

# A SHORT SURVEY ON ARITHMETIC TRANSFORMS AND THE ARITHMETIC HARTLEY TRANSFORM

Renato José de Sobral Cintra and Hélio Magalhães de Oliveira

**Abstract** - Arithmetic complexity has a main role in the performance of algorithms for spectrum evaluation. Arithmetic transform theory offers a method for computing trigonometrical transforms with minimal number of multiplications. In this paper, the proposed algorithms for the arithmetic Fourier transform are surveyed. A new arithmetic transform for computing the discrete Hartley transform is introduced: the Arithmetic Hartley transform. The interpolation process as the key to the arithmetic transform theory is also examined.

**Keywords:** Arithmetic transforms, discrete transforms, Fourier series, VLSI implementations.

**Resumo** - A complexidade aritmética ocupa um papel de destaque no desempenho de algoritmos para o cálculo de espectros. As transformadas aritméticas proporcionam um método para o cálculo de transformadas trigonométricas, minimizando-se o número de operações de multiplicação. Neste artigo, os algoritmos existentes para a transformada aritmética de Fourier são discutidos. Uma nova transformada aritmética para o cálculo da transformada discreta de Hartley é introduzida: a transformada aritmética de Hartley. O processo de interpolação é examinado com o ponto crucial das transformadas aritméticas.

**Palavras-chave:** Transformadas aritméticas, transformadas discretas, série de Fourier, implementações em VLSI.

## 1. INTRODUCTION AND HISTORICAL BACKGROUND

Despite the existence of fast algorithms for discrete transforms (e.g., fast Fourier transform, FFT), it is well known that the number of multiplications can significantly increase their computational (arithmetic) complexity. Even today, the multiplication operation consumes much more time than addition or subtraction. Table 1 brings the clock count of some mathematical operations as implemented for the Intel Pentium™ processor. Observe that multiplications and divisions can be by far more time demanding than additions, for instance. Sine and cosine function costs are also shown.

This fact stimulated the research on discrete transform algorithms that minimize the number of multiplications. The Bhatnagar's algorithm [1a], which uses Ramanujan numbers

Operation	Clock count
add	1-3
sub	1-3
fadd	1-7
fsub	1-7
mul (unsigned)	10-11
mul (signed)	10-11
div (unsigned)	17-41
div (signed)	22-46
fdiv	39
sin, cos	17-137

**Table 1.** Clock count for some arithmetic instructions carried on a Pentium™ processor. See "The Pentium Processor" by J. L. Antonakos for detailed data.

to eliminate multiplications (however, the choice of the transform blocklength is rather limited), is an example. Parallel to this, approximation approaches, which perform a trade-off between accuracy and computational complexity, have been proposed [2a, 3a, 4a]

Arithmetic transforms emerged in this framework as an algorithm for spectrum evaluation, aiming the elimination of multiplications. Thus, it would offer a lower computational complexity. The theory of arithmetic transform is essentially based on Möbius function theorems [5a], offering only trivial multiplications, i.e., multiplications by  $\{-1, 0, 1\}$ . Therefore, only addition operations (except for multiplications by scale factors) are left to computation. Beyond the computational attractiveness, arithmetic transforms turned out to be naturally suited for parallel processing and VLSI design [6, 18].

The very beginning of research on arithmetic transforms dates back to 1903 when the German mathematician Ernest Heinrich Bruns<sup>1</sup> published the *Grundlinien des wissenschaftlichen Rechnens* [3], the seminal work in this field. In spite of that, the technique remained unnoticed even among mathematicians for a long time. Forty-two years later, in Baltimore, U.S.A., the Hungarian Aurel Freidrich Wintner<sup>2</sup>, privately published a monograph entitled *An Arithmetical Approach to Ordinary Fourier Series*. This monograph presented an arithmetic method using Möbius function to calculate the Fourier series of even periodic functions.

After Wintner's monograph, the theory entered again in "hibernation" state. Not before 1988, Dr. Donald W. Tufts and Dr. Angaraih G. Sadasiv, independently, had reinvented

The authors are with the Communications Research Group - CODEC, Department of Electronics and Systems, Federal University of Pernambuco, Recife, Brazil. Emails: rjsc@ee.ufpe.br, hmo@ufpe.br. Review coordinated by Renato Baldini Filho and Jaime Portugueseis (Area Editors). Manuscript received May/19/2003, reviewed Aug/26/2003, Mar/18/2004, accepted Aug/04/2004.

<sup>1</sup>Bruns (1848-1919) got a doctorate in 1871 under supervision of Weierstrass and Kummer.

<sup>2</sup>A curious fact: Wintner was born in April 8th 1903 in Budapest, the same year Bruns had published the *Grundlinien*. Wintner died on January 15th 1958 in Baltimore.

Wintner's arithmetical procedure, reawaking the arithmetic transform.

In the quest to implement it, two other researchers played an important role: Dr. Oved Shisha of the U.R.I. Department of Mathematics and Dr. Charles Rader of Lincoln Laboratories. They were aware of Wintner's monograph and helped Tufts in many discussions. In 1988 *The Arithmetic Fourier Transform* by Tufts and Sadasiv was published in IEEE Acoustic, Speech, and Signal Processing (ASSP) Magazine [6].

Another breakthrough came in early 1990s when Emeritus Professor Dr. Irving S. Reed entered in scene. Although Dr. Reed is better recognized for his work on coding theory — since he is the main originator of the widely used Reed-Muller (1954) and Reed-Solomon (1964) codes — his interests were definitely not limited to codes. Author of hundreds of publications, Dr. Reed made important contributions to the area of signal processing. Specifically on arithmetic transforms, in 1990 Reed, Tufts and co-workers provided two fundamental contributions [15, 26].

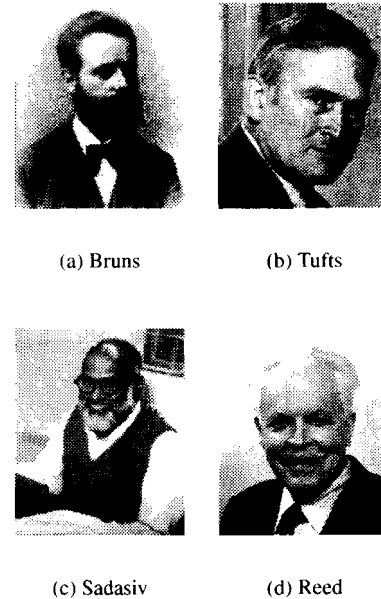
In the light of [15], a reformulated version of Tufts-Sadasiv approach, the arithmetic Fourier transform (AFT) algorithm was able to encompass a larger class of signals and to compute Fourier series of odd periodic functions as well as even periodic ones.

