$\|(A_{T_b}{}'A_{T_b})^{-1}\| = \frac{1}{\lambda_{\min}(A_{T_b}{}'A_{T_b})} \leq \frac{1}{1-\delta_{k_b}}$. Thus, we get $\|A_{\tilde{T}_d}{}'M(T_b)A_\Delta\| \leq (\theta_{\check{s},u} + \frac{\theta_{\check{s},k_b}\theta_{u,k_b}}{1-\delta_{k_b}})$. Consider the second term $\|(A_\Delta{}'M(T_b)A_\Delta)^{-1}\|$. Since $A_\Delta{}'M(T_b)A_\Delta$ is positive definite, using fact i) and (22), $\|(A_\Delta{}'M(T_b)A_\Delta)^{-1}\| = \frac{1}{\lambda_{\min}(A_\Delta{}'M(T_b)A_\Delta)} \leq \frac{1}{(1-\delta_u)-\frac{\theta^2_{u,k_b}}{1-\delta_{k_b}}}$. Using fact vi), the third term, $\|\mathrm{sgn}(x_\Delta)\|_2 = \sqrt{u}$.

Define the set, E, as $E := \{j \in (T\cup\Delta)^c : |A_j{}'\tilde{w}| > \frac{a_{k_b}(u,\check{s})\sqrt{u}}{\sqrt{\check{s}}}\}$. Notice that $|E|$ must obey $|E| < \check{s}$ since otherwise we can contradict (24) by taking $\tilde{T}_d \subseteq E$. Since $|E| < \check{s}$ and E is disjoint with $T\cup\Delta$, (24) holds for $\tilde{T}_d \equiv E$, i.e., $\|A_E{}'\tilde{w}\|_2 \leq a_{k_b}(u,\check{s})\sqrt{u}$. Also, by definition of E, $|A_j{}'\tilde{w}| \leq \frac{a_{k_b}(u,\check{s})\sqrt{u}}{\sqrt{\check{s}}}$, for all $j \notin T\cup\Delta\cup E$. Thus, $\tilde{w}$ satisfies the third condition of the lemma.

Finally, $\|\tilde{w}\|_2 \leq \|M(T_b)\|\,\|A_\Delta\|\,\|(A_\Delta{}'M(T_b)A_\Delta)^{-1}\|\sqrt{u} \leq K_{k_b}(u)\sqrt{u}$. This follows using fact v); $\|A_\Delta\| \leq \sqrt{1+\delta_u}$; and fact i) and (22). Thus, we have found a $\tilde{w}$ and E that satisfy all required conditions.

REFERENCES

[1] W. Lu and N. Vaswani, "Exact reconstruction conditions and error bounds for regularized modified basis pursuit (reg-modified-BP)," in *Proc. 44th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 7–10, 2010, pp. 763–767.

[2] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, Sep. 2010.

[3] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[4] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[5] N. Vaswani, "LS-CS-residual (LS-CS): compressive sensing on the least squares residual," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4108–4120, Aug. 2010.

[6] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," presented at the Eur. Signal Process. Conf. (EUSIPCO), Lausanne, Switzerland, Aug. 25–29, 2008.

[7] J. Y. Park and M. B. Wakin, "A multiscale framework for compressive sensing of video," presented at the Picture Coding Symp. (PCS), Chicago, IL, May 6–8, 2009.

[8] . : C. Qiu and N. Vaswani, "ReProCS: A missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise," arXiv: 1106.3286, 2011.

[9] C. Qiu and N. Vaswani, "Support-predicted modified-CS for principal components' pursuit," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, 2011, pp. 668–672.

[10] E. Candes, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Series I*, no. 346, pp. 589–592, 2008.

[11] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[12] D. Donoho and J. Tanner, "High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension," *Discr. Comput. Geom.*, vol. 35, no. 4, pp. 617–652, 2006.

[13] R. Von Borries, C. J. Miosso, and C. Potes, "Compressive sensing reconstruction with prior information by iteratively reweighted least-squares," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2424–2431, Jun. 2009.

[14] A. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Analyzing weighted 11 minimization for sparse recovery with nonuniform sparse models," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1985–2001, May 2011.

[15] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM J. Imag. Sci.*, vol. 3, pp. 462–491, 2010.

[16] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, San Diego, CA, Oct. 12–15, 2008, pp. 893–896.

[17] W. Lu and N. Vaswani, "Regularized modified BPDN for noisy sparse reconstruction with partial erroneous support and signal value knowledge," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 182–196, Jan. 2012.

[18] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, pp. 1982–2001, Apr. 2010.

[19] P. Schniter, L. Potter, and J. Ziniel, "Fast Bayesian matching pursuit: Model uncertainty and parameter estimation for sparse linear models," in *Proc. Inf. Theory Appl. (ITA)*, La Jolla, VA, Jan. 27–Feb. 1, 2008.

[20] S. Som, L. C. Potter, and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," in *Proc. 44th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 7–10, 2010, pp. 243–247.

[21] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[23] W. Lu and N. Vaswani, "Exact reconstruction conditions for regularized modified basis pursuit (reg-modified-BP)," [Online]. Available: http://arxiv.org/pdf/1108.3350.pdf, arXiv:1108.3350v1, 2011

Generalized New Mersenne Number Transforms

Said Boussakta, Monir T. Hamood, and Nick Rutter

***Abstract*—Two new number theoretic transforms named as odd and odd-squared new Mersenne number transforms are introduced for incorporation into a generalized new Mersenne number transforms (GNMNTs) suite, which are defined in finite fields modulo Mersenne primes where arithmetic operations and residue reductions are simple to implement. This suite is categorized by type, with detailed instructions regarding their derivations. An example is given which shows their suitability for the calculation of different types of convolutions, along with an analysis of their arithmetic complexities for radix-2 and split radix algorithms. This in turn shows that these new transforms are suitable for fast error free calculation of convolutions/correlations for signal processing and other applications.**

***Index Terms*—New Mersenne number transform (NMNT), number theoretic transforms (NTTs), odd new Mersenne number transform (ONMNT), odd-squared new Mersenne number transform ($\mathbf{O^2NMNT}$).**

I. INTRODUCTION

The use of number theoretic transforms (NTTs) have been firmly established within the field of signal processing [1]. This is owing to their contributing ability to perform error-free calculations over a field or a ring of integers whilst maintaining the Cyclic Convolution Property (CCP). In contrast to other methods of calculation, such as the fast Fourier transform (FFT) which involves complex arithmetic with rounding and/or truncation errors in its calculations; errors also arise in the multiplication with cosine and sine functions which are irrational, preventing exact representation in a finite precision machine [2]. Additionally, the use of NTTs have been proven to provide such

Manuscript received February 15, 2011; revised June 29, 2011 and October 17, 2011; accepted January 02, 2012. Date of publication January 26, 2012; date of current version April 13, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sofia C. Olhede.

