TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 6, December 2020, pp. 3067~3072
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i6.16100

Direct split-radix algorithm for fast computation of type-II discrete Hartley transform

Mounir Taha Hamood
Electrical Engineering Department, College of Engineering, University of Tikrit, Iraq

Article Info

Article history:

Received Mar 20, 2020
Revised Jun 3, 2020
Accepted Jun 25, 2020

Keywords:

Decimation-in-time approach
Discrete Hartley transform
Generalized DHTs
Split radix algorithm
Type-II DHT (DHT-II)

ABSTRACT

In this paper, a novel split-radix algorithm for fast calculation the discrete Hartley transform of type-II (DHT-II) is intoduced. The algorithm is established through the decimation in time (DIT) approach, and implemented by splitting a length N of DHT-II into one DHT-II of length N/2 for even-indexed samples and two DHTs-II of length N/4 for odd-indexed samples. The proposed algorithm possesses the desired properties such as regularity, inplace calculation and it is represented by simple closed form decompositions leading to considerable reductions in the arithmetic complexity compared to the existing DHT-II algorithms. Additionally, the validity of the proposed algorithm has been confirmed through analysing the arithmetic complexity by calculating the number of real additions and multiplications and associating it with the existing DHT-II algorithms.

Corresponding Author:

Mounir Taha Hamood,
Electrical Engineering Departement, College of Engineering,
University of Tikrit,
Tikrit, P O BOX 42, Iraq.
Email: m.t.hamood@tu.edu.iq

1. INTRODUCTION

The Hartley transform (HT) is an orthogonal transform that maps a real valued function into its frequency real components [1], used in diverse fields such as signal/image processing, digital communications and many other applications [2]. Although HT and the Fourier transform (FT) [3], shares equivalent properties, it allows a function to be separated into two autonomous sets of sinusoidal components; these sets are characterized in terms of negative and positive frequency components respectively. Another advantage of the HT over the FT is that the computation of the kernel of HT is exactley same as that of its inverse, so that the inverse transform differs from forward only by the scale factor, hence identical utilization of the HT can be used for signal synthesis and analysis. As we know that a real generalized discrete version of Hartley transform (GDHT) [4] can be defined analogous to the complex generalized discrete of fourier transform (GDFT) [5]. Also, it is well known that there is a fast way of computing Hartley transform that is analogous to the fast fourier transform (FFT) as there is a simple step to go from DFT to the DHT [6]. Consequently the fast algorithm developed in this paper also constitutes a fast way of arriving at GDFT. In fact the approach via the GDHT proves to be advantageous mainly because of the simplifications that results from the fact that no complex arithmetic is required when real data are being processed.

In genral, the DHT transform kernel can be extended to allow shifts in either time index, frequency index or both indexes. The resulting invertble transforms are reffered to as generalized discrete Hartley transforms (GDHTs) and are defined as;

$$X(k) = \sum_{n=0}^{N-1} x(n) cas\left(\theta \frac{(2n+n_o)(2k+k_o)}{4}\right) \qquad k = 0,1,\ldots,N\text{-}1 \tag{1}$$

where $cas(\theta) = cos\ (\theta) + sin\ (\theta), \theta = 2\pi/N$ is the kernel of the transform, and the constants n_o and k_o are the parameters that identify shifts in the frequency and time domains. By applying a different set of these parameters, a different types of GDHT can be obtained. As the input sequence $x(n)$ can be accurately retrieved from the output sequence $X(k)$, therefore $x(n)$ is completely defined by a set of coefficients $X(k)$ in diverse domain. In many realted applications, it is ordinary to specify a problem in a appropriate domain because numerous characteristics of the signals can be only revealed in particular domain. This paper deals with a particular type of GDHT when $n_o = 0$ and $k_o = 1$ that is known in literature as type-II DHT [7].