The publication of the 1992 *A VLSI Architecture for Simplified Arithmetic Fourier Transform Algorithm* by Dr. Reed and collaborators in the IEEE Transactions on ASSP [26] was another crucial slash on the subject. Indeed, that paper was previously presented at the *International Conference on Application Specific Array Processors* held in Princeton. However, the 1992 publication reached a vastly larger public, since it was published in a major journal. The new method, an enhancement of the last proposed algorithm [15], was redesigned to have a more balanced and computationally efficient performance. As a matter of fact, Reed *et al.* proved that the newly proposed algorithm was identical to Bruns' original method.

When the AFT was introduced, some concerns on the feasibility of the AFT were pointed out [10]. The main issue dealt with the number of samples required by the algorithm. However, later studies showed that the use of interpolation techniques on a sub-sampled set (e.g., zero- and first-order interpolation) could overcome these difficulties [11].

The conversion of the standard 1-D AFT into 2-D versions was just a matter of time. Many variants were proposed following the same guidelines of the 1-D case [8, 20, 39, 43, 44, 12]. Further research was carried out seeking different implementations of the AFT. An alternative method [32] proposed a "Möbius-function-free AFT". Iterative [30] and adaptative approaches [16] were also examined. In spite of that, the most popular presentations of the AFT are still those found in [15, 26].

Although the main and original motivation of the arithmetic algorithm was the computation of the Fourier Transform, further generalizations were performed and the arithmetic approach was utilized to calculate other transforms. Dr. Luc Knockaert of Department of Information Technology at Ghent University, Belgium, amplified the Bruns procedure,



**Figure 1.** Some important people in the history of the arithmetic transform algorithm (see the text).

defining a generalized Möbius transform [35, 38]. Moreover, four versions of the cosine transform was shaped in the arithmetic transform formalism [40].

Further generalization came in early 2000s with the definition of the Arithmetic Hartley Transform (AHT) [48, 47]. These works constituted an effort to make arithmetical procedure applicable for the computation of trigonometrical transforms, other than Fourier transform. In particular the AHT computes the discrete Hartley transform<sup>3</sup>: the real, symmetric, Fourier-like discrete transform defined in 1983 by Emeritus Professor Ronald Newbold Bracewell in *The Discrete Hartley Transform*, an article published in the Journal of Optical Society of America.

In 1988 and then the technological state-of-art was dramatically different from that Bruns and Wintner found. Computational facilities and digital signal processing integrated circuits made possible AFT to leave theoretical constructs and reach practical implementations. Since its inception in engineering, the AFT was recognized as tool to be implemented with VLSI techniques. Tufts himself had observed that AFT could be naturally implemented in VLSI architectures [6]. Implementations were proposed in [17, 21, 27, 22, 24, 29, 31, 18, 26, 43, 14, 19, 23, 36]. Initial applications of the AFT took place in several areas: pattern matching techniques [28], measurement and instrumentation [37, 41], auxiliary tool for computation of  $z$ -transform [34, 33], and imaging [13].

This paper is organized in two parts. In section 2, the

<sup>3</sup>Ralph Vinton Lyon Hartley (1888-1970) introduced his real integral transform in a 1942 paper published in the *Proceedings of I.R.E.* The Hartley transform relates a pair of signals  $f(t) \longleftrightarrow F(\nu)$  by

$$F(\nu) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)(\cos(\nu t) + \sin(\nu t))dt,$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\nu)(\cos(\nu t) + \sin(\nu t))d\nu.$$

mathematical evolution of the Arithmetic Fourier Transform is outlined. In section 3, a summary of the major results on the Arithmetic Hartley Transform is shown. Interpolation issues are addressed and many points of the AFT were clarified, particularly the zero-order approximation.

## 2. THE ARITHMETIC FOURIER TRANSFORM

Throughout this section, the three major breakthroughs of the arithmetic Fourier transform technique are presented. With emphasis on the theoretical groundwork, the AFT algorithms devised by Tufts, Sadasiv, Reed *et alli* are briefly surveyed.

Before describing the algorithms, it is convenient to call attention to some useful preliminary results. In this work,  $k_1|k_2$  denotes that  $k_1$  is a divisor of  $k_2$ ;  $\lfloor \cdot \rfloor$  is the floor function and  $\lceil \cdot \rceil$  is the nearest integer function.

**Lemma 2.1** *Let  $k$ ,  $k'$  and  $m$  be integers.*

$$\sum_{m=0}^{k-1} \cos\left(2\pi m \frac{k'}{k}\right) = \begin{cases} k & \text{if } k|k', \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and

$$\sum_{m=0}^{k-1} \sin\left(2\pi m \frac{k'}{k}\right) = 0. \quad (2)$$

*Proof:* Consider the expression  $\sum_{m=0}^{k-1} \left(e^{2\pi j \frac{k'}{k}}\right)^m$ . When  $k|k'$ , yields

$$\sum_{m=0}^{k-1} \left(e^{2\pi j \frac{k'}{k}}\right)^m = \sum_{m=0}^{k-1} 1 = k.$$

Otherwise,

$$\sum_{m=0}^{k-1} \left(e^{2\pi j \frac{k'}{k}}\right)^m = \frac{1 - e^{j2\pi k'}}{1 - e^{j2\pi \frac{k'}{k}}} = 0.$$

Therefore,

$$\sum_{m=0}^{k-1} e^{2\pi j m \frac{k'}{k}} = \begin{cases} k & \text{if } k|k', \\ 0 & \text{otherwise.} \end{cases}$$

Taking real and imaginary parts ends the proof. ■

**Definition 2.1 (Möbius  $\mu$ -function)** *For a positive integer  $n$ ,*

$$\mu(n) \triangleq \begin{cases} 1 & \text{if } n = 1, \\ (-1)^r & \text{if } n = \prod_{i=1}^r p_i, p_i \text{ distinct primes,} \\ 0 & \text{if } p^2|n \text{ for some prime } p. \end{cases} \quad (3)$$

An interesting lemma using the  $\mu$ -function is stated below.

**Lemma 2.2**

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & \text{if } n = 1, \\ 0 & \text{if } n > 1. \end{cases} \quad (4)$$

**Theorem 2.1 (Möbius Inversion Formula for Finite Series)**

*Let  $n$  be a positive integer and  $f_n$  a non-null sequence for  $1 \leq n \leq N$  and null for  $n > N$ . If*

$$g_n = \sum_{k=1}^{\lfloor N/n \rfloor} f_{kn}, \quad (5)$$

then

$$f_n = \sum_{m=1}^{\lfloor N/n \rfloor} \mu(m) g_{mn}. \quad (6)$$

This is the finite version of the Möbius inversion formula [5a]. A proof can be found in [15]. ■

### 2.1 TUFTS-SADASIV APPROACH

Consider a real even periodic function expressed by its Fourier series, as seen below:

$$v(t) = \sum_{k=1}^{\infty} v_k(t). \quad (7)$$

The components  $v_k(t)$  represent the harmonics of  $v(t)$ , given by:

$$v_k(t) = a_k \cdot \cos(2\pi kt), \quad (8)$$

where  $a_k$  is the amplitude of the  $k$ th harmonic.

