THE PARTITIONED EXACT FREQUENCY DOMAIN BLOCK NLMS ALGORITHM

Bernhard Nitsch
Technische Hochschule Darmstadt, Institut für Netzwerk- und Signaltheorie
Merckstr. 25, D-64283 Darmstadt, Germany
e-mail: nitsch@nesi.e-technik.th-darmstadt.de
WWW: http://www.th-darmstadt.de/fb/et/signale/nitsch.html

ABSTRACT

Acoustic echo cancellers for hands free telephony require filter impulse response sizes between 128 and 256 ms to reach a sufficient error return loss enhancement. Using a sampling rate of 16 kHz yields filter sizes of 2000 to 4000 coefficients.

For real time implementations on digital signal processors (DSPs) time domain based adaptive filters need huge processing power. Filtering in frequency subbands [5] or using block adaptive filter algorithms can reduce the algorithm complexity significantly.

In this paper the Partitioned Exact Frequency Domain Block NLMS (PEFBNLMS) algorithm is presented which is mathematically an exact formulation of the time domain NLMS algorithm. The PEFBNLMS algorithm combines a computational complexity reduction of 3 to 5 times compared to the NLMS algorithm with the same tracking ability.

1. THE PARTITIONED FREQUENCY DOMAIN BLOCK LMS ALGORITHM (PFBLMS)

Partitioning the filter impulse response is known as a method to reduce the signal delay of large block adaptive filters [2] [3]. As figure 1 illustrates the filter $\underline{w}_{N\cdot L}$ is divided into N subfilters $\underline{w}_L^{(p)}$ where each subfilter calculates the estimation signal vector $\underline{y}_B^{(p)}$.

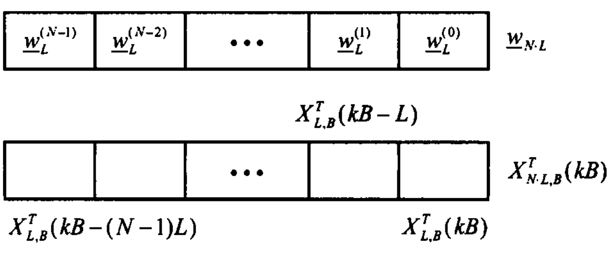

Figure 1: Partitioning the filter impulse response

$$\underline{y}_B(kB) = \sum_{p=0}^{N-1} \underline{y}_B^{(p)}(kB) = \sum_{p=0}^{N-1} X_{L,B}^T(kB - p\cdot L)\cdot \underline{w}_L^{(p)}(kB)$$

$$\underline{e}_B(kB) = \underline{d}_B(kB) - \underline{y}_B(kB)$$

with

$$\begin{aligned}
X_{L,B}(kB) &= [\ \underline{x}_L(kB-B+1)\ \cdots\ \underline{x}_L(kB)\ ] \\
\underline{x}_L^T(kB) &= [\ x(kB-L+1)\ \cdots\ x(kB)\ ] \\
\underline{e}_B^T(kB) &= [\ e(kB-B+1)\ \cdots\ e(kB)\ ] \\
\underline{d}_B^T(kB) &= [\ d(kB-B+1)\ \cdots\ d(kB)\ ] \\
\underline{w}_L^{(p)}(kB) &= [\ w^{(p+1)\cdot L-1}(kB)\ \cdots\ w^{(p\cdot L)}(kB)\ ]
\end{aligned}$$

$X_{L,B}(kB - p\cdot L)$ represents the delayed excitation signal matrix of order $(L\times B)$, $\underline{x}_B$ the excitation signal vector, $\underline{e}_B$ the error signal vector, $\underline{d}_B$ the desired signal vector and $\underline{w}_L^{(p)}$ the subfilter impulse responses. As the convolution part of the PFBLMS algorithm the correlation part can be partitioned. Using the same partition size L for convolution and correlation part yields the highest complexity reduction for fastest convergence.

$$\begin{bmatrix} \underline{w}_L^{(N-1)}(kB) \\ \vdots \\ \underline{w}_L^{(0)}(kB) \end{bmatrix} = \begin{bmatrix} \underline{w}_L^{(N-1)}(kB-B) \\ \vdots \\ \underline{w}_L^{(0)}(kB-B) \end{bmatrix} + \alpha \begin{bmatrix} X_{L,B}(kB-(N-1)L) \\ \vdots \\ X_{L,B}(kB) \end{bmatrix} \cdot \underline{e}_B(kB)$$

Using the overlap & save technique [4] convolution and correlation can be implementated efficiently with reduced numerical complexity in the frequency domain.

