IMPROVED NOISE CANCELLATION IN DISCRETE COSINE TRANSFORM DOMAIN USING ADAPTIVE BLOCK LMS FILTER

Sanjay Kumar Gupta[1], Jitendra Jain[2], Rahul Pachauri[3]

[1]*Student (M.Tech.), Electronics & Communication, Jaypee University of Engineering & Technology, Madhya Pradesh, India,* ***sanjay.newage13@gmail.com***
[2]*Student (M.Tech.), Electronics & Communication, Jaypee University of Engineering & Technology, Madhya Pradesh, India,* ***jitendrajn69@gmail.com***
[3]*Senior Lecturer, Electronics & Communication, Jaypee University of Engineering & Technology, Madhya Pradesh, India,* ***rahul.pachauri@juet.ac.in***

Abstract

This paper illustrates the advantages of Discrete Cosine Transform (DCT) when it is used in Block least mean square (BLMS) adaptive filter. DCT provides very good energy compaction in speech signals, image signals. So the signal to noise (SNR) is improved. Noise cancellation, Echo cancellation, System identification are the application of adaptive filter. We have proposed the Noise cancellation in DCT Domain and also compare with exist Fast Block LMS (FBLMS). The simulation result shows better signal to noise ratio (SNR) in DCT Domain rather than FFT Domain.

Index Terms: *Fast Block LMS (FBLMS), Least Mean Square (LMS) algorithm, DCT.*

--- *** ---

1. INTRODUCTION

It is often necessary to perform speech enhancement through removal of noise from speech processing system operating in noisy environment. The noise degrades the performance of speech and voice recognition system. There are certain applications of signal processing that require adaptive filters the length of which exceed a few hundred or even a few thousand taps. In the application of speech processing in which the noise coming with speech signal, channel equalization and echo cancellation may require adaptive filter with exceedingly long lengths. For these application LMS algorithm is computationally expensive to implement. Because filter weight is changed for each sample so it requires more time to be processed. So Block processing of data samples significantly reduces the computational complexity of adaptive filters. In Block processing, a block of samples of the filter and desired output are collected and then processed together to obtain a block of output samples and filter weight is changed for each block. And when we do it in frequency domain this is called FBLMS.

The topic Noise cancellation and many speech enhancements is widely researched and many speech algorithms make use of FFT to make it easier to remove noise embedded in the noisy speech signal. In transform domain it is easy to separate the speech energy and noise energy for example energy of white noise is uniformly spread through the entire spectrum, but the energy of speech, concentrated in certain frequencies.

We have proposed the block LMS (BLMS) algorithm in discrete cosine transform domain for noise cancellation in speech processing. DCT is most favourable for the compression of speech and image data. This is very useful in speech filtering, data compression and feature extraction. When the colored noise is mixed with speech or the signal is highly correlated then DCT exactly de-correlate the signal. DCT output consists of signal components separated in to individual frequency bands. DCT exhibits excellent energy compaction for highly correlated signals. The uncorrelated signal has its energy spread out, whereas the energy of the correlated image is packed into the low frequency region. It will be shown in this paper how DCT perform well as well as FFT. This improvement helps to reduce the residual noise present, which is musical in nature.

2. DCT

DCT is widely used in image compression because of its excellent energy compaction property. This is a useful feature for noise removal purpose too. If the speech energy can be concentrated predominantly into a few coefficients while the noise energy remains white, reduction of noise can be achieved easily. As it was shown in previous work on speech coding [1,7], DCT provides significantly higher energy compaction as compared to the FFT. DCT has the added advantage of higher spectral resolution than the FFT for the same window size. For a window size of N, the DCT has N independent spectral components while the FFT only produces N/2 + 1 independent spectral component, as the other components are just complex conjugates. This is yet another point in favor of the use of the DCT. FFT only attempt to correct the noisy amplitude but not the phase component. This actually results in an upper bound on the maximum improvement in SNR possible. If DCT is used, a higher upper bound is possible. The effect of non-corrected phase on the speech discussed in [5]. If the phase is out by more than π /8, the speech becomes rough. If the phase is replaced by random noise uniformly distributed between -π to π, a rough and completely unvoiced speech is obtained. On the other hand, if the phase is replaced by zero, the reconstructed speech sounds completely voiced and monotonous. Therefore it is not correct to view the phase as totally unimportant and especially for high levels of additive noise; the reconstructed speech quality will be affected. For DCT, the coefficients are real and can be considered to have a binary phase value. The phase will depend only on the sign of the coefficient. This provides a better degree of noise margin as unless the added noise changes the sign of the coefficient, the phase is unchanged. Therefore if strong speech energy is present in a particular coefficient, it is unlikely that the phase will be corrupted. If the noise energy is much higher than the speech energy in a particular coefficient resulting in an erroneous phase, the coefficient will be highly attenuated thus minimizing the effect of the erroneous phase. It is therefore likely that DCT would perform better than FFT.

