IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Sparse Matrix-based Low-complexity, Recursive, Radix-2 Algorithms for Discrete Sine Transforms

SIRANI M. PERERA[1], LEVI E. LINGSCH[2]
[1]Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA (e-mail: pereras2@erau.edu)
[2]Department of Aerospace Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA (e-mail: lingschl@my.erau.edu)

Corresponding author: Sirani M. Perera (e-mail: pereras2@erau.edu).

This work was supported by the Faculty Innovative Research in Science and Technology Grant 13238 at Embry-Riddle Aeronautical University, USA

ABSTRACT This paper presents factorizations of each discrete sine transform (DST) matrices of types I, II, III, and IV into a product of sparse, diagonal, bidiagonal, and scaled orthogonal matrices. Based on the proposed matrix factorization formulas, reduced multiplication complexity, recursive, and radix-2 DST-I/II/III/IV algorithms are presented. We will present the lowest multiplication complexity DST-IV algorithm in the literature. The paper fills a gap in the self-recursive, exact, and radix-2 DST-I/II/III algorithms executed via diagonal, bidiagonal, scaled orthogonal, and simple matrix factors for any input $n = 2^t$ $(t \geq 1)$. The paper establishes a novel relationship between DST-II and DST-IV matrices using diagonal and bidiagonal matrices. Similarly, a novel relationship between DST-I and DST-III matrices is proposed using sparse and diagonal matrices. These interweaving relationships among DST matrices enable us to bridge the existing factorizations of the DST matrices with the proposed factorization formulas. We present signal flow graphs to provide a layout for realizing the proposed algorithms in DST-based integrated circuit designs. Additionally, we describe an implementation of algorithms based on the proposed DST-II and DST-III factorizations within a double random phase encoding (DRPE) image encryption scheme.

INDEX TERMS Discrete sine transforms, self/completely recursive and radix-2 algorithms, complexity and performance of algorithms, sparse and orthogonal matrices, signal flow graphs

I. INTRODUCTION

Discrete sine transforms are real-valued transform matrices and a subclass of the discrete Fourier transform (DFT) matrices, extracting the imaginary part of the complex trigonometric form. The DFT matrix can be described as a Vandermonde matrix having nodes as the primitive n^{th} roots of unity, where n is a positive integer. The fast Fourier transform (FFT) is an algorithm that can be used to compute the DFT and its inverse efficiently.

For a given vector $x = [x_0, x_1, \cdots, x_{n-1}]^T \in \mathbb{R}^n$, its DST can be expressed as $y = Sx$, where S is the DST matrix. The DST matrix has four main variants, i.e., I-IV as defined below

$$S_{n-1}^{I} = \sqrt{\frac{2}{n}} \left[\sin \frac{(j+1)(k+1)\pi}{n} \right]_{j,\,k=0}^{n-2},$$

$$S_{n}^{II} = \sqrt{\frac{2}{n}} \left[\epsilon_n(j+1) \sin \frac{(j+1)(2k+1)\pi}{2n} \right]_{j,\,k=0}^{n-1},$$

$$S_{n}^{III} = \sqrt{\frac{2}{n}} \left[\epsilon_n(k+1) \sin \frac{(2j+1)(k+1)\pi}{2n} \right]_{j,\,k=0}^{n-1},$$

$$S_{n}^{IV} = \sqrt{\frac{2}{n}} \left[\sin \frac{(2j+1)(2k+1)\pi}{4n} \right]_{j,\,k=0}^{n-1},$$

where $\epsilon_n(0) = \epsilon_n(n) = \frac{1}{\sqrt{2}}$, $\epsilon_n(j) = 1$ for $j \in \{1, 2, \cdots, n-1\}$ and $n \geq 2$ is an even integer. Here, we use the superscript to denote the type of a given DST matrix and the subscript to denote the order of a DST matrix. Among the DST I-IV matrices, S_{n-1}^{I} and S_{n}^{IV} were introduced in [1], [2], and S_{n}^{II} and its inverse S_{n}^{III} were introduced in [3] to digital signal processing. The complete

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

set of even discrete cosine transform (DCT) and DST variants was originally presented in [4]. DST-II is a complementary or alternative transform to DCT-II, which is used in transform coding. Like the DFT and DCT, these DST matrices hold linearity, convolution-multiplication, orthogonality, and shift properties. Moreover, the DSTs of types II and III are related via $S_n^{III} = \left[S_n^{II}\right]^T$. In this paper, we will establish relationships between S_n^{II} and S_n^{IV} as well as S_{n-1}^{I} and S_n^{III}. The matrix factorization for DST-I in [5] used the results from [6] to decompose DST-I into the DCT and DST. Though one can find orthogonal matrix factorizations for the DCT and DST in [7], the resulting algorithms in [7] are not completely recursive and hence do not lead to simple recursive algorithms. An alternative factorization for DCT I in [8] and DST-I in [9] can be seen in [10], [11], but the factorizations in the latter papers do not solely depend on DCT I-IV or DST I-IV. Moreover, [11] has used the same factorization for DST-II and IV as in [7]. The authors in [9], [12], however, produce radix-2, completely recursive, and stable DST I-IV algorithms. Furthermore, [12] established a connection between algebraic operations used in the sparse and scaled orthogonal factorization of DST I-IV matrices and signal flow graphs. We refer to many more historical remarks on DST factorizations and the corresponding signal flow graphs in [12]. We note here that the DCT algorithms in [8], [9], [13] and the relationship between cosine and sine transform matrices in [10] can be used to obtain the DST algorithms in [12]. On the other hand, one can use an elegant factorization for Chebyshev-like Vandermonde matrices to factor the DCT and DST matrices as described in [14]. However, there are no simplest sparse factors, no explicit algorithms, and no complexity analysis to compute DCT and DST matrices by a vector in [14]. In this paper, we obtain simple factors, i.e, diagonal, bidiagonal, sparse, and scaled orthogonal factors, to compute DST matrices by a vector. Additionally, we present self-recursive/completely-recursive and radix-2 DST algorithms with the lowest multiplication complexity in the literature.

In recent years, developments in calculating the DST have widely broadened the algorithm's applications. Methods based upon the recursive nature of Chebyshev-polynomials have led to a drastic decrease in hardware complexity for applications which only require the transmission of a few key transform coefficients, creating a DST algorithm that is suitable for very large scale integrated circuits (VLSI) [15]–[17]. Similarly, a first-order moment-based approach has been developed which is able to calculate the 1-dimensional DST-II without the need for multiplications by using special accumulator arrays and without having explicit factors for the DST-II to decrease computation time for certain applications [18]. For complex inputs, sparse factorization-based methods offer an extremely reduced flop count when compared to the brute-force method. The DST-II algorithms proposed in [19] take this a step further by developing a radix-2 based even-odd output vector at each stage, but without having explicit factorization for the DST-II matrix for any n.

The development of the convolution-multiplication property for the families of DSTs and DCTs led way for the DST to become an alternative to the DFT in many filtering applications [20]. Similarly, the duality theorem proposed in [21], [22] now allows for the avoidance of calculating the inverse DST (IDST) in applications where similar signals are commonly encountered. Recently, the DST has expanded to applications including underwater signal transmission [23], [24], analysis and noise estimation of speech [25], finite impulse response (FIR) frequency filtering [26], and feature extraction of modulated signals in orthogonal frequency division multiplexing (OFDM) systems [27]. Similarly, the noise filtering capabilities have led to a new DST-based approach to measuring the volatility of stock prices [28], [29]. In hardware applications, the DST provides advantages over DCT-based approaches for statistical processes like the first-order Markov sequences with specific boundary conditions, and it achieves a lower bit error rate (BER) than the DCT for signals with low correlation [30]. It has also been shown that the DST may be implemented in the design of fractional order differentiators [31]. With the ability to develop DST algorithms with regularity, modularity, pipelining capability, and local connectivity, the DST will continue to expand its role in VLSI implementation [15]–[19], [26], [30], [31].

Although the DCT has traditionally played the pivotal role of frequency decomposition for data compression in methods like JPEG and MPEG, recent works have suggested that the DST may be applied to develop new methods of compression with increased performance for image and audio analysis as well as high-efficiency video encoding (HEVC) [22], [32]–[35]. The DST has proven to be an effective tool in the medical field as well, combining multiple images of different modes into a singular, highly detailed image for more accurate diagnoses [36], [37]. Expansion upon the fractional DST has also broadened the transform's role in the field of cyber security, particularly for applications in image encryption [38]–[40]. It was shown in [38] that the DST increases the entropy of encryption schemes that rely on processing within transform domains. This paper explores this concept further in Section VIII.

We have observed that the lowest multiplicative complexity, radix-2, and recursive DST algorithms having diagonal, bidiagonal, sparse, and scaled orthogonal matrices are missing elements in the literature. Thus, in this paper, we follow a similar approach in deriving DCT II/III algorithms in [41] and the DCT I/IV algorithms in [42] to obtain novel lowest multiplication complexity, recursive, and radix-2 DST algorithms execute via diagonal, bidiagonal, sparse, and scaled orthogonal matrices. We also obtain novel relationships between DST-II and DST-IV matrices, and DST-I and DST-III matrices in terms of diagonal, bidiagonal, and sparse

matrices.

We first introduce frequently used notation in Section II. In Section III, we obtain simple, diagonal, bidiagonal, sparse, and scaled orthogonal factors for the DST matrices. In Section IV, we derive relationships between DST-I and DST-III matrices, and also between DST-II and DST-IV matrices, using bidiagonal, diagonal, and sparse matrices. Next, in Section V, we state recursive and radix-2 DST algorithms to compute the DST matrices by a vector based on the factorization of the DST matrices described in Section III. In Section VI, we establish arithmetic complexities for computing the proposed DST algorithms. Within the same section, we compare the arithmetic complexities in computing the proposed DST algorithms with some known DST algorithms. In Section VII, we show the connection between the algebraic operations used to compute the proposed DST algorithms and the signal flow graph building blocks to provide an architecture for the DST based integrated circuit design. Finally, we describe in Section VIII an application of the proposed DST algorithms within a DRPE image encryption scheme.

II. FREQUENTLY USED NOTATIONS

First, we provide notations in [14] to distinguish and show the novelty of the proposed DST matrix factorization in comparison to the results in [14]. Next, we will introduce additional notations that we use throughout the paper.

A. FACTORIZATION FOR DST-I AND DST-II IN [14]

Here, we provide formulas in computing DST-I/II matrices in [14]. The relationship $F_n = W_Q \cdot V_Q$, where F_n is the DCT or DST matrix of types I-IV, W_Q is the weight matrix, and V_Q is the Chebyshev-like polynomial Vandermode matrix for the DCT or DST matrices, was first introduced in [43]. Later in [14], the polynomial Vandermonde matrix $V_Q = [Q_k(x_j)]_{j,k=1,0}^{n,n-1}$ was further split into self contained factors to obtain factorization for the DCT and DST matrices.

The weight matrix for DST-I in [14], [43] for odd N is given by

$$W_{Q_{S^I}} = \sqrt{\tfrac{2}{N+1}} diag\left[\sin\left(\tfrac{\pi}{N+1}\right), \sin\left(\tfrac{2\pi}{N+1}\right), \cdots, \sin\left(\tfrac{N\pi}{N+1}\right)\right]. \quad (1)$$

The weight matrix for DST-II in [14], [43] for even n is given by

$$W_{Q_{S^{II}}} = \sqrt{\tfrac{2}{n}} diag\left[\sin\left(\tfrac{\pi}{2n}\right), \sin\left(\tfrac{\pi}{n}\right), \cdots, \sin\left(\tfrac{(n-1)\pi}{2n}\right), \tfrac{1}{\sqrt{2}}\sin\left(\tfrac{\pi}{2}\right)\right]. \quad (2)$$

The self-contained factorization for the Chebyshev-like polynomial Vandermode matrix for DST-I (for odd N) and DST-II (for even n) was derived in [14] and stated as follows

$$V_{Q_{S^{I/II}}} = \begin{bmatrix} odd-even \\ permutation \end{bmatrix} \begin{bmatrix} V_{even} & \\ & V_{odd} \end{bmatrix} H_{S^{I/II}}, \quad (3)$$

where

$$H_{S^I} = \begin{bmatrix} \check{I}_{\frac{N-1}{2},\frac{N+1}{2}} & -I_{\frac{N-1}{2},\frac{N-1}{2}} \\ I_{\frac{N+1}{2},\frac{N+1}{2}} & -\widehat{I}_{\frac{N+1}{2},\frac{N-1}{2}} \end{bmatrix}, \quad (4)$$

$\widehat{I}_{\frac{N+1}{2},\frac{N-1}{2}} = \begin{bmatrix} I_{\frac{N-1}{2},\frac{N-1}{2}} \\ 0 \end{bmatrix}$, and $\check{I}_{\frac{N-1}{2},\frac{N+1}{2}} = \begin{bmatrix} I_{\frac{N-1}{2},\frac{N-1}{2}} & 0 \end{bmatrix}$, and

$$H_{S^{II}} = \begin{bmatrix} I_{\frac{n}{2},\frac{n}{2}} & -\widetilde{I}_{\frac{n}{2},\frac{n}{2}} \\ I_{\frac{n}{2},\frac{n}{2}} & \widetilde{I}_{\frac{n}{2},\frac{n}{2}} \end{bmatrix}, \quad (5)$$

$\widetilde{I}$ is the anti-diagonal matrix. Moreover, the Chebyshev-like polynomial Vandermonde matrix $V_{even} = [Q_{2j}(d_{2k})]_{j,k=1}^{\frac{N-1}{2}}$ is defined using $Q_k(d) = U_k(d)$ for DST-I and $V_{even} = [Q_{2j}(d_{2k})]_{j,k=1}^{\frac{n}{2}}$ is defined using $Q_k(d) = U_k(d) - U_{k-1}(d)$ for DST-II with respect to the classical Chebyshev polynomials $U_k(d) = \frac{\sin((k+1)\arccos(d))}{\sin(\arccos(d))}$ for $k = 0, 1, \cdots, N-1$ or $n-1$, respectively for DST-I and DST-II.