A lot of algorithms were introduced for fast calculation of the GDHTs [8-10]. Among them, the split-radix algorithm that was first proposed for the calculations of the FFT [11-13] and then developed for other transforms [14-17], has proved it gives the lowest arithmetic complexity known in literature [18, 19], that employs radix-4 decomposition to the odd-indexed samples and radix-2 decomposition to the even-indexed samples of the power-of-two samples. However, the developments of the split radix algorithms introduced for the DHT-II (SR-DHT-II) were use indirect approaches [20, 21]. Therefore, it is purpose of this paper to introduce a direct split radix algorithm for the efficient calculations of the DHT-II using decimation-in-time (DIT) approach. The paper is prepared in four sections as follows: section 2 purposes the development of the new split-radix algorithm based on DIT approach for the DHT-II. In section 3, the evalution of the proposed algorithm is studied by calculating their arithmetic complexity and associating them with the radix-2 algorithm. A conclusion is then given in section 4.

2. DEVELOPMENT OF SR-DHT-II ALGORITHM

The type-II discrete Hartley transform (DHT-II) of length N for the real valued samples $x(n)$ is given as [22]:

$$X(k) = \sum_{n=0}^{N-1} x(n) cas\left(\theta n\left(\frac{2k+1}{2}\right)\right) \qquad k = 0,1,\ldots,N\text{-}1 \tag{2}$$

where the transform length N is identified to be powers of two $N = 2^m$. The inverse DHT-II transform (known as the type-III DHT) is defined as;

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) cas\left(\theta n\left(\frac{2k+1}{2}\right)\right) \qquad n = 0,1,\ldots,N\text{-}1 \tag{3}$$

The decimation-in-time algorithm (DIT) derivation of the SR-DHT-II algorithm starts by decomposing the transformed sequence $X(k)$ into its odd $X_{od}(k)$ and even $X_{ev}(k)$ indexed sequences. Therefore (2) can be decomposed to,

$$X(k) = X_{od}(k) + X_{ev}(k) \tag{4}$$

where $X_{od}(k)$ and $X_{ev}(k)$ represents the odd- and even-indexed sequences of $X(k)$ respectively, both are of length $(N/2)$. Firstly, radix-2 algorithm for the $X_{ev}(k)$ can be written as;

$$X_{ev}(k) = \sum_{n=0}^{N/2\text{-}1} x(2n)\, cas\left(2\theta n(\frac{2k+1}{2})\right) = X_{2n}(k) \tag{5}$$

Secondly, radix-4 algorithm for the DHT-II can be developed by dividing the input samples $x(n)$ into four $(N/4)$ DHTs-II as follows:

$$X(k) = X_o(k) + X_1(k) + X_2(k) + X_3(k) \tag{6}$$

where,

$$\mathrm{X_i(k)} = \sum_{\mathrm{n=0}}^{N/4\text{-}1} \mathrm{x(4n+i)\,cas}\left(\theta(4\mathrm{n}+\mathrm{i})(\frac{2\mathrm{k}+1}{2})\right) \qquad \mathrm{i}=0,1,2,3 \tag{7}$$

Therefore, by considering the odd indexed samples only for the $X(k)$ in (6) i.e., $[X_{od}(k) = X_1(k) + X_3(k)]$, we get:

$$\begin{aligned} X_{od}(k) &= \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,cas\left(\theta(4n+1)(\tfrac{2k+1}{2})\right) + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,cas\left(\theta(4n+3)(\tfrac{2k+1}{2})\right) \\ &= \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,cas\left(4\theta n\tfrac{2k+1}{2} + \theta\tfrac{2k+1}{2}\right) + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,cas\left(4\theta n\tfrac{2k+1}{2} + 3\theta\tfrac{2k+1}{2}\right) \end{aligned} \tag{8}$$

Applying cas property given in [1] as follows;

$$\mathrm{cas}(\alpha+\beta) = \cos(\alpha)\mathrm{cas}(\beta) + \sin(\alpha)\mathrm{cas}(\text{-}\beta) \tag{9}$$

