PROPORTIONATE NLMS ALGORITHM FOR SECOND-ORDER VOLTERRA FILTERS AND ITS APPLICATION TO NONLINEAR ECHO CANCELLATION

Fabian Kuech and Walter Kellermann

Telecommunications Laboratory,
University of Erlangen-Nuremberg
Cauerstr. 7, D-91058 Erlangen, Germany
{kuech,wk}@LNT.de

ABSTRACT

The performance of linear acoustic echo cancelers degrades if nonnegligible nonlinear distortion is introduced into the echo path as, e.g., caused by low-cost loudspeaker systems driven at high volume. Adaptive second-order Volterra filters are known to effectively model nonlinear acoustic echo paths. In this contribution we propose an extension of the proportionate NLMS (PNLMS) to second-order Volterra filters. For the case that the amplitude of the input has an even probability density function a simplification of that algorithm is introduced, including an adaptation control which is crucial especially for nonstationary input. Simulation results show that the proposed algorithm leads to an increased convergence speed compared to an NLMS-based adaptive Volterra filter. An important feature of the presented approach is the inherent invariance of the performance of the adaptation with respect to a scaling of the input/output signals.

1. INTRODUCTION

The general set-up of the acoustic echo cancellation problem is shown in Figure 1. The acoustic echo canceler (AEC) seeks to

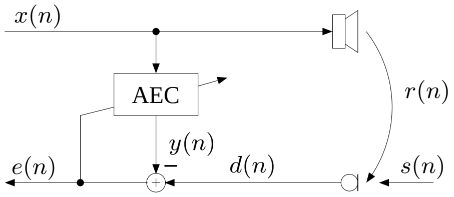

Fig. 1. General set-up of the acoustic echo cancellation problem.

minimize the power of the error signal $e(n)$ by subtracting an estimate of the echo signal $y(n)$ from the microphone signal $d(n)$. Standard approaches for the cancellation of acoustic echos rely on the assumption that the echo path to be identified can be modeled by a linear filter. However, in some practical situations loudspeaker systems introduce nonnegligible nonlinear distortions, e.g., caused by low-cost loudspeakers driven at high volume. With this nonlinear distortion, the performance of a linear acoustic echo canceler degrades and furthermore, greatly impairs quality of voice communication. Thus, nonlinear models have to be considered. A common approach to modeling the nonlinear behaviour of loudspeakers is given by finite length second-order Volterra filters [1]. However, adaptive Volterra filters applying an NLMS algorithm are known to suffer severely from slow convergence speed, especially for correlated excitation signals [2]. The recently proposed improved PNLMS [3], based on [4], provides a faster initial convergence of the adaptive filter coefficients, especially for sparse echo paths. In Figure 2 a typical quadratic Volterra kernel is shown that has been obtained from measurement of a small loudspeaker placed in an enclosure with low reverberation. Obviously, the

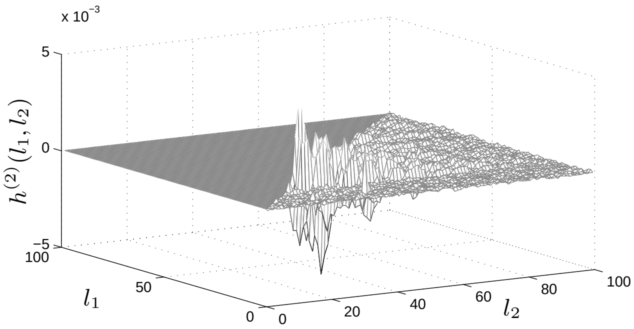

Fig. 2. Quadratic Volterra kernel measured for a small loudspeaker.

quadratic kernel can be considered as sparse and thus, a PNLMS-type adaptation algorithm for Volterra filters is desirable. In this paper a general extension of the linear PNLMS to adaptive second-order Volterra filters is proposed first, and then a simplified version for specific input signals is introduced. The suitability of the proposed approaches with respect to a nonlinear acoustic echo cancellation application is validated by simulation results.

2. SECOND-ORDER VOLTERRA FILTERS

The input/output relation of a second-order Volterra filter is given by

$$\begin{aligned} y(n) &= \sum_{l=0}^{N_1-1} h_l^{(1)}(n)x(n-l) \\ &\quad + \sum_{l_1=0}^{N_2-1} \sum_{l_2=l_1}^{N_2-1} h_{l_1,l_2}^{(2)}(n)x(n-l_1)x(n-l_2). \quad (1) \end{aligned}$$

This work was partly supported by the ANITA project funded by the European Comission under contract IST-2001-34327.