The factorization for V_{odd} was derived for DST-I (for odd N) and has the form

$$V_{odd} = G_{S^I} \begin{bmatrix} V_{even} & 0 \\ 0 & 1 \end{bmatrix} \widehat{G}_{S^I}, \quad (6)$$

where

$$G_{S^I} = \begin{bmatrix} \frac{\sin(\arccos d_2)}{\sin(\arccos d_1)} & & & \frac{(-1)^0}{\sin(\arccos d_1)} \\ \frac{\sin(\arccos d_2)}{\sin(\arccos d_3)} & \frac{\sin(\arccos d_4)}{\sin(\arccos d_3)} & & \frac{(-1)^1}{\sin(\arccos d_3)} \\ & & \ddots & \\ & \ddots & \frac{\sin(\arccos d_{N-1})}{\sin(\arccos d_{N-2})} & \frac{(-1)^{\frac{N-3}{2}}}{\sin(\arccos d_{N-2})} \\ & & \frac{\sin(\arccos d_{N-1})}{\sin(\arccos d_N)} & \frac{(-1)^{\frac{N-1}{2}}}{\sin(\arccos d_N)} \end{bmatrix}, \quad (7)$$

and

$$\widehat{G}_{S^I} = \frac{1}{2} diag\left[\frac{1}{T_1(d_1)}, \frac{1}{T_2(d_1)}, \cdots, \frac{1}{T_{\frac{N-1}{2}}(d_1)}, 1\right], \quad (8)$$

with respect to the nodes

$$\{d_k\}_{k=1}^N = \left\{\cos\left(\frac{k\pi}{N+1}\right)\right\}_{k=1}^N, \quad (9)$$

and the classical Chebyshev polynomials $T_k(d) = \cos(k \arccos d)$.

The factorization for V_{odd} was derived for DST-II (for even n) and has the form

$$V_{odd} = G_{S^{II}} V_{even} \widehat{G}_{S^{II}}, \quad (10)$$

where

$$G_{S^{II}} = \begin{bmatrix} \frac{\sin\left(\frac{\arccos d_2}{2}\right)}{\sin\left(\frac{\arccos d_1}{2}\right)} & & & \\ \frac{\sin\left(\frac{\arccos d_2}{2}\right)}{\sin\left(\frac{\arccos d_3}{2}\right)} & \frac{\sin\left(\frac{\arccos d_4}{2}\right)}{\sin\left(\frac{\arccos d_3}{2}\right)} & & \\ & & \ddots & \\ & \ddots & & \\ & & \frac{\sin\left(\frac{\arccos d_{n-2}}{2}\right)}{\sin\left(\frac{\arccos d_{n-1}}{2}\right)} & \frac{\sin\left(\frac{\arccos d_n}{2}\right)}{\sin\left(\frac{\arccos d_{n-1}}{2}\right)} \end{bmatrix}, \quad (11)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

and

$$\widehat{G}_{S^{II}} = \frac{1}{2} diag \left[\frac{1}{\cos\left(\frac{\pi}{2n}\right)}, \frac{1}{\cos\left(\frac{3\pi}{2n}\right)}, \cdots, \frac{1}{\cos\left(\frac{(n-1)\pi}{2n}\right)} \right], \tag{12}$$

with respect to the nodes

$$\{d_k\}_{k=1}^{n} = \left\{ \cos\left(\frac{k\pi}{n}\right) \right\}_{k=1}^{n}. \tag{13}$$

By using the above factorization formulas for the Chebyshev-like polynomial Vandermonde matrices and the relationship $F_n = W_Q \cdot V_Q$ in [43], the following results were obtained in [14] to factor DST-I (for odd N) and DST-II (for even n) matrices

$$S_N^I = W_{Q_{S^I}} \begin{bmatrix} odd-even \\ permutation \end{bmatrix}^{-1} \begin{bmatrix} W_{\frac{N-1}{2}}^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} S_{\frac{N-1}{2}}^I & \\ & G_{S^I}\, W_{\frac{N-1}{2}}^{-1}\, S_{\frac{N-1}{2}}^I\, \widehat{G}_{S^I} \end{bmatrix} H_{S^I}, \tag{14}$$

and

$$S_n^{II} = W_{Q_{S^{II}}} \begin{bmatrix} odd-even \\ permutation \end{bmatrix}^{-1} \begin{bmatrix} W_{\frac{n}{2}}^{-1} & \\ & I \end{bmatrix} \begin{bmatrix} S_{\frac{n}{2}}^{II} & \\ & G_{S^{II}}\, W_{\frac{n}{2}}^{-1}\, S_{\frac{n}{2}}^{II}\, \widehat{G}_{S^{II}} \end{bmatrix} H_{S^{II}}. \tag{15}$$

The factorization formulas (14) and (15) were obtained by using the factorization formula (3) with the replacement of $V_{Q_{S^{I/II}}}$ by $W_{Q_{S^{I/II}}}^{-1}\, S^{I/II}$ (from [43]), $V_{\frac{N-1}{2}}$ by $W_{\frac{N-1}{2}}^{-1} S_{\frac{N-1}{2}}^I$ and $V_{\frac{n}{2}}$ by $W_{\frac{n}{2}}^{-1} S_{\frac{n}{2}}^I$ (from [43]), respectively. Although it says in [14] that the first matrix in the factorization (3) is the odd-even permutation matrix, it must actually be the transpose (inverse) of the odd-even permutation matrix. We also note here that the factorization formulas to compute DST-III/IV matrices in [14] are not included because the proposed DST-III/IV matrix factorization formulas are completely different from the factorization formulas in [14].

B. OTHER NOTATION

We state here diagonal, bidiagonal, sparse, and orthogonal matrices which are frequently used in this paper. For a given vector $x \in \mathbb{R}^n$ and an integer $n \geq 3$, let us introduce an even-odd permutation matrix P_n by

$$P_n x = \begin{cases} [x_0, x_2, \cdots, x_{n-2}, x_1, x_3, \cdots, x_{n-1}]^T, & \text{if } n \text{ is even} \\ [x_0, x_2, \cdots, x_{n-1}, x_1, x_3, \cdots, x_{n-2}]^T, & \text{if } n \text{ is odd,} \end{cases}$$

an odd-even permutation matrix $\tilde{P}_n$ by

$$\tilde{P}_n x = \begin{cases} [x_1, x_3, \cdots, x_{n-1}, x_0, x_2, \cdots, x_{n-2}]^T, & \text{if } n \text{ is even} \\ [x_1, x_3, \cdots, x_{n-2}, x_0, x_2, \cdots, x_{n-1}]^T, & \text{if } n \text{ is odd.} \end{cases}$$

For even n s.t. $n \geq 4$, we introduce a sparse matrix

$$\bar{B}_{\frac{n}{2}} = \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & -1 \\ & \ddots & \ddots & & 1 \\ & & 1 & 1 & \vdots \\ & & & 1 & -1 \end{bmatrix}, \tag{16}$$

a bidiagonal matrix

$$B_{\frac{n}{2}} = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & 1 & \\ & & & 1 & \sqrt{2} \end{bmatrix}, \tag{17}$$

diagonal matrices

$$\bar{W}_{\frac{n}{2}} = \begin{bmatrix} diag\left[\frac{\csc(\frac{(n-2k)\pi}{2n})}{2} \right]_{k=1}^{\frac{n}{2}-1} & 0 \\ 0 & 1 \end{bmatrix} \tag{18}$$

and

$$W_{\frac{n}{2}} = diag\left[\frac{\csc(\frac{(n-2k+1)\pi}{2n})}{2} \right]_{k=1}^{\frac{n}{2}}, \tag{19}$$

sparse and orthogonal matrices

$$\tilde{H}_n = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{\frac{n}{2}} & -\widetilde{I}_{\frac{n}{2}} \\ I_{\frac{n}{2}} & \widetilde{I}_{\frac{n}{2}} \end{bmatrix}, \tag{20}$$

$$H_n = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{\frac{n}{2}} & \widetilde{I}_{\frac{n}{2}} \\ I_{\frac{n}{2}} & -\widetilde{I}_{\frac{n}{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} H_n^s,$$

$$\bar{H}_{n-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{\frac{n}{2}-1} & & -\widetilde{I}_{\frac{n}{2}-1} \\ I_{\frac{n}{2}-1} & & \widetilde{I}_{\frac{n}{2}-1} \\ & \sqrt{2} & \end{bmatrix}, \tag{21}$$

$$\widehat{H}_{n-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} I_{\frac{n}{2}-1} & & \widetilde{I}_{\frac{n}{2}-1} \\ & \sqrt{2} & \\ I_{\frac{n}{2}-1} & & -\widetilde{I}_{\frac{n}{2}-1} \end{bmatrix} = \frac{1}{\sqrt{2}} H_{n-1}^{s_1}, \tag{22}$$

$$V_n = \begin{bmatrix} 1 & & \\ & \frac{1}{\sqrt{2}} \begin{bmatrix} I_{\frac{n}{2}-1} & -I_{\frac{n}{2}-1} \\ -I_{\frac{n}{2}-1} & -I_{\frac{n}{2}-1} \end{bmatrix} & \\ & & -1 \end{bmatrix} \begin{bmatrix} \tilde{I}_{\frac{n}{2}} & \\ & J_{\frac{n}{2}} \end{bmatrix}, \tag{23}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

where $J_n = \text{diag}\left[(-1)^k\right]_{k=0}^{n}$,
a rotational-reflection matrix

$$Q_n = \begin{bmatrix} J_{\frac{n}{2}} & \\ & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} \text{diag}\, S_{\frac{n}{2}} & \left(\text{diag}\, C_{\frac{n}{2}}\right) \tilde{I}_{\frac{n}{2}} \\ -\tilde{I}_{\frac{n}{2}} \left(\text{diag}\, C_{\frac{n}{2}}\right) & \text{diag}\left(\tilde{I}_{\frac{n}{2}} S_{\frac{n}{2}}\right) \end{bmatrix}$$

$$= \begin{bmatrix} \sin\frac{\pi}{4n} & & & & & & \cos\frac{\pi}{4n} \\ & -\sin\frac{3\pi}{4n} & & & & -\cos\frac{3\pi}{4n} & \\ & & \ddots & & \iddots & & \\ & & & -\sin\frac{(n-1)\pi}{4n} & \cos\frac{(n-1)\pi}{4n} & & \\ & & & -\cos\frac{(n-1)\pi}{4n} & \sin\frac{(n-1)\pi}{4n} & & \\ & & \iddots & & \ddots & & \\ & -\cos\frac{3\pi}{4n} & & & & \sin\frac{3\pi}{4n} & \\ -\cos\frac{\pi}{4n} & & & & & & \sin\frac{\pi}{4n} \end{bmatrix}, \tag{24}$$

where

$$C_{\frac{n}{2}} = \left[\cos\frac{(2k+1)\pi}{4n}\right]_{k=0}^{\frac{n}{2}-1},$$

$$S_{\frac{n}{2}} = \left[\sin\frac{(2k+1)\pi}{4n}\right]_{k=0}^{\frac{n}{2}-1},$$

and a matrix with three liftings (we will show later that the following sparse and orthogonal matrix i.e. $\tilde{Q}_n$ is the same as the matrix Q_n with three lifting steps performed)

$$\tilde{Q}_n = \begin{bmatrix} J_{\frac{n}{2}} & J_{\frac{n}{2}} \left(\text{diag}\, T_{\frac{n}{2}}\right) \tilde{I}_{\frac{n}{2}} \\ 0 & -I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ -\tilde{I}_{\frac{n}{2}} \left(\text{diag}\, S_{\frac{n}{2}}\right) & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} \left(\text{diag}\, T_{\frac{n}{2}}\right) & \tilde{I}_{\frac{n}{2}} \\ \tilde{I}_{\frac{n}{2}} & 0 \end{bmatrix}, \tag{25}$$

where $T_{\frac{n}{2}} = \left[\tan\frac{(2k+1)\pi}{8n}\right]_{k=0}^{\frac{n}{2}-1}$.

III. SPARSE, ORTHOGONAL, BIDIAGONAL, DIAGONAL, AND SELF-CONTAINED FACTORS FOR DST MATRICES

In this section, we will propose formulas to factor DST I/II/III/IV matrices into the product of sparse, diagonal, bidiagonal, and orthogonal matrices. Among the factorization formulas, the self-contained factorization of DST-II matrix will be used to establish a factorization formula for the DST-III matrix. Also, an orthogonal matrix with three lifting steps and the factorization formula of the DST-II matrix are utilized to obtain a factorization formula for the DCT-IV matrix.

A. SPARSE, ORTHOGONAL, DIAGONAL, AND SELF-CONTAINED FACTORS TO COMPUTE DST-I MATRIX

Let us state a novel self-enclosed factorization to compute the DST-I matrix using simple, diagonal, sparse, and orthogonal matrices.

Lemma III.1. ***(Self-enclosed DST-I factorization)*** *For an even integer $n \geq 4$, the matrix S_{n-1}^I can be factored into the form*

$$S_{n-1}^I = \tilde{P}_{n-1}^T \begin{bmatrix} I_{\frac{n}{2}-1} & 0 \\ 0 & \bar{B}_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} S_{\frac{n}{2}-1}^I & & \\ & S_{\frac{n}{2}-1}^I & \\ & & \sqrt{\frac{2}{n}} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}-1} & 0 \\ 0 & \bar{W}_{\frac{n}{2}} \end{bmatrix} \bar{H}_{n-1}. \tag{26}$$

Proof. We factor the weight matrix $W_{Q_{S^I}}$ defined in the equation (1) into the product of matrices s.t.

$$W_{Q_{S^{II}}} = \sqrt{\frac{2}{n}} \tilde{P}_{n-1}^T \begin{bmatrix} W_{even} & 0 \\ 0 & W_{odd} \end{bmatrix} \tilde{P}_{n-1}, \tag{27}$$

where $W_{even} = diag\left[\sin(\frac{2k\pi}{n})\right]_{k=1}^{\frac{n}{2}-1}$ and $W_{odd} = diag\left[\sin(\frac{(2k-1)\pi}{n})\right]_{k=1}^{\frac{n}{2}}$. Now, we use the trigonometric identities to factor G_{S^I} in equation (7) and $\widehat{G}_{S^I}$ in equation (8) into the product of diagonal and sparse matrices s.t.

$$G_{S^I} = W_{odd}^{-1} \bar{B}_{\frac{n}{2}} \begin{bmatrix} W_{even} & 0 \\ 0 & 1 \end{bmatrix} \tag{28}$$

and

$$\widehat{G}_{S^I} = \frac{1}{2} \begin{bmatrix} \tilde{I}_{\frac{n}{2}-1} & \\ & 1 \end{bmatrix} \begin{bmatrix} W_h^{-1} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \tilde{I}_{\frac{n}{2}-1} & \\ & 1 \end{bmatrix}, \tag{29}$$

where $W_h = diag\left[\sin(\frac{k\pi}{n})\right]_{k=1}^{\frac{n}{2}-1}$ and $\bar{B}_{\frac{n}{2}}$ is defined in (16). Hence by (27), (28), (29), $S_{n-1}^I = W_{Q_{S^I}} V_{Q_{S^I}}$, $V_{Q_{S^I}} = \tilde{P}_{n-1}^T \begin{bmatrix} V_{even} & \\ & V_{odd} \end{bmatrix} H_{S^I}$, and using the V_{odd} defined in (6), we can obtain

$$S_{n-1}^I = \sqrt{\frac{2}{n}} \tilde{P}_{n-1}^T \begin{bmatrix} W_{even} V_{even} & 0 \\ 0 & \bar{B}_{\frac{n}{2}} \begin{bmatrix} W_{even} V_{even} & 0 \\ 0 & 1 \end{bmatrix} \bar{W}_{\frac{n}{2}} \end{bmatrix} \bar{H}_{n-1}. \tag{30}$$

