

An Efficient Multiplierless Approximation of the Fast Fourier Transform Using Sum-of-Powers-of-Two (SOPOT) Coefficients

S. C. Chan and P. M. Yiu

Abstract—This letter proposes a new multiplierless approximation of the discrete Fourier transform (DFT) called the multiplierless fast Fourier transform-like (ML-FFT) transformation. It makes use of a novel factorization to parameterize the twiddle factors in the conventional radix- 2^n or split-radix FFT algorithms as certain rotation-like matrices and approximates the associated parameters using the sum-of-powers-of-two (SOPOT) or canonical signed digits (CSD) representations. The ML-FFT converges to the DFT when the number of SOPOT terms used increases and has an arithmetic complexity of $O(N \log_2 N)$ additions, where $N = 2^m$ is the transform length. Design results show that the ML-FFT offers flexible tradeoff between arithmetic complexity and numerical accuracy in approximating the DFT.

Index Terms—Discrete Fourier transform (DFT), fast Fourier transform (FFT), radix- 2^n decimation-in-frequency (DIF), sum-of-powers-of-two (SOPOT).

I. INTRODUCTION

THE DISCRETE Fourier transform (DFT) is an important tool in digital signal processing [1]. A treasure of fast algorithms such as the Cooley–Tukey fast Fourier transform (FFT), the prime factor algorithm (PFA) FFT, and the Winograd FFT is available to compute efficiently DFT of different lengths. In this letter, a new multiplierless approximation of the DFT, called the multiplierless FFT-like transformation (ML-FFT), using the radix- 2^n FFT algorithm and the sum-of-powers-of-two (SOPOT) or canonical signed digits (CSD) representation, is proposed. Generalization to the split-radix FFT (SR-FFT) algorithm is also possible. The present work is motivated by a recent interest in the efficient realization of fast transform algorithms using either integer [6]–[8] or SOPOT [3]–[5] representations. The main objective is to provide efficient multiplierless approximation to sinusoidal transforms such as the DFT, which can be implemented efficiently either in software (say in microcontrollers with no on-chip multipliers, etc.), hardware, or very large scale integrated circuits (VLSI). Integer FFT [6] and sinusoidal transforms [7], [8] are attractive when the transform sizes are small, because the wordlength growth of the intermediate data can still be accommodated in 16–32 bits without rounding. As a result, high accuracy can

be achieved with simple integer arithmetic. However, it is very difficult to design such integer matrices when the transform length increases. Up to now, only order-8 integer FFT [8] and order-16 integer DCT [7] have been reported. The proposed multiplierless FFT-like (ML-FFT) transformation is based on the SOPOT representation of the twiddle factors in the conventional FFT algorithms using a special parameterization of the rotation-like matrices [3]. This parameterization allows us to implement both the forward and inverse transforms with the same set of SOPOT coefficients. Furthermore, by using the periodic properties of the rotation-like matrix and the twiddle factors, the number of parameters to be optimized is reduced by a factor of four and the dynamic range of the parameters is limited between zero and one, greatly improving the numerical property. Unlike integer-based transformations, multiplierless transformations with length up to 1024 and higher can be obtained without much difficulty, and they converge to their ideal counterparts when the SOPOT terms used increases. The proposed approach differs from the multiplierless DCT of [5] in the parameterization used, which has the advantages of limited dynamic range and reduced number of parameters mentioned earlier. Design results show that the proposed ML-FFT offers a flexible tradeoff between arithmetic complexity and accuracy in approximating the DFT. Another point worth mentioning is that the proposed approach is also applicable to approximate other sinusoidal transforms such as the four types of DCTs and DSTs [4] and the discrete Hartley transform (DHT) [9]. This letter is organized as follows, Section II is devoted to the proposed ML-FFT algorithm, where its principle and the parameterization of the rotation-like matrices into SOPOT coefficients are described. Section III gives the design method and several examples demonstrating its usefulness. Finally, conclusions are drawn in Section IV.