$cas(.)$ term in (8) can be simplified to:

$$\begin{aligned} X_{od}(k) &= cos\left(\theta\tfrac{2k+1}{2}\right)\sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,cas\left(4\theta n\tfrac{2k+1}{2}\right) + sin\left(\theta\tfrac{2k+1}{2}\right)\sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,cas\left(\text{-}4\theta n\tfrac{2k+1}{2}\right) \\ &+cos\left(3\theta\tfrac{2k+1}{2}\right)\sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,cas\left(4\theta n\tfrac{2k+1}{2}\right) + sin\left(3\theta\tfrac{2k+1}{2}\right)\sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,cas\left(\text{-}4\theta n\tfrac{2k+1}{2}\right) \end{aligned} \tag{10}$$

The negative indices of cas terms in (10) can be simplified to;

$$\begin{aligned} \sum_{m=0}^{N\text{-}1} x(m)\,cas\left(\text{-}\theta m\tfrac{2k+1}{2}\right) &= \sum_{n=0}^{N\text{-}1} x(m)cas\left(\theta m(\text{-}k - \tfrac{1}{2})\right) \\ &= \sum_{n=0}^{N\text{-}1} x(m)cas\left(\theta m(N - k - \tfrac{1}{2})\right) \\ &= \sum_{n=0}^{N\text{-}1} x(m)cas\left(\theta m\tfrac{2(N\text{-}k\text{-}1)+1}{2}\right) \end{aligned} \tag{11}$$

From (11) we get the relation,

$$\textstyle\sum_{\mathrm{n=0}}^{\frac{N}{4}-1} \mathrm{x(4n+i)\,cas}\left(\text{-}4\theta\mathrm{n}\frac{2\mathrm{k}+1}{2}\right) = \sum_{\mathrm{n=0}}^{\frac{N}{4}-1} \mathrm{x(4n+i)\,cas}\left(4\theta n\frac{2(\frac{N}{4}\text{-k-}1)+1}{2}\right) \tag{12}$$

Therefore $X_{od}\ (k)$ in (10) becomes;

$$\begin{aligned} X_{od}(k) &= X_{4n+1}(k)cos\left(\theta\tfrac{2k+1}{2}\right) + X_{4n+1}(\tfrac{N}{4} - k - 1)sin\left(\theta\tfrac{2k+1}{2}\right) \\ &+X_{4n+3}(k)cos\left(3\theta\tfrac{2k+1}{2}\right) + X_{4n+3}(\tfrac{N}{4} - k - 1)sin\left(3\theta\tfrac{2k+1}{2}\right) \end{aligned} \tag{13}$$

where $X_{4n+1}(k)$ and $X_{4n+3}(k)$ are two DHTs-II of length $(N/4)$, defined as:

$$\mathrm{X_{4n+1}(k)} = \sum_{\mathrm{n=0}}^{\frac{N}{4}-1} \mathrm{x(4n+1)\,cas}\left(4\theta\mathrm{n}(\frac{2\mathrm{k}+1}{2})\right) \tag{14}$$

$$\mathrm{X_{4n+3}(k)} = \sum_{\mathrm{n=0}}^{\frac{N}{4}-1} \mathrm{x(4n+3)\,cas}\left(4\theta\mathrm{n}(\frac{2\mathrm{k}+1}{2})\right) \tag{15}$$

Substituting (5) and (13) into (4), $X(k)$, we get,

$$X(k) = X_{2n}(k) + \left[X_{4n+1}(k)\,cos\left(\theta\tfrac{2k+1}{2}\right) + X_{4n+1}(\tfrac{N}{4}-k-1)\,sin\left(\theta\tfrac{2k+1}{2}\right)\right] + \left[X_{4n+3}(k)\,cos\left(3\theta\tfrac{2k+1}{2}\right) + X_{4n+3}(\tfrac{N}{4}-k-1)\,sin\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{16}$$

