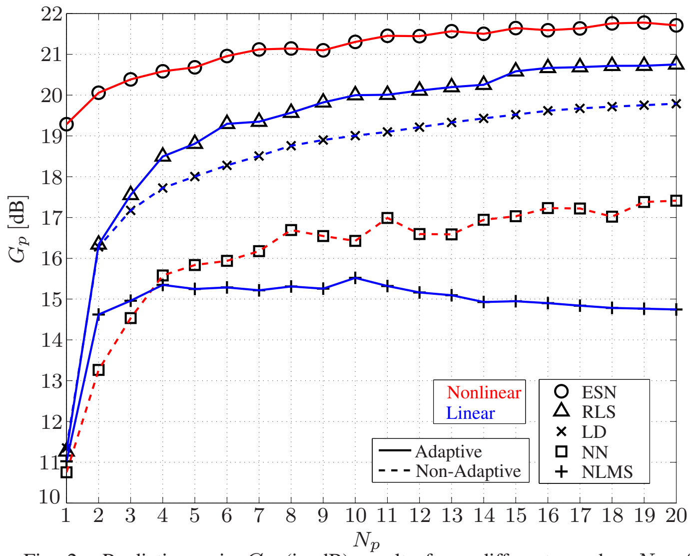

Fig. 2. Prediction gain G_p (in dB) results for a different number N_p of past speech samples (for LD: prediction order). Linear approaches (i.e., RLS, NLMS and LD) and nonlinear approaches (i.e., ESN and NN) are blue and red, respectively. The adaptive approaches (i.e., ESN, RLS and NLMS) and non-adaptive approaches (i.e., NN and LD) are shown as solid and dashed lines, respectively.

follows: For NLMS and RLS, the step size μ, the regularization parameter Δ, and the forgetting factor λ are selected depending on the number of used input samples N_p. These hyper-parameters are also selected separately to optimize the prediction gain (15) on the French and German development data. See again Table I for details on parameters for $N_p=10$. For the frame-based speech prediction (LD), a 10 ms duration frame for various linear prediction filter orders is chosen, while the frame shift is the same as the frame length[1], i.e., 10 ms. A shallow feedforward NN is also implemented for the speech prediction in an offline fashion, which is first to be trained and then to be used as the predictor with the trained NN. The NN used here has one hidden layer with a number (in the range of 20 to 40) of the neurons dependent on the number of input nodes N_p. The NN is trained and validated on a mixture of French and German speech files of the NTT database, with 80% and 20% of them constituting the training set and the development set, respectively.

B. Discussion

The simulation results with $N_p \in \{1, 2, \cdots, 20\}$ are shown in Figure 2, in which each result is averaged over 96 American English speech files. The prediction performance for the different approaches in terms of the prediction gain basically get better with increasing N_p, with the exception of NLMS having its optimum at $N_p = 10$. The ESN shows the best performance compared to all the other approaches among all the N_p values. RLS shows almost comparable gain to the ESN for large N_p; however, in small N_p conditions the ESN achieves a considerably higher prediction gain (about 8 dB when $N_p=1$). Note that both RLS and ESN approaches have virtually infinite memory due to their recurrent structure. The NN approach achieves no better prediction performance than RLS (and even the LD recursion algorithm) probably because of its non-adaptive property, although it is also a nonlinear predictor. On top of that there is no surprise that the NLMS method is also among the weak-performing ones.

From the simulation results above, it can be stated that the ESN shows exceptional performance for speech prediction, outperforming all baselines in all conditions. Even for a small number of input nodes N_p the new ESN-based speech predictor still shows strong performance. These are quite attractive properties for many applications requiring the prediction of speech.

[1]Note that for the LD approach, we employ N_p as the prediction order. This notational choice is justified by the fact that for NLMS and RLS, N_p is not only the number of used input samples, but also the prediction order as can be seen in (3) and (6). Note also that an 8 ms and 32 ms frame length/frame shift led to a lower performance.

V. Conclusions

In this paper, a nonlinear adaptive predictor using a simple echo state network (ESN) is applied to speech prediction. The output weights of the ESN are updated with an extended RLS algorithm, while the input weights and recurrent neurons stay unchanged during the prediction and do not even require training beforehand. Simulations show a prediction gain advantage of up to 8 dB compared to the best baseline method, exceeding its performance in all test conditions. Our ESN-based speech predictor can be applied in any context where speech prediction is used today.

VI. Acknowledgment

The author Ziyue Zhao would like to thank China Scholarship Council (CSC) for the financial support.

