Architectures and Algorithms for Nonlinear Adaptive Filters

V. Hegde, C. Radhakrishnan, D. J. Krusienski, and W. K. Jenkins
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA

Abstract – This paper considers series-cascade nonlinear adaptive filter architectures consisting of a linear input filter, a memoryless polynomial nonlinearity, and a linear output filter (LNL). The learning characteristics of the LNL structure are studied in terms of performance and complexity. Replacing the linear input stage and the memoryless nonlinear stage of the LNL model with a Volterra module is then considered. Adaptive algorithms are summarized for these structures and experimental examples are used to illustrate performance for the identification of an acoustic echo channel.

1. Introduction

Many nonlinear systems can be represented using one of three models shown in Figure 1 [1]. Depending on the memory size of the linear components in the LNL model, it may be desirable to combine modules of the LNL model, representing the complete system as a series-cascade of a Volterra module followed by linear filter, or as a linear filter followed by a Volterra module.

In this paper we first consider modular LNL systems with an FIR input stage and both FIR and IIR output stages. In particular, we consider the models shown in figures 2 and 3 for system identification. The nonlinear filter in figure 2 is simply a memoryless polynomial filter. Filter-1 of figure 3 is a nonlinear filter with memory, which can be implemented with a low Volterra module

2. Joint Adaptation Algorithms

Since the derivations of joint adaptation schemes for the series-cascade structures considered in this paper were published in [3, 4], the algorithms will be summarized here. The normalized least mean squares algorithm (NLMS) used in this work requires the derivative of the error-squared with respect to the filter coefficients for each module for updating the tap weights. These derivatives were obtained via the concept of back-propagation that is found extensively in the neural network literature [5].

2.1 The LNL Structure: FIR Output Stage

Table 1 summarizes the joint NLMS adaptation of the three section LNL structure with an FIR output stage. The computational requirements for this cascade structure:

Number of tap-weights:	M_1+N+M_2
Memory units:	$(M_1+N)M_2$
Additions:	$2M_1+3N+3M_2+M_1M_2+NM_2-3$
Multiplications:	$4M_1+6N+3M_2+M_1M_2+NM_2+4$

2.2 The LNL Structure: IIR Output Stage

Table 2 summarizes the joint NLMS adaptation of the LNL structure with an IIR output stage using an output error algorithm. The computational requirements for this cascade structue are as follows:

Number of tap-weights:	$M_1+N+N_a+N_b+1$
Memory units:	$(M_1+N)(N_a+N_b+1)$
Additions:	$3M_1+4N+3N_a+3N_b+(M_1+N)(N_a+N_b)$
Multiplications:	$5M_1+7N+3N_a+3N_b+(M_1+N)(N_a+N_b)+6$

2.3 Cascaded Volterra and FIR Structure

Table 3 summarizes the joint NLMS adaptation algorithm for the combined Volterra-FIR structure. The computational requirements for this structure are:

Number of tap-weights:	N_v+M_2
Memory units:	N_vM_2
Additions:	$2N_v+3M_2+N_vM_2-1$
Multiplications:	$M_v+3N_v+4M_2+N_vM_2+4$

2.4 Cascaded Volterra and IIR Structure

Table 4 summarizes the joint NLMS adaptation of the Volterra-IIR structure using the output error algorithm for adjusting the taps of the IIR stage. The computational requirements for this structure are:

Number of tap-weights:	$N_v+N_a+N_b+1$
Memory units:	$N_v(N_a+N_b+1)$
Additions:	$3N_v+3N_a+3N_b+N_aN_v+N_bN_v+2$
Multiplications:	$M_v+4N_v+3N_a+3N_b+N_aN_v+N_bN_v+7$

3. Experimental Results

Experiments were performed using the series-cascade structures from Section 2 for the identification of a nonlinear system. Figure 6 shows the model of the acoustic echo path that was used in the following experiments.

