A FAST RUNNING HARTLEY TRANSFORM ALGORITHM AND ITS APPLICATION IN ADAPTIVE SIGNAL ENHANCEMENT

Zhiyue Lin[1], Fuliang Yin[2] and Richard W. McCallum[1]

[1]University of Kansas Medical Center, Kansas City, KS 66160, USA
[2]School of Electronic and Information Engineering, Dalian University of Technology, Dalian, P.R. China
zlin@kumc.edu

ABSTRACT

A fast recursive algorithm for computation of the running discrete Hartley transform (RDHT) is presented. This method is based on the relation between the running discrete Fourier transform (RDFT) and the RDHT. The number of operations for the proposed recursive algorithm is only 2/N (N=length of the transform) of the direct computation of the RDHT. It also provides substantially computational savings compared with the recursive RDFT algorithm. The transform-domain adaptive digital filter is implemented based on the presented algorithm. The simulation results of its implementation on an adaptive line enhancer are given to demonstrate the efficiency of the presented fast algorithm.

1. INTRODUCTION

The discrete Hartley transform (DHT) directly maps a real-valued sequence to a real-valued spectrum [1]. Compared with the discrete Fourier transform (DFT), the DHT has many advantages. It is an alterative to the DFT for signal processing, such as the spectral analysis and fast convolution [2]. Fast algorithms for the DHT are, therefore, an active research area. The two traditional approaches to DHT implementation are: using a fast Fourier transform (FFT), and direct factorization of the DHT [2]-[4].

In certain signal processing applications such as transform-domain adaptive digital filtering, a data segment to be transformed is shifted ahead one sample at each time instant to update the data. It is called the running discrete Hartley transform (RDHT) if the DHT is applied in the transform-domain adaptive digital filter [5]. In this paper a fast recursive algorithm for computation of the RDHT based on the relation between the RDHT and RDFT is presented. This algorithm has speed advantages over the direct computation of the RDHT and the RDFT approach. Results of its application in the implementation of the adaptive line enhancer are given to demonstrate the efficiency of the presented fast algorithm.

2. FAST RECURSIVE ALGORITHM FOR RDHT

The running discrete Hartley transform (RDHT) of a discrete signal f_n is defined as [5]:

$$H_n(m) = \sum_{k=0}^{N-1} f_{n-k} \left(\cos\left(\frac{2\pi}{N}km\right) + \sin\left(\frac{2\pi}{N}km\right)\right) \tag{1}$$

where N is a given integer which stands for the length of the data segment. For a given n, $H_n(m)$ is the DHT in the variable k of the data segment f_{n-k} of f_n. At the nth time instant, the data segment to be transformed is $f_n, f_{n-1}, \ldots f_{n-N+1}$, and at the $(n+1)$th time instant, the data segment to be transformed is $f_{n+1}, f_n, \ldots, f_{n-N+2}$. Clearly, the data segment to be transformed is updated by one sample at each time instant.

Let $F(z)$ be the Z-transform of f_n, i.e.,

$$F(z) = \sum_{n=-\infty}^{\infty} f_n z^{-n} \tag{2}$$

Then the sequence $H_n(m)$ has Z-transform with respect to n

$$H_m(z) = \sum_{n=-\infty}^{\infty} H_n(m) z^{-n} \tag{3}$$

From equation (1), we have

$$H_m(z) = \sum_{n=-\infty}^{\infty} \left[\sum_{k=0}^{N-1} f_{n-k} \left(\cos\frac{2\pi}{N}km + \sin\frac{2\pi}{N}km\right)\right] z^{-n}$$

$$= F(z) \frac{(1 - z^{-N})(1 - z^{-1}[\cos(\frac{2\pi}{N}m) - \sin(\frac{2\pi}{N}m)])}{1 - 2\cos(\frac{2\pi}{N}m) z^{-1} + z^{-2}} \tag{4}$$

By taking the inverse Z-transform on the above equation, the following recursive equation can be obtained :

$$H_n(m) = f_n - f_{n-N} - \left[\cos\left(\frac{2\pi m}{N}\right) - \sin\left(\frac{2\pi m}{N}\right)\right] \cdot (f_{n-1} - f_{n-N-1}) + 2\cos\left(\frac{2\pi m}{N}\right) H_{n-1}(m) - H_{n-2}(m) \tag{5}$$

Similarly, for a N-length data segment f_n the running discrete Fourier transform is defined as [6]

$$Z_n(m) = \sum_{k=0}^{N-1} f_{n-k}\, e^{-j(2\pi/N)mk} \qquad (6)$$

It can be easily shown that the corresponding recursive equation for $Z_n(m)$ is

$$Z_n(m) = f_n - f_{n-N} + Z_{n-1}(m)e^{-j(2\pi/N)m} \qquad (7)$$

By comparing (1) and (6) we have

$$H_n(m) = RE[Z_n(m)] - IM[Z_n(m)] \qquad (8)$$

From (7), we can obtain

$$H_n(m) = f_n - f_{n-N} + RE[Z_{n-1}(m)e^{-j(2\pi/N)m}] - IM[Z_{n-1}(m)e^{-j(2\pi/N)m}] \qquad (9)$$