It was assumed, without loss of generality, that  $v(t)$  had unitary period and null mean ( $a_0 = 0$ ). Furthermore, consider the  $N$  first harmonics as the only significant ones, in such a way that  $v_k(t) = 0$ , for  $k > N$  (bandlimited approximation). Thus the summation of Equation (7) might be constrained to  $N$  terms.

**Definition 2.2** *The  $n$ th average is defined by*

$$S_n(t) \triangleq \frac{1}{n} \sum_{m=0}^{n-1} v\left(t - \frac{m}{n}\right), \quad (9)$$

for  $n = 1, 2, \dots, N$ .  $S_n(t)$  is null for  $n > N$ . ■

After an application of equations (7) and (8) into 9, it yielded:

$$\begin{aligned} S_n(t) &= \frac{1}{n} \sum_{m=0}^{n-1} v\left(t - \frac{m}{n}\right) \\ &= \frac{1}{n} \sum_{m=0}^{n-1} \sum_{k=1}^{\infty} a_k \cos\left(2\pi kt - 2\pi k \frac{m}{n}\right) \\ &= \frac{1}{n} \sum_{k=1}^{\infty} a_k \sum_{m=0}^{n-1} \left( \cos(2\pi kt) \cos\left(2\pi k \frac{m}{n}\right) \right. \\ &\quad \left. - \sin(2\pi kt) \sin\left(2\pi k \frac{m}{n}\right) \right) \\ &= \frac{1}{n} \sum_{k=1}^{\infty} a_k \cos(2\pi kt) \cdot \left\{ \begin{array}{l} n \text{ if } n|k, \\ 0 \text{ otherwise} \end{array} \right\} \\ &= \sum_{n|k} v_k(t) = \sum_{m=1}^{\infty} v_{mn}(t), \quad n = 1, \dots, N. \end{aligned} \quad (10)$$

Proceeding that way, the  $n$ th average could be written in terms of the harmonics of  $v(t)$ , instead of its samples (Definition 2.2). Since we assumed  $v_n(t) = 0, n > N$ , only the first  $\lfloor N/n \rfloor$  terms of Equation (10) might possibly be nonnull.

As a consequence the task was to invert Equation (10). Doing so, the harmonics could be expressed in terms of the averages,  $S_n(t)$ , which were derived from the samples of the signal  $v(t)$ . The inversion was accomplished by invoking the Möbius inversion formula.

**Theorem 2.2** *The harmonics of  $v(t)$  can be obtained by:*

$$v_k(t) = \sum_{m=1}^{\infty} \mu(m) S_{mk}(t), \quad \forall k = 1, \dots, N. \quad (11)$$

*Proof:* Some manipulation is needed. Substituting Equation (10) into Equation (11), it yields

$$\sum_{m=1}^{\infty} \mu(m) S_{mk}(t) = \sum_{m=1}^{\infty} \mu(m) \sum_{n=1}^{\infty} v_{kmn}(t). \quad (12)$$

Now it is the tricky part of the proof.

$$\begin{aligned} \sum_{m=1}^{\infty} \mu(m) \sum_{n=1}^{\infty} v_{kmn}(t) &= \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \mu(m) v_{kmn}(t) \\ &= \sum_{j=1}^{\infty} v_j(t) \left( \sum_{m|j} \mu(m) \right). \end{aligned} \quad (13)$$

According to Lemma 2.2, the inner summation can only be null if  $j/k = 1$ . In other words, the term  $v_k(t)$  is the only survivor of the outer summation and the proof is completed. ■

The following aspects of the Reed-Tufts algorithm could be highlighted [6]:

- This initial version of the AFT had a strong constraint: it could only handle even signals;
- All computations were performed using only additions (except for few multiplications due to scaling);
- The algorithm architecture was suitable for parallel processing, since each average was computed independently from the others;
- The arithmetic transform theory was based on Fourier series, instead of the discrete transform itself.

## 2.2 REED-TUFTS APPROACH

Presented by Reed *et al.* in 1990 [15], this algorithm is a generalization of Tuft-Sadasiv method. The main constraint of the latter procedure (handling only with even signals) was removed, opening path for the computation of all Fourier series coefficient of periodic functions.

Let  $v(t)$  be a real  $T$ -periodic function, whose  $N$ -term finite Fourier series is given by

$$\begin{aligned} v(t) &= a_0 + \sum_{n=1}^N a_n \cos\left(\frac{2\pi nt}{T}\right) \\ &+ \sum_{n=1}^N b_n \sin\left(\frac{2\pi nt}{T}\right), \end{aligned} \quad (14)$$

where  $a_0$  is the mean value of  $v(t)$ . The even and odd coefficients of the Fourier series are  $a_n$  and  $b_n$ , respectively.

Let  $\bar{v}(t)$  denote the signal  $v(t)$  removed of its mean value  $a_0$ . Consequently,

$$\begin{aligned} \bar{v}(t) &= v(t) - a_0 \\ &= \sum_{n=1}^N a_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^N b_n \sin\left(\frac{2\pi nt}{T}\right). \end{aligned} \quad (15)$$

A delay (shift) of  $\alpha T$  in  $\bar{v}(t)$  led to the following:

$$\begin{aligned} \bar{v}(t + \alpha T) &= \sum_{n=1}^N a_n \cos\left(2\pi n\left(\frac{t}{T} + \alpha\right)\right) \\ &+ \sum_{n=1}^N b_n \sin\left(2\pi n\left(\frac{t}{T} + \alpha\right)\right) \\ &= \sum_{n=1}^N c_n(\alpha) \cos\left(2\pi n\frac{t}{T}\right) \\ &+ \sum_{n=1}^N d_n(\alpha) \sin\left(2\pi n\frac{t}{T}\right), \end{aligned} \quad (16)$$

where  $-1 < \alpha < 1$  and

$$c_n(\alpha) = a_n \cos(2\pi n\alpha) + b_n \sin(2\pi n\alpha), \quad (17)$$

$$d_n(\alpha) = -a_n \sin(2\pi n\alpha) + b_n \cos(2\pi n\alpha). \quad (18)$$

In the sequel, the computation of the Fourier coefficients  $a_n$  and  $b_n$  based on  $c_n(\alpha)$  is outlined. Meanwhile, the formula for the  $n$ th average (Tufts-Sadasiv) was updated by the next definition.

**Definition 2.3** *The  $n$ th average is given by*

$$S_n(\alpha) \triangleq \frac{1}{n} \sum_{m=0}^{n-1} \bar{v}\left(\frac{m}{n}T + \alpha T\right), \quad (19)$$

where  $-1 < \alpha < 1$ .

Now the quantities  $c_n(\alpha)$  could be related to the averages, according to the following Theorem.