Using the following trigonometric identities,

$$\begin{aligned} cos\left(\phi+\tfrac{\pi}{2}\right) &= cos(\phi)\,cos\left(\tfrac{\pi}{2}\right) - sin(\phi)\,sin\left(\tfrac{\pi}{2}\right) = -sin(\phi) \\ sin\left(\phi+\tfrac{\pi}{2}\right) &= sin(\phi)\cos\left(\tfrac{\pi}{2}\right) + cos(\phi)\,sin\left(\tfrac{\pi}{2}\right) = cos(\phi) \\ \cos\left(\phi+\tfrac{3\pi}{2}\right) &= cos(\phi)\,cos\left(\tfrac{3\pi}{2}\right) - sin(\phi)\,sin\left(\tfrac{3\pi}{2}\right) = sin(\phi) \\ sin\left(\phi+\tfrac{3\pi}{2}\right) &= sin(\phi)\,cos\left(\tfrac{3\pi}{2}\right) + cos(\phi)\,sin\left(\tfrac{3\pi}{2}\right) = -cos(\phi) \end{aligned} \tag{17}$$

Other decompositions $X(k+N/4), X(k+N/2)$ and $X(k+3N/4)$ can be calculated, as

$$X(k+\tfrac{N}{4}) = X_{2n}(k+\tfrac{N}{4}) - \left[X_{4n+1}(k)\,sin\left(\theta\tfrac{2k+1}{2}\right) - X_{4n+1}(\tfrac{N}{4}-k-1)\,cos\left(\theta\tfrac{2k+1}{2}\right)\right] + \left[X_{4n+3}(k)\,sin\left(3\theta\tfrac{2k+1}{2}\right) - X_{4n+3}(\tfrac{N}{4}-k-1)\,cos\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{18}$$

$$X(k+\tfrac{N}{2}) = X_{2n}(k) - \left[X_{4n+1}(k)\,cos\left(\theta\tfrac{2k+1}{2}\right) + X_{4n+1}(\tfrac{N}{4}-k-1)\,sin\left(\theta\tfrac{2k+1}{2}\right)\right] - \left[X_{4n+3}(k)\,cos\left(3\theta\tfrac{2k+1}{2}\right) + X_{4n+3}(\tfrac{N}{4}-k-1)\,sin\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{19}$$

$$X(k+\tfrac{3N}{4}) = X_{2n}(k+\tfrac{N}{4}) + \left[X_{4n+1}(k)\,sin\left(\theta\tfrac{2k+1}{2}\right) - X_{4n+1}(\tfrac{N}{4}-k-1)\,cos\left(\theta\tfrac{2k+1}{2}\right)\right] - \left[X_{4n+3}(k)\,sin\left(3\theta\tfrac{2k+1}{2}\right) - X_{4n+3}(\tfrac{N}{4}-k-1)\,cos\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{20}$$

For in-place computations, other points $X(N/4-k-1), X(N/2-k-1), X(3N/4-k-1)$ and $X(N-k-1)$ need to be computed. These points can be derived using trigonometric identities given by (17) and the periodicity property of DHT-II, we get:

$$X(\tfrac{N}{4}-k-1) = X_{2n}(\tfrac{N}{4}-k-1) + \left[X_{4n+1}(k)\,cos\left(\theta\tfrac{2k+1}{2}\right) + X_{4n+1}(\tfrac{N}{4}-k-1)\,sin\left(\theta\tfrac{2k+1}{2}\right)\right] - \left[X_{4n+3}(k)\,cos\left(3\theta\tfrac{2k+1}{2}\right) + X_{4n+3}(\tfrac{N}{4}-k-1)\,sin\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{21}$$

$$X(\tfrac{N}{2}-k-1) = X_{2n}(\tfrac{N}{2}-k-1) + \left[X_{4n+1}(k)\,sin\left(\theta\tfrac{2k+1}{2}\right) - X_{4n+1}(\tfrac{N}{4}-k-1)\,cos\left(\theta\tfrac{2k+1}{2}\right)\right] + \left[X_{4n+3}(k)\,sin\left(3\theta\tfrac{2k+1}{2}\right) - X_{4n+3}(\tfrac{N}{4}-k-1)\,cos\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{22}$$