The authors are with the School of Electrical, Electronic and Computer Engineering, Newcastle University, NE1 7RU Newcastle Upon Tyne, U.K. (e-mail: s.boussakta@ncl.ac.uk; m.t.hamood@ncl.ac.uk; nick.rutter@ncl.ac.uk).

Digital Object Identifier 10.1109/TSP.2012.2186131

error-free calculations by incorporating a significantly lesser-degree of complexity than the aforementioned methods, especially when using specific NTTs which provide a simple residue reduction based upon a modulus power of two. The Fermat number transform (FNT) [3] and Mersenne number transform (MNT) [4] being notable candidates which utilize a modulus that satisfies this criteria in particular, due to using moduli of $F_t = 2^{2^t} + 1$ and $Mp = 2^p - 1$ respectively. However, while the FNT has been extensively studied and is a worthwhile candidate for the processing of error-free calculations through the use of a simplified kernel $\alpha = 2$ or $\alpha = \sqrt{2}$, it has the inconvenience of requiring an odd number of bits. Similarly, while the MNT also has a simplified kernel $(\alpha = \pm 2)$, it has the distinct disadvantage that the size N is very tightly bound to the size of p, being of either $N = p$ or $N = 2p$. This transform length is too short and is also not a power of two and as such, the Cooley–Tukey fast algorithm method cannot be used for the MNT.

Since its introduction [5], the new Mersenne number transform (NMNT) has proved to be a more flexible alternative over both the FNT and MNT techniques, offering both long transform lengths (powers of two) and flexibility due to the variable size $N = 2^m$ having a range $m = 1, 2, \ldots, p$.

While the NMNT in its present form currently has many applications in digital filtering [6], image processing [7] and encryption [8], it has so far had the restriction that there has only been a single type of NMNT, unlike other transforms [9]–[12]. Therefore, the aim of this correspondence is to introduce two new transforms named odd-NMNT (ONMNT) and odd-squared NMNT $(\mathrm{O}^2\mathrm{NMNT})$, which can be used for efficient calculation of error free convolutions/correlations for signal and image processing applications.

II. Formal Definition of NMNT Types

This section introduces two new Mersenne number transforms, namely the odd and odd-squared NMNTs, the calculation of their transform parameters, the proof of the forward/inverse transforms and the skew-cyclic convolution property (SCC).

A. NMNT and Preliminary

For the sake of clarity, the original NMNT transform is briefly described here. The NMNT $X(k)$ of an integer sequence $x(n)$ of transform length $N = 2^m$ for $m = 1, 2, \ldots, p$ is defined as [5]

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n)\ \beta(nk) \right\rangle_{Mp} \qquad k = 0, 1, \ldots, N-1 \tag{1}$$

where $\langle . \rangle_{Mp}$ denotes modulo Mp, $Mp = 2^p - 1$ is a Mersenne prime for $p = 2, 3, 5, 7, 13, 17, 19, \ldots$, etc and the transform kernel β is given by

$$\begin{aligned} \beta(n) &= \beta_1(n) + \beta_2(n) \\ \beta_1(n) &= \langle \mathrm{Re}(\alpha_1 + j\alpha_2)^n \rangle_{Mp} \\ \beta_2(n) &= \langle \mathrm{Im}(\alpha_1 + j\alpha_2)^n \rangle_{Mp} \end{aligned} \tag{2}$$

where

$$\begin{aligned} \alpha_1 &= \pm \langle 2^q \rangle_{Mp}, \\ \alpha_2 &= \pm \langle -3^q \rangle_{Mp} \quad \text{and} \quad q = 2^{p-2}. \end{aligned} \tag{3}$$

Re(.) and Im(.) denote real and imaginary parts of the enclosed terms respectively. For transform lengths equal to $\frac{N}{d}$, β_1 and β_2 can be calculated as

$$\begin{aligned} \beta_1(n) &= \left\langle \mathrm{Re}\left((\alpha_1 + j\alpha_2)^d\right)^n \right\rangle_{Mp} \\ \beta_2(n) &= \left\langle \mathrm{Im}\left((\alpha_1 + j\alpha_2)^d\right)^n \right\rangle_{Mp} \end{aligned} \tag{4}$$

where $d = (\frac{2^{p+1}}{N})$ is an integer power of two and the term $(\alpha_1 + j\alpha_2)$ is of order 2^{p+1}. The inverse NMNT is defined as

$$x(n) = \left\langle N^{-1} \sum_{k=0}^{N-1} X(k)\ \beta(nk) \right\rangle_{Mp} \qquad n = 0, 1, \ldots, N-1. \tag{5}$$

From (5), it is clear that the NMNT has the same inverse and except for the scale factor N^{-1}, there is no need to distinguish between the forward and inverse transforms.

B. Odd New Mersenne Number Transform

The forward odd NMNT (ONMNT) of an integer sequence $x(n)$ for transform length $N = 2^m$ for $m = 1, 2, \ldots, p-1$ is defined as

$$X_o(k) = \left\langle \sum_{n=0}^{N-1} x(n)\ \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} \qquad k = 0, 1, \ldots, N-1. \tag{6}$$

Let $M_{nk} = \beta\left(\frac{n(2k+1)}{2}\right)$ for $0 \le n,\ k \le N-1$ be the elements of the ONMNT matrix. According to (6), $\mathbf{M}$ can be written as

$$\mathrm{M} = \begin{bmatrix} \beta(0) & \beta(0) & \beta(0) & \beta(0) & \cdots & \beta(0) \\ \beta(\frac{1}{2}) & \beta(\frac{3}{2}) & \beta(\frac{5}{2}) & \beta(\frac{7}{2}) & \cdots & \beta(\frac{2N-1}{2}) \\ \beta(1) & \beta(3) & \beta(5) & \beta(7) & \cdots & \beta(2N-1) \\ \beta(\frac{3}{2}) & \beta(\frac{9}{2}) & \beta(\frac{15}{2}) & \beta(\frac{21}{2}) & \cdots & \beta\left(\frac{3(2N-1)}{2}\right) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta(\frac{N-1}{2}) & \beta(\frac{3(N-1)}{2}) & \beta(\frac{5(N-1)}{2}) & \beta(\frac{7(N-1)}{2}) & \cdots & \beta(\frac{(2N-1)(N-1)}{2}) \end{bmatrix}. \tag{7}$$

For matrix $\mathbf{M}$ to be orthogonal, its transpose should be equal to its inverse. In order to achieve this, the inverse ONMNT matrix is obtained by multiplying the transpose of $\mathbf{M}$ by a scale factor N^{-1}. Therefore, the inverse ONMNT is defined as

$$x(n) = \left\langle N^{-1} \sum_{k=0}^{N-1} X_o(k)\ \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} \qquad n = 0, 1, \ldots, N-1. \tag{8}$$