By simplifying $\tilde{I}_{\frac{n}{2}-1} W_h^{-1} \tilde{I}_{\frac{n}{2}-1}$ and H_{S^I} (defined in (4)), we can obtain $\bar{W}_{\frac{n}{2}}$ in (18) and $\bar{H}_{n-1}$ in (21), respectively. By following [43], we can obtain $S_{\frac{n}{2}-1}^I = \sqrt{\frac{4}{n}} W_{even} V_{even}$. Thus, we substitute this relationship into the equation (30) and modify the last entry in the (2,2) block of the middle matrix in (30). This is modified to fix the scaling because the $\left(\frac{n}{2}, \frac{n}{2}\right)$ entry of $\widehat{G}_{S^I}$ is $\frac{1}{\sqrt{2}}$ and scaling factor of $S_{\frac{n}{2}-1}^I$ is $\sqrt{\frac{4}{n}}$. Then, we add a scaling factor of $\frac{1}{\sqrt{2}}$ to $\bar{H}_{n-1}$ (i.e. modify the $\left(n-1, \frac{n}{2}\right)$ entry of $\bar{H}_{n-1}$) to obtain the factorization proposed in (26). □

Remark III.2. *By comparing the factorizations of the DST-I matrix in (14) and (26) (i.e. distinguishing the proposed self-contained factorization of the DST-II matrix from the DST-I factorization in [14]) one can observe that* ***(a).*** *the weight matrix is congregated in (26) as opposed to scattered weight matrices in (14),* ***(b).*** *the weight matrix $W_{Q_{S^I}}$ in (2) was factored into the product of permutation and odd/even weight matrices in (27),* ***(c).*** *the matrix G_{S^I} in (29) has been simplified and factored into the product of sparse and diagonal matrices as opposed to the matrix G_{S^I} in (7),* ***(d).*** *the matrix $\widehat{G}_{S^I}$ in (29) has been simplified and factored into the product of diagonal matrices as opposed to $\widehat{G}_{S^I}$ in (8),* ***(f).*** *the matrix $\bar{H}_{n-1}$ in (26) is orthogonal as opposed to the nonorthogonal matrix H_{S^I} in (4). Thus, the proposed self-contained factorization of the DST-I matrix in (26) is in the simplest form as opposed to the factorization of the DST-I*

matrix in (14).

Remark III.3. *The factorization of the DST-I matrix established in Lemma III.1 also yields to the following alternative factorization formula through a permutation*

$$S^I_{n-1} = P^T_{n-1}\begin{bmatrix} \bar{B}_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}-1} \end{bmatrix}\begin{bmatrix} S^I_{\frac{n}{2}-1} & & \\ & \sqrt{\frac{2}{n}} & \\ & & S^I_{\frac{n}{2}-1} \end{bmatrix}\begin{bmatrix} \bar{W}_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}-1} \end{bmatrix}\widehat{H}_{n-1}, \tag{31}$$

where P_{n-1} is an even-odd permutation matrix and $\widehat{H}_{n-1}$ is defined in (22).

B. BIDIAGONAL, DIAGONAL, ORTHOGONAL, AND SELF-CONTAINED FACTORS TO COMPUTE DST-II MATRIX

Let us state a novel self-contained factorization to compute the DST-II matrix using sparse, bidiagonal, diagonal, and orthogonal matrices.

Lemma III.4. ***(Self-enclosed DST-II factorization)*** *For an even integer $n \geq 4$, the matrix S^{II}_n can be factored into the form*

$$S^{II}_n = \tilde{P}^T_n\begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & B_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} S^{II}_{\frac{n}{2}} & 0 \\ 0 & S^{II}_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & W_{\frac{n}{2}} \end{bmatrix}\tilde{H}_n. \tag{32}$$

Proof. We factor the weight matrix $W_{Q_{S^{II}}}$ defined in the equation (2) into the product of matrices s.t.

$$W_{Q_{S^{II}}} = \sqrt{\frac{2}{n}}\tilde{P}^T_n\begin{bmatrix} W_{even} & 0 \\ 0 & W_{odd} \end{bmatrix}\tilde{P}_n, \tag{33}$$

where $W_{even} = diag\{\left[\sin(\frac{k\pi}{n})\right]_{k=1}^{\frac{n}{2}-1}, \frac{1}{\sqrt{2}}\}$ and $W_{odd} = diag\left[\sin(\frac{(2k-1)\pi}{2n})\right]_{k=1}^{\frac{n}{2}}$. Now, we can use the inverse, reciprocal, and cofunction trigonometric identities to factor $G_{S^{II}}$ in the equation (11) and $\widehat{G}_{S^{II}}$ in the equation (12) into the product of bidiagonal, diagonal, and sparse matrices s.t.

$$G_{S^{II}} = W^{-1}_{odd}B_{\frac{n}{2}}W_{even} \;\; and \;\; \widehat{G}_{S^{II}} = \tfrac{1}{2}\tilde{I}_{\frac{n}{2}}W^{-1}_{odd}\tilde{I}_{\frac{n}{2}}, \tag{34}$$

where $B_{\frac{n}{2}}$ is defined in (17). While factoring the matrix $G_{S^{II}}$, the $\left(\frac{n}{2},\frac{n}{2}\right)$ entry of $B_{\frac{n}{2}}$ is modified into $\sqrt{2}$ because the $\left(\frac{n}{2},\frac{n}{2}\right)$ entry of W_{even} is defined as $\frac{1}{\sqrt{2}}$ in (33). Hence by (33), (34), $S^{II}_n = W_{Q_{S^{II}}}V_{Q_{S^{II}}}$, $V_{Q_{S^{II}}} = \tilde{P}^T_n\begin{bmatrix} V_{even} & \\ & V_{odd} \end{bmatrix}H_{S^{II}}$, and using the V_{odd} defined in (10), we can obtain

$$S^{II}_n = \sqrt{\tfrac{2}{n}}\tilde{P}^T_n\begin{bmatrix} W_{even}V_{even} & 0 \\ 0 & \frac{1}{2}B_{\frac{n}{2}}W_{even}V_{even}\tilde{I}_{\frac{n}{2}}W^{-1}_{odd}\tilde{I}_{\frac{n}{2}} \end{bmatrix}\tilde{H}_n, \tag{35}$$

where $\tilde{H}_n$ is given in (20). By following [43], we can obtain

$$S^{II}_{even} = \sqrt{\frac{4}{n}}W_{even}V_{even}. \tag{36}$$

Thus, from the equations (35) and (36) we get

$$\begin{aligned} S^{II}_n &= \tilde{P}^T_n\begin{bmatrix} S^{II}_{\frac{n}{2}} & 0 \\ 0 & \frac{1}{2}B_{\frac{n}{2}}S^{II}_{\frac{n}{2}}\tilde{I}_{\frac{n}{2}}W^{-1}_{odd}\tilde{I}_{\frac{n}{2}} \end{bmatrix}\tilde{H}_n, \\ &= \tilde{P}^T_n\begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & B_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} S^{II}_{\frac{n}{2}} & 0 \\ 0 & S^{II}_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & \frac{1}{2}\tilde{I}_{\frac{n}{2}}W^{-1}_{odd}\tilde{I}_{\frac{n}{2}} \end{bmatrix}\tilde{H}_n. \end{aligned} \tag{37}$$

Finally, by using the above equation and the trigonometric identities to simplify the formula $\tilde{I}_{\frac{n}{2}}W^{-1}_{odd}\tilde{I}_{\frac{n}{2}}$, we can obtain the factorization formula proposed in (32). □

Remark III.5. *By comparing the factorizations of the DST-II matrix in (15) and (32) (i.e. distinguishing the proposed self-contained factorization of the DST-II matrix from the DST-II factorization in [14]) one can observe that **(a).** there are not many weight matrices spread in (32) like in (15), **(b).** a factorization for $W_{Q_{S^{II}}}$ is presented via a permutation and odd/even weight matrices in (33) as opposed to $W_{Q_{S^{II}}}$ in (2), **(c).** a factorization for $G_{S^{II}}$ is presented via a simple bidiagonal and diagonal matrices in (34) as opposed to $G_{S^{II}}$ in (11), **(d).** a factorization for $\widehat{G}_{S^{II}}$ is presented via antidiagonal and diagonal matrices in (34) as opposed to $\widehat{G}_{S^{II}}$ in (12), **(e).** an orthogonal matrix $\tilde{H}_n$ is given in (20) as opposed to the scaled orthogonal matrix $H_{S^{II}}$ in (15), and **(f).** identified the exact permutation matrix for $V_{Q_{S^{II}}}$ within the Lemma III.4 as opposed to $V_{Q_{S^{II}}}$ in (3). Thus, we have expressed an exact, simplified, and self-contained factorization formula to compute the DST-II matrix using bidiagonal, diagonal, sparse, and orthogonal factors in (32) as opposed to (15).*

Remark III.6. *The factorization of the DST-II matrix established in Lemma III.1 also yields to the following alternative factorization formula through a permutation*

$$S^{II}_n = P^T_n\begin{bmatrix} B_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} S^{II}_{\frac{n}{2}} & \\ & S^{II}_{\frac{n}{2}} \end{bmatrix}\begin{bmatrix} W_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix}H_n, \tag{38}$$

where P_n is an even-odd permutation matrix and $H_n = \frac{1}{\sqrt{2}}\begin{bmatrix} I_{\frac{n}{2}} & \tilde{I}_{\frac{n}{2}} \\ I_{\frac{n}{2}} & -\tilde{I}_{\frac{n}{2}} \end{bmatrix}$.

C. BIDIAGONAL, DIAGONAL, ORTHOGONAL, AND SELF-CONTAINED FACTORS TO COMPUTE DST-III MATRIX

Let us state a novel self-contained factorization to compute the DST-III matrix using sparse, bidiagonal, diagonal, and orthogonal matrices.

Corollary III.7. ***(Self-enclosed DST-III factorization)*** *For an even integer $n \geq 4$, the matrix S^{III}_n can be factored into*

the form

$$S_n^{III} = H_n^T \begin{bmatrix} W_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} S_{\frac{n}{2}}^{III} & 0 \\ 0 & S_{\frac{n}{2}}^{III} \end{bmatrix} \begin{bmatrix} B_{\frac{n}{2}}^T & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix} P_n. \tag{39}$$

Proof. This is trivial by Lemma III.4, $S_n^{III} = \left[S_n^{II}\right]^T$, and Remark III.6. □

D. SPARSE, ORTHOGONAL, DIAGONAL, BIDIAGONAL, AND RECURSIVE FACTORS TO COMPUTE DST-IV MATRIX

We propose a novel factorization formula to compute the DST-IV matrix using bidiagonal, diagonal, sparse, and orthogonal matrices. The proposed factorization of the DST-IV matrix is based on the self-enclosed DST-II factorization in Lemma III.4. Before we proceed, let us recall the DST-IV matrix factorization formula in [9], [12] s.t.

$$S_n^{IV} = P_n^T V_n \begin{bmatrix} S_{\frac{n}{2}}^{II} & 0 \\ 0 & S_{\frac{n}{2}}^{II} \end{bmatrix} Q_n. \tag{40}$$

To derive a novel factorization formula to compute the DST-IV matrix, we will use the above factorization formula and apply three lifting steps [44] for the rotation matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ and the rotation/reflection matrix $\begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$. Let us first derive a factorization formula for the rotational/rotational-reflection butterfly matrix Q_n defined in (24).

Lemma III.8. ***(Butterfly matrix Q_n is the same as the matrix with three lifting steps $\tilde{Q}_n$)*** *Let $n \geq 2$ be an even integer. The matrix with three lifting steps, $\tilde{Q}_n$, is the same as the rotational/rotational-reflection matrix Q_n and can be factored in the form:*

$$\tilde{Q}_n = \begin{bmatrix} J_{\frac{n}{2}} & J_{\frac{n}{2}} diag(T_{\frac{n}{2}})\tilde{I}_{\frac{n}{2}} \\ 0 & -I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ -\tilde{I}_{\frac{n}{2}} diag(S_{\frac{n}{2}}) & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} diag(T_{\frac{n}{2}}) & \tilde{I}_{\frac{n}{2}} \\ \tilde{I}_{\frac{n}{2}} & 0 \end{bmatrix}, \tag{41}$$

where $T_{\frac{n}{2}} = \left[\tan\frac{(2k+1)\pi}{8n}\right]_{k=0}^{\frac{n}{2}-1}$.

Proof. Let us consider the block products in $\tilde{Q}_n$ explicitly:

$$\tilde{Q}_n = \left[\begin{array}{c|c} J_{\frac{n}{2}}\,\mathrm{diag}(T_{\frac{n}{2}})(2I_{\frac{n}{2}} - \mathrm{diag}(S_{\frac{n}{2}})\mathrm{diag}(T_{\frac{n}{2}})) & J_{\frac{n}{2}}(I_{\frac{n}{2}} - \mathrm{diag}(T_{\frac{n}{2}})\mathrm{diag}(S_{\frac{n}{2}}))\tilde{I}_{\frac{n}{2}} \\ \hline -\tilde{I}_{\frac{n}{2}}(I_{\frac{n}{2}} - \mathrm{diag}(T_{\frac{n}{2}})\mathrm{diag}(S_{\frac{n}{2}})) & \tilde{I}_{\frac{n}{2}}\,\mathrm{diag}(S_{\frac{n}{2}})\tilde{I}_{\frac{n}{2}} \end{array}\right]. \tag{42}$$

To verify the equivalency of $\tilde{Q}_n$ and Q_n, we will consider the corresponding blocks in both matrices.