$$X(\tfrac{3N}{4}-k-1) = X_{2n}(\tfrac{N}{4}-k-1) - \left[X_{4n+1}(k)\,cos\left(\theta\tfrac{2k+1}{2}\right) + X_{4n+1}(\tfrac{N}{4}-k-1)\,sin\left(\theta\tfrac{2k+1}{2}\right)\right] - \left[X_{4n+3}(k)\,cos\left(3\theta\tfrac{2k+1}{2}\right) - X_{4n+3}(\tfrac{N}{4}-k-1)\,sin\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{23}$$

$$X(N-k-1) = X_{2n}(\tfrac{N}{2}-k-1) - \left[X_{4n+1}(k)\,sin\left(\theta\tfrac{2k+1}{2}\right) - X_{4n+1}(\tfrac{N}{4}-k-1)\,cos\left(\theta\tfrac{2k+1}{2}\right)\right] - \left[X_{4n+3}(k)\,sin\left(3\theta\tfrac{2k+1}{2}\right) - X_{4n+3}(\tfrac{N}{4}-k-1)\,cos\left(3\theta\tfrac{2k+1}{2}\right)\right] \tag{24}$$

From decompositions (16) and (18)-(24), it is clearly that this algorithm processes data in groups of eight points, specifically $X(k)$, $X(k+N/4)$, $X(k+N/2)$, $X(k+3N/4), X(N/4-k-1)$, $X(N/2-k-1)$, $X(3N/4-k-1)$ and $X(N-k-1)$. The index k is in the range $0 \le k \le N/8-1$, with the first 4-points, found for $k=0$, becomes $X(0), X(N/4), X(N/2)$ and $X(3N/4)$. The algorithm butterfly contains a special indexing scheme known as retrograde [23, 24], i.e., when the negative indices of samples $X(N/4-k-1), X(N/2-k-1), X(3N/4-k-1)$ and $X(N-k-1)$ are decremented, the positive indices of samples $X(k), X(k+N/4), X(k+N/2)$ and $X(k+3N/4)$ are incremented. The resultant in-place butterfly structure for this algorithm is shown in Figure 1.

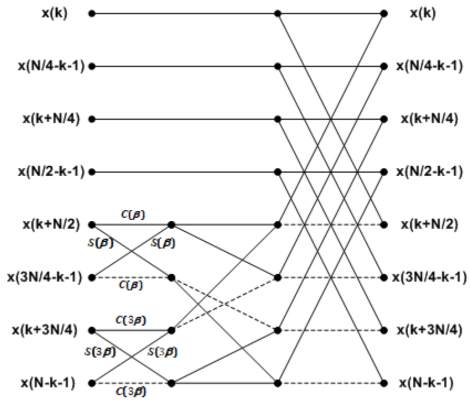

Figure 1. An in-place butterfly of the SR-DHT-II algorithm; where $C(\beta) = cos(\pi(2k + 1)/N)$ and $S(\beta) = sin((\pi(2k + 1)/N)$, dotted and solid lines stand for subtraction and additions respectively

3. ARITHMETIC COMPLEXITY

In (16) and (18-24) represent the decompositions of the developed DHT-II split-radix algorithm. For in-place computation, a double butterfly is used as shown in Figure 1. Therefore, the trivial arithmetic operations (multiplying by -1, 0, 1) will be removed; the resultant butterfly calculates 8 points and requires 16 additions and 8 multiplications. Therefore for transform length $N = 2^m$, the split-radix DHT-II needs $log_2 N$ rounds of butterflies' computation and each round uses $2N$ additions and N multiplications. Additionally, one $N/2$ and two $N/4$ length DHTs-II must be computed, thus the whole split radix DHT-II fulfills the recurrences:

$$M(N) = N + M(\tfrac{N}{2}) + 2M(\tfrac{N}{4})$$
$$A(N) = 2N + A(\tfrac{N}{2}) + 2A(\tfrac{N}{4}) \qquad (25)$$