One dimensional DCT is given as follows: [1,8]

Forward transform of a sequence $\{x(n), 0 \leq n \leq N-1\}$ is given by,

$$X(k) = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \quad 0 \leq k \leq N-1 \tag{1}$$

Where,

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}} \ for\ 1 \leq k \leq N-1$$

The inverse Transform is given by,

$$x(n) = \sum_{k=0}^{N-1} \alpha(k) X(k) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \quad 0 \leq k \leq N-1 \tag{2}$$

3. DCT BASED BLOCK LMS ADAPTIVE FILTER (DBBLAF)

3.1. Fast Block LMS Algorithm

Consider a BLMS based adaptive filter that takes an input sequence x(n), which is portioned into non-overlapping blocks of length P each by means of a serial to parallel converter, and the block of data is so produced are applied to an FIR filter of length N, one block at a time. The tap weights of the filter are updated after the collection of each block of data samples, so that the adaption of the filter proceeds on a block-by-block basis rather than on a sample by sample basis as in conventional LMS algorithm. [3, 7]

$$W(i+1) = W(i) + \mu[RW - P] \tag{3}$$

Where, $R = \frac{1}{L}\sum_{r=0}^{L-1} x(iL+r)x^t(iL+r)$

$$P = \frac{1}{L}\sum_{r=0}^{L-1} x(iL+r)d(iL+r)$$

L is the Block length, R is the autocorrelation matrix and P, cross correlation vector.