II. ML-FFT ALGORITHM

The DFT $X[k]$ of a finite-length sequence $x[n]$ of length N is

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \text{ for } k = 0, 1, \dots, N-1 \quad (2.1a)$$

where $W_N = e^{-j(2\pi/N)}$ and $j = \sqrt{-1}$. Its inverse, called the inverse DFT (IDFT), is given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \text{ for } n = 0, 1, \dots, N-1. \quad (2.1b)$$

Manuscript received August 16, 2001; revised June 10, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giovanni Ramponi.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong (e-mail: scchan@eee.hku.hk; pmyiu@eee.hku.hk).

Digital Object Identifier 10.1109/LSP.2002.803408.

If N is a power-of-two number, the DFT can be computed by the radix-2 ($n = 2, 4, 8, \dots$) or the split-radix FFT. Without loss of generality, let us consider the decimation-in-frequency (DIF) FFT algorithm. From (2.1a), it can be shown that the even-numbered frequency samples can be computed via an $(N/2)$ -point DFT of the sequence $\{x[n] + x[n + N/2] \cdot W_N^n\}$ as follows:

$$\begin{aligned} X[2r] &= \sum_{n=0}^{N-1} x[n] W_N^{n(2r)} \\ &= \sum_{n=0}^{(N/2)-1} x[n] W_N^{2nr} \\ &\quad + \sum_{n=0}^{(N/2)-1} x[n + N/2] W_N^{2r(n+(N/2))} \\ &= \sum_{n=0}^{(N/2)-1} \{x[n] + x[n + N/2]\} W_{N/2}^{nr}. \quad (2.2) \end{aligned}$$

Similarly, the odd-numbered frequency samples can be computed as an $(N/2)$ -point DFT of the sequence $\{[x(n) - x(n + N/2)] \cdot W_N^n\}$ as shown in the following:

$$\begin{aligned} X[2r + 1] &= \sum_{n=0}^{N-1} x[n] W_N^{n(2r+1)} \\ &= \sum_{n=0}^{(N/2)-1} x[n] W_N^{n(2r+1)} \\ &\quad + \sum_{n=0}^{(N/2)-1} x[n + N/2] W_N^{(2r+1)(n+(N/2))} \\ &= \sum_{n=0}^{(N/2)-1} \{x[n] - x[n + N/2]\} W_N^n \cdot W_{N/2}^{nr}. \quad (2.3) \end{aligned}$$

W_N^n is called the twiddle factor. If N is a power-of-two number, then the DFT can be reduced successively using the above decomposition to eventually $(N/2)$ two-point DFTs. The arithmetic complexity is $N \cdot \log_2 N$ complex additions and $(N/2) \cdot \log_2 N$ complex multiplications (including possible trivial multiplications). Fig. 1 shows the flow graph of the eight-point DFT computed using the DIF FFT algorithm. The proposed ML-FFT algorithm approximates the twiddle factor multiplication, which can be implemented as rotations, using the SOPOT representation and a certain parameterization of the rotation-like matrix. More precisely, let $c = (x + j \cdot y)$ be a complex number. The multiplication of c with $\exp(-j\theta)$, $p = c \cdot \exp(-j\theta)$, can be written as

$$\begin{bmatrix} p_r \\ p_i \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \mathbf{R}_\theta \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.4)$$

where

$$\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix}$$

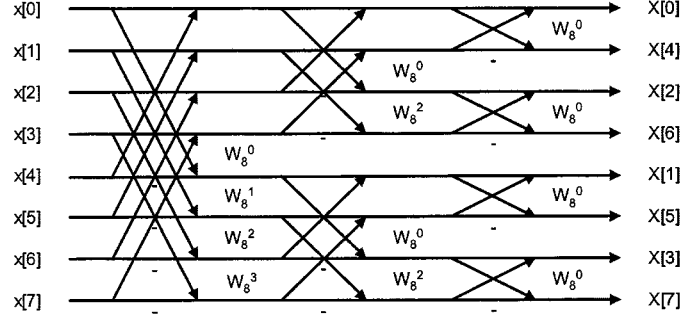


Fig. 1. Flow graph for computing an eight-point DFT using the DIF radix-2 FFT algorithm ($W_N = e^{-j(2\pi/N)}$).