It can be proven that for the real-valued data sequence f_n the following is valid:

$$RE[Z_{n-1}(m)] = \frac{1}{2}[H_{n-1}(N-m) + H_{n-1}(m)] \qquad (10)$$

$$IM[Z_{n-1}(m)] = \frac{1}{2}[H_{n-1}(N-m) - H_{n-1}(m)] \qquad (11)$$

Combining (10) and (11), we have

$$Z_{n-1}(m) = \frac{1}{2}[H_{n-1}(N-m) + H_{n-1}(m)] + j\frac{1}{2}[H_{n-1}(N-m) + H_{n-1}(m)] \qquad (12)$$

Now from (9) we finally have

$$H_n(m) = f_n - f_{n-N} + \cos(\frac{2\pi m}{N})H_{n-1}(m) + \sin(\frac{2\pi m}{N})H_{n-1}(N-m) \qquad (13)$$

The above equation is the proposed recursive algorithm for $H_n(m)$.

It can be seen from the above equation that only two real multiplications and additions are required, respectively, for the computation of each RDHT coefficient. One additional addition is needed to compute N-length RDHT values over each data segment. Thus, the total number of real multiplications and additions is $2N$ and $2N+1$, respectively. For the direct computation of an N-length RDHT coefficient using equation (1), the number of real multiplications and additions is $N \times N$ and $N(2N-1)$, respectively. Obviously, the number of operations using the proposed recursive algorithm is only $2/N$ of the direct computation and is also faster than the use of the directly deduced recursive equation (5). Unlike the recursive equation (7) for the RDFT, the proposed recursive algorithm is real and no complex arithmetic is involved. This represents a considerable saving in computational complexity over the RDFT.

3. APPLICATIONS OF THE RDHT IN TRANSFORM-DOMAIN ADAPTIVE DIGITAL FILTERING

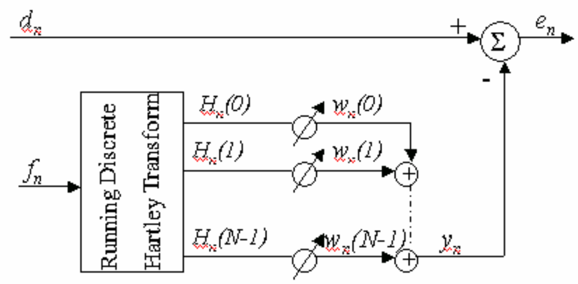


Fig.1. Structure of a transform-domain adaptive filter

It is known that the time-domain adaptive filter using the least mean-square (LMS) algorithm converges slowly, especially when the eigenvalue spread of the input autocorrelation matrix is large [6], [7]. An approach to acceleration the convergence rate is to somehow transform the filter input signal into another signal with the corrersponding autocorrelation matrix having a small eigenvalue spread. This can be achieved by performing the adaptive filtering in the orthogonal domain. In the transform-domain LMS adaptive digital filter, the eigenvalue spread is reduced by whitening the power spectrum of the input signal [6]. The structure of the transform-domain adaptive filter is shown in Fig 1. d_n in Fig.1 is the primary input, f_n is the reference input, and $H_n(m)$ is the RDHT coefficients of the input data segment f_n, f_{n-1}, ..., f_{n-N+1} computed by the proposed recursive algorithm. The output of the filter, y_n is

$$y_n = \sum_{i=0}^{N-1} H_n(i)w_n(i) \qquad (14)$$

where $w_n(i)$ is the ith filter weight at the nth time instant. The weighted output is subtracted from the primary input to form an error signal:

$$e_n = d_n - y_n \qquad (15)$$

The LMS algorithm for the adaptation of the filter weights is written as follows:

$$w_{n+1}(i) = w_n(i) + \frac{\mu}{||H_n(i)||} e_n H_n(i) \qquad (16)$$

where μ is a small positive constant, called step size, controlling the rate of convergence and

$$\| H_n(i) \| = \frac{1}{N} \sum_{i=0}^{N-1} | H_n(i) |^2 \quad (17)$$

It denotes the energy content of the input signal over the length of the filter. The LMS adaptive constant in the above equation is time-varying and is inversely proportional to the input energy.