To obtain a compact vector representation of (1) we define

$$\mathbf{x}_1(n) = [x(n),\, x(n-1),\, \dots ,x(n-N_1+1)]^T , \quad (2)$$

$$\mathbf{h}_1(n) = \left[h_0^{(1)}(n),\, h_1^{(1)}(n) \dots ,h_{N_1-1}^{(1)}(n)\right]^T , \quad (3)$$

$$\mathbf{x}_2(n) = \big[x^2(n),\, x(n)x(n-1), \dots ,x(n)x(n-N_2+1), \\ x^2(n-1),\, \dots ,\, x^2(n-N_2+1)\big]^T , \quad (4)$$

$$\mathbf{h}_2(n) = \Big[h_{0,0}^{(2)}(n),\, h_{0,1}^{(2)}(n),\, \dots ,h_{0,N_2-1}^{(2)}(n), \\ h_{1,1}^{(2)}(n),\, \dots ,h_{N_2-1,N_2-1}^{(2)}(n)\Big]^T , \quad (5)$$

$$\mathbf{h}(n) = \left[\mathbf{h}_1^T(n)\; \mathbf{h}_2^T(n)\right]^T , \quad (6)$$

$$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n)\; \mathbf{x}_2^T(n)\right]^T . \quad (7)$$

The vector length of $\mathbf{x}_1(n)$ and $\mathbf{h}_1(n)$ is $L_1 = N_1$, the length of $\mathbf{x}_2(n)$ and $\mathbf{h}_2(n)$ is $L_2 = N_2(N_2+1)/2$, and the length of $\mathbf{x}(n)$ and $\mathbf{h}(n)$ is $L = L_1 + L_2$. Then, (1) can be expressed by

$$y(n) = \mathbf{h}^T(n)\mathbf{x}(n) \quad (8)$$

in a compact way.

3. PNLMS ALGORITHM FOR ADAPTIVE SECOND-ORDER VOLTERRA FILTERS

3.1. PNLMS algorithm for linear adaptive filters

The PNLMS algorithm for linear adaptive filters in its improved version according to [3] can be summarized as follows:

$$e_1(n) = d(n) - \mathbf{h}_1^T(n)\mathbf{x}_1(n), \quad (9)$$

$$\mathbf{h}_1(n+1) = \mathbf{h}_1(n) + \mu\frac{\mathbf{K}(n)\mathbf{x}_1(n)e_1(n)}{\mathbf{x}_1^T(n)\mathbf{K}(n)\mathbf{x}_1(n)}, \quad (10)$$

$$\mathbf{K}(n) = \mathrm{diag}\left\{k_0(n),\, \dots ,\, k_{L_1-1}(n)\right\}, \quad (11)$$

$$k_l(n) = \frac{(1-\alpha)}{2L_1} + (1+\alpha)\frac{\left|h_l^{(1)}(n)\right|}{2\left\|\mathbf{h}_1(n)\right\|_1}, \quad (12)$$

where the 1-norm of $\mathbf{h}_1(n)$ is defined as

$$\left\|\mathbf{h}_1(n)\right\|_1 = \sum_{l=0}^{L_1-1}\left|h_l^{(1)}(n)\right| . \quad (13)$$

From the definition of $k_l(n)$ in (11) we notice that the first term corresponds to an NLMS update, whereas the second term introduces the idea of making the update of a coefficient proportionate to its significance compared to the other coefficients of the adaptive filter. The significance of a coefficient is described here as the ratio of its absolute value compared to 1-norm of the filter vector. For stationary input $x(n)$, the statistics of the excitation of the elements of $\mathbf{h}_1(n)$ is equal for each element. Thus, the second term in (12) can also be interpreted as being proportionate to the relative contribution of the coefficient to the output signal $y(n)$.