The weight update equation is,

$$W(i+1) = W(i) + \mu_B \frac{1}{L}\sum_{r=0}^{L-1} x(iL+r)e(iL+r) \tag{4}$$

The error vector is,

$$e(iL+r) = d(iL+r) - y(iL+r) \tag{5}$$

The output is,

$$y(iL+r) = W^t(i)x(iL+r) \tag{6}$$

Define the matrix (kth block),

$$X(i) = [X(iL)\ X(iL+1)\ \dots\ X(iL+L-1)]^T \tag{7}$$

The column Vectors are,

$$d(i) = [d(iL)\ d(iL+1)\ \dots\ d(iL+L-1)]^T \tag{8}$$

$$y(i) = [y(iL)\ y(iL+1)\ \dots\ y(iL+L-1)]^T \tag{9}$$

$$e(i) = [e(iL)\ e(iL+1)\ \dots\ e(kL+L-1)]^T \tag{10}$$

$$Block\ Interval = \frac{2}{4L\mu_B\lambda_{\max}} \tag{11}$$

$$Sample\ Interval = \frac{2}{4\mu_B\lambda_{\max}} \tag{12}$$

Step size should be bounded as,

$$0 < \mu_B < \frac{2}{L\lambda_{\max}} \tag{13}$$

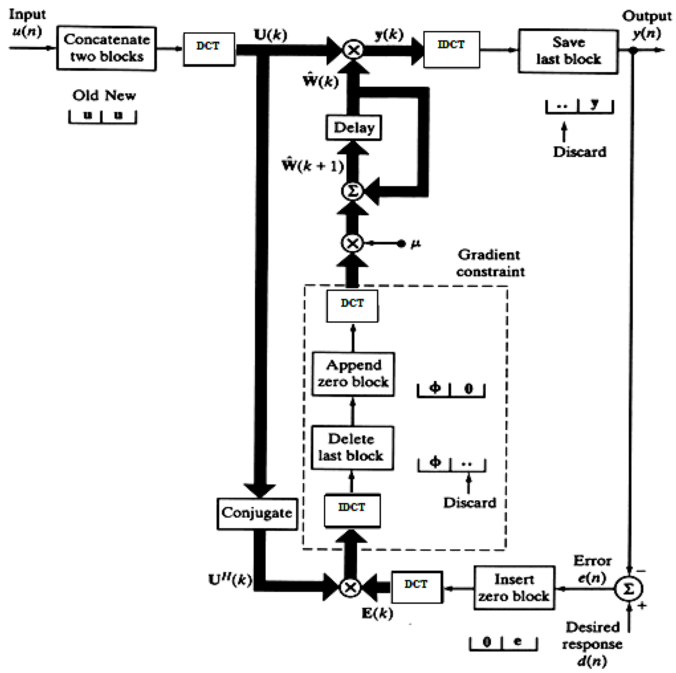


Fig-1 shows the DBBLAF Adaptive filter, [6]

3.2. DBBLAF Algorithm

Input: Tap weight vector, $w_f(k)$

Signal power estimates $\sigma_{x,F}^2(k-1)s$

Extended input vector,

$x(k) = [x(kL-N+1) \dots\dots\dots\dots x(kL+L-1)]^T$

Desired output vector,

$d(k) = [d(kL-N+1) \dots\dots\dots\dots d(kL+L-1)]^T$

Output: Filter output

$y(k) = [y(kL-N+1) \dots\dots\dots y(kL+L-1)]^T$

Tap weight vector update $w_f(k+1)$

1. Filtering:

$$x_f(k) = DCT(x(k))$$

$$y(k) = \text{the last element of } IDCT(x_f(k) \otimes w_f(k))$$

2. Error estimation:

$$e(k) = d(k) - y(k)$$

3. Step normalization:

$$\sigma_{x_{F,i(k)}}^2 = \beta\sigma_{x_{F,i(k-1)}}^2 + (1-\beta)\left|x_{F,i}(k)\right|^2$$

$$\mu(k) = [\mu_0(k)\,\mu_1(k) \,\dots\dots\dots\, \mu_{N'-1}(k)]^T$$

$$\mu_i(k) = \frac{\mu_0}{\sigma_{F,i(k)}^2}$$

4. Tap weight adaption:

$$e_f(k) = DCT\begin{pmatrix} 0 \\ e(k) \end{pmatrix}$$

$$w_f(k+1) = w_f(k) + 2\mu(k)x_f^*(k) \otimes e_f(k)$$

Note:

- N is the filter length, L is the block length
- N'=N+L-1
- $\otimes$ denotes the element by element multiplication
- σ_i^2 is the power estimates of the sample of filter input in the frequency domain.

4. ADAPTIVE NOISE CANCELLATION

Separating a signal from additive noise is a common problem in signal processing. Fig-2 [4, 6] shows the adaptive noise cancellation technique. This approach is viable only when an additional reference input is available containing noise n_1, which is correlated with the original corrupting noise n_0. In figure the adaptive filter receives the reference noise, filter it, and subtract the result from the noisy "primary input," $s+n_0$. For the adaptive filter, the noisy input $s+n_0$ acts as the desired response. The "system output" acts as the error for the adaptive filter.

One might think that some prior knowledge of the signal *'s'* or of the noises n_0 and $n_1$would be necessary before the filter could adapt to produce the noise-cancelling signal *y*. A simple argument will show, however, that little or no prior knowledge of *s*, n_0, or n_1 of their interrelationships is required.