We begin with the (1,1) block of $\tilde{Q}_n$ in (42) s.t.

$$\begin{aligned} &J_{\frac{n}{2}}\mathrm{diag}(T_{\frac{n}{2}})(2I_{\frac{n}{2}} - \mathrm{diag}(S_{\frac{n}{2}})\mathrm{diag}(T_{\frac{n}{2}})) \\ &= J_{\frac{n}{2}}\mathrm{diag}\left[\left(\tan\tfrac{(2k+1)\pi}{8n}\right)\left(2 - \tan\tfrac{(2k+1)\pi}{8n}\sin\tfrac{(2k+1)\pi}{4n}\right)\right]_{k=0}^{\frac{n}{2}-1} \\ &= J_{\frac{n}{2}}\mathrm{diag}\left[2\tan\frac{(2k+1)\pi}{8n}\cos^2\frac{(2k+1)\pi}{8n}\right]_{k=0}^{\frac{n}{2}-1} \\ &= J_{\frac{n}{2}}\mathrm{diag}\left[\sin\frac{(2k+1)\pi}{4n}\right]_{k=0}^{\frac{n}{2}-1} \\ &= J_{\frac{n}{2}}\mathrm{diag}(S_{\frac{n}{2}}). \end{aligned} \tag{43}$$

Second, we consider the (1,2) block of $\tilde{Q}_n$ in (42) s.t.

$$\begin{aligned} &J_{\frac{n}{2}}(I_{\frac{n}{2}} - \mathrm{diag}(T_{\frac{n}{2}})\mathrm{diag}(S_{\frac{n}{2}}))\tilde{I}_{\frac{n}{2}} \\ &= J_{\frac{n}{2}}\mathrm{diag}\left[1 - \tan\frac{(2k+1)\pi}{8n}\sin\frac{(2k+1)\pi}{4n}\right]_{k=0}^{\frac{n}{2}-1}\tilde{I}_{\frac{n}{2}} \\ &= J_{\frac{n}{2}}\mathrm{diag}\left[1 - 2\sin^2\frac{(2k+1)\pi}{8n}\right]_{k=0}^{\frac{n}{2}-1}\tilde{I}_{\frac{n}{2}} \\ &= J_{\frac{n}{2}}\mathrm{diag}C_{\frac{n}{2}}\tilde{I}_{\frac{n}{2}}. \end{aligned} \tag{44}$$

Similarly, the (2,1) block of $\tilde{Q}_n$ in (42) can be expressed as

$$\begin{aligned} &-\tilde{I}_{\frac{n}{2}}(I_{\frac{n}{2}} - \mathrm{diag}(T_{\frac{n}{2}})\mathrm{diag}(S_{\frac{n}{2}})) \\ &= -\tilde{I}_{\frac{n}{2}}\mathrm{diag}\left[1 - \tan\frac{(2k+1)\pi}{8n}\sin\frac{(2k+1)\pi}{4n}\right]_{k=0}^{\frac{n}{2}-1} \\ &= -\tilde{I}_{\frac{n}{2}}\mathrm{diag}\left[1 - 2\sin^2\frac{(2k+1)\pi}{8n}\right]_{k=0}^{\frac{n}{2}-1}\tilde{I}_{\frac{n}{2}} \\ &= -\tilde{I}_{\frac{n}{2}}\mathrm{diag}C_{\frac{n}{2}}. \end{aligned} \tag{45}$$

Finally, the (2,2) block of $\tilde{Q}_n$ in (42) can be expressed as

$$\begin{aligned} &\tilde{I}_{\frac{n}{2}}\mathrm{diag}(S_{\frac{n}{2}})\tilde{I}_{\frac{n}{2}} \\ &= \mathrm{diag}\left[\sin\frac{(n-2k-1)\pi}{4n})\right]_{k=0}^{\frac{n}{2}-1} \\ &= \mathrm{diag}(\tilde{I}_{\frac{n}{2}}S_{\frac{n}{2}}). \end{aligned} \tag{46}$$

Thus, from (43), (44), (45), and (46), we prove the claim. □

Next, we use the immediate result to state a novel recursive factorization formula to compute the DST-IV matrix using the DST-II, sparse, and orthogonal matrices. We recall here that DST-II matrix can further be factored into the product of sparse, bidiagonal, diagonal, and orthogonal matrices as shown in Lemma III.4 followed by Remark III.6.

Lemma III.9. ***(Recursive DST-IV factorization)*** *For an even integer $n \geq 4$, the matrix S_n^{IV} can be factorized into the form*

$$S_n^{IV} = P_n^T V_n \left[\begin{array}{c|c} S_{\frac{n}{2}}^{II} & 0 \\ \hline 0 & S_{\frac{n}{2}}^{II} \end{array}\right] \tilde{Q}_n, \tag{47}$$

where V_n is defined in (23) and $\tilde{Q}_n$ is defined in (41).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

Proof. Lemma III.8 shows that $\tilde{Q}_n$ in (41) is the same as Q_n in (24). This result together with the explicit derivation for the factorization of the DST-IV matrix in [9], [12] gives the result (47). □

Remark III.10. *The DCT I-IV matrix factorization proposed in [41], [42] and the interweaving relationships among cosine and sine matrices in [10] can be utilized to obtain variants of the proposed DST I-IV matrix factorization formulas.*

IV. RELATIONSHIP AMONG DST MATRICES AND CONNECTION AMONG DST FACTORIZATIONS IN THE LITERATURE

We first present a relationship between DST-I and DST-III matrices and next between the DST-II and DST-IV matrices using bidiagonal, diagonal, and sparse matrices. After these relationships are established, we will observe connections among the proposed DST matrix factorizations with the existing factorization of the DST matrices.

Lemma IV.1. ***(Relationship between $S^I_{\frac{n}{2}-1}$ and $S^{III}_{\frac{n}{2}}$)*** *For an even integer $n \geq 4$, the matrix $S^{III}_{\frac{n}{2}}$ is related to the matrix $S^I_{\frac{n}{2}-1}$ in the following way*

$$S^{III}_{\frac{n}{2}} = \bar{B}_{\frac{n}{2}} \begin{bmatrix} S^I_{\frac{n}{2}-1} & \\ & \sqrt{\frac{2}{n}} \end{bmatrix} \bar{W}_{\frac{n}{2}}. \tag{48}$$

Proof. Let us consider the product $A = \bar{B}_{\frac{n}{2}} \begin{bmatrix} S^I_{\frac{n}{2}-1} & \\ & \sqrt{\frac{2}{n}} \end{bmatrix} \bar{W}_{\frac{n}{2}}$ explicitly:

$$A = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & -1 \\ & \ddots & \ddots & & 1 \\ & & 1 & 1 & \vdots \\ & & & 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{\sin\frac{2\pi}{n}}{\sin\frac{(n-2)\pi}{2n}} & \frac{\sin\frac{4\pi}{n}}{\sin\frac{(n-4)\pi}{2n}} & \cdots & \frac{\sin\frac{(n-2)\pi}{n}}{\sin\frac{\pi}{2n}} & 0 \\ \frac{\sin\frac{4\pi}{n}}{\sin\frac{(n-2)\pi}{2n}} & \frac{\sin\frac{8\pi}{n}}{\sin\frac{(n-4)\pi}{2n}} & \cdots & \frac{\sin\frac{2(n-2)\pi}{n}}{\sin\frac{\pi}{2n}} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{\sin\frac{(n-2)\pi}{n}}{\sin\frac{(n-2)\pi}{2n}} & \frac{\sin\frac{2(n-2)\pi}{n}}{\sin\frac{(n-4)\pi}{2n}} & \cdots & \frac{\sin\frac{2(\frac{n}{2}-1)^2\pi}{n}}{\sin\frac{\pi}{2n}} & 0 \\ 0 & 0 & \cdots & 0 & \sqrt{2} \end{bmatrix} \tag{49}$$

Let us evaluate the elements of A. First, we consider elements in the first row, but the first to the second-to-last column of (49), i.e., $j = 0$ and $k = 0, 1, \ldots, \frac{n}{2} - 2$.

$$\begin{aligned} [A]_{0,k} &= \frac{1}{\sqrt{n}} \cdot \frac{\sin\frac{2(k+1)\pi}{n}}{\sin\frac{(n-2(k+1))\pi}{2n}} \\ &= \frac{2}{\sqrt{n}} \cdot \sin\frac{(k+1)\pi}{n} \\ &= \left[S^{III}_{\frac{n}{2}}\right]_{0,k}. \end{aligned} \tag{50}$$

Second, we consider elements in the last row, but the first to the second-to-last column of (49), i.e., $j = \frac{n}{2} - 1$ and $k = 0, 1, \ldots, \frac{n}{2} - 2$.

$$\begin{aligned} [A]_{\frac{n}{2}-1,k} &= \frac{1}{\sqrt{n}} \cdot \frac{\sin\frac{2(\frac{n}{2}-1)(k+1)\pi}{n}}{\sin\frac{(n-2(k+1))\pi}{2n}} \\ &= \frac{2}{\sqrt{n}} \cdot \frac{\sin\frac{(\frac{n}{2}-1)(k+1)\pi}{n}\cos\frac{(\frac{n}{2}-1)(k+1)\pi}{n}}{\cos\frac{(k+1)\pi}{n}} \\ &= \frac{2}{\sqrt{n}} \cdot \sin\frac{(k+1)\pi}{n} \\ &= \left[S^{III}_{\frac{n}{2}}\right]_{\frac{n}{2}-1,k}. \end{aligned} \tag{51}$$

Third, we observe elements in the last column of (49) (from the first to last row), i.e., $j = 0, 1, \ldots, \frac{n}{2} - 1$ and $k = \frac{n}{2} - 1$.

$$\begin{aligned} [A]_{j,\frac{n}{2}-1} &= \frac{1}{\sqrt{n}} \cdot (-1)^j \cdot \sqrt{2} \\ &= \frac{2}{\sqrt{n}} \cdot \frac{1}{\sqrt{2}} \cdot \sin\frac{(2j+1)\pi}{2} \\ &= \left[S^{III}_{\frac{n}{2}}\right]_{j,\frac{n}{2}-1}. \end{aligned} \tag{52}$$

Finally, we consider elements in the second to second-to-last row and the first to second-to-last column of (49), i.e., $j = 1, 2, \ldots, \frac{n}{2} - 2$ and $k = 0, 1, \ldots, \frac{n}{2} - 2$.

$$\begin{aligned} [A]_{j,k} &= \frac{1}{\sqrt{n}} \left(\frac{\sin\frac{2j(k+1)\pi}{n} + \sin\frac{2(j+1)(k+1)\pi}{n}}{\sin\frac{(n-2(k+1))\pi}{2n}} \right) \\ &= \frac{2}{\sqrt{n}} \left(\frac{\sin\frac{(2j+1)(k+1)\pi}{n}\cos\frac{(k+1)\pi}{n}}{\cos\frac{(k+1)\pi}{n}} \right) \\ &= \frac{2}{\sqrt{n}} \cdot \sin\frac{(2j+1)(k+1)\pi}{n} \\ &= \left[S^{III}_{\frac{n}{2}}\right]_{j,k}. \end{aligned} \tag{53}$$

Thus, from (50), (51), (52), and (53), we prove the claim. □

The immediate result is true for any even integer $n \geq 2$ satisfying the following relationship.

Corollary IV.2. ***(Relationship between S^I_{n-1} and S^{III}_n)*** *For an even integer $n \geq 2$, the matrix S^{III}_n is related to the matrix S^I_{n-1} in the following way*

$$S^{III}_n = \bar{B}_n \begin{bmatrix} S^I_{n-1} & \\ & \frac{1}{\sqrt{n}} \end{bmatrix} \bar{W}_n. \tag{54}$$

Proof. This is trivial from the Lemma IV.1. □

Lemma IV.3. ***(Relationship between $S^{II}_{\frac{n}{2}}$ and $S^{IV}_{\frac{n}{2}}$)*** *For an even integer $n \geq 4$, the matrix $S^{IV}_{\frac{n}{2}}$ is related to the matrix $S^{II}_{\frac{n}{2}}$ in the following way*

$$S^{IV}_{\frac{n}{2}} = B_{\frac{n}{2}} S^{II}_{\frac{n}{2}} [W_c]_{\frac{n}{2}}. \tag{55}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

Proof. Let us consider the product $M = B_{\frac{n}{2}} S^{II}_{\frac{n}{2}} [W_c]_{\frac{n}{2}}$ explicitly

$$M = \frac{1}{\sqrt{n}} \cdot \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & 1 & 1 & \\ & & & 1 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{\sin\frac{\pi}{n}}{\sin\frac{(n-1)\pi}{2n}} & \frac{\sin\frac{3\pi}{n}}{\sin\frac{(n-3)\pi}{2n}} & \cdots & \frac{\sin\frac{(n-1)\pi}{n}}{\sin\frac{\pi}{2n}} \\ \frac{\sin\frac{2\pi}{n}}{\sin\frac{(n-1)\pi}{2n}} & \frac{\sin\frac{6\pi}{n}}{\sin\frac{(n-3)\pi}{2n}} & \cdots & \frac{\sin\frac{2(n-1)\pi}{n}}{\sin\frac{\pi}{2n}} \\ \frac{\sin\frac{3\pi}{n}}{\sin\frac{(n-1)\pi}{2n}} & \frac{\sin\frac{9\pi}{n}}{\sin\frac{(n-3)\pi}{2n}} & \cdots & \frac{\sin\frac{3(n-1)\pi}{n}}{\sin\frac{\pi}{2n}} \\ \vdots & \vdots & & \vdots \\ \frac{\sin\frac{(\frac{n}{2}-1)\pi}{n}}{\sin\frac{(n-1)\pi}{2n}} & \frac{\sin\frac{3(\frac{n}{2}-1)\pi}{n}}{\sin\frac{(n-3)\pi}{2n}} & \cdots & \frac{\sin\frac{(\frac{n}{2}-1)(n-1)\pi}{n}}{\sin\frac{\pi}{2n}} \\ \frac{\sqrt{2}/2}{\sin\frac{(n-1)\pi}{2n}} & \frac{\sqrt{2}/2}{\sin\frac{(n-3)\pi}{2n}} & \cdots & \frac{\sqrt{2}/2}{\sin\frac{\pi}{2n}} \end{bmatrix} \tag{56}$$

Let us evaluate the elements of M. First, we consider elements in the first row of (56), i.e., $j = 0$ and $k = 0, 1, \cdots, n/2-1$.

$$\begin{aligned} [M]_{0,k} &= \frac{1}{\sqrt{n}} \frac{\sin\left(\frac{(2k+1)\pi}{n}\right)}{\sin\left(\frac{(n-2k-1)\pi}{2n}\right)} \\ &= \frac{2}{\sqrt{n}} \frac{\sin\left(\frac{(2k+1)\pi}{2n}\right)\cos\left(\frac{(2k+1)\pi}{2n}\right)}{\cos\left(\frac{(2k+1)\pi}{2n}\right)} \\ &= \left[S^{IV}_{\frac{n}{2}}\right]_{0,k}. \end{aligned} \tag{57}$$

Second, we consider elements between the second and the $(n/2-2)^{th}$ rows of (56), i.e., $j = 1, 2, \cdots, n/2-2$ and $k = 0, 1, \cdots, n/2-1$.

$$\begin{aligned} [M]_{j,k} &= \frac{1}{\sqrt{n}} \frac{\sin\left(\frac{j(2k+1)\pi}{n}\right) + \sin\left(\frac{(j+1)(2k+1)\pi}{n}\right)}{\sin\left(\frac{(n-2k-1)\pi}{2n}\right)} \\ &= \frac{2}{\sqrt{n}} \frac{\sin\left(\frac{(2j+1)(2k+1)\pi}{2n}\right)\cos\left(\frac{(2k+1)\pi}{2n}\right)}{\cos\left(\frac{(2k+1)\pi}{2n}\right)} \\ &= \frac{2}{\sqrt{n}} \sin\left(\frac{(2j+1)(2k+1)\pi}{2n}\right) \\ &= \left[S^{IV}_{\frac{n}{2}}\right]_{j,k}. \end{aligned} \tag{58}$$

Finally, we consider elements in the last row of (56), i.e., $j = n/2-1$ and $k = 0, 1, \cdots, n/2-1$.

$$\begin{aligned} [M]_{\frac{n}{2}-1,k} &= \frac{1}{\sqrt{n}} \frac{1+\sin\left(\frac{(\frac{n}{2}-1)(2k+1)\pi}{n}\right)}{\sin\left(\frac{(n-2k-1)\pi}{2n}\right)} \\ &= \frac{1}{\sqrt{n}} \frac{1+\cos\left(\frac{(2k+1)\pi}{n}\right)}{\cos\left(\frac{(2k+1)\pi}{2n}\right)} \\ &= \frac{2}{\sqrt{n}} \cos\left(\frac{(2k+1)\pi}{2n}\right) \\ &= \frac{2}{\sqrt{n}} \sin\left(\frac{(n-1)(2k+1)\pi}{2n}\right) \\ &= \left[S^{IV}_{\frac{n}{2}}\right]_{\frac{n}{2}-1,k}. \end{aligned} \tag{59}$$

Thus, from (57), (58), and (59), we prove the claim. □

The immediate result is true for any even integer $n \geq 2$ satisfying the following relationship.