is the rotation-like matrix that we are interested in and $p = (p_r + j \cdot p_i)$. The main difficulty in constructing a multiplierless transformation using the SOPOT representation of \mathbf{R}_θ is that the coefficients of the matrix transformation and its inverse cannot, in general, be expressed in terms of SOPOT coefficients. If $\cos \theta$ and $\sin \theta$ in (2.4) are expressed directly in terms of SOPOT coefficients, say α and β , then the inverse of

$$\tilde{\mathbf{R}}_\theta = \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix}$$

is

$$\tilde{\mathbf{R}}_\theta^{-1} = \frac{1}{\sqrt{\alpha^2 + \beta^2}} \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix}.$$

As α and β are SOPOT coefficients, the term $(\alpha^2 + \beta^2)^{-1/2}$ cannot, in general, be expressed as SOPOT coefficients. The basic idea of the proposed ML-FFT is to parameterize \mathbf{R}_θ and its inverse as follows:

$$\begin{aligned} \mathbf{R}_\theta &= \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix} = \mathbf{R}_\theta^{-1} \\ &= \begin{bmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ \sin \theta & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \tan(\theta/2) \\ 0 & -1 \end{bmatrix}. \quad (2.5) \end{aligned}$$

Since these factorizations of \mathbf{R}_θ and \mathbf{R}_θ^{-1} involve the same set of coefficients, i.e., $\sin \theta$ and $\tan(\theta/2)$, they can be directly quantized to SOPOT coefficients to form

$$\mathbf{R}_\theta \approx \mathbf{S}_\theta = \begin{bmatrix} 1 & -\beta_\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \alpha_\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta_\theta \\ 0 & -1 \end{bmatrix} \quad (2.6)$$

where α_θ and β_θ are, respectively, SOPOT approximations to $\sin \theta$ and $\tan(\theta/2)$ having the form $\alpha_\theta = \sum_{k=1}^t a_k 2^{b_k}$, where $a_k \in \{-1, 1\}$, $b_k \in \{-r, \dots, -1, 0, 1, \dots, r\}$; r is the range of the coefficients, and t is the number of terms used in each coefficient. In practice, it is usually limited to a small number so that the multiplication can be implemented with limited number of additions and shifts. Since the coefficient multiplications are now replaced by a limited number of additions, the transformation can be realized efficiently without multipliers. Interested readers are referred to [3]–[5], [10] for more references on efficient realization of digital filters using SOPOT coefficients. Direct application of (2.6) might lead to large dynamic range for $\tan(\theta/2)$ when θ is close to π . Although the factorization of \mathbf{R}_θ in (2.5) is unique for a given value of θ , the periodic property

of \mathbf{R}_θ can be utilized to construct slight variation of this factorization so as to reduce the dynamic range of the coefficients, as well as the number of parameters to be optimized. More precisely, it can be shown that the quantities \mathbf{R}_θ , $\mathbf{R}_{\theta+\pi/2}$, $\mathbf{R}_{\pi-\theta}$, $\mathbf{R}_{\pi/2-\theta}$, and $\mathbf{R}_{\pi+\theta}$ can be expressed completely in terms of any one of the following quantities:

$$\text{for } \frac{\pi}{4} \leq \theta < \frac{\pi}{2}, \quad \mathbf{R}_{\pi/2-\theta} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \mathbf{R}_\theta \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2.7a)$$

$$\text{for } \frac{\pi}{2} \leq \theta < \frac{3\pi}{4}, \quad \mathbf{R}_{\theta+\pi/2} = \mathbf{R}_\theta \cdot \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (2.7b)$$

$$\text{for } \frac{3\pi}{4} \leq \theta < \pi, \quad \mathbf{R}_{\pi-\theta} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \mathbf{R}_\theta \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.7c)$$

$$\text{for } \theta \geq \pi, \quad \mathbf{R}_{\pi+\theta} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \mathbf{R}_\theta. \quad (2.7d)$$