The proof that (8) is the inverse of (6) can be obtained by assuming that $\tilde{x}(n)$ is the inverse ONMNT of $X_o(k)$. Therefore, we need to prove that $\tilde{x}(n)$ is equal to $x(n)$ as follows:

$$\begin{aligned} \tilde{x}(n) &= \left\langle N^{-1} \sum_{k=0}^{N-1} X_o(k)\ \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} \\ &= \left\langle N^{-1} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} x(m)\ \beta\left(\frac{m(2k+1)}{2}\right) \right) \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} \\ &= \left\langle N^{-1} \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} \beta\left(\frac{m(2k+1)}{2}\right) \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp}. \end{aligned} \tag{9}$$

Using the orthogonality property of the $\beta(.)$ function which was proved in [5], the second summation of (9) can be written as

$$\left\langle \sum_{k=0}^{N-1} \beta\left(\frac{m(2k+1)}{2}\right) \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} = \begin{cases} N & \text{if } m = n \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Substituting (10) into (9), we get

$$\tilde{x}(n) = \left\langle N^{-1} \sum_{m=0}^{N-1} x(m) \quad \begin{cases} N & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \right\rangle_{Mp} = x(n). \tag{11}$$

Therefore, (6) and (8) form a transform pair.

C. Odd-Squared New Mersenne Number Transform $\mathrm{O^2NMNT}$

The forward $\mathrm{O^2NMNT}$ and its inverse of an integer sequence $x(n)$ and transform length $N = 2^m$ where $m = 1, 2, \ldots, p-2$ is defined as

$$X_{o^2}(k) = \left\langle \sum_{n=0}^{N-1} x(n)\, \beta\left(\frac{(2n+1)(2k+1)}{4}\right) \right\rangle_{Mp} \quad k = 0, 1, \ldots, N-1. \tag{12}$$

and

$$x(n) = \left\langle N^{-1} \sum_{k=0}^{N-1} X_{o^2}(k)\, \beta\left(\frac{(2n+1)(2k+1)}{4}\right) \right\rangle_{Mp} \quad n = 0, 1, \ldots, N-1. \tag{13}$$

The proof that (12) and (13) form a transform pair can be shown by following the same procedure shown in (9)–(11). Also, it should be noted that the forward and inverse $\mathrm{O^2NMNT}$s are exactly the same except from the scale factor N^{-1}.

D. Calculation of the Transform Parameters

As shown from the definition of the ONMNT and $\mathrm{O^2NMNT}$, it is required to calculate the half and quarter index values of the transform parameters (i.e., $\beta_1(\frac{n}{2})$ and $\beta_2(\frac{n}{2})$ for the ONMNT and $\beta_1(\frac{n}{4})$ and $\beta_2(\frac{n}{4})$ for the $\mathrm{O^2NMNT}$). These values can be calculated from the definitions of β_1 and β_2 given in (4) as follows:

i) The transform length N of ONMNT is defined by $N = 2^m$, therefore the value of d for maximum transform length ($N_{\max} = 2^{p-1}$) is equal to $d = \left(\frac{2^{p+1}}{2^{p-1}}\right) = 4$. Therefore, the ONMNT parameters $\beta_1\left(\frac{n}{2}\right)$ and $\beta_2\left(\frac{n}{2}\right)$ for $N_{\max}$ can be calculated as

$$\begin{aligned} \beta_1\left(\frac{n}{2}\right) &= \left\langle \mathrm{Re}\left((\alpha_1 + j\alpha_2)^2\right)^n \right\rangle_{Mp} \\ \beta_2\left(\frac{n}{2}\right) &= \left\langle \mathrm{Im}\left((\alpha_1 + j\alpha_2)^2\right)^n \right\rangle_{Mp}. \end{aligned} \tag{14}$$

ii) Similarly, the transform length N of $\mathrm{O^2NMNT}$ is defined by $N = 2^m, 0 < m < p-1$ and therefore, the value of d for the maximum transform length ($N_{\max} = 2^{p-2}$) for the $\mathrm{O^2NMNT}$ transform is equal to $d = \left(\frac{2^{p+1}}{2^{p-2}}\right) = 8$. Likewise, for a given p and N, the values of $\mathrm{O^2NMNT}$ parameters $\beta_1\left(\frac{n}{4}\right)$ and $\beta_2\left(\frac{n}{4}\right)$ have the same values of the ONMNT parameters $\beta_1\left(\frac{n}{2}\right)$ and $\beta_2\left(\frac{n}{2}\right)$ at length $2N$.

III. Fast Algorithms for GNMNTs

Since the transform lengths of the GNMNTs are powers of two, it is possible to develop fast algorithms such as the radix-2, radix-4 and split-radix, where the decimation can be done either in time (DIT) or in frequency (DIF).

A. Radix-2 DIT Algorithm for the ONMNT

Dividing the input sequence into its odd and even parts, (6) can be decomposed to $X(k) = X_1(k) + X_2(k)$, where $X_1(k)$ and $X_2(k)$ are given by:

$$X_1(k) = \left\langle \sum_{n=0}^{\frac{N}{2}-1} x(2n)\, \beta((2k+1)n) \right\rangle_{Mp} = X_{2n}(k) \tag{15}$$

and

$$X_2(k) = \left\langle \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, \beta\left(\left(\frac{2k+1}{2}\right)(2n+1)\right) \right\rangle_{Mp}. \tag{16}$$

Using the following NMNT identity that was proved in [5]:

$$\beta(a+b) = \beta_1(a)\beta(b) + \beta_2(a)\beta(-b). \tag{17}$$

The $\beta(.)$ term in (16) can be simplified as

$$\begin{aligned} \beta\left(\left(\frac{2k+1}{2}\right)(2n+1)\right) = \Big\langle & \beta_1\left(\frac{2k+1}{2}\right) \\ & \beta\left((2k+1)\, n\right) + \beta_2\left(\frac{2k+1}{2}\right) \beta\left(-(2k+1)n\right) \Big\rangle_{Mp}. \end{aligned} \tag{18}$$

Therefore, (16) can be written as

$$\begin{aligned} X_2(k) = \Big\langle & \beta_1\left(\frac{2k+1}{2}\right) \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, \beta\left((2k+1)n\right) \\ & + \beta_2\left(\frac{2k+1}{2}\right) \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, \beta\left(-(2k+1)n\right) \Big\rangle_{Mp}. \end{aligned} \tag{19}$$

Substituting (15) and (19) into, we obtain the recursive formula for the radix-2 ONMNT:

$$\begin{aligned} X(k) = \Big\langle & X_{2n}(k) + \beta_1\left(\frac{2k+1}{2}\right) X_{2n+1}(k) \\ & + \beta_2\left(\frac{2k+1}{2}\right) X_{2n+1}(N-k-1) \Big\rangle_{Mp} \end{aligned} \tag{20}$$

where $X_{2n}(k)$ and $X_{2n+1}(k)$ can be identified as the $\left(\frac{N}{2}\right)$-point ONMNTs of the even and odd parts of $x(n)$, respectively. Another point of the DIT decomposition $X\left(k + \frac{N}{2}\right)$ can be computed as follows:

$$\begin{aligned} X\left(k + \frac{N}{2}\right) = \Big\langle & X_{2n}(k) - \beta 1\left(\frac{2k+1}{2}\right) X_{2n+1}(k) \\ & - \beta_2\left(\frac{2k+1}{2}\right) X_{2n+1}(N-k-1) \Big\rangle_{Mp}. \end{aligned} \tag{21}$$

Combining the four points together, produces an in-place radix-2 ONMNT butterfly as shown in Fig. 1.