Corollary IV.4. ***(Relationship between S^{II}_n and S^{IV}_n)*** *For an even integer $n \geq 2$, the matrix S^{IV}_n is related to the matrix S^{II}_n in the following way*

$$S^{IV}_n = B_n S^{II}_n [W]_n. \tag{60}$$

Proof. This is trivial from the Lemma IV.3. □

A. CONNECTION BETWEEN THE PROPOSED DST FACTORIZATION AND THE EXISTING DST FACTORIZATIONS

In this section, we utilize the relationship among DST matrices established in Section IV to identify a relationship among the proposed DST factorization and the existing factorization of DST matrices. The relationship between DST-II and DST-IV will be utilized to establish a connection between the proposed DST-II factorization with the existing DST-II factorization in [7], [9], [11], [12], [45]–[47]. We will also show the DST-III factorization in [7], [9]–[12], [46] can be seen as the proposed self-contained DST-III factorization. The relationship between DST-I and DST-III will be utilized to establish a connection between the proposed DST-I factorization with the existing DST-I factorization in [7], [9]–[12], [46], [47].

The following result shows that the proposed factorization of the DST-I matrix can be converted into the existing DST-I factorization in [7], [9]–[12], [46], [47].

Corollary IV.5. ***(Connection between traditional DST-I and the proposed DST-I)*** *For an even integer $n \geq 4$, the matrix S^I_{n-1} can be factorized into the form*

$$S^I_{n-1} = P^T_{n-1} \begin{bmatrix} S^{III}_{\frac{n}{2}} & 0 \\ 0 & S^I_{\frac{n}{2}-1} \end{bmatrix} \widehat{H}_{n-1}. \tag{61}$$

Proof. This follows immediately from Lemma III.1, Remark III.3, and Lemma IV.1. □

The following result shows that the proposed factorization of the DST-II matrix can be converted into the existing DST-II factorization in [7], [9], [11], [12], [45]–[47].

Corollary IV.6. ***(Connection between traditional DST-II and the proposed DST-II)*** *For an even integer $n \geq 4$, the matrix S^{II}_n can be factored into the form*

$$S^{II}_n = P^T_n \begin{bmatrix} S^{IV}_{\frac{n}{2}} & 0 \\ 0 & S^{II}_{\frac{n}{2}} \end{bmatrix} H_n. \tag{62}$$

Proof. This follows immediately from Lemma III.4, Remark III.6, and Lemma IV.3. □

The following result shows that the proposed factorization of the DST-III matrix can be converted into the existing DST-III factorization in [7], [9]–[12], [46].

Corollary IV.7. ***(Connection between traditional DST-III and the proposed DST-III)*** *For an even integer $n \geq 4$, the*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

matrix S_n^{III} can be factored into the form

$$S_n^{III} = H_n^T \begin{bmatrix} S_{\frac{n}{2}}^{IV} & 0 \\ 0 & S_{\frac{n}{2}}^{III} \end{bmatrix} P_n. \tag{63}$$

Proof. Note that in Lemma IV.3, it was shown that $B_{\frac{n}{2}} S_{\frac{n}{2}}^{II} [W_c]_{\frac{n}{2}} = S_{\frac{n}{2}}^{IV}$. This, together with the facts $\left[S_n^{II}\right]^T = S_n^{III}$ and $\left[S_n^{IV}\right]^T = S_n^{IV}$, gives the result. □

Remark IV.8. *The derivation of the novel self-contained DST-I/II/III factorizations, respectively (26), (32), and (39) in this paper, are simpler than the existing interdependent factorizations for the DST-I/II/III matrices. Moreover, the proposed DST factorizations for the types I/II/III can be seen as a real variant of the self-contained factorization of the DFT matrices in [48], [49]. Also, the proposed DST-I/II/III factorizations are easy to implement and establish a bridge to the existing DST factorizations in [7], [9]–[12], [45]–[47]. We will see in Section V that the proposed algorithms are self-recursive as opposed to completely recursive algorithms in the literature. Moreover, the new algorithms require only simple, sparse, scaled orthogonal, diagonal, and bidiagonal matrices to compute the DST matrices by a vector as shown in the next section.*

V. SELF/COMPLETELY RECURSIVE AND RADIX-2 DST-I/II/III/IV ALGORITHMS

The factorizations of DST I-IV matrices established in Section III lead to self/completely recursive and radix-2 algorithms. These algorithms can be used to compute DST matrices by a vector using simple, sparse, diagonal, bidiagonal, and orthogonal matrices. In this section, we present low-complexity, recursive, and radix-2 DST I-IV algorithms. Similar to [12], we remove the scaling factor $\frac{1}{\sqrt{2}}$ from $H_n, \widehat{H}_{n-1}$, V_n, S_2^{II}, and S_2^{III} for further reduction of multiplication complexity. Compared to most of the existing algorithms, the proposed algorithms are easy to implement and attain the lowest multiplication complexity.

Before stating the algorithms, let us define the following notation to denote diagonal and bidiagonal matrices which will be used hereafter for $n \geq 4$.

$$\bar{B}_{n-1}^s = \begin{bmatrix} \bar{B}_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}-1} \end{bmatrix}, \quad \bar{W}_{n-1}^s = \begin{bmatrix} \bar{W}_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}-1} \end{bmatrix}$$
$$B_n^s = \begin{bmatrix} B_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix}, \quad W_n^s = \begin{bmatrix} W_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix}. \tag{64}$$

Also, from now onwards we call $\sqrt{2}V_n = \tilde{V}_n$. We recall here from the notations that $\sqrt{2}\widehat{H}_{n-1} = H_{n-1}^s$ and $\sqrt{2}H_n = H_n^s$.

Now we compute the scaled orthogonal DST I-IV matrices by a vector via $y = \sqrt{n}S_{n-1}^I x$, $y = \sqrt{n}S_n^{II} x$, $y = \sqrt{n}S_n^{III} x$, and $y = \sqrt{n}S_n^{IV} x$, respectively. Here we call the corresponding DST I-IV algorithms $sin1(x, n-1)$, $sin2(x, n)$, $sin3(x, n)$, and $sin4(x, n)$, respectively.

Algorithm V.1. $\mathbf{sin1}(\mathbf{x}, n-1)$
Input: $n = 2^t (t \geq 1)$, $n_1 = \frac{n}{2}$, $x \in \mathbb{R}^{n-1}$.
1) *If* $n = 2$, *then*
$y := \sqrt{2}x$.
2) *If* $n \geq 4$, *then*
$$u := H_{n-1}^{s_1} x,$$
$$[v_j]_{j=0}^{n-2} := \bar{W}_{n-1}^s u,$$
$$z_1 := sin1\left([v_j]_{j=0}^{n_1-2}, n_1 - 1\right),$$
$$z_2 := [v_{n_1-1}],$$
$$z_3 := sin1\left([v_j]_{j=n_1}^{n-2}, n_1 - 1\right),$$
$$w := \bar{B}_{n-1}^s \left(z_1^T, z_2, z_3^T\right)^T,$$
$$y := P_{n-1}^T w.$$
Output: $y = \sqrt{n}S_{n-1}^I x$.

Algorithm V.2. $sin2(x, n)$
Input: $n = 2^t (t \geq 1)$, $n_1 = \frac{n}{2}$, $x \in \mathbb{R}^n$.
1) *If* $n = 2$, *then*
$$y := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x.$$
2) *If* $n \geq 4$, *then*
$$u := H_n^s x,$$
$$[v_j]_{j=0}^{n-1} := W_n^s u,$$
$$z_1 := sin2\left([v_j]_{j=0}^{n_1-1}, n_1\right),$$
$$z_2 := sin2\left([v_j]_{j=n_1}^{n-1}, n_1\right),$$
$$w := B_n^s \left(z_1^T, z_2^T\right)^T,$$
$$y := P_n^T w.$$
Output: $y = \sqrt{n}S_n^{II} x$.

Algorithm V.3. $sin3(x, n)$
Input: $n = 2^t (t \geq 1)$, $n_1 = \frac{n}{2}$, $x \in \mathbb{R}^n$.
1) *If* $n = 2$, *then*
$$y := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x.$$
2) *If* $n \geq 4$, *then*
$$u := P_n x,$$
$$[v_j]_{j=0}^{n-1} := [B^s]_n^T u,$$
$$z_1 := sin3\left([v_j]_{j=0}^{n_1-1}, n_1\right),$$
$$z_2 := sin3\left([v_j]_{j=n_1}^{n-1}, n_1\right),$$
$$w := [W^s]_n \left(z_1^T, z_2^T\right)^T,$$
$$y := [H^s]_n^T w.$$
Output: $y = \sqrt{n}S_n^{III} x$.

Algorithm V.4. $sin4(x, n)$
Input: $n = 2^t (t \geq 1)$, $n_1 = \frac{n}{2}$, $x \in \mathbb{R}^n$.
1) *If* $n = 2$, *then*
$$y := \begin{bmatrix} \sqrt{2}\sin\frac{\pi}{8} & \sqrt{2}\cos\frac{\pi}{8} \\ \sqrt{2}\cos\frac{\pi}{8} & -\sqrt{2}\sin\frac{\pi}{8} \end{bmatrix} x.$$

2) *If* $n \geq 4$, *then*

$$
\begin{aligned}
[u_j]_{j=0}^{n-1} &:= \tilde{Q}_n\, x,\\
z_1 &:= sin2\left([u_j]_{j=0}^{n_1-1}, n_1\right),\\
z_2 &:= sin2\left([u_j]_{j=n_1}^{n-1}, n_1\right),\\
w &:= \left(\tilde{V}_n\right)\left(z_1^T, z_2^T\right)^T,\\
y &:= P_n^T w.
\end{aligned}
$$

Output: $y = \sqrt{n}\, S_n^{IV} x$.

VI. ARITHMETIC COMPLEXITY OF THE DST-I/II/III/IV ALGORITHMS

The arithmetic complexity of an algorithm tells us the number of additions/subtractions and multiplications/divisions required to execute the algorithm. If we compute an order $n \times n$ DST matrix by a vector in the usual way, it requires $\mathcal{O}(n^2)$ operations. Using the proposed DST algorithms, it is possible to compute the DST matrices by a vector using $\mathcal{O}(n \log n)$ complexity with a significant reduction of multiplication cost.

Here we compute the number of additions (say $\#a$) and multiplications (say $\#m$) required to produce $y = \sqrt{n} S_{n-1}^I x$ and $y = \sqrt{n} S^{II/III/IV}\ x$. Note that we do not count multiplication by ± 1.

A. LOW-COMPLEXITY DST-I/II/III/IV ALGORITHMS

Here we analyze the arithmetic complexity of the self-recursive and radix-2 DST-I/II/III algorithms and the completely-recursive and radix-2 DST-IV algorithm presented in Section V.

