SPEECH ENHANCEMENT USING FAST HARTLEY TRANSFORM

Chennamsetti . S P , Munish SR , CA Jalaludeen,
Agere Systems India Pvt Ltd,
Unit 4 , Innovator block, ITPL, Bangalore, India,
Tel : +91-80-8098208 ; Fax : 91-80-8411 839
e-mail : *sivaramprasad@agere.com*

ABSTRACT

In automotive hands free communication, speech is mostly corrupted by engine and other background noises. Engine noise can be removed using frequency domain adaptive filtering as proposed by Benesty [1, 2]. Subband speech processing can be used to further enhance the speech signal by suppressing residual and background noise, as proposed by Diethorn [3]. For both these algorithms, significant computational savings can be obtained by using faster time-to-frequency transformation techniques and reducing order of calculations in the frequency domain. Benesty and Diethorn [1, 3] have formulated the noise canceller and noise suppressor algorithms using Fast Fourier Transform (FFT). In this paper, we reformulate the problem using Fast Hartley Transform (FHT) for both the adaptive noise canceller and subband noise suppressor algorithms,which reduces the computational complexity to half that of FFT based implementation.

1. INTRODUCTION

The Fast Fourier transform uses two properties of the discrete Fourier transform to reduce the order of computations. The first property is that the kernel for the Fourier transform is periodic. The second property is the shift rule, which states that shifting a function in time domain is equivalent to multiplying the function by a complex exponential in the frequency domain. The Hartley transform also has similar properties. The Hartley kernel is periodic, and shifting a function in time domain is equivalent to real multiplication with real kernel. Since Hartley transform uses only real numbers, it requires only half the number of multiplication compared to FFT.

The main difference between the Discrete Fourier Transform (DFT) and the Discrete Hartley Transform (DHT) is the core kernel [5]. For the DHT, the kernel is real unlike the complex exponential kernel of the DFT. The analysis equation for the Discrete Hartley Transform for an N-point sequence is given by Equation (1).

$$X(k) = \sum_{n=0}^{N-1} x(n)\,[\cos(2\pi kn/N) + \sin(2\pi kn/N)]$$

$$\text{for} \quad k = 0,\ldots\ldots,N-1 \qquad \ldots\ldots(1)$$

This results in the replacement of complex multiplications (each complex multiplication requires four real multiplications and two real additions) in a DFT by real multiplications in a DHT. For the DHT, this computation involves only two real multiplications and one real addition. The Hartley transform removes redundancy in the Fourier domain by repacking the numbers through the relation

$$\text{DHT}(k) = \text{Re}\,[\text{DFT}(k)] - \text{Im}\,[\text{DFT}(k)] \quad \ldots (2)$$

There exists an inexpensive mapping of coefficients from the Hartley domain to the Fourier domain, which can be used to convert the output of a DHT to the traditional DFT coefficients. Equation (3) gives the relation of DFT coefficients to the DHT coefficients for a N-point DFT computation.

$$\text{Re}(\text{DFT}(k)) = 0.5\,(\text{DHT}(k) + \text{DHT}(N-k))$$

$$\text{Im}(\text{DFT}(k)) = 0.5\,(\text{DHT}(k) - \text{DHT}(N-k))$$

$$\ldots\ldots\ldots\ldots(3)$$

However, instead of using the FHT as a means of computing the DFT, operations like convolution, cross-correlation and auto-correlation can be done in the Hartley domain itself. For example, the equation for convolution in the Fourier space is

$$Y_f(k) = X_f(k) * H_f(k) \qquad \ldots\ldots(4)$$

This involves four multiplication's, two additions and four memory accesses for each k. The convolution formula in Hartley space is

$$Y_h(k) = 0.5 * [\, X_h(k) * (\, H_h(k) + H_h(-k)\,) + X_h(-k) * (\, H_h(k) + H_h(-k)\,)\,] \quad ....(5)$$

The equation involves three multiplications, three additions and four memory accesses for each k. Though the number of arithmetic operations are comparable, the memory requirement is halved since only real arrays need to be stored and manipulated.