**Theorem 2.3** *The coefficients  $c_n(\alpha)$  are computed via Möbius inversion formula for finite series and are expressed by*

$$c_n(\alpha) = \sum_{l=1}^{\lfloor N/n \rfloor} \mu(l) S_{ln}(\alpha). \quad (20)$$

*Proof:* Substituting the result of Equation (16) into Equation (19):

$$\begin{aligned} S_n(\alpha) &= \sum_{k=1}^N c_k(\alpha) \frac{1}{n} \sum_{m=0}^{n-1} \cos\left(\frac{2\pi km}{n}\right) \\ &+ \sum_{k=1}^N d_k(\alpha) \frac{1}{n} \sum_{m=0}^{n-1} \sin\left(\frac{2\pi km}{n}\right). \end{aligned} \quad (21)$$

A direct application of Lemma 2.1 yields

$$S_n(\alpha) = \sum_{l=1}^{\lfloor N/n \rfloor} c_{ln}(\alpha). \quad (22)$$

Invoking the Möbius inversion formula for finite series, the theorem is proved. ■

Finally, the main result could be derived.

**Theorem 2.4 (Reed-Tufts)** *The Fourier series coefficients  $a_n$  and  $b_n$  are computed by*

$$a_n = c_n(0), \quad (23)$$

$$b_n = (-1)^m c_n \left( \frac{1}{2^{k+2}} \right) \quad n = 1, \dots, N, \quad (24)$$

where  $k$  and  $m$  are determined by the factorization  $n = 2^k(2m + 1)$ .

*Proof:* For  $\alpha = 0$ , using Equation (17), it is straightforward to show that  $a_n = c_n(0)$ . For  $\alpha = \frac{1}{2^{k+2}}$  and  $n = 2^k(2m + 1)$ , there are two sub-cases:  $m$  even or odd.

- For  $m = 2q$ ,  $n = 2^k(4q + 1)$ . Therefore,

$$2\pi n\alpha = 2\pi \frac{2^k(4q + 1)}{2^{k+2}} = 2\pi q + \frac{\pi}{2}. \quad (25)$$

Consequently, substituting this quantity into Equation (17), yields

$$\begin{aligned} c_n \left( \frac{1}{2^{k+2}} \right) &= a_n \cos \left( 2\pi q + \frac{\pi}{2} \right) \\ &\quad + b_n \sin \left( 2\pi q + \frac{\pi}{2} \right) \\ &= b_n. \end{aligned} \quad (26)$$

- For  $m = 2q + 1$ ,  $n = 2^k(4q + 3)$ . It follows that

$$2\pi n\alpha = 2\pi \frac{2^k(4q + 3)}{2^{k+2}} = 2\pi q + \frac{3\pi}{2}. \quad (27)$$

Again invoking the Equation (17), the following expression is derived.

$$\begin{aligned} c_n \left( \frac{1}{2^{k+2}} \right) &= a_n \cos \left( 2\pi q + \frac{3\pi}{2} \right) \\ &\quad + b_n \sin \left( 2\pi q + \frac{3\pi}{2} \right) \\ &= -b_n. \end{aligned} \quad (28)$$

Joining these two sub-cases, it is easy to verify that

$$b_n = (-1)^m c_n \left( \frac{1}{2^{k+2}} \right). \quad (29)$$

The number of real multiplications and additions of this algorithm were given by [15]

$$M_R(N) = \frac{3}{2}N, \quad (30)$$

and

$$A_R(N) = \frac{3}{8}N^2, \quad (31)$$

respectively, where  $N$  is the blocklength of the transform.

### 2.3 REED-SHIH (SIMPLIFIED AFT)

Introduced by Reed *et al.* [18], this algorithm is an evolution of that one developed by Reed and Tufts. Surprisingly, in this new method, the averages were re-defined in accordance to the theory created by H. Bruns [3] in 1903.

**Definition 2.4 (Bruns Alternating Average)** *The  $2n$ th Bruns alternating average,  $B_{2n}(\alpha)$ , is defined by*

$$B_{2n}(\alpha) \triangleq \frac{1}{2n} \sum_{m=0}^{2n-1} (-1)^m \cdot v \left( \frac{m}{2n}T + \alpha T \right). \quad (32)$$

Invoking the definition of  $c_n$ , applying Theorem 2.3 and Definition 2.3, the following theorem was derived.

**Theorem 2.5** *The coefficients  $c_n(\alpha)$  are given by the Möbius inversion formula for finite series as*

$$c_n(\alpha) = \sum_{l=1,3,\dots}^{\lfloor \frac{N}{n} \rfloor} \mu(l) \cdot B_{2nl}(\alpha). \quad (33)$$

*Proof:* See [26]. ■

Since a relation between the signal samples and the Bruns alternating averages was obtained, as well as an expression connecting the Bruns alternating averages to the  $c_n$  coefficients, was available, few points were missing to compute the Fourier series coefficients. Actually, it remained to derive an expression that could relate the Fourier series coefficients ( $a_n$  and  $b_n$ ) and the coefficients  $c_n$ . Examining Equation (17), two conditions were distinguishable:

- $a_n = c_n(0)$ ;
- $b_n = c_n \left( \frac{1}{4n} \right)$ .

Those were the final relations. Calling Theorem 2.5, the next result was obtained.

**Theorem 2.6 (Reed-Shih)** *The Fourier series coefficients  $a_n$  and  $b_n$  are computed by*

$$a_0 = \frac{1}{T} \int_0^T v(t)dt, \quad (34)$$

$$a_n = \sum_{l=1,3,5,\dots}^{\lfloor \frac{N}{n} \rfloor} \mu(l) B_{2nl}(0), \quad (35)$$

$$b_n = \sum_{l=1,3,5,\dots}^{\lfloor \frac{N}{n} \rfloor} \mu(l) (-1)^{\frac{l-1}{2}} B_{2nl} \left( \frac{1}{4nl} \right), \quad (36)$$

for  $n = 1, \dots, N$ .

*Proof:* The proof is similar to the proof of Theorem 2.4. ■

For a blocklength  $N$ , the multiplicative and additive complexities were given by

$$M_R(N) = N, \quad (37)$$

and

$$A_R(N) = \frac{1}{2}N^2, \quad (38)$$

respectively.

The AFT algorithm proposed by Reed-Shih presented some advancements:

- The computation of both  $a_n$  and  $b_n$  had been readjusted, having roughly the same computational effort. The algorithm became even more balanced than Reed-Tufts algorithm;
- The algorithm was naturally suited to a parallel processing implementation;
- It was computationally less complex than Reed-Tufts algorithm.

## 2.4 AN EXAMPLE

In this subsection, some comments to an example of the Reed-Shih algorithm are done. Let  $v(t)$  be a signal with period  $T = 1$  s. Consider the computation of the Fourier series coefficients up to the 5th harmonic.