3.2. Extension to second-order Volterra filters

Regarding the definition of $\mathbf{x}(n)$ in (7), it is obvious that the statistics of the elements of $\mathbf{x}(n)$ are in general not equal, implying that the statistics of the excitation of different elements of $\mathbf{h}(n)$ are in general not equal for each coefficient. Assuming stationary input in the following, we obtain for the second-order moment of the j-th element of $\mathbf{x}(n)$, denoted by $x_j(n)$:

$$\sigma_j^2 = \mathcal{E}\{x_j^2(n)\} \quad (14)$$
$$= \begin{cases} \mathcal{E}\{x^2(n)\}, & \text{if } x_j(n) \in \mathbf{x}_1(n) \\ \mathcal{E}\{x^2(n-l_1)x^2(n-l_2)\}, & \text{if } x_j(n) \in \mathbf{x}_2(n). \end{cases}$$

As a consequence, a coefficient can have a significant value on the one hand, but its contribution to the output signal is only marginal on the other hand, if the power of the corresponding excitation signal $x_j(n)$ is very small. Therefore, a straightforward extension of the linear PNLMS to Volterra filters, i.e., replacing $\mathbf{x}_1(n)$ by $\mathbf{x}(n)$, and replacing $\mathbf{h}_1(n)$ by $\mathbf{h}(n)$ in (9)-(12), would not result in a suitable PNLMS algorithm for Volterra filters. Aiming at an algorithm that promotes the update of coefficients with large significance with respect to the computation of the output signal, we introduce a signal dependent norm of the filter coefficients, i.e.,

$$\left|h_j(n)\right|_x = \left|h_j(n)\right|\sigma_j, \quad (15)$$

where $\sigma_j = \sqrt{\mathcal{E}\{x_j^2(n)\}}$ and $h_j(n)$ denotes the j-th element of $\mathbf{h}(n)$. Accordingly, we introduce the signal dependent 1-norm of the coefficient vector $\mathbf{h}(n)$

$$\left\|\mathbf{h}(n)\right\|_x = \sum_{l=0}^{L-1}\left|h_l(n)\right|_x . \quad (16)$$

With these signal dependent norms we define the diagonal matrix

$$\mathbf{G}(n) = \mathrm{diag}\left\{g_0(n),\, \dots ,\, g_{L-1}(n)\right\}, \quad (17)$$

where

$$g_l(n) = \frac{(1-\alpha)}{2L} + (1+\alpha)\frac{\left|h_l(n)\right|_x}{2\left\|\mathbf{h}(n)\right\|_x}. \quad (18)$$

Comparing (12) with (18) we notice that $\mathbf{G}(n)$ can be considered as the step-size adjustment matrix of the PNLMS algorithm according to [3] applied to $\tilde{h}_l(n) = h_l(n)\sigma_j$. Introducing the diagonal matrix

$$\mathbf{S} = \mathrm{diag}\left\{\sigma_0,\, \dots ,\, \sigma_{L-1}\right\}, \quad (19)$$

we define

$$\tilde{\mathbf{h}}(n) = \mathbf{S}\mathbf{h}(n), \quad (20)$$

$$\tilde{\mathbf{x}}(n) = \mathbf{S}^{-1}\mathbf{x}(n), \quad (21)$$

and rewrite the computation of $y(n)$ according to (8) as

$$y(n) = \tilde{\mathbf{h}}^T(n)\tilde{\mathbf{x}}(n). \quad (22)$$

The update equation of $\tilde{\mathbf{h}}(n)$ applying a PNLMS-type algorithm according to [3] is then given by

$$\tilde{\mathbf{h}}(n+1) = \tilde{\mathbf{h}}(n) + \mu\frac{\mathbf{G}(n)\tilde{\mathbf{x}}(n)e(n)}{\tilde{\mathbf{x}}^T(n)\mathbf{G}(n)\tilde{\mathbf{x}}(n)}. \quad (23)$$

Finally, the update equation for the Volterra filter coefficients $\mathbf{h}(n)$ is obtained by left-multiplying (23) by $\mathbf{S}^{-1}$ and taking the definitions (20), (21) into account, i.e.,

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu\frac{\widetilde{\mathbf{G}}(n)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\widetilde{\mathbf{G}}(n)\mathbf{x}(n)}, \quad (24)$$

where the abbreviation

$$\widetilde{\mathbf{G}}(n) = \mathbf{S}^{-1}\mathbf{G}(n)\mathbf{S}^{-1} \tag{25}$$

for the normalized version of $\mathbf{G}(n)$ has been introduced. For an implementation of the proposed algorithm one can either compute $y(n)$ according to (22) and apply the update equation for $\tilde{\mathbf{h}}(n)$ according to (23), or use the corresponding combination of (8) and (24), instead.