Modeling the loudspeaker: The loudspeaker was modeled with an FIR filter with memory length 8 in series-cascade with a memoryless nonlinearity of order 5 [2]. The first coefficient of the FIR filter was fixed at value 1.0 and the remaining 7 coefficients were generated randomly and were kept small, so that the gain due to the memory of this FIR filter is about 25%. This makes sure that if the magnitude of the input to the loudspeaker is within 0.8 the magnitude of the input to the static nonlinearity is within 1.0. The amplifier was modeled by the nonlinear function $f(x) = x - 0.5x^3 + 0.02x^5$, which approximates a hyperbolic tangent. The amplifier is nearly linear for an input [-0.3, 0.3], but starts saturating for inputs outside of this range.

0-7803-7576-9/02$17.00 © 2002 IEEE

Modeling the linear part of the echo path: The envelope of the impulse response of the room typically has a peak, and decays exponentially on either side. In the following experiments, where an FIR filter is used at the output stage, a filter of length 64 was used to model the echo path. A small random number was added to each coefficient. For experiments where the IIR filter is used at the output stage, an all-pass filter with two complex conjugate poles at 0.9e +/-j60 and two zeros at 1.1111e +/-j60 was used to model the echo path [4]. A babble signal was used as a test input signal [3]. The instantaneous magnitude of the input signal was set to be in the range [0.3, 0.8], so that the loudspeaker operates in its nonlinear region. A Gausian noise of variance 0.001 was added to the echo signal, corresponding to a SNR of 21 dB for the echo signal.

Experiment 1: As a baseline experiment an FIR filter of length 71 was used for acoustic echo cancellation. During the first 20,000 iterations the babble input was de-amplified by 0.375 so that the input remained in the range [-0.3,0.3], and the loudspeaker operated in its linear region. For the next 20,000 iterations the de-amplification was removed and the loudspeaker was driven into its nonlinear region. The echo return loss (ERLE) [3] plot for this simulation is shown in figure 7. Whereas the FIR filter achieved an ERLE of 22.5 dB in the linear region, in the nonlinear region it was able to achieve an ERLE of only 17 dB.

Experiment 2: The LNL cascade structure was used for echo cancellation with input and output filter lengths of $M_1 = 8$ and $M_2 = 64$. The polynomial filter had a nonlinearity of order $N = 5$. The total number of taps required for this filter is 77. The ERLE plot is shown figure 8, where it is seen that this structure converged to approximately 20 dB after 30,000 iterations, i.e., it achieved about 2.5 dB lower ERLE as compared with operating in the linear region

Next we considered the LNL cascade with a memoryless polynomial filter of order 5 sandwiched between an FIR input filter and an IIR output filter. In this case the IIR stage was adapted with an output error algorithm. The FIR filter had a memory of length 8, whereas the IIR filter had 2 zeros and 2 poles. The ERLE plot of this experiment is also plotted in figure 8. This filter reached an ERLE of only 14 dB after 20,000 iterations with the output error scheme, which appears to be convergence to a local minimum.

Experiment 3: Next the series-cascade of a Volterra filter and an FIR was tested. The Volterra filter had a memory of $M_1 = 8$ and a nonlinearity order of 3, thereby requiring 164 filter coefficients. The length of the FIR filter was $M_2 = 64$. The ERLE plot of this structure is shown figure 9. This structure converged to 21 dB after 30,000 iterations, which is 1.5 dB lower than the ERLE obtained in the linear region.

Next he cascade of a Volterra filter followed by an IIR using an output error algorithm was tested. The Volterra filter had order 3 and memory length 8. The IIR filter had 2 zeros and 2 poles. The ERLE plot of this filter using the output error scheme is also plotted in figure 9. This architecture obtained an ERLE of 22 dB with the output error scheme.

Experiment 4: Both of the cases from the above examples with the IIR output stage were tested again using the equation error method for adaptation of the IIR stage. The ERLE plots for these two cases are compared in figure 10, where it appears that the equation error method is effective in avoiding convergence to local minima. Both of these cases achieved the near optimum ERLE of approximately 22 dB.

4. Conclusion

Of the structures considered in this paper, the LNL architecture appears to perform best with respect to convergence and computational complexity. The Volterra-FIR series-cascade structure achieved an ERLE that is comparable to that of the LNL structure, although its convergence rate was often slower, and its computation complexity higher. The equation error method for adaptatin of the IIR output stage alleviates the difficulty of convergence to local minima.