According to Reed-Shih algorithm, the coefficients  $a_n$  and  $b_n$  of the Fourier series of  $v(t)$  were expressed by

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_2(0) \\ B_4(0) \\ B_6(0) \\ B_8(0) \\ B_{10}(0) \end{bmatrix} \quad (39)$$

and

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_2(\frac{1}{4}) \\ B_4(\frac{1}{8}) \\ B_6(\frac{1}{12}) \\ B_8(\frac{1}{16}) \\ B_{10}(\frac{1}{20}) \end{bmatrix} \quad (40)$$

Comparing these formulations with the ones of Reed-Tufts algorithm, one may note the balance in the computation of  $a_n$  and  $b_n$ . Both coefficients were obtained through similar matrices. A table relating Bruns alternative averages,  $B_n(\alpha)$ , with the necessary time samples to compute it, could be constructed.

Bruns averages	Sample time (s)
$B_2(0)$	$0, \frac{1}{2}$
$B_4(0)$	$0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$
$B_6(0)$	$0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}$
$B_8(0)$	$0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}$
$B_{10}(0)$	$0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}$
$B_2(\frac{1}{4})$	$\frac{1}{4}, \frac{3}{4}$
$B_4(\frac{1}{8})$	$\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}$
$B_6(\frac{1}{12})$	$\frac{1}{12}, \frac{1}{4}, \frac{5}{12}, \frac{7}{12}, \frac{3}{4}, \frac{11}{12}$
$B_8(\frac{1}{16})$	$\frac{1}{16}, \frac{3}{16}, \frac{5}{16}, \frac{7}{16}, \frac{9}{16}, \frac{11}{16}, \frac{13}{16}, \frac{15}{16}$
$B_{10}(\frac{1}{20})$	$\frac{1}{20}, \frac{3}{20}, \frac{1}{4}, \frac{7}{20}, \frac{9}{20}, \frac{11}{20}, \frac{13}{20}, \frac{3}{4}, \frac{17}{20}, \frac{19}{20}$

Table 2. Necessary samples for the Bruns alternating averages.

Table 2 shows that at least 40 non-uniform time samples of  $v(t)$  were necessary to exactly compute the Bruns alternating averages, and then the Fourier series coefficients.

At this point, some observations were relevant:

- This algorithm is not naturally suited for uniform sampling.
- A uniform sampler utilized to obtain all the necessary samples would need a sampling rate too high. In the example illustrated here, a 120 Hz clock should be required to sample the necessary points for the computation of the Fourier series of a 1 Hz bandlimited signal.

Certainly these observations appear to be disturbing and seems to jeopardize the feasibility of the whole procedure. However, it is important to stress that this procedure furnishes the exact computation of the Fourier series coefficients.

An empirical solution to circumvent this problem is to interpolate. An interpolation based on uniformly sampled points could be used to estimate the sample values required by AFT. Of course, this procedure inherently introduces computation errors.

For example, assuming that the 1Hz signal  $v(t)$  was sampled by a clock with period  $T_0 = \frac{1}{10}$  s. Hence, the following sample points were available:

$$v(0), v\left(\frac{1}{10}\right), v\left(\frac{2}{10}\right), v\left(\frac{3}{10}\right), v\left(\frac{4}{10}\right), \\ v\left(\frac{5}{10}\right), v\left(\frac{6}{10}\right), v\left(\frac{7}{10}\right), v\left(\frac{8}{10}\right), v\left(\frac{9}{10}\right).$$

Table 2 shows, for example, that the computation of  $B_4(0)$  requires — among other samples —  $v(\frac{1}{4})$ , which is clearly not available. To overcome this difficulty, a rounding operation could be introduced. Thus, the sample  $v(\frac{3}{10})$  could be used whenever the algorithm called  $v(\frac{1}{4})$  ( $\lceil 10 \frac{1}{4} \rceil / 10 = 3/10$ ). This rounding operation is also known as zero-order interpolation.

The accuracy of the AFT algorithm is deeply associated with the sampling period  $T_0$ . If more precision is required, then one should expect to increase sampling rate, resulting in the introduction of smaller errors due to the interpolation scheme. Higher order of interpolation (e.g. first-order interpolation) could also be used to obtain more accurate estimations of the Fourier series coefficients. The following trade-off is quite clear accuracy versus order of interpolation.

However, for signals sampled at Nyquist rate (or close to), zero-order interpolation already leads to good results [15]. A detailed error analysis of interpolation schemes can be found in [9, 15, 40, 34]. Further comments can be found in [18].

## 3. A NEW ARITHMETIC TRANSFORM

Besides its numerical appropriateness [6a], the discrete Hartley transform (DHT) has proved along the years to be an important tool with several applications, such as biomedical image compression, OFDM/CDMA systems, and ADSL transceivers. Searching the literature, no mention about a possible "Arithmetic Hartley Transform" to compute the DHT was found.

In this section, a condensation of the main results of the Arithmetic Hartley Transform is outlined. The method used to define the AHT turned out to furnish a new insight into the arithmetic transform. In particular, the role of interpolation was clarified. Additionally, it was mathematically shown that interpolation is a pivotal issue in arithmetic transforms. Indeed it determines the transform.

A new approach to arithmetic transform is adopted. Instead of considering uniformly sampled points extracted from a continuous signal  $v(t)$ , the AHT is based on the purely discrete signal. Thus, the starting point of the development is the discrete transform definition, not the series expansion, as it was done in the AFT algorithm. This approach is philosophically appealing, since in a final analysis a discrete transform relates two set of points, not continuous functions.

Let  $\mathbf{v}$  be an  $N$ -dimensional vector with real elements. The DHT establishes a pair denoted by  $\mathbf{v} = [v_0, v_1, \dots, v_{N-1}]^T \leftrightarrow \mathbf{V} = [V_0, V_1, \dots, V_{N-1}]^T$ , where the elements of the transformed vector  $\mathbf{V}$  (i.e., Hartley spectrum) are defined by [6a]

$$V_k \triangleq \frac{1}{N} \sum_{i=0}^{N-1} v_i \cdot \text{cas} \left( \frac{2\pi ki}{N} \right), \quad k = 0, 1, \dots, N-1, \quad (41)$$

where  $\text{cas } x \triangleq \cos x + \sin x$  is Hartley's "cosine and sine" kernel. The inverse discrete Hartley transform is then [6a]

$$v_i = \sum_{k=0}^{N-1} V_k \cdot \text{cas} \left( \frac{2\pi ki}{N} \right), \quad i = 0, 1, \dots, N-1. \quad (42)$$

**Lemma 3.1 (Fundamental Property)** *The function  $\text{cas}(\cdot)$  satisfies*

$$\sum_{m=0}^{k-1} \text{cas} \left( 2\pi m \frac{k'}{k} \right) = \begin{cases} k & \text{if } k|k', \\ 0 & \text{otherwise.} \end{cases} \quad (43)$$

*Proof:* It follows directly from Lemma 2.1. ■

Similarly to the AFT theory, it was necessary to define averages  $S_k$ , calculated from the time-domain elements. The averages were computed by

$$S_k \triangleq \frac{1}{k} \sum_{m=0}^{k-1} v_{m \frac{N}{k}}, \quad k = 1, \dots, N-1. \quad (44)$$

It is interesting to note that this definition required fractional index sampling (!). Analogously to the AFT methods, this fact seemed to make further considerations impracticable, since only integer index samples were available. This subtle question is to be treated in the sequel. Meanwhile, the fractional indexes will be treated in mathematical development without concerns.