Note that for the pure linear case, i.e., $\mathbf{x}(n) = \mathbf{x}_1(n)$ and $\mathbf{h}(n) = \mathbf{h}_1(n)$ (implying $L_1 = L$ and $e_1(n) = e(n)$), the proposed algorithm simplifies to the linear PNLMS algorithm according to Section 3.1, i.e. [3], as in this case

$$\frac{|h_{1,l}(n)|_x}{2\,\|\mathbf{h}_1(n)\|_x} = \frac{|h_{1,l}(n)|}{2\,\|\mathbf{h}_1(n)\|_1} \tag{26}$$

and thus $\mathbf{G}(n) = \mathbf{K}(n)$. Observing that $\mathbf{S} = \sigma_0\mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix, the update equation for $\mathbf{h}_1(n)$ according to (24) simplifies to (10).

3.3. Simplified version for non-general input

The PNLMS algorithm presented in the last section requires the estimation of $\mathbf{S}$, implying the estimation of second- and fourth-order moments of the input signal $x(n)$ which is usually unreliable for short observation intervals as required for nonstationary input signals such as speech. If we assume that the elements of $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ are approximately orthogonal,i.e.,[1]

$$\mathcal{E}\{x(n-i)x(n-j)x(n-l)\} \approx 0, \tag{27}$$

the linear and quadratic Volterra kernels can be adapted separately. In the following we present a simplification of the general approach according to Section 3.2 which does not rely on the estimate of $\mathbf{S}$. As already discussed in the last section, the separate adaptation of the linear kernel $\mathbf{h}_1(n)$ can be performed according to (10) which corresponds to [3]. In general, (26) does not hold for the quadratic Volterra kernel $\mathbf{h}_2(n)$, as the elements of $\mathbf{x}_2(n)$ are not just samples of the same signal taken at different time instants, as it can be seen from (4). However, if we introduce the approximation

$$\mathcal{E}\left\{x^2(n-i)x^2(n-j)\right\} \approx \mathcal{E}\left\{x^4(n-i)\right\} \tag{28}$$

for $i, j \in \{0, \dots, N_2 - 1\}$, we obtain

$$\frac{|h_{2,l}(n)|_x}{2\,\|\mathbf{h}_2(n)\|_x} \approx \frac{|h_{2,l}(n)|}{2\,\|\mathbf{h}_2(n)\|_1}. \tag{29}$$

Then, the separate adaptation of $\mathbf{h}_2(n)$ can be performed using the corresponding version of (10).

Although (27) implies that the Volterra kernels can be adapted separately from each other, unstable behaviour of the adaptive Volterra filter has been observed in simulation, especially in the beginning of the adaptation or after a change of the echo path. The reason for this behaviour is that the error introduced by a misadjusted linear kernel acts as a distortion for the adaptation of the quadratic kernel and vice versa. Thus, a control of the adaptation is crucial. With $e_1(n) = d(n) - \mathbf{h}_1(n)^T\mathbf{x}_1(n)$ we define

$$\overline{e}_1^2(n) = \lambda\overline{e}_1^2(n-1) + (1-\lambda)e_1^2(n), \tag{30}$$

$$\overline{e}^2(n) = \lambda\overline{e}^2(n-1) + (1-\lambda)e^2(n), \tag{31}$$

where the forgetting factor λ is chosen close to 1. The error signal

$$\widehat{e}_1(n) = \begin{cases} e_1(n), & \text{if } \overline{e}_1^2(n) < \overline{e}^2(n) \\ e(n), & \text{if } \overline{e}_1^2(n) \geq \overline{e}^2(n), \end{cases} \tag{32}$$

is then used for the adaptation of the linear kernel, whereas the error signal $\widehat{e}_2(n) = e(n)$ applies for the adaptation of the quadratic kernel. $\widehat{e}_1(n)$ also represents the signal transmitted to the far-end. The simplified version of the algorithm can be summarized by

$$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) + \mu\frac{\mathbf{K}_i(n)\mathbf{x}_i(n)\widehat{e}_i(n)}{\mathbf{x}_i^T(n)\mathbf{K}_i(n)\mathbf{x}_i(n)}, \tag{33}$$

$$\mathbf{K}_i(n) = \operatorname{diag}\{k_{i,0}(n), \dots, k_{i,L_i-1}(n)\}, \tag{34}$$

$$k_{i,l}(n) = \frac{(1-\alpha)}{2L_i} + (1+\alpha)\frac{|h_{i,l}(n)|}{2\,\|\mathbf{h}_i(n)\|_1}, \tag{35}$$

for $i \in \{1, 2\}$, where $h_{i,l}(n)$ denotes the l-th element of the coefficient vector $\mathbf{h}_i(n)$. With the above adaptation control, the quadratic Volterra kernel acts as a shadow filter that is only used for the echo cancellation, if a reduction of the power of the residual error is achieved, whereas otherwise, the AEC corresponds to a linear adaptive filter.