B. Split-Radix Algorithm for the ONMNT

The split-radix algorithm is one of the most efficient algorithms for computing fast transforms. It applies radix-2 decomposition to the even samples and radix-4 decomposition to the odd samples. Therefore $X_o(k)$ can be divided into even-indexed samples $X_o^{\mathrm{ev}}(k)$ and odd-indexed samples $X_o^{\mathrm{od}}(k)$, each of length $\frac{N}{2}$ given by

$$X_o(k) = \left\langle X_o^{\mathrm{ev}}(k) + X_o^{\mathrm{od}}(k) \right\rangle_{Mp} \tag{22}$$

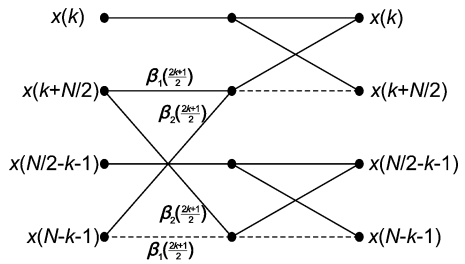

Fig. 1. An in-place butterfly of the radix-2 ONMNT DIT algorithm; where solid and dotted lines stand for addition and subtraction, respectively.

where $X_o^{\mathrm{ev}}(k)$ is as given by (15), and it is equal to $X_{2n}(k)$, while $X_o^{\mathrm{od}}(k)$ can be developed by applying radix-4 algorithm to the odd-indexed samples as follows:

$$X_o^{\mathrm{od}}(k) = \left\langle \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,\beta\left((4n+1)\left(\frac{2k+1}{2}\right)\right) + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\beta\left((4n+3)\left(\frac{2k+1}{2}\right)\right)\right\rangle_{Mp} \quad (23)$$

Applying the NMNT identity given in (17), $X_o^{\mathrm{od}}(k)$ can be further decomposed as

$$\begin{aligned} X_o^{\mathrm{od}}(k) = \Bigg\langle & \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,\beta\left(4n\left(\frac{2k+1}{2}\right)\right)\beta_1\left(\frac{2k+1}{2}\right) \\ & + \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,\beta\left(-4n\left(\frac{2k+1}{2}\right)\right)\beta_2\left(\frac{2k+1}{2}\right) \\ & + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,\beta\left(4n\left(\frac{2k+1}{2}\right)\right)\beta_1\left(3\frac{2k+1}{2}\right) \\ & + \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,\beta\left(-4n\left(\frac{2k+1}{2}\right)\right)\beta_2\left(3\frac{2k+1}{2}\right)\Bigg\rangle_{Mp}. \end{aligned} \quad (24)$$

Equation (24) can be written as

$$\begin{aligned} X_o^{\mathrm{od}}(k) = \Bigg\langle & X_{4n+1}(k)\beta_1\left(\frac{2k+1}{2}\right) + X_{4n+1}\left(\frac{N}{4}-k-1\right)\beta_2\left(\frac{2k+1}{2}\right) \\ & + X_{4n+3}(k)\beta_2\left(\frac{3(2k+1)}{2}\right) + X_{4n+3}\left(\frac{N}{4}-k-1\right)\beta_2\left(\frac{3(2k+1)}{2}\right)\Bigg\rangle_{Mp} \end{aligned} \quad (25)$$

where $X_{4n+1}(k)$ and $X_{4n+3}(k)$ are two ONMNTs of length $\frac{N}{4}$, defined as

$$X_{4n+1}(k) = \left\langle \sum_{n=0}^{\frac{N}{4}-1} x(4n+1)\,\beta\left(4n\left(\frac{2k+1}{2}\right)\right)\right\rangle_{Mp} \quad (26)$$

$$X_{4n+3}(k) = \left\langle \sum_{n=0}^{\frac{N}{4}-1} x(4n+3)\,\beta\left(4n\left(\frac{2k+1}{2}\right)\right)\right\rangle_{Mp}. \quad (27)$$

Substituting (15) and (25) into (22), $X_o(k)$ can be written as

$$\begin{aligned} X_o(k) = \Bigg\langle & X_{2n}(k) + \Bigg[X_{4n+1}(k)\beta_1\left(\frac{2k+1}{2}\right) + X_{4n+1}\left(\frac{N}{4}-k-1\right)\beta_2\left(\frac{2k+1}{2}\right)\Bigg] \\ & + \Bigg[X_{4n+3}(k)\beta_1\left(\frac{3(2k+1)}{2}\right) + X_{4n+3}\left(\frac{N}{4}-k-1\right)\times\beta_2\left(\frac{3(2k+1)}{2}\right)\Bigg]\Bigg\rangle_{Mp}. \end{aligned} \quad (28)$$

Using the NMNT identities [5], other DIT decompositions can be derived, as follows:

$$\begin{aligned} X_o\left(k+\frac{N}{4}\right) = \Bigg\langle & X_{2n}\left(k+\frac{N}{4}\right) - \Bigg[X_{4n+1}(k)\beta_2\left(\frac{2k+1}{2}\right) - X_{4n+1}\left(\frac{N}{4}-k-1\right)\beta_1\left(\frac{2k+1}{2}\right)\Bigg] \\ & + \Bigg[X_{4n+3}(k)\beta_2\left(\frac{3(2k+1)}{2}\right) - X_{4n+3}\left(\frac{N}{4}-k-1\right)\beta_1\left(\frac{3(2k+1)}{2}\right)\Bigg]\Bigg\rangle_{Mp} \end{aligned} \quad (29)$$

$$\begin{aligned} X_o\left(k+\frac{N}{2}\right) = \Bigg\langle & X_{2n}(k) - \Bigg[X_{4n+1}(k)\beta_1\left(\frac{2k+1}{2}\right) + X_{4n+1}\left(\frac{N}{4}-k-1\right)\beta_2\left(\frac{2k+1}{2}\right)\Bigg] \\ & \Bigg[X_{4n+3}(k)\beta_1\left(\frac{3(2k+1)}{2}\right) + X_{4n+3}\left(\frac{N}{4}-k-1\right)\beta_2\left(\frac{3(2k+1)}{2}\right)\Bigg]\Bigg\rangle_{Mp} \end{aligned} \quad (30)$$

$$\begin{aligned} X_o\left(k+\frac{3N}{4}\right) = \Bigg\langle & X_{2n}\left(k+\frac{N}{4}\right) + \Bigg[X_{4n+1}(k)\beta_2\left(\frac{2k+1}{2}\right) - X_{4n+1}\left(\frac{N}{4}-k-1\right)\beta_1\left(\frac{2k+1}{2}\right)\Bigg] \\ & - \Bigg[X_{4n+3}(k)\beta_2\left(\frac{3(2k+1)}{2}\right) - X_{4n+3}\left(\frac{N}{4}-k-1\right)\beta_1\left(\frac{3(2k+1)}{2}\right)\Bigg]\Bigg\rangle_{Mp}. \end{aligned} \quad (31)$$

Combining the eight points together produces an in-place split-radix ONMNT butterfly as shown in Fig. 2.