where $A(N)$ and $M(N)$ stand for the number of real additions and multiplications respectively. Solving the complexity relations in (25), by repeatedly substitution of the initial values $M(4) = 2$, $M(4) = 6$ and $A(4) = 12$, $A(8) = 24$, we get the closed form complexity:

$$M(N) = \left(\tfrac{2}{3}\right) N\log_2 N - \left(\tfrac{11}{18}\right) N - \left(\tfrac{8}{9}\right)(-1)^m$$
$$A(N) = \left(\tfrac{4}{3}\right) N\log_2 N - \left(\tfrac{19}{18}\right) N - \left(\tfrac{4}{9}\right)(-1)^m \qquad (26)$$

Comparing the computational complexity of the radix-2 FHT algorithm [25] with this algorithm based on the same implementaion, shows an important saving in the amount of arithmetic complexity can be achieved as listed in Table 1. The saving is up to 10% in the number of additions and around 29% in the number of multiplications.

Table 1. Counts in the arithmetic complexities of split-radix and radix-2 DHT-II algorithms

Transform Length	Split-raix DHT-II algorithm (A)		Radix−2 DHT-II algorithm (B)		Saving % (A/B)	
N	*Mults.*	*Adds.*	*Mults.*	*Adds.*	*Mults.*	*Adds.*
8	12	24	12	24	0	0
16	32	68	40	72	20	5.55
32	88	180	112	192	21.42	6.25
64	216	444	288	480	25	7.50
128	520	1060	704	1152	26.13	7.98
256	1208	2460	1664	2688	27.40	8.48
512	2760	5604	3840	6144	28.12	8.78
1024	6200	12572	8704	13824	28.76	9.05

4. CONCLUSION

This paper presents the development of a novel direct algorithm for the fast calculation of the type-II DHT using split-radix DIT approach. The algorithm has been implemented and its arithmetical complexity has been analyzed. Comparisons between the developed algorithm and the basic radix-2 algorithm, based on the amount of the arithmetical operations have been performed. This comparison has revael that the developed split radix algorithm involve much less arithmetical complexity than radix-2 algorithm (10% saving in the number of additions and around 29% saving in the number of multiplications have been obtained). The proposed algorithm has regular and simple butterfly framework, and has in place computations. Therefore it enables us to develop and improve the performance of the multidimensional split-vector radix (SVR) algorithms for the multidimensional (MD) GDHT for higher dimensions with regard to accesses to the lookup table and to the number of twiddle factor estimations without any increase in the structural or computational complexities.