Lemma VI.1. *For a given* $n = 2^t$ $(t \geq 2)$, *the arithmetic complexity in computing the DST-I algorithm, i.e.,* $sin1(x, n-1)$ *is given by*

$$
\begin{aligned}
\#a\,(\textit{DST-I}, n-1) &= 2nt - 4n + 4,\\
\#m\,(\textit{DST-I}, n-1) &= \frac{1}{2} nt.
\end{aligned} \tag{65}
$$

Proof. Referring to the DST-I algorithm, we get

$$
\begin{aligned}
\#a(\text{DST-I}, n-1) = \{&2 \cdot \#a\left(\text{DST-I}, \frac{n}{2} - 1\right) + \#a\left(H_{n-1}^{s_1}\right)\\
&+ \#a\left(\bar{W}_{n-1}^{s}\right) + \#a\left(\bar{B}_{n-1}^{s}\right)\},\\
\#m(\text{DST-I}, n-1) = \{&2 \cdot \#m\left(\text{DST-I}, \frac{n}{2} - 1\right) + \#m\left(H_{n-1}^{s_1}\right)\\
&+ \#m\left(\bar{W}_{n-1}^{s}\right) + \#m\left(\bar{B}_{n-1}^{s}\right)\}.
\end{aligned} \tag{66}
$$

Following the structures of $H_{n-1}^{s_1}$, $\bar{W}_{n-1}^{s}$, and $\bar{B}_{n-1}^{s}$, we get

$$
\begin{aligned}
\#a\left(H_{n-1}^{s_1}\right) &= n-2, \qquad \#m\left(H_{n-1}^{s_1}\right) = 1,\\
\#a\left(\bar{W}_{n-1}^{s}\right) &= 0, \qquad \#m\left(\bar{W}_{n-1}^{s}\right) = \tfrac{n}{2} - 1,\\
\#a\left(\bar{B}_{n-1}^{s}\right) &= n-2, \qquad \#m\left(\bar{B}_{n-1}^{s}\right) = 0.
\end{aligned}
$$

Using the above result, we can rewrite (66) as

$$
\#a(\text{DST-I}, n-1) = 2 \cdot \#a\left(\text{DST-I}, \frac{n}{2} - 1\right) + 2n - 4.
$$

$$
\#m(\text{DST-I}, n-1) = 2 \cdot \#m\left(\text{DST-I}, \frac{n}{2} - 1\right) + \frac{n}{2}.
$$

Since $n = 2^t$, the above simplifies to the first order linear difference equations with respect to $t \geq 2$ s.t.

$$
\#a(\text{DST-I}, 2^t - 1) - 2 \cdot \#a\left(\text{DST-I}, 2^{t-1} - 1\right) = 2^{t+1} - 4.
$$

$$
\#m(\text{DST-I}, 2^t - 1) - 2 \cdot \#m\left(\text{DST-I}, 2^{t-1} - 1\right) = 2^{t-1}.
$$

Solving the above first order linear difference equations using the initial conditions $\#a\ (\text{DST-I}, 1) = 0$ and $\#m\ (\text{DST-I}, 1) = 1$, we can obtain

$$
\begin{aligned}
\#a(\text{DST-I}, 2^t - 1) &= 2nt - 4n + 4,\\
\#m(\text{DST-I}, 2^t - 1) &= \frac{1}{2} nt.
\end{aligned}
$$

□

Lemma VI.2. *For a given* $n = 2^t$ $(t \geq 2)$, *the arithmetic complexity in computing the DST-II algorithm* $sin2(x, n)$ *is given by*

$$
\begin{aligned}
\#a\,(\textit{DST-II}, n) &= \frac{3}{2} nt - n + 1,\\
\#m\,(\textit{DST-II}, n) &= \frac{1}{2} nt - 1.
\end{aligned} \tag{67}
$$

Proof. Referring to the DST-II algorithm $sin2(x, n)$, we get

$$
\begin{aligned}
\#a(\text{DST-II}, n) = \{&2 \cdot \#a\left(\text{DST-II}, \frac{n}{2}\right) + \#a\left(H_n^s\right) + \#a\left(W_n^s\right)\\
&+ \#a\left(B_n^s\right)\},\\
\#m(\text{DST-II}, n) = \{&2 \cdot \#m\left(\text{DST-II}, \frac{n}{2}\right) + \#m\left(H_n^s\right) + \#m\left(W_n^s\right)\\
&+ \#m\left(B_n^s\right)\}.
\end{aligned} \tag{68}
$$

Following the structures of H_n^s, W_n^s, and B_n^s, we get

$$
\begin{aligned}
\#a\left(H_n^s\right) &= n, \qquad \#m\left(\bar{H}_n\right) = 0,\\
\#a\left(W_n^s\right) &= 0, \qquad \#m\left(W_n^s\right) = \tfrac{n}{2},\\
\#a\left(B_n^s\right) &= \tfrac{n}{2} - 1, \qquad \#m\left(B_n^s\right) = 1.
\end{aligned}
$$

Using the above result, we can rewrite (68) as

$$
\#a(\text{DST-II}, n) = 2 \cdot \#a\left(\text{DST-II}, \frac{n}{2}\right) + \frac{3}{2} n - 1,
$$

and

$$
\#m(\text{DST-II}, n) = 2 \cdot \#m\left(\text{DST-II}, \frac{n}{2}\right) + \frac{n}{2} + 1.
$$

Since $n = 2^t$, the above simplifies to the first order linear difference equations with respect to $t \geq 2$ s. t.

$$
\#a(\text{DST-II}, 2^t) - 2 \cdot \#a\left(\text{DST-II}, 2^{t-1}\right) = 3 \cdot 2^{t-1} - 1.
$$

and

$$
\#m(\text{DST-II}, 2^t) - 2 \cdot \#m\left(\text{DST-II}, 2^{t-1}\right) = 2^{t-1} + 1.
$$

Solving the above first order linear difference equations using the initial conditions $\#a\ (\text{DST-II}, 2) = 2$ and $\#m\ (\text{DST-II}, 2) = 0$, we can obtain

$$
\#a(\text{DST-II}, 2^t) = \frac{3}{2} nt - n + 1.
$$

and

$$\#m(\text{DST-II}, 2^t) = \frac{1}{2}nt - 1.$$

Hence the result. □

Corollary VI.3. *For a given $n = 2^t$ $(t \geq 2)$, the arithmetic complexity in computing the DST-III algorithm $sin3(x, n)$ is given by*

$$\begin{aligned} \#a\ (\textit{DST-III}, n) &= \frac{3}{2}nt - n + 1, \\ \#m\ (\textit{DST-III}, n) &= \frac{1}{2}nt - 1. \end{aligned} \tag{69}$$

Proof. This is trivial as the factorization of the DST-III matrix is obtained using the factorization of DST-II matrix with the help of the transpose property. □

Remark VI.4. *The arithmetic complexity of the proposed DST-II/III algorithms is exactly the same as that of the lowest multiplication complexity DCT-II/III algorithms in [42]. This is due to the fact that the DST-II/III and DCT-II/III matrices are related to each other via transpositions, permutations, and sign flips [50]–[53].*

Lemma VI.5. *For a given $n = 2^t$ $(t \geq 2)$, the arithmetic complexity in computing the DST-IV recursively via $sin4(\mathbf{x}, n)$ and $sin2(\mathbf{x}, n)$ algorithms is given by*

$$\begin{aligned} \#a\ (\textit{DST-IV}, n) &= \frac{3}{2}nt, \\ \#m\ (\textit{DST-IV}, n) &= \frac{1}{2}nt + n. \end{aligned} \tag{70}$$

Proof. The number of additions and multiplications required to compute the DST-IV algorithm can be found by using the number of additions and multiplications required to compute the DST-II algorithm at $\frac{n}{2}$.
By using the DST-IV algorithm, we get

$$\begin{aligned} \#a(\text{DST-IV}, n) = \{2 \cdot \#a\left(\text{DST-II}, \frac{n}{2}\right) + \#a\left(\tilde{Q}_n\right) \\ + \#a\left(\tilde{V}_n\right)\}, \\ \#m(\text{DST-IV}, n) = \{2 \cdot \#m\left(\text{DST-II}, \frac{n}{2}\right) + \#m\left(\tilde{R}_n\right) \\ + \#m\left(\tilde{V}_n\right)\}. \end{aligned} \tag{71}$$

The structures of $\tilde{Q}_n$ and $\tilde{V}_n$ gives us

$$\begin{aligned} \#a\left(\tilde{Q}_n\right) = \tfrac{3}{2}n, \quad \#m\left(\tilde{Q}_n\right) = \tfrac{3}{2}n, \\ \#a\left(\tilde{V}_n\right) = n - 2, \quad \#m\left(\tilde{V}_n\right) = 2. \end{aligned} \tag{72}$$

Following the arithmetic complexity in computing DST-II algorithm in Lemma VI.2 and using the equations (72) and (71), we get

$$\begin{aligned} \#a(\text{DST-IV}, n) = 2\left(\frac{3}{2} \cdot \frac{n}{2} \cdot (t-1) - \frac{n}{2} + 1\right) + \frac{5}{2}n - 2. \\ \#m(\text{DST-IV}, n) = 2\left(\frac{1}{2} \cdot \frac{n}{2} \cdot (t-1) - 1\right) + \frac{3}{2}n + 2. \end{aligned} \tag{73}$$

Simplifying the above yields the multiplication complexity of the modified DST-IV algorithm as in (70). □

Remark VI.6. ***(a)*** *The paper [54] presents the total lowest arithmetic operations in calculating DST-IV using the DFT and DCT-IV while leaving an open question: whether the DST-IV algorithm in [54] leads to practical gains in performance on real computers. To compute DST-IV, the authors of [54] used twiddle factors but these factors can be overlapped with other operations in the CPU [55]. Also to compute DST-IV, authors in [54] have used scaled factors of the form $\pm 1 \pm i\tan(2\pi k/n)$ and $\pm\cot(2\pi k/n) \pm i$, where $i^2 = -1$ but these factors contain singular functions. Hence, the practical gains of the lowest flop count for DST-IV calculation in [54] may lead to a numerically ill-posed problem. We will show in Section VI-B, that the proposed DST-IV algorithm attains the lowest multiplication complexity algorithm to compute the DST-IV matrix by a vector.*
(b) *The arithmetic complexity of the proposed DST-IV is exactly the same as that of lowest multiplication complexity DCT-IV in [42] because DST-IV is exactly equivalent to a DCT-IV in which the outputs are reversed and every other input is multiplied by -1 (or vice-versa) [50], [54].*

B. NUMERICAL ILLUSTRATIONS FOR THE COMPLEXITY OF DST-I/II/III/IV ALGORITHMS

In this section, we compare the arithmetic complexity of the proposed DST I-IV algorithms with the most existing DST algorithms in the literature. Tables 1 through 4 provide the number of additions and multiplications necessary to compute the DST I-IV algorithms for input sizes ranging from $n = 8$ to $n = 4096$.

Although the total flop count of the DST-I algorithm in [47] is lower than all other DST-I algorithms in [7], [10], [12] and the proposed algorithm, the **sin1**(**x**,n) algorithm is presented using simple, sparse, diagonal, and scaled orthogonal matrices as opposed to the DST-I factorization in [47].

Based on the counts in computing DST-II/III algorithms, the proposed DST-II/III algorithms have the lowest multiplication counts. We recall here that the authors in [53] calculated the lowest flop count split-radix DST-II/III algorithms. By considering the fact that the authors in [53] have reduced the flop count only by cutting off multiplication counts while preserving the same addition counts as in [56], the explicit multiplication count in computing split-radix DST-II/III algorithms in [53] is given via $(7/18)nt + (10/27)n - (1/9)(-1)^t t + (7/54)(-1)^t + 1/2$. In this situation, the split-radix DST-II/III algorithms attain the lowest multiplication counts for $n \geq 32$ but without using simple, sparse, diagonal, bidiagonal, and scaled orthogonal matrices. Furthermore, the split-radix DST-II/III algorithms leave a question based on the practical gains in performance on real computers as the algorithms in [53] use singular functions and hence may lead

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

TABLE 1: Number of additions and multiplications required to compute DST-I algorithms.

n	**sin1**(**x**,n)		[12]		[47]		[10]		[7]		[57]	
	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m
8	20	12	18	9	27	5	18	14	19	5	20	5
16	68	32	58	26	72	17	58	43	62	17	68	17
32	196	80	160	73	185	49	160	118	175	51	196	49
64	516	192	408	186	458	129	408	299	456	141	516	129
128	1284	448	990	457	1099	321	990	726	1129	367	1284	321
256	3076	1024	2326	1082	2572	769	2326	1707	2698	913	3076	769
512	7172	2304	5340	2505	5901	1793	5340	3926	6283	2195	7172	1793
1024	16388	5120	12052	5690	13326	4097	12052	8875	14348	5141	16388	4097
2048	36868	11264	26842	12745	29711	9217	26842	19798	32269	11799	36868	9217
4096	81924	24576	59145	28128	65552	20481	59154	43691	71694	26649	81924	20481

TABLE 2: Number of additions and multiplications required to compute DST-II algorithms.

n	**sin2**(**x**,n)		[12]		[47], [52], [57]–[59]		[7]		[56]	
	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m
8	29	11	26	16	29	12	29	13	29	13
16	81	31	72	46	81	32	83	35	81	33
32	209	79	186	112	209	80	219	91	209	81
64	513	191	456	270	513	192	547	227	513	193
128	1217	447	1082	624	1217	448	1315	547	1217	449
256	2817	1023	2504	1422	2817	1024	3075	1283	2817	1025
512	6401	2303	5690	3184	6401	2304	7043	2947	6401	2305
1024	14337	5119	12744	7054	14337	5120	15875	6659	14337	5121
2048	31745	11263	28218	15472	31745	11264	35331	14851	31745	11265
4096	69633	24575	61896	33678	69633	24576	77827	32771	69633	24577

TABLE 3: Number of additions and multiplications required to compute DST-III algorithms.

n	**sin3**(**x**,n)		[12]		[47], [52], [58], [59]		[57]		[7]		[56]	
	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m
8	29	11	26	16	29	12	34	12	29	13	29	13
16	81	31	72	46	81	32	98	32	83	35	81	33
32	209	79	186	112	209	80	258	80	219	91	209	81
64	513	191	456	270	513	192	642	192	547	227	513	193
128	1217	447	1082	624	1217	448	1538	448	1315	547	1217	449
256	2817	1023	2504	1422	2817	1024	3586	1024	3075	1283	2817	1025
512	6401	2303	5690	3184	6401	2304	8194	2304	7043	2947	6401	2305
1024	14337	5119	12744	7054	14337	5120	18434	5120	15875	6659	14337	5121
2048	31745	11263	28218	15472	31745	11264	40962	11264	35331	14851	31745	11265
4096	69633	24575	61896	33678	69633	24576	90114	24576	77827	32771	69633	24577

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

TABLE 4: Number of additions and multiplications required to compute DST-IV algorithms.

n	**sin4**(**x**,n)		[12]		[47]		[60]		[7]		[56]		[57]	
	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m	#a	#m
8	36	20	30	30	36	20	-	-	38	22	32	22	35	21
16	96	48	82	66	96	48	-	-	104	56	84	56	95	49
32	240	112	206	158	240	112	-	-	264	136	208	130	239	113
64	576	256	498	354	576	256	576	274	640	320	500	300	575	257
128	1344	576	1166	798	1344	576	2176	1058	1504	736	1168	670	1343	577
256	3072	1280	2674	1762	3072	1280	8448	4162	3456	1664	2676	1488	3071	1281
512	6912	2816	6030	3870	6912	2816	-	-	7808	3712	6032	3258	6911	2817
1024	15360	6144	13426	8418	15360	6144	131584	65794	17408	8192	13428	7092	15359	6145
2048	33792	13312	29582	18206	33792	13312	-	-	38400	17920	29584	15318	33791	13313
4096	73728	28672	64626	39138	73728	28672	-	-	83968	38912	64628	32920	73727	28673

to ill-posed problems.