Also, the FHT is a bilateral transform. Like DFT, it uses the same functional form for both the forward and inverse transforms.So hartley domain transforms like FHT works well for processing real valued signals.

2. SPEECH ENHANCEMENT FOR HANDS FREE COMMUNICATION SYSTEMS

The speech enhancement algorithm can be mainly divided in two parts – Adaptive noise canceller and subband noise suppressor. The Noise canceller is a correlation canceller, which uses the reference noise signal to eliminate noise from the speech. The adaptive filtering is followed by subband speech enhancement technique as described by Diethorn [3], which is referred to as noise suppressor. Our recommendation of the noise canceller is a Hartley domain adaptive filter which is computationally more efficient compared to Fourier domain.

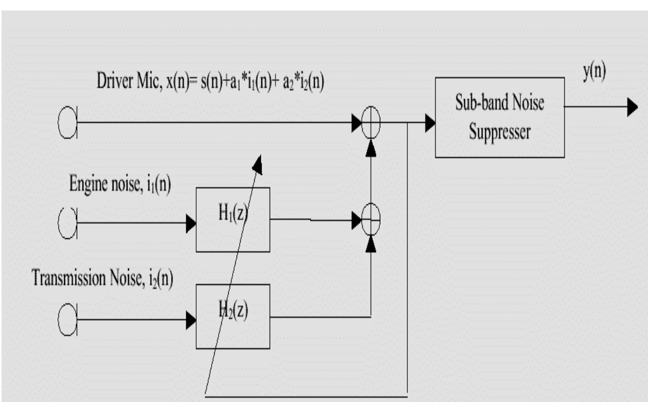


Fig1:Block Diagram for speech enhancement for hands-free communication system

2.1 ADAPTIVE NOISE CANCELLER

Adaptive filters play an important role in signal processing. In situations where we need to identify and track unknown and time-varying channels, adaptive filters have proven to be an effective tool [2]. There are roughly two classes of adaptive filters. One class of filters operates in the time domain, using sample-by-sample processing. The other class consists of filters that work in frequency domain, using block processing. There are two reasons for the use of frequency domain adaptive filters. One is block implementation of FIR filter using the FFT, which allows efficient use of fast convolution and fast correlation in the algorithm. The other reason is to improve the estimation accuracy of the standard LMS algorithm. The arithmetic complexity can be significantly reduced using Fast Hartley Transform instead of Fast Fourier Transform. The complete algorithm of Hartley domain adaptive noise canceller is given below .

Initialization:

$\mathbf{W}(0)$ = 2M-by-1 null vector

$P_i(0) = \delta_i, \quad i = 0, \ldots\ldots\ldots, 2M-1$

Notations:

0 = M-by-1 null vector

FHT = Fast Hartley transformation

IFHT = Inverse Fast Hartley transformation

α = adaptation constant

Computation: For each new block of M input samples, compute

$$\mathbf{U}(k) = \mathrm{diag}\{\, \mathrm{FHT}[\, u(kM-M), \ldots, u(kM-1), u(kM), \ldots, u(kM+M-1)\,]^T \}$$

$\mathbf{y}(k)$ = last M elements of IFHT [$\mathbf{U}(k)\mathbf{W}(k)$]

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{y}(k)$$

$$\mathbf{E}(k) = \mathrm{FHT}[\, \mathbf{0} \;\; \mathbf{e}(k)\,]$$

$$P_i(k) = \gamma P_i(k-1) + 0.5\,(1-\gamma)\,(\, U_i(k)^2 + U_i(N-k)^2\,)$$

$$i = 0,1,2,,\ldots\ldots 2M-1$$

$\mathbf{D}(k)$ = diag $[P_0^{-1}(k), P_1^{-1}(k), \ldots\ldots P_{2M-1}^{-1}(k)]$
$\phi(k)$ = first M elements of IFHT $[\mathbf{D}(k)\, \mathbf{U}^H(k)\, \mathbf{E}(k)]$

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \alpha \text{ FHT } [\ \phi(k)\ \ \mathbf{0}\ ]$$