Convolution part:

$$\underline{y}_B^{(p)}(kB) = [0_{B,M-B}\ I_{B,B}]\cdot F_{M,M}^{-1} \cdot [\underline{\tilde{X}}_M(kB-p\cdot L)\otimes \underline{\tilde{W}}_M^{(p)}(kB)]$$

with the delayed excitation signal spectrum vector

$$\underline{\tilde{X}}_M(kB-p\cdot L)=F_{M,M}\cdot \underline{x}_M(kB-p\cdot L)$$

and the subfilter transfer function

$$\underline{\tilde{W}}_M^{(p)}(kB)=F_{M,M}\cdot\begin{bmatrix} R_{L,L}\cdot \underline{w}_L^{(p)}(kB)\\ 0_{M-L,M-L}\end{bmatrix}$$

$$R_{L,L}=\begin{bmatrix} 0 & 0 & \cdots & 0 & 1\\ 0 & 0 & \cdots & 1 & 0\\ & & \ddots & & \\ 0 & 1 & \cdots & 0 & 0\\ 1 & 0 & \cdots & 0 & 0\end{bmatrix}.$$

Where $F_{M,M}$ resp. $F_{M,M}^{-1}$ are the Fourier matrix resp. the inverse Fourier matrix, $0_{B,M-B}$ the zero matrix and $I_{B,B}$ the identity matrix. The operator $\otimes$ performs an element wise multiplication of two complex valued vectors.

In the same way as the convolution part the compensator vector update of the correlation part is calculated in the frequency domain

$$\begin{aligned} X_{L,B}(kB-p\cdot L)\cdot\underline{e}_B(kB)= &\\ &[R_{L,L}\ 0_{L,M-L}]\cdot F_{M,M}^{-1}\\ &\cdot[\underline{\tilde{X}}_M^*(kB-p\cdot L)\otimes\underline{\tilde{E}}_M(kB)]\end{aligned}$$

with the error signal spectrum vector

$$\underline{\tilde{E}}_M(kB)=F_{M,M}\cdot\begin{bmatrix} 0_{M-B,M-B}\\ \underline{e}_B(kB)\end{bmatrix}.$$

If we assume that $\frac{L}{B}$ is an integer the excitation signal spectrums $\underline{\tilde{X}}_M(kB-p\cdot L)$ need not to be calculated for each subfilter in the convolution and correlation part. By using a tap delay line in frequency domain the instant spectrum $\underline{\tilde{X}}_M(kB)$ can be reused for subfilters of higher order (p) in following block sampling times.

2. THE PARTITIONED EXACT FREQUENCY DOMAIN BLOCK NLMS ALGORITHM (PEFBNLMS)

For correlated input processes like speech the PFBLMS algorithm shows a poor tracking behavior because the sub diagonal elements of the error signal correction matrix $\breve{S}_{B,B}(kB)$ are not small compared with main diagonal elements [1]. To improve the convergence speed we correct the error signal vector of the PFBLMS algorithm $\underline{e}_B^{PFBLMS}$ in such a way that this vector is equal to the NLMS algorithm error signal vector $\underline{e}_B^{NLMS}$. Using the corrected error signal vector $\underline{e}_B^{NLMS}$ in the correlation part of the adaptive filter results the same tracking behavior as the NLMS algorithm.

Starting from the NLMS algorithm equations in the time domain

$$e(k)=y(k)-\underline{x}_{NL}^T(k)\cdot\underline{w}_{NL}(k-1)$$

$$\underline{w}_{NL}(k)=\underline{w}_{NL}(k-1)+\frac{\alpha(k)}{\|\underline{x}_{NL}(k)\|^2}\cdot\underline{x}_{NL}(k)\cdot e(k)$$