An application of the inverse Hartley transform (Equation (42)) in Equation (44) offered:

$$S_k = \frac{1}{k} \sum_{m=0}^{k-1} \sum_{k'=0}^{N-1} V_{k'} \cdot \text{cas} \left( \frac{2\pi k' m N}{k} \right). \quad (45)$$

Rearranging the summation order, simplifying, and calling

Lemma 3.1, it yielded:

$$\begin{aligned} S_k &= \frac{1}{k} \sum_{k'=0}^{N-1} V_{k'} \sum_{m=0}^{k-1} \text{cas} \left( 2\pi m \frac{k'}{k} \right) \\ &= \sum_{s=0}^{\lfloor (N-1)/k \rfloor} V_{sk}. \end{aligned} \quad (46)$$

For simplicity and without loss of generality, consider a signal  $\mathbf{v}$  with zero mean value, i.e.,  $\frac{1}{N} \sum_{i=0}^{N-1} v_i = 0$ . Clearly, this consideration has no influence on the values of  $V_k$ ,  $k \neq 0$ . An application of the modified Möbius inversion formula for finite series [15] was sufficient to obtain the final theorem to derive the Arithmetic Hartley Transform. According to Theorem 2.1, the following result could be stated:

**Theorem 3.1 (Reed et alli)** *If*

$$S_k = \sum_{s=1}^{\lfloor (N-1)/k \rfloor} V_{sk}, \quad 1 \leq k \leq N-1, \quad (47)$$

then

$$V_k = \sum_{l=1}^{\lfloor (N-1)/k \rfloor} \mu(l) S_{kl}, \quad (48)$$

where  $\mu(\cdot)$  is Möbius function. ■

To illustrate its usage, consider an 8-point DHT. Using Theorem 3.1, Equation (48), the spectral analysis is given by:

$$\begin{aligned} V_1 &= S_1 - S_2 - S_3 - S_5 + S_6 - S_7, \\ V_2 &= S_2 - S_4 - S_6, \\ V_3 &= S_3 - S_6, \\ V_4 &= S_4, \\ V_5 &= S_5, \\ V_6 &= S_6, \\ V_7 &= S_7. \end{aligned}$$

The component  $V_0 = V_8$  can be computed directly from the given samples, since it represents the mean value of the signal  $V_0 = \frac{1}{8} \sum_{m=0}^7 v_m$ . In Figure 2, a diagram of this computation is shown.

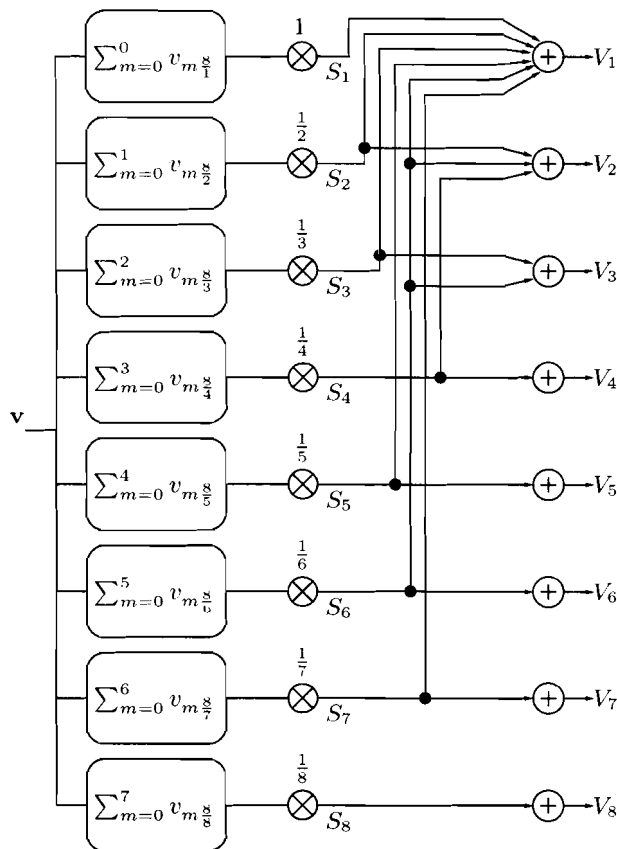
The above theorem and equations completely specified how to compute the discrete Hartley spectrum. Additionally, the inverse transformation could also be established. The following was straightforward.

**Corollary 1** *Inverse discrete Hartley transform components can be computed by*

$$v_i = \sum_{l=1}^{\lfloor (N-1)/i \rfloor} \mu(l) \sigma_{il}, \quad (49)$$

where  $\sigma_i \triangleq \frac{1}{i} \sum_{m=0}^{i-1} V_{m \frac{N}{i}}$ ,  $i = 1, \dots, N-1$ . ■

The original Arithmetic Fourier Transform had identical equations to those just derived for the Hartley transform (compare Equation (10) and Equation (47)). A question arises: since the equations were the same, which spectrum was actually being evaluated? Fourier or Hartley spectrum?



**Figure 2.** Diagram for computing the AHT for  $N = 8$ . The boxes compute the averages and the multipliers implement the scaling operation. The third layer accounts for the arithmetic computation based on Möbius functions.

A clear understanding of underlying arithmetic transform mechanisms will be possible in the next section. Once more the reader is asked to put this question aside for a while, allowing further developments to be derived.

To sum it up, at this point two major questions were accumulated: (i) How to handle with fractional indexes? and (ii) How could same formulae result in different spectra? Interestingly, both questions had the same answer.

The arithmetic transform algorithm could be summarized in four major steps:

1. Index generation, i.e., calculating the indexes of necessary samples ( $m \frac{N}{k}$ );
2. Fractional index samples handling, which requires interpolation;
3. Computation of averages:  $S_k \triangleq \frac{1}{k} \sum_{m=0}^{k-1} v_{m \frac{N}{k}}$ ;
4. Computation of spectrum by Möbius Inversion Formula:  $V_k = \sum_{l=1}^{\lfloor (N-1)/k \rfloor} \mu(l) S_{kl}$ .

In the rest of this paper, the step two was addressed. In the sequel, a mathematical method, explaining the importance of the interpolation process in the arithmetic algorithms, was derived.

### 3.1 INTERPOLATION

Usual arithmetic theory deals with spectrum approximations via zero- or first-order interpolation [15, 26, 33]. In this section, it is shown that an interpolation process based on the known components (integer index samples) characterizes the definition of the fractional index components,  $v_r$ ,  $r \notin \mathbb{N}$ . This analysis allows a more encompassing perception of the interpolation mechanisms and gives mathematical tools for establishing validation constraints to such interpolation process. In addition, brief comments on the trade-off between accuracy and computational cost required by interpolation process close the section.