References

[1] A. E. Nordsjo and L. H. Zetterberg, Identification of certain time-varying nonlinear Wiener and Hammerstein systems, *IEEE Trans. on Sig. Proc.*, VOL 49, NO. 3, pp 577-592, March 2001.

[2] F. X. Y. Gao and W. M. Snelgrove., "Adaptive Linearization of Loudspeakers," Proc. ICASSP'91, pp. 3589-3592.

[3] V. Hedge, "Series-Cascade Adaptive Filters for Identification of nonlinear Systems," Masters Thesis, Department of Electrical Engineering, The Pennsylvania State University, August 2001.

[4] V. Hegde C. Radhakrishnan, D. Krusienski, and W. K Jenkins, Series-Cascade Nonlinear Adaptive Filters, *Proceedings of the Midwest Symposium on Circuits and Systems,* Tulsa, OK, August 2002, to appear.

[5] Bose N. and Liang P., *Neural Networks: Graphs and Algorithms*, McGraw-Hill Book Company, N.Y., 1996.

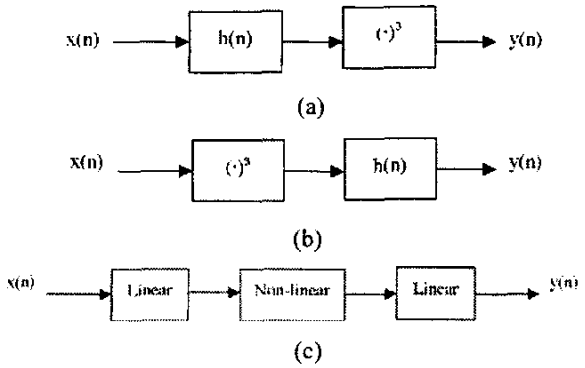

Figure 1. (a) Wiener, (b) Hammerstein, and (c) LNL models.

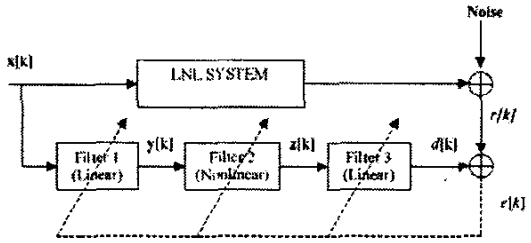

Figure 2. System id. with an LNL model.

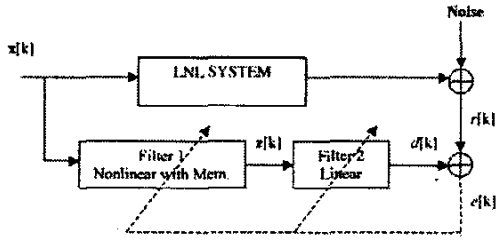

Figure 3. System id. using Volterra-FIR filter structures.

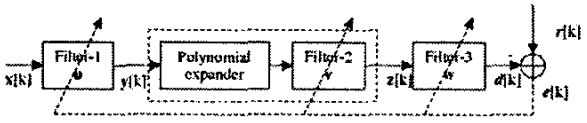

Figure 4. Details of the LNL model.

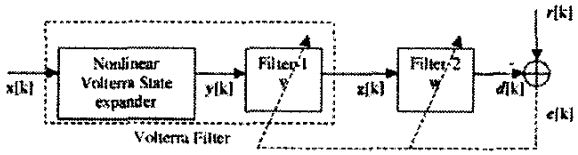

Figure 5. Details of the Volterra - FIR structure.

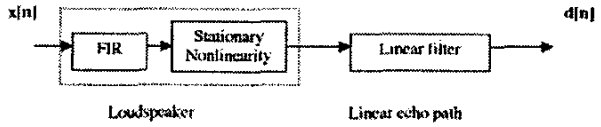

Figure 6. Model of the acoustic echo path.

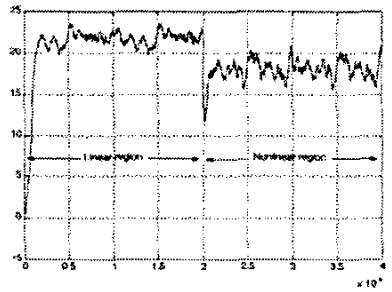

Figure 7. ERLE of 71-Tap FIR acoustic echo canceler.