REFERENCES

[1] R. N. Bracewell, "The Hartley transform," *Oxford University Press, Inc*., 1986.
[2] G. Bi, *et al.*, "Fast generalized DFT and DHT algorithms," *Elsevier Signal Processing Journal,* vol. 65, no. 3, pp. 383-390, 1998.
[3] H. Sorensen, D. Jones, C. Burrus, and M. Heideman, "On computing the discrete Hartley transform," *IEEE Transactions on Acoustics, Speech and Signal Processing.,* vol. 33, no. 5, pp. 1231-1238, 1985.
[4] H. Neng-Chung, I. C. Hong, and O. K. Ersoy, "Generalized discrete Hartley transforms," *IEEE Transactions on Signal Processing,* vol. 40, no. 12, pp. 2931-2940, 1992.
[5] V. Britanak and K. R. Rao, "The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transforms computation," *Elsevier Signal Processing Journal,* vol. 79, no. 2, pp. 135-150, 1999.
[6] M. T. Hamood, "New Decimation-In-Time Fast Hartley Transform Algorithm," *International Journal of Electrical and Computer Engineering,* vol. 6, no. 4, pp. 1654-1661, 2016.
[7] G. Bi and Y. Zeng, "Transforms and Fast Algorithms for Signal Analysis and Representations," *Birkhäuser Boston*, 2012.
[8] K. Jones, "Design and parallel computation of regularised fast Hartley transform," *IEE Proceedings-Vision, Image and Signal Processing,* pp. 70-78, 2006.
[9] G. A. Shah, *et al.*, "A new fast radix-2 decimation-in-frequency algorithm for computing the discrete Hartley transform," *1st International Conference on Computational Intelligence, Communication Systems and Networks,* pp. 363-368, 2009.
[10] M. T. Hamood and S. Boussakta, "New radix-based FHT algorithm for computing the discrete Hartley transform," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 1581-1584, 2011.
[11] P. Duhamel and H. Hollmann, "Split radix' FFT algorithm," *Electronics Letters,* vol. 20, no. 1, pp. 14-16, 1984.
[12] S. C. Chan, *et al.*, "Split vector-radix fast Fourier transform," *IEEE T. Signal. Proces.,* vol. 40, pp. 2029-39, 1992.
[13] W. C. Yeh, *et al.*, "High-speed and low power split-radix FFT," *IEEE T. Signal. Proces.,* vol. 51, no. 3, pp. 864-74, 2003.
[14] N. Anupindi, *et al.*, "Split-radix FHT algorithm for real-symmetric data," *Electron. Lett.,* vol. 26, no. 23, pp. 1973-75, 1990.
[15] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "An efficient split-radix FHT algorithm," *Proceedings of the 2004 International Symposium on Circuits and Systems,* 2004.
[16] M. T. Hamood, *et al.*, "Decimation-in-frequency split-radix algorithm for computing new Mersenne number transform," *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 298-302, 2010.
[17] D. F. Chiper, "A Structured Dual Split-Radix Algorithm for the Discrete Hartley Transform of Length 2^N," *Springer Circuits, Systems, and Signal Processing Journal,* vol. 37, no. 9, pp. 290-304, January 01 2018.
[18] P. Duhamel and M. Vetterli, "Fast Fourier transforms: A tutorial review and a state of the art," *Elsevier Signal Processing Journal,* vol. 19, no. 4, pp. 259-299, 1990.
[19] H. Sorensen, M. Heideman, and C. Burrus, "On computing the split-radix FFT," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 34, no. 1, pp. 152-156, 1986.
[20] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A new split-radix FHT algorithm for length-$q*2^m$ DHTs," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 51, no. 10, pp. 2031-2043, 2004.
[21] K. Jones, "The Regularized Fast Hartley Transform," *Springer Science*, 2009.
[22] Z. Wang, G. A. Jullien, *et al.*, "The generalized discrete W transform and its application to interpolation," *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems & Computers,* 1992.
[23] T. Bortfeld and W. Dinter, "Calculation of multidimensional Hartley transforms using one-dimensional Fourier transforms," *IEEE Transactions on Signal Processing,* vol. 43, no. 5, pp. 1306-1310, 1995.
[24] O. Nibouche, S. Boussakta, M. Darnell, and M. Benaissa, "Algorithms and pipeline architectures for 2-D FFT and FFT-like transforms," *Elsevier Digital Signal Processing Journal,* vol. 20, no. 4, pp. 1072-1086, 2010.
[25] M. T. Hamood, "Fast Algorithm for Computing the Discrete Hartley Transform of Type-II," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI),* vol. 4, no. 2, pp. 120-125, 2016.

BIOGRAPHIES AUTHOR

Mounir Taha Hamood received the B.Sc. degree in electrical engineering from University of Technology, Baghdad, Iraq in 1990 and the M.Sc. degree in electronic and communications engineering from Al-Nahrain University, Baghdad, Iraq in 1995. He graduated from Newcastle University, Newcastle upon Tyne, U.K in 2012 with the PhD degree in communications and signal processing. His doctoral research was in the development of efficient algorithms for fast computation of discrete transforms. From 2012 to 2016 He was a lecturer in signal processing for communication at Tikrit University. He is currently an associate professor of signal processing at the Department of Electrical Engineering, College of Engineering, Tikrit University, Tikrit, Iraq. His research interest includes discrete transforms, fast algorithms for digital signal processing in one and multidimensional applications, and communication systems.