The enhancement and detection of a coherent sinusoid in noise occurs in many applications of signal processing. The adaptive line enhancer (ALE) is one possible solution to this problem [8]. Fig.2 shows a block diagram of the ALE implemented by the transform-domain adaptive digital filter using the proposed algorithm. The ALE primary input is d_n and the ALE reference input is a delayed version of the primary input signal, i.e.,

$$f_n = d_{n-\Delta} \quad (18)$$

The delayed input signal used as reference decorrelates noise component of the input signal.

4. RESULTS

Two simulation results of the transform-domain ALE are provided in the following to demonstrate the efficiency of the proposed method. Due to the limited space, the successful applications of the proposed algorithm to real data are presented in [11].

4.1. Example 1

The ALE primary input is a sinusoidal signal with power, γ, corrupted by additive white noise v_n with zero mean [5]

$$d_n = \sqrt{2\gamma} \cos(\frac{2\pi}{25} n) + v_n \quad (19)$$

We chose the order of the filter N=32, step size μ= 0.1, Δ=2 and γ=1. The original signal is shown in Fig. 3a, the filter input with additive noise in Fig. 3b, and the enhanced output in Fig. 3c. The enhancement of the sinusoidal signal is more clearly observed in the power spectra of the filter input and output signals as shown in Fig. 3d. It can be seen that a reduction of about 30 dB of noise level is achieved, whereas the original signal is not affected.

4.2. Example 2

A sinusoid corrupted by multiplicative noise also occurs in several signal processing applications [9], [10]. Consider that the ALE primary input is a sinusoidal signal corrupted by multiplicative noise b_n

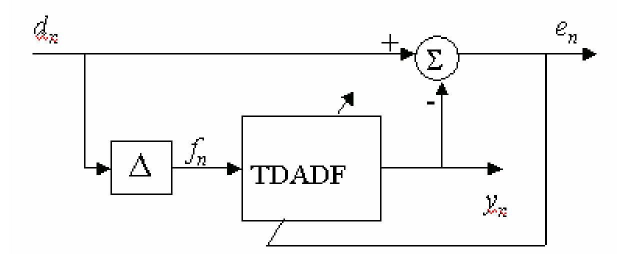


Fig. 2. Block diagram of the adaptive line enhancer; TDADF, transform-domain adaptive digital filter.

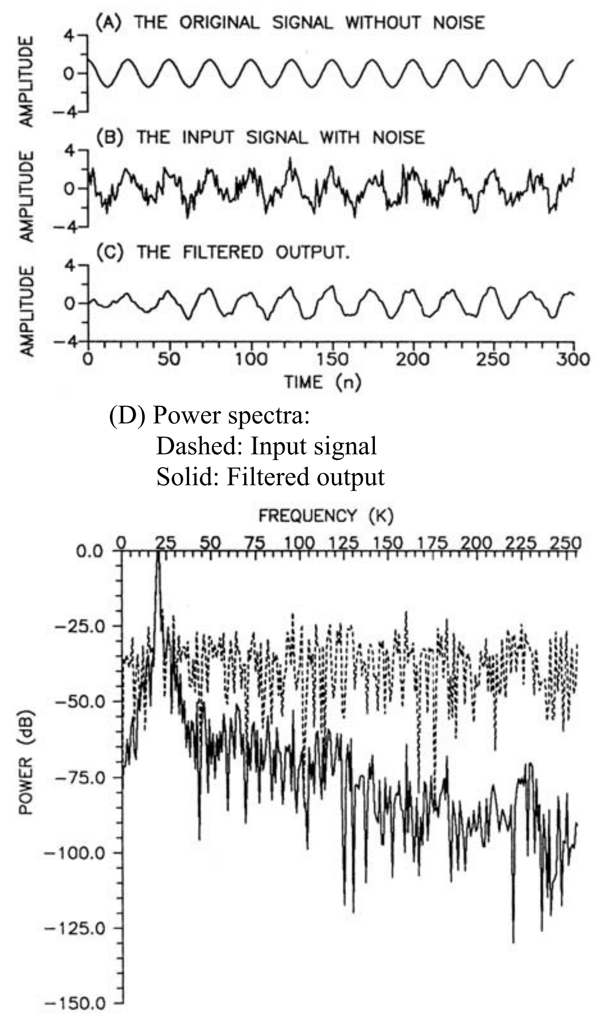
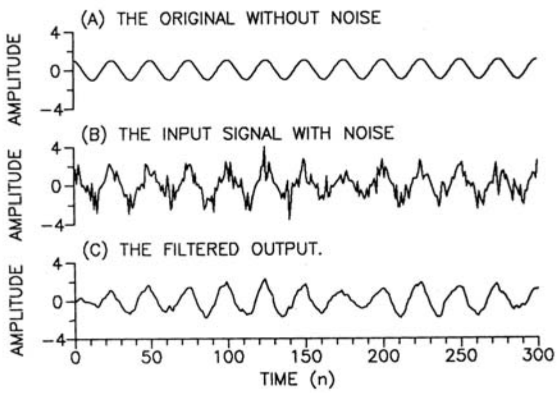