we write the error signal $e(k)$ for sampling times from $kB-B+1$ to kB

$$\begin{aligned} e(kB-B+1)&=y(kB-B+1)\\ &\quad-\underline{x}_{NL}^T(kB-B+1)\cdot\underline{w}_{NL}(kB-B)\\ e(kB-B+2)&=y(kB-B+2)\\ &\quad-\underline{x}_{NL}^T(kB-B+2)\cdot\underline{w}_{NL}(kB-B)\\ &\quad-\frac{\alpha(kB-B+1)}{\|\underline{x}_{NL}(kB-B+1)\|^2}\cdot e(kB-B+1)\\ &\quad\cdot\underline{x}_{NL}^T(kB-B+2)\cdot\underline{x}_{NL}(kB-B+1)\\ &\ldots\\ e(kB)&=y(kB)\\ &\quad-\underline{x}_{NL}^T(kB)\cdot\underline{w}_{NL}(kB-B)\\ &\quad-\frac{\alpha(kB-B+1)}{\|\underline{x}_{NL}(kB-B+1)\|^2}\cdot e(kB-B+1)\\ &\quad\cdot\underline{x}_{NL}^T(kB)\cdot\underline{x}_{NL}(kB-B+1)\\ &\quad-\ldots\\ &\quad-\frac{\alpha(kB-1)}{\|\underline{x}_{NL}(kB-1)\|^2}\cdot e(kB-1)\\ &\quad\cdot\underline{x}_{NL}^T(kB)\cdot\underline{x}_{NL}(kB-1).\end{aligned}$$

This system of equations can be rewritten in matrix form

$$\begin{aligned}\underline{e}_B^{NLMS}(kB)&=[I_{B,B}+S_{B,B}(kB)]^{-1}\\ &\quad\cdot[\underline{y}_B(kB)-X_{NL,B}^T(kB)\cdot\underline{w}_{NL}(kB)]\\ &=[I_{B,B}+\breve{S}_{B,B}(kB)]^{-1}\\ &\quad\cdot\underline{\breve{e}}_B^{PFBLMS}(kB)\end{aligned}$$

with the normalized error signal vector of the PFBLMS algorithm

$$\begin{aligned}&\breve{e}^{PFBLMS}(kB)=\\ &\begin{bmatrix}\frac{\alpha(kB-B+1)}{\|\underline{x}_{NL}(kB-B+1)\|^2} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \frac{\alpha(kB)}{\|\underline{x}_{NL}(kB)\|^2}\end{bmatrix}\\ &\cdot[\underline{y}_B(kB)-X_{NL,B}^T(kB)\cdot\underline{w}_{NL}(kB)]\end{aligned}$$

and the lower triangle error signal correction matrix

$$\breve{S}_{B,B}(kB)=$$

$$\begin{bmatrix} 0 & 0 & \cdots & 0\\ \hat{s}_{xx}(kB-B+2,-1) & 0 & \cdots & 0\\ \hat{s}_{xx}(kB-B+3,-2) & \hat{s}_{xx}(kB-B+3,-1) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \hat{s}_{xx}(kB,-B+1) & \hat{s}_{xx}(kB,-B+2) & \cdots & 0\end{bmatrix}$$

B	L	M	N	L_{AKF}	MR	AC
128	128	256	16	128	2.41	0.30
128	128	256	16	64	2.39	0.27
128	384	512	6	128	3.47	0.25
128	384	512	6	64	3.46	0.22
256	256	512	8	256	2.81	0.29
256	256	512	8	128	2.78	0.22
256	768	1024	3	256	3.94	0.25
256	768	1024	3	128	3.92	0.19

Table 1: Relative algorithm complexity (AC) and relative memory requirement (MR) of the PEFBNLMS- compared with the NLMS algorithm

which contains autocorrelation coefficients

$$\hat{s}_{xx}(k,l) = \underline{x}_L^T(k) \cdot \underline{x}_L(k+l).$$

For the error vector correction an estimation of autocorrelation coefficients $\hat{s}_{xx}(k,l)$ and the inversion of the lower triangle matrix $[I_{B,B} + \check{S}_{B,B}(kB)]$ must be carried out. The most expensive operation concerning memory occupation and computational complexity represents the autocorrelation function update which can be calculated recursively.

$$\hat{s}_{xx}(k,l) = \hat{s}_{xx}(k-1,l) + x(k) \cdot x(k+l) - x(k-L) \cdot x(k-L+l)$$

To avoid an instable behavior due to an accumulation of numerical round-off errors the autocorrelation function $\hat{s}_{xx}(k,l)$ must be reinitialized periodically.
Combining the estimation of $\hat{s}_{xx}(k,l)$ and the matrix inversion the memory occupation can be reduced to $2 \cdot B$ words necessary to store the autocorrelation coefficients.
Additional complexity reduction can be achieved if less autocorrelation coefficients L_{AKF} are used for the error vector correction. The tracking ability of the PEFBNLMS algorithm is still sufficient if we use $L_{AKF} = \frac{B}{2}$ coefficients in case of speech as excitation signal.
In table 1 the numerical complexity measured by the number of multiplications and the memory requirement of the PEFBNLMS algorithm is compared with the NLMS algorithm. For compensator impulse response sizes between 2000 and 2300 coefficients and block sizes between 128 to 256 samples the PEFBNLMS algorithm needs up to 5 times less computational power than the NLMS algorithm.