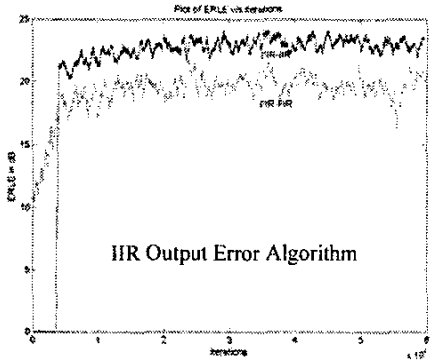

Figure 8. ERLE of the LNL structure.

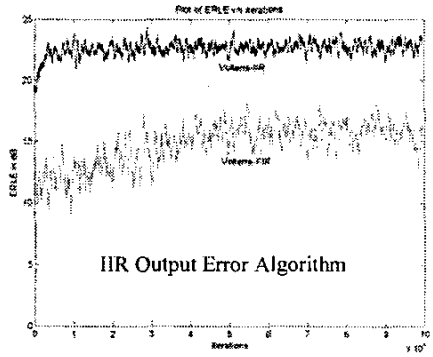

Figure 9. ERLE of Volterra and Linear Output Filters.

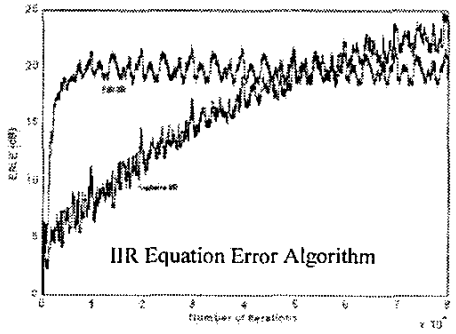

Figure 10. ERLE of Volterra and Linear Output Filters.

Table 1. Joint NLMS update for FIR +memoryless polynomial filter + FIR cascade

Operation	Additions	Multiplications
1. $x[k] = [\,x[k], x[k-1],\ldots, x[k-M_1+1]\,]^T$	-	-
2. $y[k] = u^T[k].x[k]$	M_1-1	M_1
3. $y[k] = [\,y[k], y[k]^2,\ldots, y[k]^N]^T$	-	N-1
4. $z[k] = v^T[k].y[k]$	N-1	N
5. $z[k] = [\,z[k], z[k-1],\ldots, z[k-M_2+1]\,]^T$	-	-
6. $d[k] = w^T[k].z[k]$	M_2-1	M_2
7. $e[k] = r[k] - d[k]$	1	-
8. $a[k] = [1, 2y[k], 3y[k]^2,\ldots, Ny[k]^{N-1}]$	-	N-1
9. $b[k] = v^T[k].a[k]$	N-1	N
10. $x_p[k] = b[k].x[k]$	-	M_1
11. $X[k] = [x_p[k], x_p[k-1],\ldots, x_p[k-M_2+1]]$	-	-
12. $p[k] = X[k].w[k]$	$M_1(M_2-1)$	M_1M_2
13. $Y[k] = [\,y[k], y[k-1],\ldots, y[k-M_2+1]]$	-	-
14. $q[k] = Y[k].w[k]$	$N(M_2-1)$	NM_2
15. $u[k+1] = u[k] + (\alpha_u / (\|p[k]\|^2 + \delta)).p[k].e[k]$	$2M_1$	$2M_1+2$
16. $v[k+1] = v[k] + (\alpha_v / (\|q[k]\|^2 + \delta)).q[k].e[k]$	2N	2N+2
17. $w[k+1] = w[k] + (\alpha_w / (\|z[k]\|^2 + \delta)).z[k].e[k]$	$2M_2$	$2M_2+2$