C. Fast Algorithms for the $\mathrm{O}^2\mathrm{NMNT}$

Converting the $\mathrm{O}^2\mathrm{NMNT}$ to ONMNT by multiplying both sides of the $\mathrm{O}^2\mathrm{NMNT}$ given in (12) by $2\beta_1\left(\frac{(2k+1)}{4}\right)$, we obtain

$$\left[2\beta_1\left(\frac{2k+1}{4}\right)\right]X_{o^2}(k) = \left\langle 2\sum_{n=0}^{N-1} x(n) \times\beta_1\left(\frac{2k+1}{4}\right)\beta\left(\frac{(2k+1)(2n+1)}{4}\right)\right\rangle_{Mp}. \quad (32)$$

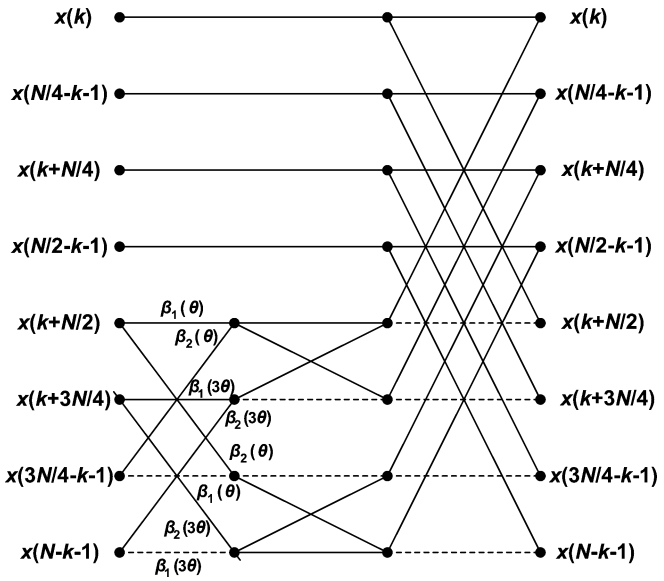

Fig. 2. An in-place butterfly of the split-radix ONMNT DIT algorithm; where $\boldsymbol{\theta} = \frac{(2k+2)}{2}$ solid and dotted lines stand for addition and subtraction, respectively.

From (17), we can obtain the following relation:

$$\beta(a-b)+\beta(a+b)=2\beta_1(a)\beta(b). \tag{33}$$

Substituting (33) into (32) with $a=\frac{(2k+1)}{4}$ and $b=\frac{(2k+1)(2n+1)}{4}$, then for $a-b=\frac{(2k+1)n}{2}$ and $a+b=\frac{(2k+1)(n+1)}{2}$, we get

$$\left[2\beta_1\left(\frac{2k+1}{4}\right)\right]X_{o2}(k)=\left\langle\sum_{n=0}^{N-1}x(n)\,\beta\left(\frac{(2k+1)n}{2}\right)+\sum_{n=0}^{N-1}x(n)\,\beta\left(\frac{(2k+1)(2n+1)}{2}\right)\right\rangle_{Mp}. \tag{34}$$

Using (17) and rearranging, (34) can be written as

$$\left[2\beta_1\left(\frac{2k+1}{4}\right)\right]X_{o2}(k)=\left\langle\sum_{n=0}^{N-1}\left[x(n)+x(n-1)\right]\beta\left(\frac{(2k+1)n}{2}\right)\right\rangle_{Mp}. \tag{35}$$

The right-hand side of (35) is exactly an ONMNT of length N.

Since the sequence $x(n)$ is anti-periodic, then $x(-1)=-x(N-1)$ and as such, (35) can be written as

$$X_{o2}(k)=\left\langle\bar{\beta}_1\left(\frac{2k+1}{4}\right)\left\{\left[x(0)-x(N-1)\right]+\sum_{n=1}^{N-1}\left[x(n)+x(n-1)\right]\beta\left(\frac{(2k+1)n}{2}\right)\right\}\right\rangle_{Mp} \tag{36}$$

where

$$\bar{\beta}_1\left(\frac{2k+1}{4}\right)=\left[2\beta_1\left(\frac{2k+1}{4}\right)\right]^{-1} \tag{37}$$

and $[.]^{-1}$ denotes the multiplicative inverse of the enclosed term.

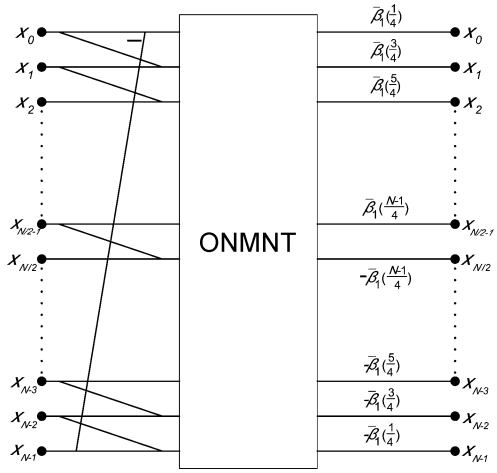

Fig. 3. Fast Algorithm for the $\mathbf{O^2NMNT}$ based on the ONMNT.

Therefore, a length N $\mathrm{O^2NMNT}$ is reduced to a length N ONMNT with just N pre-additions and N post-multiplications. The signal flow graph for this algorithm according to (36) is shown in Fig. 3.

D. Arithmetic Complexity

In general, a radix-2 ONMNT algorithm requires $\log_2 N$ stages, where each stage involves $\frac{N}{4}$ butterflies and each butterfly calculates four points together involving 4 multiplications and 6 additions. Therefore, the calculation of the whole transform satisfies the following equations:

$$\begin{aligned}M_o(N)&=2M_o\left(\frac{N}{2}\right)+N\\A_o(N)&=2A_o\left(\frac{N}{2}\right)+\frac{3N}{2}\end{aligned} \tag{38}$$

where $M_o(N)$ and $A_o(N)$ denotes the total number of integer multiplications and additions required to calculate length N ONMNT respectively. The initial values are taken for a four point ONMNT are $M_o(4)=2$ and $A_o(4)=6$.