By considering the fact that the authors in [53] have reduced the flop count in computing DST-IV matrix by a vector only by cutting off multiplication counts, the explicit multiplication count in computing the split-radix DST-IV algorithm in [53] is given via $(5/9)nt + (37/27)n + (2/9)(-1)^t t - (10/27)(-1)^t - 2$. Even though the total flop count of the split-radix DST-IV algorithm in [54] is lower than all the other DST-IV algorithms in the literature, the multiplication counts of the proposed $\mathbf{sin4}(\mathbf{x}, n)$ algorithm and the DST-IV algorithm in [47] is the lowest in the literature. Moreover, the proposed DST-IV algorithm is presented using simple, sparse, diagonal, bidiagonal, and scaled orthogonal matrices as opposed to the DST-IV factorization in [47] and any other existing DST-IV algorithms in the literature. Note that if the factorization of the DCT or DST does not preserve orthogonality, the resulting DCT or DST algorithms lead to inferior numerical stability [61].

To sum up, the proposed self/completely recursive and radix-2 **sin2**(**x**,n), **sin3**(**x**,n), and **sin4**(**x**,n) algorithms can be computed via simple, sparse, diagonal, bidiagonal, and scaled orthogonal matrices having the lowest multiplication counts in the literature.

VII. SIGNAL FLOW GRAPHS FOR DST-I/II/III/IV ALGORITHMS

Signal flow graphs can be used to represent a physical system and its controllers with the help of a system of linear equations. We present signal flow graphs of the proposed DST algorithms while illustrating the connection between system components, associated equations, and the simplicity of the proposed factorizations of the DST I-IV matrices. The signal flow graphs are drawn using decimation-in-frequency. It is possible to convert these signal flow graphs into decimation-in-time flow graphs.

For a given input signal x, we present signal flow graphs

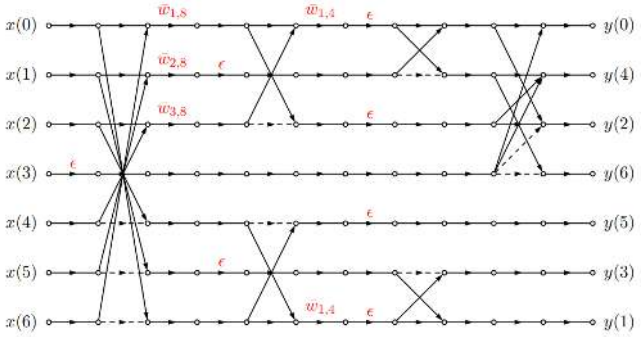


FIGURE 1: Signal flow graph for the 7-point scaled DST-I algorithm, i.e., $\sqrt{8}S_7^I$.

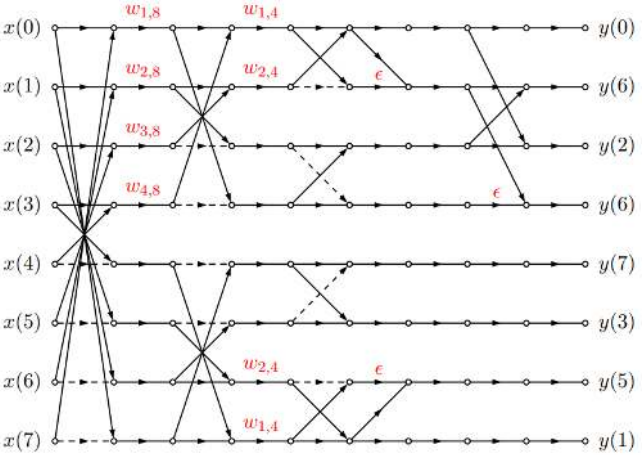


FIGURE 2: Signal flow graph for the 8-point scaled DST-II algorithm, i.e., $\sqrt{8}S_8^{II}$.

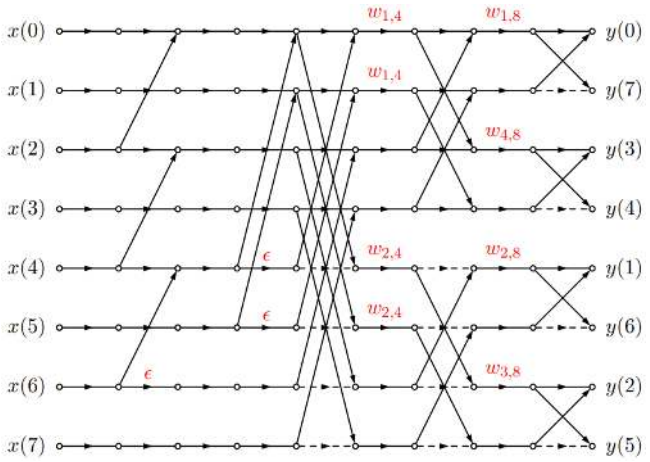


FIGURE 3: Signal flow graph for the 8-point scaled DST-III algorithm, i.e., $\sqrt{8}S_8^{III}$.

for the output signal $y = \sqrt{n}S_{n-1}^{I}\ x$, $y = \sqrt{n}S_n^{II}\ x$, $y = \sqrt{n}S_n^{III}\ x$, and $y = \sqrt{n}S_n^{IV}\ x$. Fig. 1, Fig. 2, Fig. 3 and Fig. 4 present the signal flow graphs for the $msin1(x, n-1)$, $msin2(x, n)$, $msin3(x, n)$, and $msin4(x, n)$ algorithms, respectively. The notation in the figures are defined as $\epsilon := \sqrt{2}$, $\bar{w}_{j,k} := \frac{1}{2}\csc\frac{(k-2j)\pi}{2k}$, $w_{j,k} := \frac{1}{2}\csc\frac{(k-2j+1)\pi}{2k}$, $t_{j,k} = \tan\frac{(2j+1)\pi}{8k}$, $s_{j,k} = \sin\frac{(2j+1)\pi}{4k}$, and the dotted lines indicate a multiplication by factor -1.

VIII. ENCRYPTION USING DST ALGORITHMS

In [62], a $4f$ encryption scheme was proposed which introduces a chaotic mapping to the established double random phase encoding (DPRE) scheme originally proposed in [63]–[65]. The DPRE process applies to a phase image a random phase mask followed by a Fourier transform twice in series. While the DPRE scheme maximizes the entropy of the encrypted image, the linearity of the scheme creates vulnerabilities [66], [67]. Thus, [62] proposed an encryption scheme that applies a chaotic pixel mapping based on the famous 3D Lorenz System after the first Fourier transform. The Lorenz system, also commonly known as the Lorenz path or Lorenz attractor, was first described in [68]. The path itself is the solution to the set of ordinary differential equations

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x),\\ \frac{dy}{dt} &= \rho x - y - xz,\\ \frac{dz}{dt} &= xy - \beta z,\end{aligned}$$

where σ, ρ, and β are system parameters and the initial conditions are $x(0) = x_0$, $y(0) = y_0$, $z(0) = z_0$. Given time to diverge, this set of equations will produce completely different paths with even the slightest difference in the system parameters. This chaotic nature, along with the relative speed by which the set of equations may be solved, makes the Lorenz system a valuable tool for mapping and permutation techniques.

In the following, we describe image encryption based on the DPRE encryption technique in [62] and the proposed DST algorithms.

1) An image with pixel values $f(x, y)$, where x and y are locations in the spatial domain, is converted into a phase image $p_0(x, y) = e^{i\pi f(x,y)}$.
2) A random phase mask is applied to each pixel as defined by $p_1(x, y) = p_0(x, y)e^{2\pi i r(x,y)}$, where r is a random matrix having the same size as the original image.
3) The image is transformed into the Fourier domain using the 2D DFT and the proposed **sin2** algorithm in 2D.
4) A chaotic mapping technique is applied based on the 3D Lorenz system. The Lorenz system is solved using Matlab's *ode45* function. First, the image is decomposed into $n \times n$ blocks, where n is a multiple of the total length of the image. The red, green, and blue color channels of each block are associated with the position vectors of the Lorenz path in time. These blocks are then sorted in ascending order row-major order based on the value of the associated Lorenz system position. This produces a new image p_L.
5) A random phase mask is applied to each pixel as defined by $p_2(u, v) = p_L(u, v)e^{2\pi i s(u,v)}$, where s is a random matrix of the same size as the original image, u and v are positions in the Fourier domain.
6) The proposed **sin2** algorithm and the DFT are applied to p_2 to produce the fully encrypted image.

The final result is an image of high entropy with a Gaussian distribution of pixel values across all channels. To decrypt the image, the process is simply performed in reverse using the same random phase mask keys, Lorenz system parameters, and the inverse of the respective transform. For the decryption of the image, the proposed 2D **sin3** algorithm and the 2D inverse-DFT are used. For a block length n less than or equal to $\frac{1}{8}^{th}$ of the total image length (i.e. $n = 32$ for a 256×256 image), attempts to decrypt the image using Lorenz system initial conditions with even the most minute difference will fail. Blocks which are $\frac{1}{4}^{th}$ the total image length (i.e. $n = 64$ for a 256×256 image) will reveal some information if transformed into the Fourier domain using the DFT; however, we will show below that images transformed using the proposed DST algorithms will withstand these attacks. Figure 5 shows the original image which was used for testing purposes and four attempts to decrypt using an incorrect key for the Lorenz path. The corresponding image histograms are also displayed. The Lorenz parameter key used to encrypt the image is

$$[\sigma = 10, \beta = \frac{8}{3}, \rho = 28] \tag{74}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3120051, IEEE Access

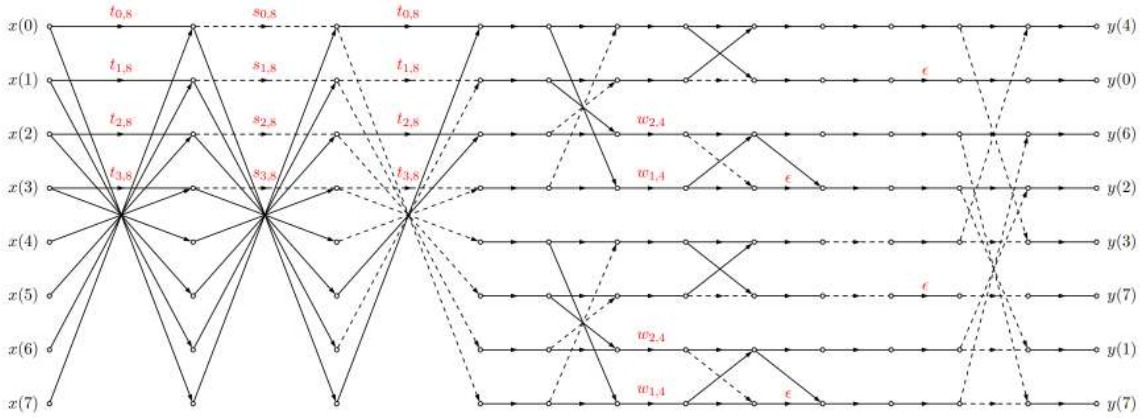


FIGURE 4: Signal flow graph for the 8-point scaled DST-IV algorithm, i.e., $\sqrt{8}S_8^{IV}$.

and the decryption attack is simulated using the key

$$[\sigma = 10.000000000000001, \beta = \frac{8}{3}, \rho = 28] \qquad (75)$$

As shown in Figures 5.c through 5.f, the encryption scheme remains secure for small block sizes when transformed into the Fourier domain through either the proposed **sin2** algorithm or the DFT. This remains true for the encryption scheme employing larger mapping blocks when the proposed **sin2** algorithm is used, as shown by Figures 5.g and 5.h. When the encryption scheme uses these block sizes with the DFT, however, the scheme starts to become liable to attack. As illustrated by Figure 5.j, the decryption attack does not yield an even distribution of pixel intensities across the channels. Likewise, Figure 5.i starts to reveal the characteristic blue and red nose of the baboon in the original image.

IX. CONCLUSION

We have proposed novel factorization formulas to compute DST I-IV matrices using the product of simple, sparse, diagonal, bidiagonal, and scaled orthogonal matrices. The proposed DST matrix factorizations are used to establish self/completely recursive and radix-2 DST algorithms. We have also established a novel algebraic relationship between DST-II and DST-IV matrices using diagonal and bidiagonal matrices and described connections between traditional DST-II/III algorithms and the proposed DST-II/III algorithms. Moreover, we have established another novel algebraic relationship between DST-I and DST-III matrices using sparse and diagonal matrices. We have also described connections between traditional DST-I algorithms and the proposed DST-I algorithm. The arithmetic complexity of the proposed algorithms is analyzed to show that the proposed algorithms are much more efficient than the existing algorithms. Specifically, the DST-IV algorithm attains the lowest multiplication complexity for all transform matrix sizes $n \geq 8$ in the literature. The DST-II/III algorithms are the lowest multiplication complexity, radix-2, and self-recursive algorithms and execute using simple, sparse, diagonal, bidiagonal, and scaled orthogonal matrices for the transform matrix sizes $n \geq 8$. The presented signal flow graphs illustrate the simplicity of the proposed DST algorithms and provide a foundational framework for the DST-based integrated circuits. The robust performance of the proposed algorithms within software applications is explored through implementation within a DPRE encryption scheme with a chaotic Lorenz path mapping. The proposed DST algorithms in 2D are able to increase encryption performance when compared to an implementation using the 2D DFT.

REFERENCES

[1] A. K. Jain, "A sinusoidal family of unitary transform," *IEEE. Trans. Pattern Anal. Mach. Intell.*, 1:356-365, 1979.

[2] A. K. Jain, "A fast Karhunen-Loeve transform for a class of stochastic processes", *IEEE. Trans. Commun.*, 24:1023-1029, 1976.

[3] H. B. Kekre and J. K. Solanki, "Comparative performance of various trigonometric unitary transforms for transform image coding," *International Journal of Electronics*, 44(3):305-315, 1978.

[4] Z. Wang and B. R. Hunt, "The discrete W transform," *Applied Mathematics and Computation*, 16:19-48, 1985.

[5] P. C. Yip and K. R. Rao, "A fast computational algorithm for the discrete sine transform," *IEEE Trans. Commun.*, 28:304-307, 1980.

[6] W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Comm.*, 25: 1004-1009, 1977.

[7] Z. Wang, "Fast algorithms for the discrete W transform and the discrete Fourier transform," *IEEE Trans. Acoust. Speech Signal Process*, 32:803-816, 1984.

[8] S. M. Perera, "Signal Processing based on Stable radix-2 DCT I-IV Algorithms having Orthogonal Factors," *Electronic Journal of Linear Algebra*, 31:362-380, 2016.

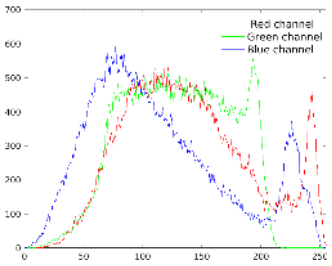
[9] S. M. Perera and V. Olshevsky, "Fast and Stable Algorithms for Discrete Sine Transformations having Orthogonal Factors," in M.G. Cojocaru, I. S. Kotsireas, R. N. Makarov, R. V. N. Melnik, and H. Shodiev (Eds.), *Interdisciplinary Topics in Applied Mathematics, Modeling and Computational Science*, Springer International, Switzerland, 117:347-354, 2015.