Table 2. Joint NLMS FIR +memoryless polynomial + IIR cascade: Output Error

Operation	Additions	Multiplications
1. $x[k] = [\,x[k], x[k-1],\ldots, x[k-M_1+1]\,]^T$	-	-
2. $y[k] = u^T[k].x[k]$	M_1-1	M_1
3. $y[k] = [\,y[k], y[k]^2,\ldots, y[k]^N]^T$	-	N-1
4. $z[k] = v^T[k].y[k]$	N-1	N
5. $z[k] = [z[k],\ldots z[k-N_b],\ d[k-1],\ldots, d[k-N_a]]^T$	-	-
6. $d[k] = w^T[k].z[k]$	N_a+N_b	N_a+N_b
7. $e[k] = r[k] - d[k]$	1	-
8. $a[k] = [1, 2y[k], 3y[k]^2,\ldots, Ny[k]^{N-1}]$	-	N-1
9. $b[k] = v^T[k].a[k]$	N-1	N
10. $x_p[k] = b[k].x[k]$	-	M_1
11. $X[k] = [x_p[k],\ldots,x_p[k-N_b], p[k-1],\ldots,p[k-N_a]]$	-	-
12. $p[k] = X[k].w[k]$	$M_1(N_a+N_b)$	$M_1(N_a+N_b+1)$
13. $Y[k] = [y[k],\ldots,y[k-N_b], q[k-1],\ldots,q[k-N_a]]$	-	-
14. $q[k] = Y[k].w[k]$	$N(N_a+N_b)$	$N(N_a+N_b+1)$
15. $u[k+1] = u[k] + (\alpha_u / (\|p[k]\|^2 + \delta)).p[k].e[k]$	$2M_1$	$2M_1+2$
16. $v[k+1] = v[k] + (\alpha_v / (\|q[k]\|^2 + \delta)).q[k].e[k]$	2N	2N+2
17. $w[k+1] = w[k] + (\alpha_w / (\|z[k]\|^2 + \delta)).z[k].e[k]$	$2(N_a+N_b)+2$	$2(N_a+N_b)+4$

Table 3. Joint NLMS Volterra filter + FIR filter cascade.

Operation	Additions	Multiplications
1. $y[k] = [\,y_0[k], y_1[k], \ldots, y_{N_v-1}[k]\,]^T$	-	M_v
2. $z[k] = v^T[k].y[k]$	N_v-1	N_v
3. $z[k] = [\,z[k], z[k-1],\ldots, z[k-M+1]\,]^T$	-	-
4. $d[k] = w^T[k].z[k]$	M_2-1	M_2
5. $e[k] = r[k] - d[k]$	1	-
6. $Y[k] = [\,y[k], y[k-1],\ldots, y[k-M+1]]$	-	-
7. $q[k] = Y[k].w[k]$	$(M_2-1)N_v$	M_2N_v
8. $v[k+1] = v[k] + (\alpha_v / (\|q[k]\|^2 + \delta)).q[k].e[k]$	$2N_v$	$2N_v+2$
9. $w[k+1] = w[k] + (\alpha_w / (\|z[k]\|^2 + \delta)).z[k].e[k]$	$2M_2$	$2M_2+2$

Table 4. Joint NLMS for Volterra Filter + IIR Filter Cascade: Output Error

Operation	Additions	Multiplications
1. $y[k] = [\,y[k], y[k-1],\ldots, y[k-N+1]\,]^T$	-	M_v
2. $z[k] = v^T[k].y[k]$	N_v-1	N_v
3. $z[k] = [z[k],\ldots,z[k-N_b],\ d[k-1],\ldots d[k-N_a]]^T$	-	-
4. $d[k] = w^T[k].z[k]$	N_a+N_b	N_a+N_b+1
5. $e[k] = r[k] - d[k]$	1	-
6. $Y[k] = [\,y[k],\ldots,y[k-N_b], q[k-1],\ldots q[k-N_a]]$	-	-
7. $q[k] = Y[k].w[k]$	$N_v(N_a+N_b)$	$N_v(N_a+N_b+1)$
8. $v[k+1] = v[k] + (\alpha_v / (\|q[k]\|^2 + \delta)).q[k].e[k]$	$2N_v$	$2N_v+2$
9. $w[k+1] = w[k] + (\alpha_w / (\|z[k]\|^2 + \delta)).z[k].e[k]$	$2(N_a+N_b)+2$	$2(N_a+N_b)+4$