References

[1] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. John Wiley & Sons, 2006.
[2] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.
[3] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1984.
[4] T. Bäckström, *Speech Coding: Code Excited Linear Prediction*. Springer, 2017.
[5] C. Erdmann, P. Vary, K. Fischer, W. Xu, M. Marke, T. Fingscheidt, I. Varga, M. Kaindl, C. Quinquis, B. Kövesi, and D. Massaloux, "A Candidate Proposal for A 3GPP Adaptive Multi-Rate Wideband Speech Codec," in *Proc. of ICASSP*, vol. 2, Salt Lake City, UT, USA, May 2001, pp. 757–760.
[6] F. Pflug and T. Fingscheidt, "Delayless Soft-Decision Decoding of High-Quality Audio Transmitted Over AWGN Channels," in *Proc. of ICASSP*, Prague, Czech Republic, May 2011, pp. 489–492.
[7] ——, "Robust Ultra-Low Latency Soft-Decision Decoding of Linear PCM Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2324–2336, Nov 2013.
[8] P. Bauer, J. Abel, and T. Fingscheidt, "HMM-Based Artificial Bandwidth Extension Supported by Neural Networks," in *Proc. of IWAENC*, Juan-les-Pins, France, Sep. 2014, pp. 1–5.
[9] J. Markel and A. Gray, *Linear Prediction of Speech*. Springer Science & Business Media, 2013, vol. 12.
[10] J. Wung, S. Miyabe, and B. Juang, "Speech Enhancement Using Minimum Mean-Square Error Estimation and A Post-Filter Derived from Vector Quantization of Clean Speech," in *Proc. of ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 4657–4660.

[11] S. Elshamy, N. Madhu, W. Tirry, and T. Fingscheidt, “Two-Stage Speech Enhancement With Manipulation of the Cepstral Excitation,” in *Proc. of HSCMA*, San Francisco, CA, USA, Mar. 2017, pp. 106–110.
[12] ——, “Instantaneous A Priori SNR Estimation by Cepstral Excitation Manipulation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 8, pp. 1592–1605, May 2017.
[13] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. John Wiley & Sons, 2005, vol. 40.
[14] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, New Jersey: Prentice-Hall, 2002.
[15] N. Levinson, “The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction,” *Studies in Applied Mathematics*, vol. 25, no. 1-4, pp. 261–278, Apr. 1946.
[16] J. Durbin, “The Fitting of Time-Series Models,” *Revue de l’Institut International de Statistique*, vol. 28, no. 3, pp. 233–244, 1960.
[17] J. Nagumo and A. Noda, “A Learning Method for System Identification,” *IEEE Transactions on Automatic Control*, vol. 12, no. 3, pp. 282–287, Jun. 1967.
[18] D. Gabor, W. Wilby, and R. Woodcock, “A Universal Non-Linear Filter, Predictor and Simulator Which Optimizes Itself by a Learning Process,” *Proceedings of the IEE-Part B: Electronic and Communication Engineering*, vol. 108, no. 40, pp. 422–435, Jul. 1961.
[19] S. Haykin and L. Li, “Nonlinear Adaptive Prediction of Nonstationary Signals,” *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 526–535, Feb. 1995.
[20] A. Husssain, A. Jameel, D. Al-Jumeily, and R. Ghazali, “Speech Prediction Using Higher Order Neural Networks,” in *Proc. of Innovations in Information Technology (IIT)*, Al Ain, United Arab Emirates, Dec. 2009, pp. 294–298.
[21] N. Tishby, “A Dynamical Systems Approach to Speech Processing,” in *Proc. of ICASSP*, Albuquerque, NM, USA, Apr. 1990, pp. 365–368.
[22] R. Dillon and C. Manikopoulos, “Neural Net Nonlinear Prediction for Speech Data,” *Electronics Letters*, vol. 27, no. 10, pp. 824–826, May 1991.
[23] D. Mandic, J. Chambers *et al.*, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Wiley Online Library, 2001.
[24] J. Baltersee and J. Chambers, “Nonlinear Adaptive Prediction of Speech with a Pipelined Recurrent Neural Network,” *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2207–2216, Aug. 1998.
[25] D. Stavrakoudis and J. Theocharis, “A Recurrent Fuzzy Neural Network for Adaptive Speech Prediction,” in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, Montreal, QC, Canada, Jan. 2007, pp. 2056–2061.
[26] ——, “Pipelined Recurrent Fuzzy Neural Networks for Nonlinear Adaptive Speech Prediction,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1305–1320, Sep. 2007.
[27] P. Werbos, “Backpropagation Through Time: What It Does and How to Do It,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[28] R. Williams and D. Zipser, “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks,” *Neural Computation*, vol. 1, no. 2, pp. 270–280, Summ. 1989.
[29] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[30] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *arXiv preprint arXiv:1412.3555*, Dec. 2014.
[31] H. Jaeger and H. Haas, “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
[32] H. Jaeger, “The ‘Echo State’ Approach to Analysing and Training Recurrent Neural Networks,” GMD Report 148, German National Research Center for Information Technology, Tech. Rep., 2001.
[33] ——, “Adaptive Nonlinear System Identification With Echo State Networks,” in *Proc. of Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, Dec. 2003, pp. 609–616.
[34] S. Haykin, A. Sayed, J. Zeidler, P. Yee, and P. Wei, “Adaptive Tracking of Linear Time-Variant Systems by Extended RLS Algorithms,” *IEEE Transactions on Signal Processing*, vol. 45, no. 5, pp. 1118–1128, May 1997.
[35] “Multi-Lingual Speech Database for Telephonometry,” NTT Advanced Technology Corporation (NTT-AT), 1994.
[36] H. Jaeger, “Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the ‘Echo State Network’ Approach,” GMD Report 159, German National Research Center for Information Technology, Tech. Rep., 2002.
[37] M. Ozturk, D. Xu, and J. Principe, “Analysis and Design of Echo State Networks,” *Neural Computation*, vol. 19, no. 1, pp. 111–138, Jan. 2007.