Using these relationships, it is possible to limit θ within the range $[0, \pi/4]$. Thus, the dynamic range of the parameters to be approximated by the SOPOT representation is always limited to the range $[0, 1]$, and the number of parameters to be optimized can be reduced by a factor of four. The former allows us to limit the dynamic range of the parameters even if the transform length is very large, resulting in better numerical property. The approximations $\mathbf{S}_{\theta+\pi/2}$, $\mathbf{S}_{\pi-\theta}$, $\mathbf{S}_{\pi/2-\theta}$, and $\mathbf{S}_{\pi+\theta}$ of their respective rotation-like matrices can be generated from \mathbf{S}_θ by replacing \mathbf{R} with \mathbf{S} in (2.7). As all the twiddle factors in the radix-2 DIF FFT algorithm are just a subset of those in the first stage, there are altogether $N/2$ twiddle factors, including W_N^0 . Using the proposed parameterization, which has two parameters per rotation, there are approximately $2 \times (N/8)$ parameters in an N -point ML-FFT. The SOPOT twiddle factors for the radix-4, radix-8, and split-radix ML-FFTs can be generated similarly.

III. DESIGN METHOD AND EXAMPLES

The random search algorithm proposed in [4] is used to perform the discrete optimization. More precisely, a random vector with all its elements bounded by ± 1 is first multiplied by a scaling factor s , and is added to the parameter vector containing the real-valued α_θ and β_θ . It is then quantized to the nearest SOPOT coefficients. The objective function is then evaluated for this SOPOT candidate. The one with the best performance at a given number of additions is recorded. The search continues until the maximum allowed number of trials is exceeded. The scale factor controls the size of the neighborhood to be searched. A number of solutions with different tradeoffs between implementation complexity and performance are then obtained. Without loss of generality, the Frobenius norm of the derivation of the ML-FFT from its ideal counterpart is used as the objective function

$$E_{\text{OBJ}} = \sqrt{\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} (f_{j,k} - \hat{f}_{j,k}) \cdot \overline{(f_{j,k} - \hat{f}_{j,k})}} \quad (3.1)$$

where $\hat{f}_{j,k}$ and $f_{j,k}$ are the (j, k) th entries of the ML-FFT matrix and its ideal counterpart, respectively. Table I shows

TABLE I
FROBENIUS NORM OF THE ERROR MATRIX IN APPROXIMATING THE DFT BY THE ML-FFT USING DIRECT QUANTIZATION

Number of channels of the multiplier-less radix-2 FFT				
	32-channel	64-channel	128-channel	256-channel
8 terms	-107 dB	-98 dB	-90 dB	-81 dB
7 terms	-95 dB	-85 dB	-76 dB	-67 dB
6 terms	-69 dB	-63 dB	-57 dB	-50 dB
5-terms	-53 dB	-45 dB	-39 dB	-31 dB
4-terms	-20 dB	-14 dB	-8 dB	-2 dB

TABLE II
THE OBJECTIVE FUNCTION, THE FROBENIUS NORM, THE AVERAGE NUMBER OF SOPOT TERMS, AND THE COMPUTATIONAL COMPLEXITY FOR 8-, 16-, 32-, 64-, AND 128-CHANNEL REAL-VALUED FFT AND ML-FFT (MULT: MULTIPLICATIONS; ADD.: ADDITIONS)