3. REAL-TIME MEASUREMENTS

To compare the tracking ability of the PEFBNLMS and the NLMS algorithm both algorithms were implemented in an echo compensation environment working with a sampling rate of 8 kHz as shown in figure 2. We use the floating point DSP TMS320C44 of Texas Instruments.

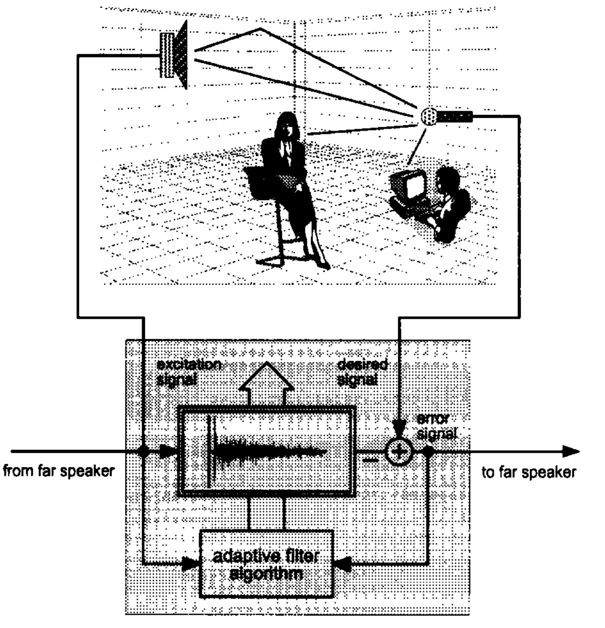

Figure 2: Echo compensation environment

For reproducible measurements the loudspeaker-room-microphone system (LRMS) was simulated by two measured impulse responses of an office room which were truncated after 920 coefficients. To illustrate the tracking behavior a switch between the two impulse responses was performed after 14 seconds.

The algorithms were tested with speech as excitation signal which contains several speech pauses. To simulate room background noise white noise was added to the microphone signal. The SNR over time can be examined in figure 3.

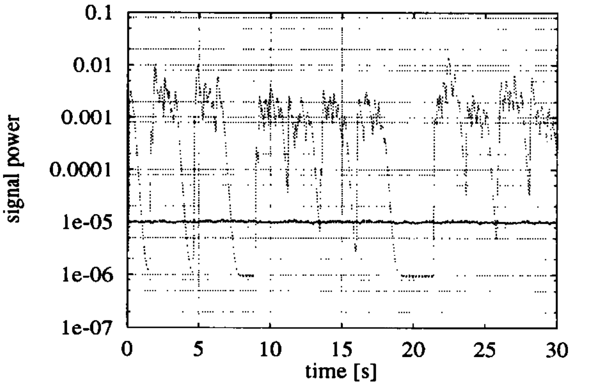

Figure 3: SNR of the desired signal

... LRMS output signal power
– background noise signal power

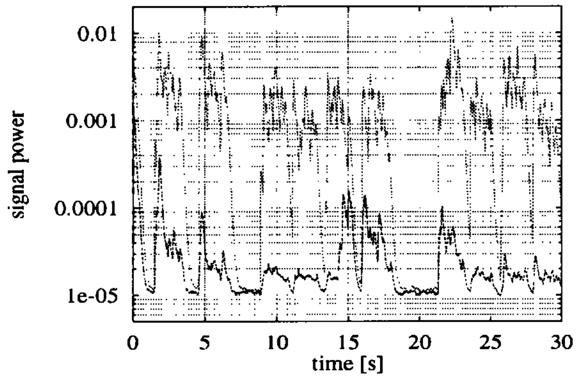

Figure 4: Convergence speed of the PEFBNLMS algorithm

$L = 192 \quad B = 64 \quad N = 5$
$M = 256 \quad L_{AKF} = 64 \quad \alpha = 1.0$
··· desired signal power
– error signal power