Assume that s, n_0, n_1 and *y* are statistically stationary and have zero means. Assume that *s* is uncorrelated with n_0 and n_1 and suppose that n_1 is correlated with n_0. That output is [4]

$$\epsilon = s + n_0 - y \qquad (14)$$

Squaring, one obtains

$$\epsilon^2 = s^2 + (n_0 - y)^2 + 2s(n_0 - y) \qquad (15)$$

Taking expectation of both sides of equation 14, and realizing that *s* is uncorrelated with n_0 and with *y*, yields

$$E[\epsilon^2] = E[s^2] + E[(n_0-y)^2] + 2E[s(n_0-y)]$$
$$= E[s^2] + E[(n_0-y)^2] \qquad (16)$$

Adapting the filter to minimize $E[\epsilon^2]$ will not affect the signal power $E[s^2]$. Accordingly, the minimum output power is

$$E_{min}[\epsilon^2] = E[s^2] + E_{min}[(n_0-y)^2] \qquad (17)$$

When the filter is adjusted so that $E[s^2]$ is minimized, therefore $E[(n_0 - y)^2]$, is also minimized. The filter output y is then best least-squares estimate of the primary noise n_0. Moreover, when $E[(n_0 - y)^2]$ is minimized, $E[(\epsilon - y)^2]$ is also minimized, since, from equation 5,

$$(\epsilon - s) = (n_0 - y) \qquad (18)$$

Adjusting or adapting the filter to minimize the total output power is tantamount to causing the output ϵ to be a best least-square estimate of the signal s for the given structure and adjustability of the adaptive filter and for the given reference input. [4]

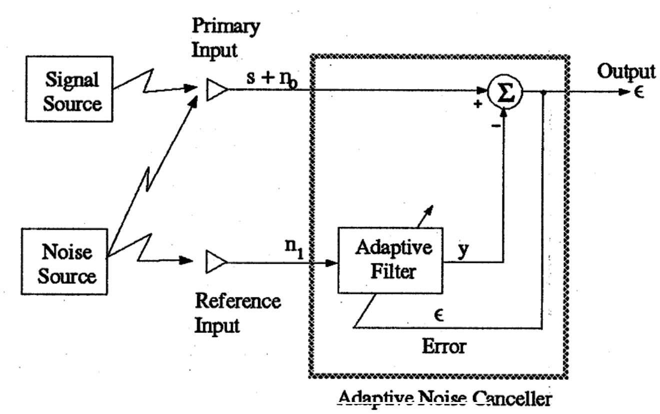


Fig-2: Noise Cancellation

5. SIMULATION RESULT

In the following simulation, the speech signal and engine noise is recorded. We add both the signals with white noise. We have taken the filter length like 8, 16, 32, 64, 128 and different step size. And it is filtered through FBLMS and DBBLAF and calculates the input and output SNR.

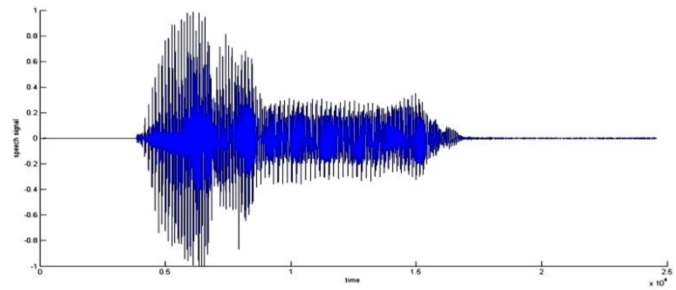
From the simulation, we may see the DBBLAF provide the high SNR.

Table 1 shows the comparison of FBLMS and DBBLAF filtering with different-2 step size and filter length.

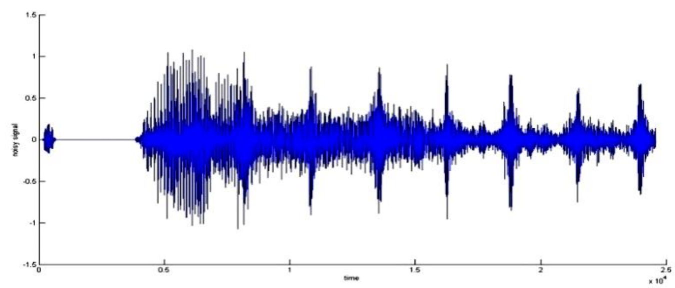
Table-1: Comparison of FBLMS and DBBLAF

Step Size	Input SNR	Output SNR					
		FBLMS			DBBLAF		
		N=16	N=32	N=64	N=16	N=32	N=64
0.004	2.73	17.32	13.37	10.03	18.62	15.20	11.91
0.008	2.73	16.58	12.31	12.35	17.44	16.98	14.31
0.01	2.73	14.25	11.16	10.20	16.49	15.33	15.06
0.02	2.73	11.80	9.20	8.35	15.01	14.35	13.05