	Frobenius Norm	Average number of SOPOT terms	Computational complexity using real-valued radix-2 FFT		Computational complexity using real-valued radix-4 FFT		Computational complexity using SOPOT coefficients	
			MULT.	ADD.	MULT.	ADD.	ADD. for radix-2	ADD. for radix-4
8-channel	-64 dB	4.500	20	58	-	-	84	-
16-channel	-53 dB	4.250	68	162	36	146	252	224
32-channel	-45 dB	4.375	196	418	-	-	756	-
64-channel	-43 dB	4.738	516	1026	324	930	2094	1920
128-channel	-41 dB	4.938	1284	2434	-	-	6727	-

E_{OBJ} (in decibels) obtained by directly rounding the coefficients to the SOPOT representations without performing any optimization. It can be seen that the precision of the ML-FFT is directly related to the number of SOPOT terms used, which in turn affects its arithmetic complexity. Table II shows E_{OBJ} , the average SOPOT terms, and the computational complexities for the 8-, 16-, 32-, 64-, and 128-channel ML-FFT obtained using the random search algorithm. The numbers of SOPOT terms are limited to five to simplify hardware implementation, though it is not necessary, say, in software implementation. The arithmetic complexities of the radix-4 FFT and ML-FFT are also shown as a comparison. It can be seen that the number of additions required for the ML-FFT is approximately two times that of the radix-2 FFT algorithm for an averaged number of SOPOT terms per coefficient of five. In general, if S is the average number of the SOPOT terms per coefficient, then from (2.6), the arithmetic complexity of the radix- 2^n ML-FFT algorithm derived from a radix- 2^n FFT algorithm with an arithmetic complexity of $M_{\text{radix-}2^n}(N)$ complex multiplications and $A_{\text{radix-}2^n}(N)$ complex additions is approximately

$$A_{\text{ML-radix-}2^n}(N) = 3S \cdot M_{\text{radix-}2^n}(N) + 2 \cdot A_{\text{radix-}2^n}(N)$$

real additions. For simplicity, we have ignored those trivial multiplications in the real-valued algorithm. As an example, the radix-2 FFT algorithm has an arithmetic complexity of $M_{\text{radix-}2}(N) = (N/2) \cdot \log_2 N$ and $A_{\text{radix-}2}(N) = N \cdot \log_2 N$. Then, the corresponding complexity of the ML-FFT algorithm is $3(N/2) \cdot \log_2 N + 2N \cdot \log_2 N$ real additions. Further saving can be obtained by using the radix-4 (as illustrated in

TABLE III
THE SOPOT COEFFICIENTS OF THE 32-CHANNEL ML-FFT

R_θ	$\alpha_\theta (\sin \theta)$	$\beta_\theta (\tan(\theta / 2))$
$R_{1/32}$	$2^{-10} + 2^{-8} + 2^{-5} + 2^{-4}$	$-2^{-13} + 2^{-7} - 2^{-4} + 2^{-2}$
$R_{2/32}$	$-2^{-11} - 2^{-8} + 2^{-6} - 2^{-4} + 2^{-2}$	$2^{-7} + 2^{-3} + 2^{-2}$
$R_{3/32}$	$2^{-11} - 2^{-9} - 2^{-7} + 2^{-4} + 2^{-2}$	$-2^{-12} + 2^{-10} - 2^{-7} + 2^{-4} + 2^{-1}$
$R_{4/32}$	$2^{-7} + 2^{-5} + 2^{-3} + 2^{-2}$	$2^{-8} - 2^{-6} - 2^{-5} + 2^{-2} + 2^{-1}$