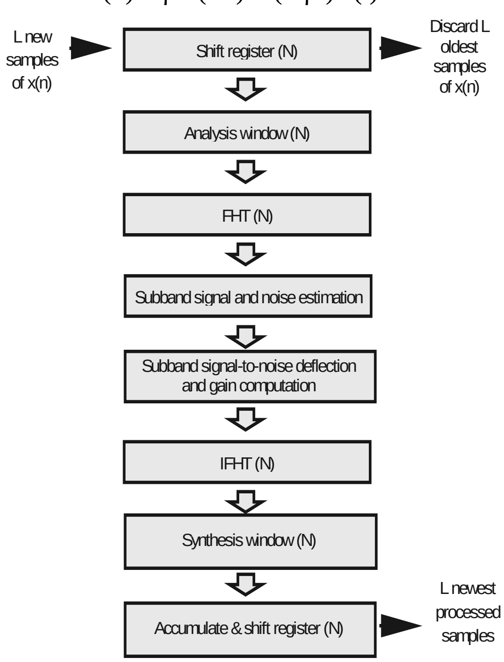
where $\mathbf{W}$ is estimate of filter weight vector, u is the input signal and $U_i(k)$ are Hartley coefficients of input, $\mathbf{e}$ is error signal , P is the energy estimate. The output of the adaptive filter is then fed to subband noise suppressor, which estimates the signal and noise powers, and uses a heuristic mechanism to modify the subband signal gains.

2.2 SUBBAND NOISE SUPPRESSOR

The architecture of subband noise suppressor discussed in [3] is reproduced in the Fig 2. As shown in Fig.2, i/p samples are passed through an analysis filter ban*k*, which is implemented using an analysis window (which acts as the prototype filter for the filter ban*k*) and the FHT. The o/p subband time series is given to a ban*k* of voice activity detectors. Each detector estimates the long and short-term average levels of the time-series envelope in the subband. The long-term average estimates the stationary or noise component, and when speech is present, the short-term average estimates the signal level.

The signal and noise estimates are computed using the nonlinear single-pole recursions given below –

$$s(i) = \alpha\, s(i-1) + (1-\alpha)\, k(i)$$
$$n(i) = \beta\, n(i-1) + (1-\beta)\, k(i)$$



Fig(2) Bloc*k* diagram for subband noise suppressor.

where $s(i)$ and $n(i)$ are the signal and noise estimates at subband time index i, and recursion constants α and β are given by

$$\alpha = \begin{cases} \alpha_a, \text{ if } k(i) > s(i-1) \\ \alpha_d, \text{ if } k(i) < s(i-1) \end{cases}$$

$$\beta = \begin{cases} \beta_a, \text{ if } k(i) > n(i-1) \\ \beta_d, \text{ if } k(i) < n(i-1) \end{cases}$$

where $k(i) = 0.5 \cdot \text{sqrt}\, [\, x(i)^2 + x(N-i)^2\,]$ which is magnitude spectrum of input signal and $x(i)$ is FHT of input signal.

Estimates $s(i)$ and $n(i)$ are updated with the magnitude of the subband time-series sample, $k(i)$, at each new sampling interval. α and β ta*k*e on different "attac*k*" and "decay" values depending on the relationship of $k(i)$ to the current estimate. Deflection ratios and gain computations in each subband are calculated as per algorithm given by Diethorn [3].