#### 3.1.1 IDEAL INTERPOLATION

What does a fractional index discrete signal component really mean? The value of  $v_r$  for a noninteger value  $r$ ,  $r \notin \mathbb{N}$ , could be computed by

$$v_r = \sum_{k=0}^{N-1} V_k \text{cas} \left( \frac{2\pi kr}{N} \right) = \sum_{i=0}^{N-1} v_i \sum_{k=0}^{N-1} \text{cas} \left( \frac{2\pi ki}{N} \right) \text{cas} \left( \frac{2\pi kr}{N} \right). \quad (50)$$

Defining the *Hartley weighting function* by

$$w_i(r) \triangleq \sum_{k=0}^{N-1} \text{cas} \left( \frac{2\pi ki}{N} \right) \text{cas} \left( \frac{2\pi kr}{N} \right), \quad (51)$$

the value of the signal at fractional indexes could be found utilizing an  $N$ -order interpolation expressed by:

$$v_r \triangleq \sum_{i=0}^{N-1} w_i(r) \cdot v_i. \quad (52)$$

It is clear that each transform kernel could be associated to a different weighting function. Consequently, a different interpolation process for each weighting function is required. In the arithmetic transform formalism, the difference from one transform to another resides in its interpolation process.

It can be shown that weighting functions make the Equation  $\sum_{i=0}^{N-1} w_i(r) = 1$  to hold. If  $r$  is an integer number, then the orthogonality properties of  $\text{cas}(\cdot)$  function [6a] make  $w_r(r) = 1$  and  $w_i(r) = 0$  ( $\forall i \neq r$ ). Therefore, no interpolation is needed.

After some trigonometrical manipulation, the interpolation weights for several kernels could be expressed by closed formulae. As stated before, there is a weighting function for each transform. Let  $\text{Sa}(\cdot)$  be the sampling function,

$$\text{Sa}(x) \triangleq \begin{cases} \sin(x)/x, & x \neq 0, \\ 1, & x = 0. \end{cases} \quad (53)$$



**Proposition 1** An  $N$ -point transform has interpolation weighting functions given by  
 Cosine Kernel

$$w_i(r) = \frac{1}{2N} + \frac{N-1/2}{N} \left\{ \frac{1}{2} \frac{\text{Sa}\left(\frac{N-1/2}{N} 2\pi(i-r)\right)}{\text{Sa}\left(\pi(i-r)/N\right)} + \frac{1}{2} \frac{\text{Sa}\left(\frac{N-1/2}{N} 2\pi(i+r)\right)}{\text{Sa}\left(\pi(i+r)/N\right)} \right\}.$$

Sine Kernel

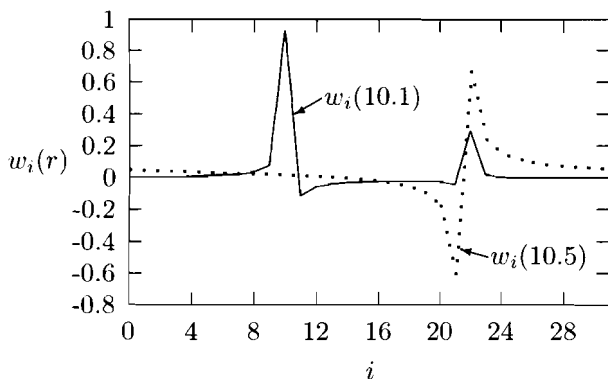
$$w_i(r) = \frac{N-1/2}{N} \left\{ \frac{1}{2} \frac{\text{Sa}\left(\frac{N-1/2}{N} 2\pi(i-r)\right)}{\text{Sa}\left(\pi(i-r)/N\right)} - \frac{1}{2} \frac{\text{Sa}\left(\frac{N-1/2}{N} 2\pi(i+r)\right)}{\text{Sa}\left(\pi(i+r)/N\right)} \right\}.$$

Hartley Kernel

$$w_i(r) = \frac{1}{2N} + \frac{N-1/2}{N} \frac{\text{Sa}\left(\frac{N-1/2}{N} 2\pi(i-r)\right)}{\text{Sa}\left(\pi(i-r)/N\right)} + \frac{1}{2N} \cot\left(\frac{\pi(i+r)}{N}\right) - \frac{1}{2N} \frac{\cos\left(\frac{N-1/2}{N} 2\pi(i+r)\right)}{\sin\left(\pi(i+r)/N\right)}.$$

To exemplify, Figure 3 shows two weighting functions used to compute  $v_{10.1}$  and  $v_{10.5}$  in the arithmetic Hartley transform. Figure 4 shows the weighting profile for the cosine and Hartley kernels. These functions were calculated by closed formulae. Note in Figure 4 that the parameters  $N = 16$ ,  $r = 0, 0.1 \dots 15$ , and  $i = 0 \dots 15$  were employed. Observe that  $i \in \mathbb{N}$  and  $r \in \mathbb{R}$ . The maximum values are achieved at  $i = 0$  or  $i = N/2 = 8$  (central peak) and they corresponds to the unity. The value of the local maxima are exactly  $1/2$ , occurring when  $r$  is integer. Note that each curve is almost null everywhere, except at the vicinities of  $i \approx r$  or  $i \approx N - r$ .

With this proposition, the mathematical description of the AHT algorithm is completed. The derived formulae furnish the *exact* value of the spectral components. On the other hand, the computational complexity of the ideal interpolation implementation is similar to the direct implementation, i.e., computing the transform by its plain definition:  $V_k = \frac{1}{N} \sum_{i=0}^{N-1} v_i \cos\left(\frac{2\pi}{N} ki\right)$ . To tackle this issue, a non-ideal interpolation scheme was envisioned.



**Figure 3.** Hartley weighting functions used to interpolate  $v_{10.1}$  and  $v_{10.5}$  ( $N = 32$  blocklength).

### 3.1.2 NON-IDEAL INTERPOLATION

According to the index generation ( $m \frac{N}{k}$ ), the number  $R$  of points that require interpolation is upper bounded by  $R \leq$

$\sum_{d|N} d - 1$ . This sum represents the number of samples with fractional index. Consequently, this approach could be attractive for large non-prime blocklength  $N$  with great number of factors, because it would require a smaller number of interpolations.

The next task was to find simpler formulae for the weighting functions, assuming large blocklength. Instead of using the exact weighting functions, the limit  $N \rightarrow \infty$  was examined and used to derive asymptotic approximations of the weighting function.