TABLE IV
THE SOPOT COEFFICIENTS OF THE 128-CHANNEL ML-FFT

θ	$\alpha_\theta (\sin \theta)$	$\beta_\theta (\tan(\theta / 2))$
$R_{1/128}$	$2^{-16} + 2^{-13} + 2^{-10} + 2^{-7} + 2^{-6}$	$-2^{-18} + 2^{-12} + 2^{-9} - 2^{-6} + 2^{-4}$
$R_{2/128}$	$2^{-14} + 2^{-12} + 2^{-9} + 2^{-6} + 2^{-5}$	$2^{-13} + 2^{-12} + 2^{-8} + 2^{-5} + 2^{-4}$
$R_{3/128}$	$2^{-15} - 2^{-11} - 2^{-8} + 2^{-6} + 2^{-4}$	$2^{-12} - 2^{-9} + 2^{-7} + 2^{-6} + 2^{-3}$
$R_{4/128}$	$-2^{-13} + 2^{-10} + 2^{-8} - 2^{-5} + 2^{-3}$	$2^{-16} - 2^{-12} + 2^{-7} - 2^{-4} + 2^{-2}$
$R_{5/128}$	$-2^{-18} + 2^{-14} + 2^{-12} - 2^{-9} + 2^{-3}$	$2^{-14} - 2^{-12} + 2^{-10} - 2^{-7} + 2^{-2}$
$R_{6/128}$	$2^{-17} - 2^{-13} - 2^{-7} + 2^{-5} + 2^{-3}$	$2^{-12} + 2^{-10} + 2^{-7} + 2^{-5} + 2^{-2}$
$R_{7/128}$	$-2^{-12} + 2^{-9} - 2^{-6} + 2^{-4} + 2^{-3}$	$2^{-10} - 2^{-7} + 2^{-5} + 2^{-4} + 2^{-2}$
$R_{8/128}$	$-2^{-12} - 2^{-8} + 2^{-6} - 2^{-4} + 2^{-2}$	$-2^{-20} - 2^{-13} + 2^{-7} - 2^{-3} + 2^{-1}$
$R_{9/128}$	$-2^{-15} - 2^{-9} + 2^{-7} - 2^{-5} + 2^{-2}$	$-2^{-13} - 2^{-9} - 2^{-7} - 2^{-4} + 2^{-1}$
$R_{10/128}$	$-2^{-19} + 2^{-11} + 2^{-2}$	$-2^{-12} + 2^{-10} + 2^{-9} - 2^{-5} + 2^{-1}$
$R_{11/128}$	$-2^{-13} - 2^{-11} - 2^{-8} + 2^{-5} + 2^{-2}$	$-2^{-15} + 2^{-11} - 2^{-9} + 2^{-6} + 2^{-1}$
$R_{12/128}$	$2^{-11} - 2^{-9} - 2^{-7} + 2^{-4} + 2^{-2}$	$-2^{-14} + 2^{-10} - 2^{-7} + 2^{-4} + 2^{-1}$
$R_{13/128}$	$2^{-12} + 2^{-9} + 2^{-6} + 2^{-4} + 2^{-2}$	$-2^{-17} + 2^{-9} - 2^{-5} + 2^{-3} + 2^{-1}$
$R_{14/128}$	$2^{-12} - 2^{-9} - 2^{-6} - 2^{-3} + 2^{-1}$	$-2^{-12} + 2^{-9} + 2^{-7} + 2^{-3} + 2^{-1}$
$R_{15/128}$	$2^{-10} + 2^{-9} + 2^{-7} + 2^{-3} + 2^{-2}$	$-2^{-12} - 2^{-6} + 2^{-4} + 2^{-3} + 2^{-1}$
$R_{16/128}$	$2^{-12} + 2^{-7} + 2^{-5} + 2^{-3} + 2^{-2}$	$2^{-8} - 2^{-6} - 2^{-5} + 2^{-2} + 2^{-1}$

Table II) and the SR-FFT algorithms. The details are omitted due to page length limitations. For illustration purposes, the parameters of the proposed 32- and 128-channel radix-2 DIF ML-FFT are also shown in Tables III and IV, respectively. Finally, it is noted from Table II that fewer than five average number of SOPOT terms per coefficient are needed to give a E_{OBJ} below -40 dB. By increasing the number of SOPOT terms, the error between the ML-FFT and the ideal DFT can be reduced to an arbitrary small value. This allows one to choose an appropriate tradeoff between implementation complexity and numerical accuracy required in a given application. For example, in DFT filter bank, only the frequency characteristics of the final filter banks are of major concern. Therefore, a lower average SOPOT term of say two to three will suffice for a variety of applications [3], [4]. As an illustration, the frequency response of the 32-channel ML-FFT mentioned earlier in Table III is shown in Fig. 2. Its frequency response is almost visually identical to the ideal DFT. On the other hand, if the ML-FFT is used to compute circular convolutions with high accuracy, then a larger number of SOPOT terms is needed.