The split-radix ONMNT algorithm also requires $\log_2 N$ stages, where each stage involves $\frac{N}{8}$ butterflies and each butterfly calculates eight points together involving 8 multiplications and 16 additions. Therefore, the calculation of the whole transform satisfies the following equations:

$$\begin{aligned}M_o(N)&=M_o\left(\frac{N}{2}\right)+2M_o\left(\frac{N}{4}\right)+N\\A_o(N)&=A_o\left(\frac{N}{2}\right)+2A_o\left(\frac{N}{4}\right)+2N.\end{aligned} \tag{39}$$

For the $\mathrm{O^2NMNT}$ algorithm described in Section III-C, the number of integer multiplications $M_{o2}(N)$ and additions $A_{o2}(N)$, can be calculated by adding N multiplications and N additions to those of the ONMNT:

$$\begin{aligned}M_{o2}(N)&=M_o(N)+N\\A_{o2}(N)&=A_o(N)+N.\end{aligned} \tag{40}$$

Analysis of the radix-2 and split-radix arithmetic complexities over different transform lengths for the ONMNT and $\mathrm{O^2NMNT}$ algorithms is shown in Tables I and II respectively.

TABLE I
OPERATION COUNTS FOR RADIX-2 AND SPLIT-RADIX ONMNT ALGORITHMS

N	Radix-2 ONMNT		Split-radix ONMNT	
	Mults.	*Adds.*	*Mults.*	*Adds.*
8	12	24	12	24
16	40	72	32	68
32	112	192	88	180
64	288	480	216	444
128	704	1152	520	1060
256	1664	2688	1208	2460
512	3840	6144	2760	5604
1024	8704	13824	6200	12572

TABLE II
OPERATION COUNTS FOR RADIX-2 AND SPLIT-RADIX
$\mathbf{O^2NMNT}$ ALGORITHMS

N	Radix-2 O^2NMNT		Split-radix O^2NMNT	
	Mults.	*Adds.*	*Mults.*	*Adds.*
8	20	32	20	32
16	56	88	48	84
32	144	224	120	212
64	352	544	280	508
128	832	1280	648	1188
256	1920	2944	1464	2716
512	4352	6656	3272	6116
1024	9728	13824	7224	13596

IV. SKEW-CYCLIC CONVOLUTION PROPERTY FOR GNMNTS

The skew-cyclic convolution (SCC) $y(n)$ of two sequences $x(n)$ and $h(n)$ for $n = 0, 1, \ldots, N-1$ is defined as

$$y(n) = \sum_{\ell=0}^{n} x(\ell)\, h(n-\ell) - \sum_{\ell=n+1}^{N-1} x(\ell)\, h(N+n-\ell). \tag{41}$$

SCC is an efficient tool for fast computation of the cyclic (CC) and linear convolutions [13]. Also, the SCC is used for computing discrete cosine transforms (DCTs) [14], [15]. Therefore, the SCC can be computed by mapping it to CC [16] (and vice versa), which can then be computed by fast CC algorithms. Furthermore, we can calculate the linear convolution using both SCC and CC. For example, let $y_{\rm cc}(n)$ and $y_{\rm scc}(n)$ be the output of the cyclic and skew cyclic convolutions, respectively, for $n = 0, 1, \ldots, N-1$ and let $y(m) = [y_1(n)\ y_2(n)]$ be the output of the linear convolution for $m = 0, 1, \ldots, 2N-1$. Then $y(m)$ can be computed from $y_{\rm cc}(n)$ and $y_{\rm scc}(n)$ as follows:

$$\begin{aligned} y_1(n) &= \frac{1}{2}\left[y_{\rm CC}(n) + y_{\rm SCC}(n)\right] \\ y_2(n) &= \frac{1}{2}\left[y_{\rm CC}(n) - y_{\rm SCC}(n)\right]. \end{aligned} \tag{42}$$

A significant advantage of this method is that the linear convolution can be calculated by combining two types of circular convolutions (SCC and CC) each of length N instead of the traditional method based on padding with zeros, which requires $2N$ point transforms.

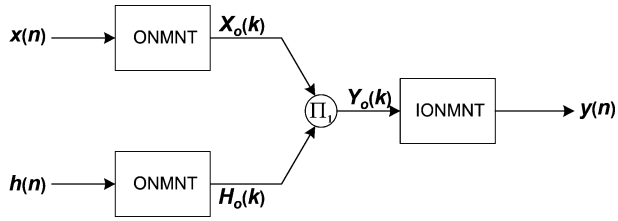

Fig. 4. Fast SCC using the ONMNT.

For error-free calculations, the SCC can be efficiently computed either by ONMNT or $\rm O^2NMNT$.

A. SCC Property for the ONMNT

Let $Y_o(k)$, $X_o(k)$, and $H_o(n)$ be the ONMNT of $y(n)$, $x(n)$ and $h(n)$ respectively. The relationship between these sequences can be written as:

$$\begin{aligned} Y_o(k) &= X_o(k)\ \Pi_1\ H_o(k) \\ &= X_o(k) \otimes H_o^{\rm ev}(k) + X_o(N-k-1) \otimes H_o^{\rm od}(k) \end{aligned} \tag{43}$$

where $\otimes$ is point-by-point multiplication, $H_o^{\rm ev}(k)$ and $H_o^{\rm od}(k)$, stand for even and odd parts of $H_o(k)$ respectively and are given by

$$\begin{aligned} H_o^{\rm ev}(k) &= \left\langle \frac{(H_o(k) + H_o(N-k-1))}{2} \right\rangle_{Mp} \\ H_o^{\rm od}(k) &= \left\langle \frac{(H_o(k) - H_o(N-k-1))}{2} \right\rangle_{Mp}. \end{aligned} \tag{44}$$

The process for calculating the SCC using the ONMNT is shown in Fig. 4, where the operator $\prod_1$ is as defined by (43).