3.4. Signal gain invariance

Neglecting the nonlinearity of the quantization, the analog-to-digital (A/D) conversion of a continuous time signal $s_c(t)$ (e.g. measured as voltage) to its corresponding discrete time representation $s(n)$ can be expressed by

$$s(n) = \frac{s_c(nT)}{S_o}, \tag{36}$$

where $\frac{1}{T}$ is the sampling rate and S_o is the maximum input. This representation will be useful for the following considerations. Different A/D-converters can have different values for S_o, resulting in different values of the discrete time signal for the same analog signal $s_c(t)$, i.e.,

$$\breve{s}(n) = \frac{s_c(nT)}{\breve{S}_o} = C\,s(n). \tag{37}$$

In the following the $\breve{}$ over a signal indicates a scaling according to (37). However, an adaptive algorithm should be invariant with respect to a different scaling of the discrete time signals, in order to be independent from the hardware used for the signal acquisition. Here, gain invariance of an adaptive algorithm for echo cancellation is understood as

$$\frac{\mathcal{E}\left\{d^2(n)\right\}}{\mathcal{E}\left\{e^2(n)\right\}} \stackrel{!}{=} \frac{\mathcal{E}\left\{\breve{d}^2(n)\right\}}{\mathcal{E}\left\{\breve{e}^2(n)\right\}}, \tag{38}$$

i.e., the relative echo attenuation is invariant with respect to a scaling of the discrete time signals. Regarding that $\breve{\mathbf{x}}_1(n) = C\,\mathbf{x}_1(n)$ and $\breve{\mathbf{x}}_2(n) = C^2\,\mathbf{x}_2(n)$ if $\breve{x}(n) = Cx(n)$, we have

$$\breve{y}(n) = \mathbf{v}_1^T(n)\breve{\mathbf{x}}_1(n) + \mathbf{v}_2^T(n)\breve{\mathbf{x}}_2(n) \tag{39}$$

$$C\,y(n) = C\,\mathbf{h}_1^T(n)\mathbf{x}_1(n) + C\,\mathbf{h}_2^T(n)\mathbf{x}_2(n) = \mathbf{h}_1^T(n)\breve{\mathbf{x}}_1(n) + \frac{1}{C}\mathbf{h}_2^T(n)\breve{\mathbf{x}}_2(n). \tag{40}$$

[1]Assuming that $x(k)$ is an i.i.d. process, (27) is fulfilled if the probability density function (PDF) of the amplitude of $x(k)$ is an even function, i.e. $f_X(x) = f_X(-x)$. Although this assumption is unrealistic for speech input, experimental results indicate that (27) is sufficiently fulfilled in practical situations.

Comparing (39) with (40), the requirements for (38) are given by

$$\mathbf{h}_1(n) = \mathbf{v}_1(n) \ \wedge\ \mathbf{h}_2(n) = C\,\mathbf{v}_2(n) \quad \forall n. \tag{41}$$

It is straightforward to show that the update equation (24) of the PNLMS algorithm for second-order Volterra filters fulfills (41), if the initialization of the coefficient vectors is chosen such that $\mathbf{h}_1(0) = \mathbf{v}_1(0)$ and $\mathbf{h}_2(0) = C\,\mathbf{v}_2(0)$ (which is, e.g., achieved by initializing all coefficients with zero) and the same value for the step-size parameter μ is used. Accordingly, it can be shown that the simplified update version (33) is signal gain invariant if $\mathbf{h}_1(0) = \mathbf{v}_1(0)$ and $\mathbf{h}_2(0) = C\,\mathbf{v}_2(0)$. The NLMS algorithm for Volterra filters usually referred to in the literature (see e.g. [2]) is given by the update equation

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}. \tag{42}$$

Note that the behaviour of the NLMS according to (42) is not input gain invariant, i.e., it does not fulfill (38), whereas applying the proposed general PNLMS with $\alpha = -1$ represents a gain invariant version of the NLMS. As a consequence, the performance of the NLMS algorithm according to [2] degrades for specific values of C, e.g., if $\mathcal{E}\{\breve{x}^2(n)\} \ll \mathcal{E}\{\breve{x}^4(n)\}$ holds.