[10] G. Plonka and M. Tasche, "Fast and Numerically stable algorithms for discrete cosine transforms," *Linear Algebra and its Applications*, 394:309-345, 2005.

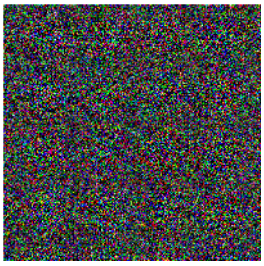
[11] V. Britanak, P. C. Yip, and K. R. Rao, "Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations," Academic Press, Great Britain, 2007.

[12] S. M. Perera, "Signal Flow Graph Approach to Efficient and Forward Stable DST Algorithms," *Linear Algebra and Its Applications*, 542: 360-390, 2018.

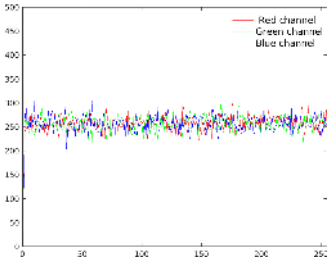
(a) Original baboon image of size 256×256.



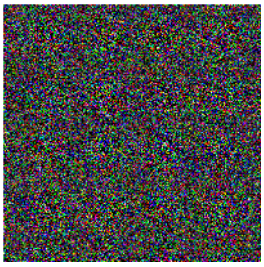
(b) Histogram of the original baboon image.



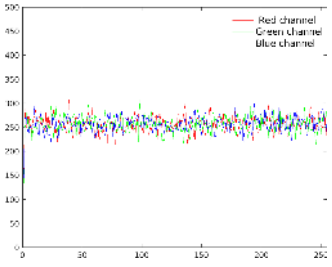
(c) Decrypted baboon image using the proposed DST-II/III algorithms with 8×8 blocks.



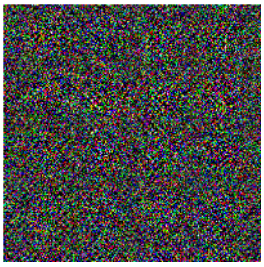
(d) Histogram of encrypted baboon image using the proposed DST-II/III algorithms with 8×8 blocks.



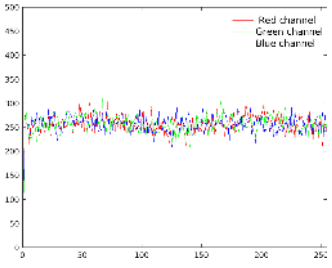
(e) Decrypted baboon image using the DFT with 8×8 blocks.



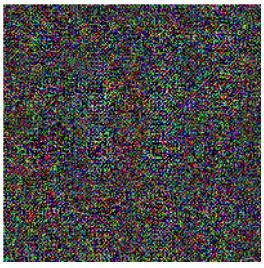
(f) Histogram of encrypted baboon image using DFT with 8×8 blocks.



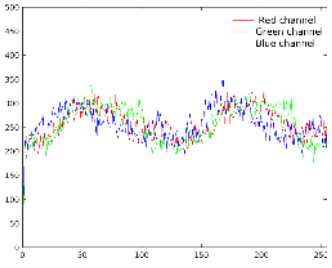
(g) Decrypted baboon image using the proposed DST-II/III algorithms with 64×64 blocks.



(h) Histogram of encrypted baboon image using the proposed DST-II/III algorithms with 64×64 blocks.



(i) Decrypted baboon image using the DFT with 64×64 blocks.



(j) Histogram of encrypted baboon image using the DFT with 64×64 blocks.

FIGURE 5: Comparison of encryption performance using the proposed DST-II/III algorithms and the DFT.

[13] S. M. Perera, A. Madanayake, N. Dornback, and N. Udayanga, "Design and Digital Implementation of Fast and Recursive DCT II-IV Algorithms," *Circuits, System, and Signal Processing*, 38(2): 529-555, 2019.

[14] A. Olshevsky, V. Olshevsky, and J. Wang, "A comrade-matrix-based derivation of the eight versions of fast cosine and sine transforms," in *Fast Algorithms for Structured Matrices: Theory and Applications*, CONM/323:119-150, AMS publications, May 2003.

[15] P. Dahiya and P. Jain, "Realization of first-order structure for recursive algorithm of discrete Sine transform," *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-5, 2017, DOI: 10.1109/ICCCNT.2017.8204014.

[16] P. Jain, B. Kumar, and S. B. Jain, "Discrete sine transform and its inverse - realization through recursive algorithms," *International Journal of Circuit Theory and Applications*, 36(4), 2008.

[17] P. Jain and A. Jain, "Regressive structures for computation of DST-II and its inverse," *ISRN Electronics*, 2012:1-4, 2012.

[18] T. Luo and J. G. Liu, "A fast algorithm for discrete sine transform using first-order moment," *2011 International Conference on Image Analysis and Signal Processing*, pp. 10-15, 2011, DOI: 10.1109/IASP.2011.6108988.

[19] M. N. Murty, "Radix-2 algorithms for implementation of type-II discrete cosine transform and discrete sine transform," *International Journal of Engineering Research and Applications*, 3(3):602-608, 2013.

[20] V. G. Reju, S. N. Koh, and Y. Soon, "Convolution using discrete sine and cosine transforms," *IEEE Signal Processing Letters*, 14(7):445-448, 2007.

[21] B. N. Madhukar and S. Jain, "A duality theorem for the discrete sine transform - IV (DST - IV)," *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2016, doi:10.1109/icaccs.2016.7586322.

[22] B. N. Madhukar and S. Jain, "A duality theorem for the discrete sine transform (DST)," *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), IEEE*, pp. 156-160, 2015.

[23] T. Binesh, M. H. Supriya and P. R. S. Pillai, "Discrete Sine Transform based HMM underwater signal classifier," *2011 International Symposium on Ocean Electronics*, pp. 152-156, 2011, doi: 10.1109/SYMPOL.2011.6170513.

[24] K. Ramadan, A. S. Fiky, M. I. Dessouky, F. E. A. El-Samie, "Equalization and carrier frequency offset compensation for UWA-OFDM communication systems based on the discrete sine transform," *Digital Signal processing*, 90:142-149, 2019.

[25] S. Dhamija and P. Jain, "Comparative Analysis for Discrete Sine Transform as a suitable method for noise estimation," *IJCSI International Journal of Computer Science Issues*, 8(5):162-164, 2011.

[26] C. Tseng and S. Lee, "Closed-form design of FIR frequency selective filter using discrete sine transform," *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 591-594, 2016, doi: 10.1109/APCCAS.2016.7804039.

[27] M. E. Keshk, M. Abd El-Naby, S. Elrabie, M. I. Dessouky, and F. E. Abd El-Samie, "Digital modulation recognition in OFDM systems using support vector machine classifier," *CiiT International Journal of Networking and Communication Engineering*, 5(12):516-525, 2013.

[28] G. Curci and F. Corsi, "Discrete sine transform for multi-scales realized volatility measures," *Quantitative Finance*, 12(2):263-279, 2012.

[29] G. Curci and F. Corsi, "A Discrete Sine Transform Approach for Realized Volatility Measurement," *National Centre of Competence in Research Financial Valuation and Risk Management*, 2006.

[30] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad and T. Stouraitis, "Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST," in *IEEE Transactions on Circuits and Systems I*: Regular Papers, 52(6):1125-1137, 2005, doi: 10.1109/TCSI.2005.849109.

[31] C. Tseng and S. Lee, "Design of digital fractional order differentiator using discrete sine transform," *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1-9, 2013, doi: 10.1109/APSIPA.2013.6694388.

[32] V. Britanak, "The fast DCT-IV/DST-IV computation via the MDCT," *Signal Processing*, 83(8):1803-1813, 2003, DOI:10.1016/S0165-1684(03)00109-9.

[33] R. K. Chivukula and Y. A. Reznik, "Fast computing of discrete cosine and sine transforms of types VI and VII," *Proc. SPIE 8135, Applications of Digital Image Processing XXXIV*, 8135:39-48, 2011, DOI:10.1117/12.903685.

[34] A. Al-Fayadh and H. Majid, "Singular value decomposition based classified vector quantization image compression method using discrete sine

transform," *ARPN Journal of Engineering and Applied Sciences*, 11:12883-12891, 2016.

[35] M. J. Kim and Y. L. Lee, "Discrete sine transform-based interpolation filter for video compression," *Symmetry (Basel)*, 9(11), 2017, DOI:10.3390/sym9110257.

[36] V. P. S. Naidu, M. Divya, and P. Mahalakshmi, "Multi-modal image fusion using multi-resolution discrete sine transform," *Control and Data Fusion e-Journal*, 1(2):13-26, 2017.

[37] A. P. James and B. V. Dasarathy, "Medical image fusion: a survey of the state of the art," *Elsevier*, 19:4-19, 2014.

[38] E. A. Naeem, M. M. Abd Elnaby and M. M. Hadhoud, "Chaotic image encryption in transform domains," *2009 International Conference on Computer Engineering & Systems*, pp. 71-76, 2009, DOI: 10.1109/ICCES.2009.5383309.

[39] S. C. Pei and M. H. Yeh, "The discrete fractional cosine and sine transforms," *IEEE Transactions on Signal Processing*, 49(6):1198-1207, 2001, DOI: 10.1109/78.923302.

[40] H. Zhao, Q. Ran, G. Ge, J. Ma and L. Tan, "Image Encryption Based on Random Fractional Discrete Cosine and Sine Transforms," *2009 First International Workshop on Education Technology and Computer Science*, pp. 804-808, 2009, DOI: 10.1109/ETCS.2009.183.

[41] S. M. Perera and J. Liu, "Lowest Complexity Self Recursive Radix-2 DCT II/III Algorithms," *SIAM J. Matrix Analysis and Applications*, 39(2): 664-682, 2018.

[42] S. M. Perera and J. Liu, "Complexity Reduction, Self/Completely Recursive, Radix-2 DCT I/IV Algorithms," *Elsevier Journal of Computational Applied Mathematics*, 379, 112936: 1-16 2020.

[43] T. Kailath and V. Olshevsky. "Displacement structure approach to discrete trigonometric transform based preconditioners of G.Strang and T.Chan types," *Calcolo*, 33:191-208, 1996.

[44] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, 4(3):247-269, 1999.

[45] V. Britanak, "New generalized conversion method of the MDCT and MDST coefficients in the frequency domain for arbitrary symmetric windowing function," *Digital Signal Processing*, 23:1783-1797, 2013.

[46] S. M. Perera and V. Olshevsky, "Stable, recursive and fast algorithms for discrete sine transformations having orthogonal factors," *J. Coupled System Multiscale Dynamics*, 1(3):358-371, 2013.

[47] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs," *IEEE Transactions on Signal Processing*, 56(4):1502-1521, 2008.

[48] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp. 19:297-301*, 1965.

[49] R. Yavne, "An economical method for calculating the discrete Fourier transform," in *Proceedings of the AFIPS Fall Joint Computer Conference*, 33:115-125, 1968.

[50] S. C. Chan, K. L. Ho, "Direct methods for computing discrete sinusoidal transforms," *IEEE Proceedings F (Radar and Signal Processing)*, 137(6):433-442, 1990.

[51] Z. Wang, "A fast algorithm for the discrete sine transform implemented by the fast cosine transform," *IEEE Trans. Acoust. Speech Signal Process.*, 30(5):814-815, 1982.

[52] P. Lee, F.-Y. Huang, "Restructured recursive DCT and DST algorithms," *IEEE Trans. Signal Process.*, 42 (7): 1600-1609, 1994.

[53] X. Shao and S. G. Johnson, "Type-II/III DCT/DST algorithms with reduced number of arithmetic operations," *Signal Processing*, 88:1553-1564, 2008.

[54] X. Shao and S. G. Johnson, "Type-IV DCT, DST and MDCT algorithms with reduced number of arithmetic operations," *Signal Processing*, 88(6):1313-1326, 2008.

[55] S. G. Johnson and M. Frigo, "A Modified Split-Radix FFT With Fewer Arithmetic Operations," *IEEE Trans. Signal Process.*, 55(1):111-119, 2007.

[56] Z. Wang, "On computing the Fourier and cosine transforms," *IEEE Trans. Acoust. Speech Signal Process.*, 33:1341-1344, 1985

[57] P. Yip and K. R. Rao, "Fast decimation-in-time algorithms for a family of discrete sine and cosine transforms," *Circuits, Syst., Signal Processing*, 3:387-408, 1984.

[58] O. Ersoy and N. C. Hu, "A unified approach to the fast computation of all discrete trigonmetric transforms," in *Proc. IEEE Itl. Conf. Acoust., Speech, Signal Processing*, 1843-1846, 1987.

[59] Z. Cvetkovic and M. V. Popovic, "New fast recursive algorithms for the computation of discrete cosine and sine transforms," *IEEE Trans. Signal Processing*, 40(8):2083-2086, 1992.

[60] V. Britanak, "New universal rotation-based fast computational structure for an efficient implementation of the DCT-IV/DST-IV and analysis/synthesis MDCT/MDST filter banks," *Signal Processing*, 89:2213-2232, 2009.

[61] M. Tasche and H. Zeuner, in G. Anastassiou (Ed.), "Handbook of Analytic-Computational Methods in Applied Mathematics," 357-406, Chapman and Hall/CRC press, Boca Raton 2000.

[62] N. Sharma, I. Saini, A. K. Yadav, and P. Singh, "Phase-image encryption based on 3D-Lorenz chaotic system and double random phase encoding," *3D Res*, 8:39, 2017.

[63] B. Javadi, "Optical and digital techniques for information security," Berlin: Springer, 2006.

[64] B. Javadi, et al., "Roadmap on optical security," *Journal of Optics*, 18(8): 083001, 2016.

[65] P. Refregier and B. Javadi, "Optical image encryption based on input plane and Fourier plane random encoding," *Optical Letters*, 20(7): 767-769, 1995.

[66] P. Kumar, J. Joseph, and K. Singh, "Known-plaintext attack-free double random phase-amplitude optical encryption: Vulnerability to impulse function attack," *Journal of Optics*, 14(4): 045401, 2012.

[67] P. Kumar, J. Joseph, and K. Singh, "Double random phase encoding based optical encryption systems using some linear canonical transforms: Weaknesses and countermeasures," in J. J. Healy et al. (Eds.), *Linear canonical transforms*, New York: Springer, pp. 367, 2016.

[68] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, 20(2): 130-141, 1963.