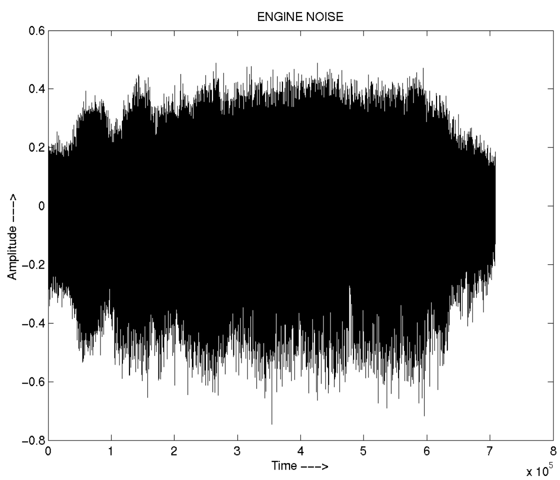
3.QUANTITATIVE PERFORMANCE

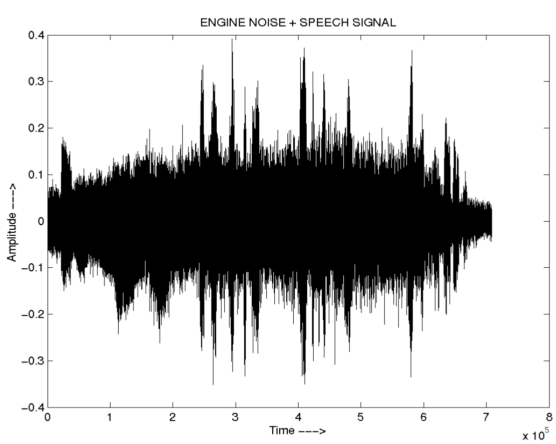
We have implemented the speech enhancement algorithm with FFT and FHT. The performance of FHT implementation was found to be exactly same as FFT implementation. Tables given below give comparison of the implementation complexity using FFT and FHT respectively, on the basis of number of (real) floating point multiplications additions and memory usage. For Noise canceller frame size is 128 samples and for noise suppressor frame size is 32 .

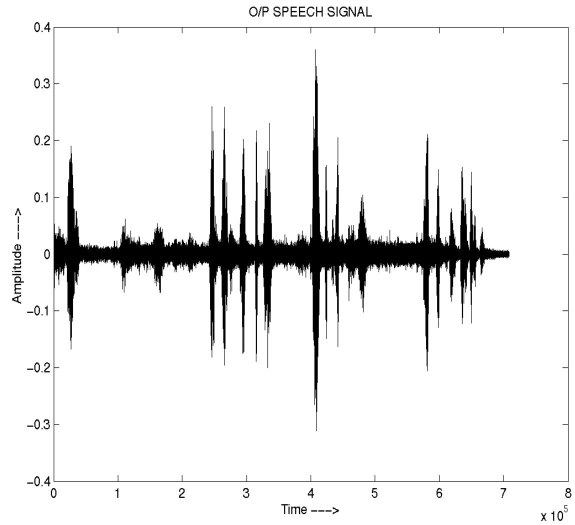
	Noise Canceller using FFT	Noise Canceller using FHT
No. of Floating point multiplication's	249856	133120
No. of floating point additions	352256	185334
Memory usage (in Kbytes)	118.74	77.824

	Noise Suppressor using FFT	Noise Suppressor using FHT
No. of Floating point multiplications	1600	864
No. of floating point additions	2336	1184
Memory usage (in Kbytes)	1.28	1.28

It is clear from the above tables that the computational complexity of FHT based implementation is approximately half of that of FFT based implementation. Memory requirement is also considerably less for FHT based implementation. Simulation results for proposed speech enhancement scheme for sampling frequency 16Khz and frame size is 128 are given below .







4. CONCLUSION

In this paper, we have discussed the use of FHT in effective speech enhancement algorithm .The results indicate that memory requirements and computational complexity are greatly reduced by using FHT in the algorithm compared to FFT implementation

5. REFERENCES

[1] J. Benesty et al, Advances in Network and Acoustic Echo cancellation , Springer-Verlog ,Berlin 2001.

[2] Simon Haykin, *Adaptive Filter Theory* , Prentice hall, 3rd edition , 3rd edition 1996.

[3] E. J. Diethorn, “A Subband noise-Reduction method for enhancing speech in telephony & teleconferencing”, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1997.

[4] Ronald F. Ullmann, “An Algorithm for Fast Hartley Transform”. Technical Report ,Stanford university,1984.

[5] Aravinda Ganapathiraju, “Report on Analysis and Characterization of Fast Fourier Transform Algorithms”, Institute for Signal and Information Processing , Mississippi State University ,Technical report ,1999.