4. SIMULATION RESULTS

To evaluate the performance of the different versions of the PNLMS algorithm for second-order Volterra filters we present simulation results obtained for an acoustic echo cancellation application. In the first experimental set-up the echo path has been modeled by a second-order Volterra filter (with memory lengths of 400 taps and 80 taps for the linear and quadratic kernel, respectively), where the quadratic kernel is shown in Figure 2. To evaluate the performance of the adaptive algorithms we consider

$$ERLE = 10\,\log_{10} \frac{\mathcal{E}\{y^2(n)\}}{\mathcal{E}\{e^2(n)\}}\,. \tag{43}$$

The $ERLE$ graphs resulting from a linear PNLMS [3] (with $\alpha = 0$), a linear NLMS, the proposed general PNLMS for second-order Volterra filters with $\alpha = 0$, a second-order Volterra filter using NLMS adaptation, i.e. $\alpha = -1$, are shown in Figure 3. The excitation has been colored noise and an SNR of 35dB has been preset. For a fair comparison, the step-size parameters μ for the linear kernels and the linear approaches have been chosen equally. The step-sizes for the quadratic kernels have been the same for both, NLMS and PNLMS algorithm. It can be noticed that the proposed PNLMS for second-order Volterra filters clearly outperforms a NLMS-based Volterra filter in terms of convergence speed. It can also be seen that a remarkable increase of achievable echo attenuation is obtained by taking the nonlinearities in the echo path into account. The $ERLE$ graphs presented in Figure 4 base on recorded speech data from a low-cost loudspeaker in an enclosure with low reverberation, where the proposed simplified version of the PNLMS including the adaptation control has been applied for $\alpha = 0$. The adaptation control has also been used for $\alpha = -1$ which corresponds to a respective NLMS algorithm for Volterra filters. The simplified PNLMS algorithm for Volterra filters shows an increased convergence speed compared to a corresponding NLMS-based Volterra filter. The increase of echo attenuation obtained for the Volterra filter with PNLMS adaptation compared to the linear PNLMS algorithm is also significant, especially during periods of high excitation levels.

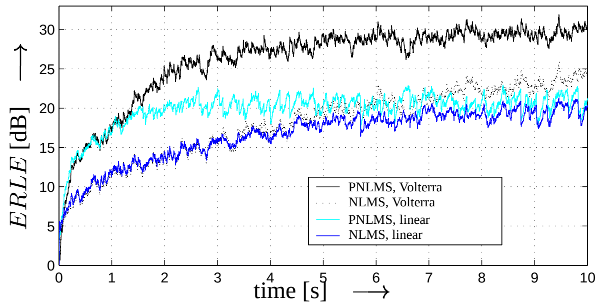

Fig. 3. $ERLE$ for linear and nonlinear echo cancellation using different adaptive approaches for stationary colored noise input.

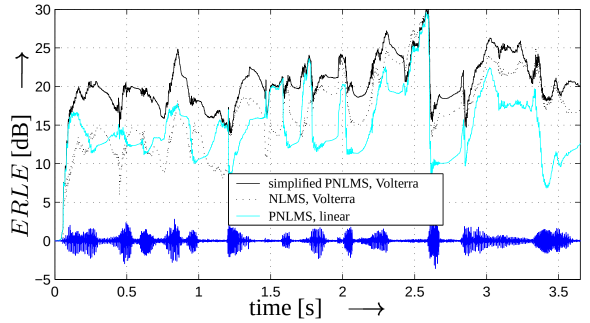

Fig. 4. $ERLE$ for linear and nonlinear echo cancellation using different adaptive approaches. The excitation has been speech.

5. CONCLUSION

We have presented an extension of the linear PNLMS algorithm to second-order Volterra filters. A simplified version of the general approach, where the linear and the quadratic kernels are adapted separately has been introduced for the case that the amplitude of input signal has an even PDF, including a simple but essential adaptation control. The input gain invariance of the performance of the approaches has been discussed. Simulation results have shown that the proposed algorithms provide an increased convergence speed compared to the NLMS algorithm for Volterra filters and an increased echo attenuation compared to a pure linear AEC for a nonlinear echo path.

6. REFERENCES

[1] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filter," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Phoenix, March 1999.

[2] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, pp. 10–26, July 1991.

[3] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, May 2002, pp. 1881–1884.

[4] D. L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancelers," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 8, no. 5, pp. 508–518, Sept. 2000.