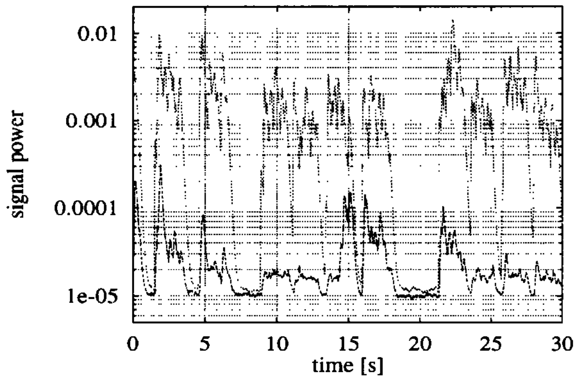

Figure 5: Convergence speed of the NLMS algorithm

$L = 940 \quad \alpha = 1.0$
··· desired signal power
– error signal power

Figures 4 to 6 depict the desired- and errror signal powers over time. For a better clearness all curves are smoothed by a first order IIR-filter

$$\overline{x}^2(k) = \lambda \cdot \overline{x}^2(k-1) + (1-\lambda) \cdot x^2(k), \quad \lambda = 0.999.$$

A comparison of figures 4 and 5 indicates the comparability of convergence speed of the PEFBNLMS and the NLMS algorithm. Due to periodical reinitialisation of the autocorrelation coefficients of the correction matrix $\check{S}_{B,B}(kB)$ the PEFBNLMS algorithms shows a stable behavior for hours even after the measurement time of 30 seconds.
Using less than B autocorrelation coefficients for the error signal correction yields additional algorithm complexity reduction. As figure 6 illustrates the convergence speed is not reduced significantly if we use $\frac{B}{2}$ autocorrelation coefficients instead of B.

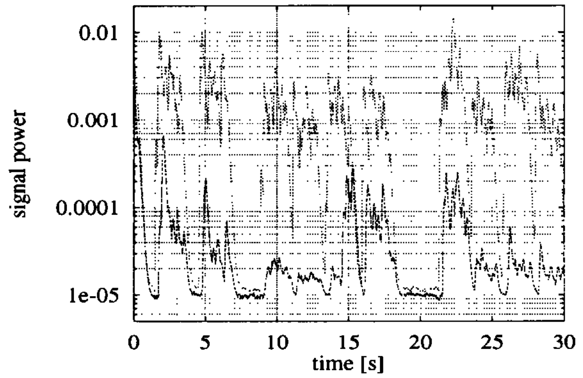

Figure 6: Convergence speed of the PEFBNLMS algorithm

$L = 192 \quad B = 64 \quad N = 5$
$M = 256 \quad L_{AKF} = 32 \quad \alpha = 1.0$
··· desired signal power
– error signal power

4. CONCLUSIONS

In this paper a mathematical exact formulation of the time domain NLMS algorithm was derived for the frequency domain. The PEFBNLMS algorithm combines a reduced numerical complexity with the same tracking ability as the NLMS algorithm.
The theoretical results were reconsidered by real-time measurements in an acoustic echo compensation environment. It was shown that the PEFBNLMS algorithm behave stable if the error signal correction matrix was reinitialized periodically.

REFERENCES

[1] Jacob Benesty and Pierre Duhamel. A Fast Exact Least Mean Square Adaptive Algorithm. *IEEE Trans. on Signal Processing*, volume 40, December 1992.

[2] G.P.M. Egelmeers. *Real Time Realization Concepts of Large Adaptive Filters.* PhD thesis, Technical University Eindhoven, 1995.

[3] P. Estermann. *Adaptive Filter im Frequenzbereich: Analyse und Entwurfsstrategie.* PhD thesis, Eidgenössische Technische Hochschule Zürich, 1996.

[4] Gregory A. Clark, Sydney R. Parker and Sanjit K. Mitra. A Unified Approach to Time- and Frequency-Domain Realizations of FIR Adaptive Digital Filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, volume 31, October 1983.

[5] B. Nitsch and S. Binde. Stabilizing the LFTF algorithm by Leakage control. In *Proc. EUSIPCO-96, Eighth European Conference On Signal Processing*, pages 1389–1392, September 1995.