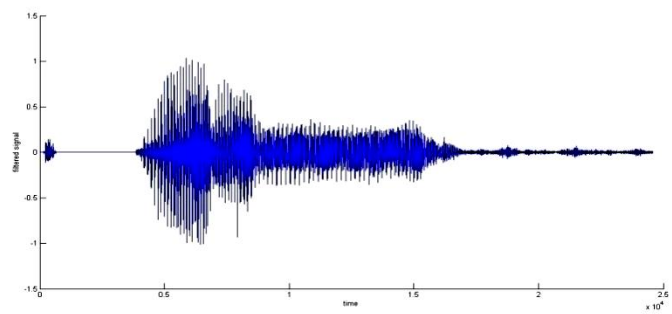
In Fig-3, a shows the Speech signal, b shows the engine noise added with speech signal, c shows the FBLMS output and d shows the DBBLAF output with N=16, β=.99.



(a)



(b)



(c)

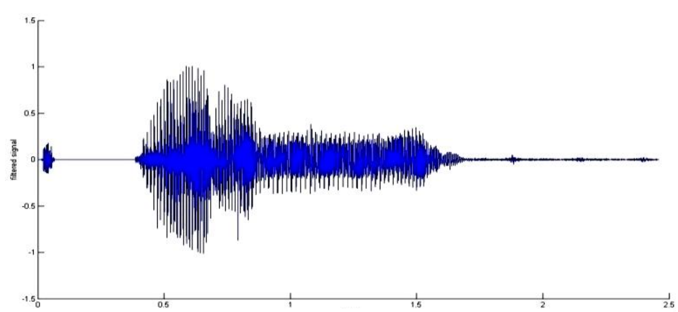


Fig-3: Simulation result:

(a) Speech Signal, (b) Signal with engine noise,

(c) FBLMS Result, (d) DBBLAF Result

6. CONCLUSION

This paper clearly shows the feasibility of using DCT for noise cancellation. The use DBBLAF generates better results as compared to FBLMS. The improvement is also more noticeable for engine noise and white noise. Although the algorithm is simulated with the DCT, it should also be applicable to other types of transform such as the wavelet transform, Fractional Fourier transform.

REFERENCES

[1] N. Ahmed, T. Natarajan, K.R. Rao, "Discrete cosine transform", *IEEE Trans.* Comput. C-23 (1974) 90-93.

[2] Earl R. Ferrara, "Fast Implementation of LMS Adaptive Filters", *IEEE Transactions* on acoustics, speech, and signal processing, vol. Assp-28, no. 4, august 1980.

[3] S.S. Narayan and A.M. Peterson, "Frequency domain least mean square algorithm", *Proc. IEEE*, 69(1):124-126, January 1981.

[4] G.A. Clark, S.K. Mitra and S.R. Parker, "Block Implementation Of Adaptive Digital Filters", *IEEE Trans. Circuit Syst.*, Vol CAS-28, PP. 584-592, June 1981.

[5] B. Widraw and S.D. Stearns, "Adaptive signal Processing", Prentice-Hall, Englewood Cliffs, NJ, 1985.

[6] P. Vary, "Noise suppression by spectral magnitude estima-tion - Mechanism and theoretical limits", Signal Processing 8 (1985) 387-400.

[7] S. Haykin, " Adaptive Filter Theory", Englewood Cliffs, NJ: Prentice-Hall, 1986.

[8] B. Farhang-Boroujeny, "Adaptive Filters: Theory and Applications",Chichester, Ed. Wiley, 1998.

[9] Ing Yann Soon, Soo Ngee Koh, Chai Kiat Yeo, "Noisy speech enhancement using discrete cosine transform", *ELSEVIER Science*, Speech communication 24 (1998) 249-257.

[10] D.P. Das, G. Panda and S.M. Kuo, "New block filtered-X LMS algorithms for active noise control systems", IET Signal Process., 2007, 1, (2), pp. 73–81.