Fig. 3. An example of simulation results of the adaptive transform-domain line enhancer. (A) The original signal; (B) the filter input with white noise; (C) the enhanced output; (D) power spectra of the filter input and enhanced output signals.

$$d_n = (\sqrt{2\gamma} + b_n) \cos(\frac{2\pi}{25} n) \quad (20)$$



(D) Power spectra:
Dashed: Input signal
Solid: Filtered output

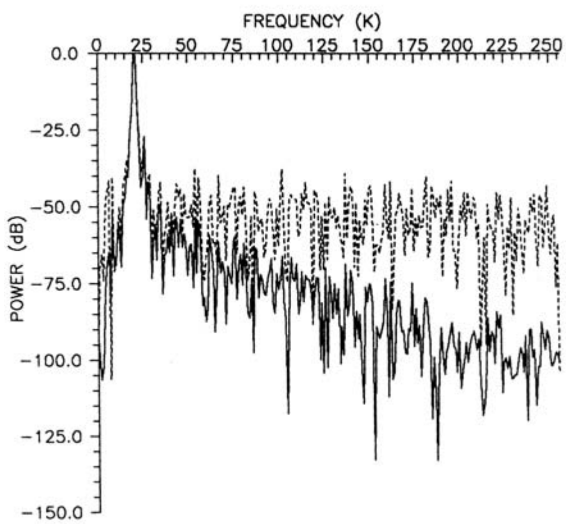


Fig. 4. Example 2 of simulation results of the adaptive transform-domain line enhancer. (A) The original signal; (B) the filter input corrupted by multiplicative white noise; (C) the enhanced output; (D) power spectra of the filter input and enhanced output signals.

For the purpose of comparison to Example 1, assume that b_n is the same as v_n in Example 1, i.e., white noise with zero mean. Similarly as in Example 1, we chose γ=1, Δ=2, N=32, and step size μ= 0.1. Fig. 4 shows the simulation results. The original signal is shown in Fig. 4a, the filter input with multiplicative noise in Fig. 4b, and the enhanced output in Fig. 4c. The enhancement of the sinusoidal signal is more clearly observed in the power spectra of the filter input and output signals as shown in Fig. 4d. It can be seen that the filter output is similar to that in Example 1 so that the proposed method is not only efficiency for enhancement of a sinusoid corrupted by additive white noise but also for multiplicative noise.

4. CONCLUSION

In this paper, a fast recursive algorithm for computation of RDHT is proposed. The computation complexity of the proposed algorithm is only 2/N of the direct computation of a RDHT. Unlike the recursive RDFT, the fast recursive RDHT algorithm is real and no complex arithmetic is involved. Thus this algorithm has speed advantages over the direct computation of the RDHT and RDFT approach. The simulation results demonstrate that the adaptive transform-domain line enhancer using the proposed fast RDHT is efficiency for reducing the power levels of both additive noise and multiplicative noise.

5. REFERENCES

[1] R. N. Bracewell, "Discrete Hartley transform," *J. Opt. Soc. Amer.*, vol. 3, No. 12, pp. 1832-1835, Dec. 1983.

[2] H.-J. Meckelberg and D. Lipka, "Fast Hartley transform algorithm," *Electron. Lett.*, vol. 21, pp. 341-343, Apr. 11, 1985.

[3] R. N. Bracewell, "The fast Hartley transform," *Proc. IEEE*, vol. 72, No. 8, pp. 1010-1018, Aug. 1984.

[4] H. S. Hou, "The fast Hartley transform algorithm," *IEEE Trans. Comput.*, vol. 36, pp. 147-156, 1987.

[5] P. K. Bondyopadhyay, "Application of running Hartley transform in adaptive digital filtering," *Proc. IEEE*, vol. 76, pp. 1370-1372, 1988.

[6] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain LMS algorithm,"*IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-31, pp. 609-615, June 1983.

[7] B. Widrow *et al.*, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151-1162, Aug. 1976.

[8] B. Widrow *et al.*, "Adaptive noise canceling: Principle and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, Dec. 1975.

[9] M. Ghogho, M. Ibnkahla, and N. J. Bershad, "Analytic behavior of the LMS adaptive line enhancer for sinusoids corrupted by multiplicative and additive noise," *Trans. Acoust. Speech, Signal Processing*, vol. ASSP-46, pp. 2386-2393, Sept., 1998.

[10] A. Swami, "Multiplicative noise models: Parameter estimation using cumulants," *Signal Process.*, vol. 36, pp. 355-373, April 1994.

[11] Z. Lin, F. Yin and R.W. McCallum, "A combined method for recovery of gastric slow waves from electrogastrograms," submitted to *Med. Biol. Eng. Comput.* 2004.