It is necessary that the modulus Mp must be chosen so that the convolution result does not exceed Mp. One suggested upper bound is given by [3], [5]

$$|y(n)| \le |x(n)|_{max} \sum_{n=0}^{N-1} |h(n)| \le \frac{Mp}{2}. \tag{45}$$

The proof of the SCC property for the ONMNT as given by (43), is carried out by assuming that the sequences $x(n)$ and $h(n)$ are anti-periodic [17] and $y(n)$ is the skew-cyclic convolution. Therefore, (41) can be written as

$$y(n) = \sum_{\ell=0}^{N-1} h(\ell)\, x(n-\ell). \tag{46}$$

The ONMNT of (46) is

$$\begin{aligned} Y_o(k) &= \left\langle \sum_{n=0}^{N-1} y(n)\ \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp} \\ &= \left\langle \sum_{n=0}^{N-1} \sum_{\ell=0}^{N-1} h(\ell)\ x(n-\ell) \beta\left(\frac{n(2k+1)}{2}\right) \right\rangle_{Mp}. \end{aligned} \tag{47}$$

Changing the variable $m = n - l$ in (47), we get

$$Y_o(k) = \left\langle \sum_{m=-\ell}^{N-1-\ell} \sum_{\ell=0}^{N-1} h(\ell)\ x(m)\ \beta\left(\frac{(m+\ell)(2k+1)}{2}\right) \right\rangle_{Mp}. \tag{48}$$

Applying (17), we get

$$Y_o(k) = \left\langle \sum_{m=-\ell}^{N-1-\ell} x(m)\beta\left(\frac{m(2k+1)}{2}\right)\sum_{\ell=0}^{N-1} h(\ell) \times \beta_1\left(\frac{\ell(2k+1)}{2}\right) + \sum_{m=-\ell}^{N-1-\ell} x(m)\beta\left(-\frac{m(2k+1)}{2}\right) \times \sum_{\ell=0}^{N-1} h(\ell)\,\beta_2\left(\frac{\ell(2k+1)}{2}\right)\right\rangle_{Mp}. \quad (49)$$

Equation (49) can be written as:

$$Y_o(k) = \left\langle X_o(k)\sum_{\ell=0}^{N-1} h(\ell)\,\beta_1\left(\frac{\ell(2k+1)}{2}\right) + X_o(N-k-1)\sum_{\ell=0}^{N-1} h(\ell)\beta_2\left(\frac{\ell(2k+1)}{2}\right)\right\rangle_{Mp}. \quad (50)$$

Using the fact that $\beta(\pm n) = \beta_1(n) \pm \beta_2(n)$, we obtain the following relations:

$$\beta_1(n) = \frac{1}{2}\left[\beta(n) + \beta(-n)\right]$$
$$\beta_2(n) = \frac{1}{2}\left[\beta(n) - \beta(-n)\right]. \quad (51)$$

Substituting (51) into (50), we get

$$Y_o(k) = \frac{1}{2}\left\langle X_o(k)\left[\sum_{\ell=0}^{N-1} h(\ell)\,\beta\left(\frac{(2k+1)\ell}{2}\right) + \sum_{\ell=0}^{N-1} h(\ell)\,\beta\left(-\frac{(2k+1)\ell}{2}\right)\right] + X_o(N-k-1) \times \left[\sum_{\ell=0}^{N-1} h(\ell)\,\beta\left(\frac{(2k+1)\ell}{2}\right) - \sum_{\ell=0}^{N-1} h(\ell)\,\beta\left(-\frac{(2k+1)\ell}{2}\right)\right]\right\rangle_{Mp}. \quad (52)$$

From the definition of the ONMNT given in (6), we have

$$H_o(k) = \left\langle \sum_{\ell=0}^{N-1} h(\ell)\,\beta(\frac{(2k+1)\ell}{2})\right\rangle_{Mp}$$
$$H_o(N-k-1) = \left\langle \sum_{\ell=0}^{N-1} h(\ell)\,\beta\left(-\frac{(2k+1)\ell}{2}\right)\right\rangle_{Mp}. \quad (53)$$

Substituting (53) into (52) yields

$$Y_o(k) = \frac{1}{2}\left\langle X_o(k)\left[H_o(k) + H_o(N-k-1)\right] + X_o(N-k-1)\left[H_o(k) - H_o(N-k-1)\right]\right\rangle_{Mp}. \quad (54)$$

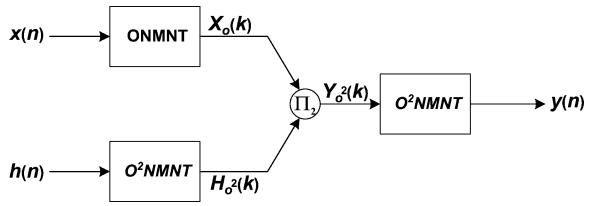

Fig. 5. Fast SCC using the $\mathbf{O^2NMNT}$ based on (56).

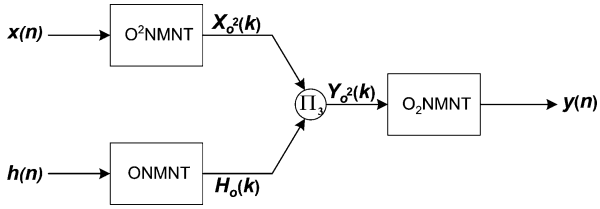

Fig. 6. Fast SCC using the $\mathbf{O^2NMNT}$ based on (57).

Using the definition of $H_o^{\rm ev}(k)$ and $H_o^{\rm od}(k)$ given in (44), we get the proof of the SCC property for the ONMNT given in (43), as follows:

$$Y_o(k) = \left\langle X_o(k)\left[\frac{1}{2}\left(H_o(k) + H_o(N-k-1)\right)\right] + X_o(N-k-1)\left[\frac{1}{2}\left(H_o(k) - H_o(N-k-1)\right)\right]\right\rangle_{Mp}$$
$$= X_o(k) \otimes H_o^{\rm ev}(k) + X_o(N-k-1) \otimes H_o^{\rm od}(k) \quad (55)$$

B. SCC Property for the $\mathrm{O^2NMNT}$

Let $Y_{o2}(k)$, $X_{o2}(k)$ and $H_{o2}(k)$ be the $\mathrm{O^2NMNT}$ of $y(n)$, $x(n)$ and $h(n)$ respectively. In this case, the relationship between these sequences becomes

$$Y_{o2}(k) = X_o(k)\,\Pi_2\,H_{o2}(k)$$
$$= X_o(N-k-1) \otimes H_{o2}^{\rm ev}(k) + X_o(k) \otimes H_{o2}^{\rm od}(k). \quad (56)$$

or

$$Y_{o2}(k) = X_{o2}(k)\,\Pi_3\,H_o(k)$$
$$= X_{o2}(k) \otimes H_o^{\rm ev}(k) - X_{o2}(N-k-1) \otimes H_o^{\rm od}(k) \quad (57)$$

where $H_o^{\rm ev}(k)$ and $H_o^{\rm od}(k)$ in (56) stand for even and odd parts of $H_{o2}(k)$, respectively, and are given by

$$H_{o2}^{\rm ev}(k) = \left\langle \frac{(H_{o2}(k) + H_{o2}(N-k-1))}{2}\right\rangle_{Mp}$$
$$H_{o2}^{\rm od}(k) = \left\langle \frac{(H_{o2}(k) - H_{o2}(N-k-1))}{2}\right\rangle_{Mp} \quad (58)$$

and $H_o^{\rm ev}(k)$ and $H_o^{\rm od}(k)$ in (57) are as given by (44).