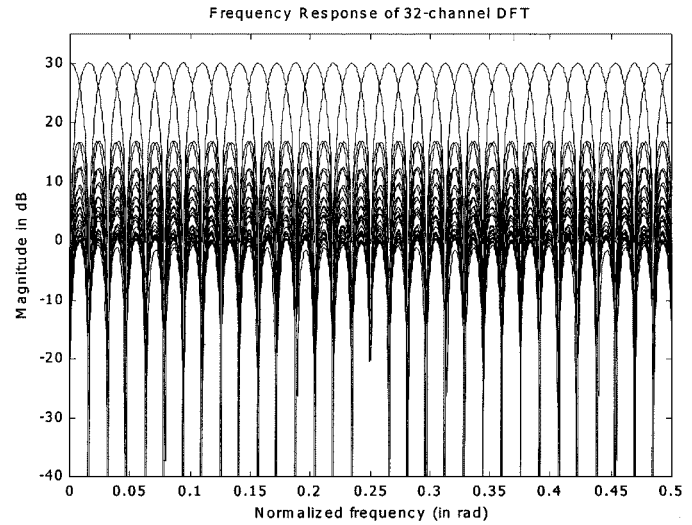


Fig. 2. Frequency response of the 32-channel ML-FFT using radix 2 DIF FFT algorithm.

IV. CONCLUSION

A new multiplierless approximation of the FFT called the multiplierless FFT-like transformation is proposed. It parameterizes the twiddle factors in the conventional radix- 2^m or split-radix FFT algorithms as certain rotation-like matrices and approximates the associated parameters using the SOPOT or canonical signed digits representations. The ML-FFT converges to the FFT when the number of SOPOT terms used increases and has an arithmetic complexity of $O(N \log_2 N)$ additions, where $N = 2^m$ is the transform length. Design results show that the ML-FFT offers flexible tradeoff between arithmetic complexity and numerical accuracy in approximating the FFT.

REFERENCES

- [1] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [3] S. C. Chan, W. Liu, and K. L. Ho, "Perfect reconstruction modulated filter banks with sum-of-powers-of-two coefficients," in *Proc. ISCAS*, Geneva, Switzerland, May 28–31, 2000.
- [4] S. C. Chan and P. M. Yiu, "Multiplier-less discrete sinusoidal and lapped transforms using sum-of-powers-of-two (SOPOT) coefficients," in *Proc. ISCAS*, Sydney, Australia, May 6–9, 2001.
- [5] T. D. Tran, "A fast multiplier-less block transform for image and video compression," in *Proc. ICIP*, vol. 3, 1999, pp. 822–826.
- [6] S. C. Pei and J. J. Ding, "The integer transforms analogous to discrete trigonometric transforms," *IEEE Trans. Signal Processing*, vol. 48, pp. 3345–3364, Dec. 2000.
- [7] W. K. Cham and Y. T. Chan, "An order-16 integer cosine transform," *IEEE Trans. Signal Processing*, vol. 39, pp. 1205–1208, May 1991.
- [8] A. Nallanathan, S. C. Chan, and T. S. Ng, "Lapped orthogonal transform with integer coefficients," in *Proc. 42nd Midwest Symp. Circuits and Systems*, vol. 2, 2000, pp. 1065–1068.
- [9] S. C. Chan, "Multiplier-less approximation of DFT and DHT," Dept. Elect. Electron. Eng., The Univ. Hong Kong, Int. Rep., May 2000.
- [10] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete power-of-two coefficient space," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-31, pp. 583–591, June 1983.