The proof of the SCC property for the $\mathrm{O^2NMNT}$ can be derived following the same procedure shown in (47)–(55) for the case of the ONMNT.

Figs. 5 and 6 show the process for calculating the SCC using the $\mathrm{O^2NMNT}$, where the operators $\prod_2$ and $\prod_3$ are as defined by (56) and (57), respectively.

V. Example of the Calculation of Convolutions Using the ONMNT

To prove the validity of the new transforms, including their skew cyclic convolution property and the developed algorithms, an example is given in this section. For the sake of demonstration and without loss of generality, it is required to use the ONMNT to calculate the cyclic convolutions (SCC and CC) and the linear convolution for the following two 16-point integer sequences generated randomly:

$$x(n) = [7,3,12,7,1,5,12,1,9,11,10,1,14,6,8,1]$$

and

$$h(n) = [7,2,4,1,5,9,7,11,8,1,3,13,13,14,4,14].$$

Choosing $Mp = 2^{13} - 1$, from (3), the initial values of α_1 and α_2 can be calculated as $\alpha_1 = \langle 2^{2^{11}} \rangle_{Mp} = 128$ and $\alpha_2 = \langle -3^{2^{11}} \rangle_{Mp} = 181$ are used in order to obtain the new values for the transform length $N = 16$ using (14).

Since $d = 512$, applying (14), we get

$$\beta_1 = \frac{1}{2}\langle \mathrm{Re}(128 + j181)^{512} \rangle_{Mp} = -3077$$

and

$$\beta_2 = \frac{1}{2}\langle \mathrm{Im}(128 + j181)^{512} \rangle_{Mp} = 647.$$

Using these parameters, $x(n)$ and $h(n)$ are transformed into the ONMNT domain, producing the following sequences:

$$X(k) = \begin{bmatrix} 315,7212,7187,1940,2291,5668,1949,7629, \\ 4752,4645,6175,3694,1848,6600,40,3695 \end{bmatrix}$$

and

$$H(k) = \begin{bmatrix} 1317,7814,5397,4796,973,3345,6974,7654, \\ 7175,7861,3476,3535,257,3053,7187,3017 \end{bmatrix}.$$

First, the skew-cyclic convolution of $x(n)$ and $h(n)$ can be calculated using the skew convolution property of the ONMNT as described in Section IV. The skew convolution output will be

$$y_{\mathrm{scc}}(n) = \begin{bmatrix} -636, -781, -545, -564, -559, -576, -229, \\ -246, -134, -168,167,62,395,598,755,749 \end{bmatrix}.$$

Second, the cyclic convolution of $x(n)$ and $h(n)$ can be calculated either by using the convolution property of the NMNT [5] or by converting the SCC to CC as described in [16], producing

$$y_{\mathrm{cc}}(n) = \begin{bmatrix} 734,851,781,748,773,886,711,832, \\ 852,882,709,752,655,830,783,749 \end{bmatrix}.$$

Finally, the linear convolution of the sequences $x(n)$ and $h(n)$ is calculated from $y_{\mathrm{scc}}(n)$ and $y_{\mathrm{cc}}(n)$ as given in (42), hence the desired convolution result is given as

$$y(n) = \begin{bmatrix} 49,35,118,92,107,155,241,293,359,357,438,407, \\ 525,714,769,749,685,816,663,656,666,731,470, \\ 539,493,525,271,345,130,116,14 \end{bmatrix}.$$

Therefore, it is clear that a $2N-1$ point linear convolution is calculated by circularly convolving two N length sequences, using an ONMNT of length N.

VI. Conclusion

This paper has presented two new NTTs using the Mersenne numbers where arithmetic operations and residue reduction are known to be simpler than other moduli (equivalent to 1's complement). These new types of generalized NMNT are named odd and odd-squared NMNTs that can be added to the NTT family. These transforms have the skew cyclic convolution property, long transform lengths powers of two, and are amenable to fast algorithms. Hence, they are suitable for fast calculation of error-free convolutions/correlations for signal and image processing applications. Furthermore, the presented transforms can be used to calculate the original NMNT with reduced arithmetic complexity. Future work will focus on the applications of the GNMNTs beyond the calculation of convolutions, the development of other fast algorithms, implementation of new encryption systems, and extending the development to multidimensional GNMNTs for image and three-dimensional applications.

References

[1] J. H. McClellen and C. M. Rader, *Number Theory in Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall PTR, 1979.

[2] A. V. Oppenheim and C. J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," *Proc. IEEE*, vol. 60, pp. 957–976, 1972.

[3] R. Agarwal and C. Burrus, "Fast convolution using Fermat number transforms with applications to digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, no. 2, pp. 87–97, 1974.

[4] C. M. Rader, "Discrete convolutions via Mersenne transforms," *IEEE Trans. Comput.*, vol. C-21, pp. 1269–1273, 1972.

[5] S. Boussakta and A. G. J. Holt, "New transform using the Mersenne numbers," *Proc. Inst. Electr. Eng.—Vision, Image, Signal Process.*, vol. 142, pp. 381–388, 1995.

[6] S. Boussakta and A. G. J. Holt, "Filtering employing a new transform," *Proc. OCEANS*, vol. 1, pp. I/547–I/553, 1994.

[7] S. Boussakta and A. G. J. Holt, "Number theoretic transforms and their applications in image processing," *Adv. Imag. Electron Phys.*, vol. 111, pp. 1–90, 1999.

[8] D. Kehil and Y. Ferdi, "Signal encryption using new Mersenne number transform," in *Proc. 7th Int. Symp. Commun. Syst. Netw. Digit. Signal Process. (CSNDSP)*, 2010, pp. 736–740.

[9] G. Bongiovanni, P. Corsini, and G. Frosini, "One-dimensional and two-dimensional generalised discrete Fourier transforms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 1, pp. 97–99, 1976.

[10] V. Britanak and K. R. Rao, "The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transforms computation," *Signal Process.*, vol. 79, pp. 135–150, 1999.

[11] O. K. Ersoy, *Fourier-Related Transforms, Fast Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall PTR, 1997.

[12] Z. Wang, G. A. Jullien, and W. C. Miller, "The generalized discrete W transform and its application to interpolation," *Signal Process.*, vol. 36, pp. 99–109, 1994.

[13] J. N. Madihally, "Linear convolution using skew-cyclic convolutions," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 173–176, 2007.

[14] W. Li, "A new algorithm to compute the DCT and its inverse," *IEEE Trans. Signal Process.*, vol. 39, no. 6, pp. 1305–1313, 1991.

[15] D. Slawecki and W. Li, "DCT/IDCT processor design for high data rate image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 135–146, 1992.

[16] B. Arambepola and P. Rayner, "Discrete transforms over polynomial rings with applications in computing multidimensional convolutions," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 407–414, 1980.

[17] S. A. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1038–1051, 1994.