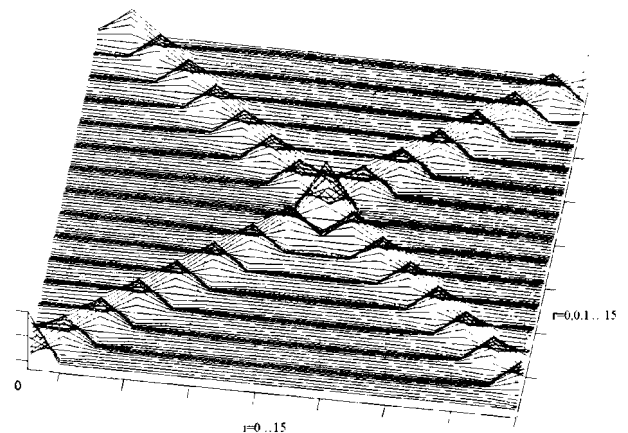
**Proposition 2** A continuous approximation for the interpolation weighting function for sufficiently large  $N$  is given by:  
 Cosine Kernel

$$\hat{w}_i(r) \approx \frac{\text{Sa}(2\pi(i-r))}{2} + \frac{\text{Sa}(2\pi(i+r))}{2} \quad \text{Sine Kernel}$$

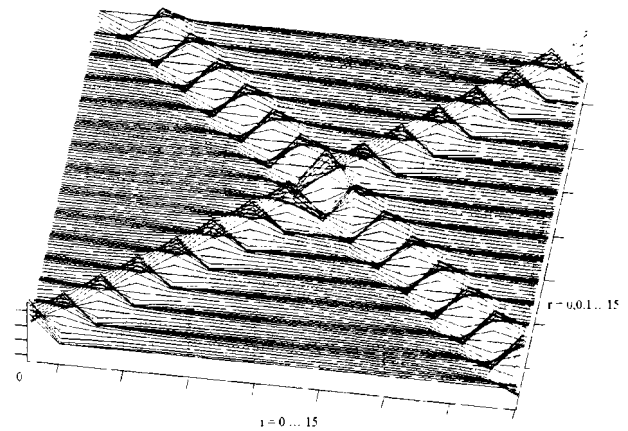
$$\hat{w}_i(r) \approx \frac{\text{Sa}(2\pi(i-r))}{2} - \frac{\text{Sa}(2\pi(i+r))}{2}$$

Hartley Kernel

$$\hat{w}_i(r) \approx \text{Sa}(2\pi(i-r)) + \frac{1-\cos 2\pi r}{2\pi(i+r)}.$$



(a) Cosine Kernel



(b) Hartley Kernel

**Figure 4.** These curve families represent the weighting profile  $w_i(r)$  for the cosine kernel (a) and Hartley kernel (b).

In terms of the Hilbert transform and the  $\text{Sa}(\cdot)$  function, the asymptotic weighting function for Hartley kernel could be given by

$$\hat{w}_i(r) \approx \text{Sa}(2\pi(i-r)) - \mathcal{Hil}\left\{\text{Sa}(2\pi(i+r))\right\}, \quad (54)$$

or alternatively,

$$\hat{w}_i(r) \approx \text{Sa}(2\pi(i-r)) - \text{Ca}(2\pi(i+r)) - \mathcal{Hil}\left\{\delta(2(r+i))\right\}, \quad (55)$$

where  $\mathcal{Hil}$  denotes the Hilbert transform,  $\text{Ca}(x) \triangleq \frac{\cos x}{x}$ , for  $x \neq 0$ , is the co-sampling function and  $\delta(x)$  is the Dirac impulse.

**Zero-order Interpolation.** Rounding the fractional index provides the zero-order interpolation. The estimated (interpolated) signal  $\hat{v}_j$  was then expressed by  $\hat{v}_j = v_{[j]}$ , where  $[ \cdot ]$  is a function which rounds off its argument to its nearest integer. Examining the asymptotic behavior of the weighting function for cosine kernel, we derive the following results:

$$\begin{aligned} \hat{w}_i(r) &\approx 0, \quad \forall i \neq [r], N - [r], \\ \hat{w}_{[r]}(r) &\approx \frac{\text{Sa}(2\pi([r]-r))}{2} \approx \frac{1}{2}, \\ \hat{w}_{N-[r]}(r) &\approx \frac{\text{Sa}(2\pi([r]-r))}{2} \approx \frac{1}{2}. \end{aligned} \quad (56)$$

Under the above assumptions, the Equation (52) furnishes:

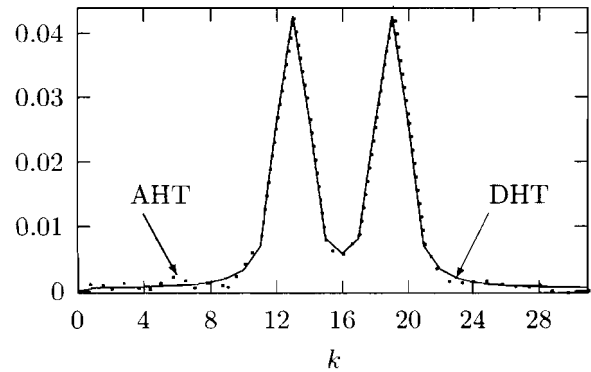
$$\begin{aligned} \hat{v}_r &\approx w_{[r]}(r)v_{[r]} + w_{N-[r]}(r)v_{N-[r]} \\ &\approx \frac{1}{2}v_{[r]} + \frac{1}{2}v_{N-[r]}. \end{aligned} \quad (57)$$

Thus, for even signal ( $v_k = v_{N-k}$ ), the approximated value of the interpolated sample is roughly given by  $\hat{v}_r \approx v_{[r]}$ . It is straightforward to see that the influence of odd part of the signal vanishes in the zero-order interpolation. Besides zero-order interpolation is "blind" to the odd component of a signal. In fact, as show by the set of Equations (56), zero-order interpolation is an (indeed good) approximation to the cosine asymptotic weighting function. This puts some light on the role of the zero-order interpolations and its relation to the earliest versions of the arithmetic Fourier transform, which could not analyze odd periodic signals.

Zero-order interpolation was intuitively used in previous work by Tufts, Reed *et alli* [15, 6, 26]. Hsu, in his Ph.D. dissertation, derives an analysis of first-order interpolation effect [33].

**Interpolation Order.** Let  $M_m$  be a set with the  $m$  ( $< N$ ) most significant coefficients  $w_i(r)$ . For zero-order interpolation,  $m = 1$ . Increasing the value of  $m$ , the interpolation process would gradually be improved, because more coefficients would be retained. Proceeding in this way, the following calculation performs a non-ideal interpolation:

$$\hat{v}_r = \frac{1}{\eta} \sum_{i \in M_m} w_i(r) \cdot v_i, \quad (58)$$



**Figure 5.** The discrete Hartley transform computed by definition (solid line) and by arithmetic transform algorithm (dotted line). Simulation data:  $f(t) = \cos(90\pi t) (t - \frac{1}{2})^2$ ,  $t = 0 \dots 1$ ,  $N = 32$ .

where  $\eta \triangleq \sum_{j \in M_m} w_j(r)$  is a normalization factor.

Figure 5 presents a 32-point discrete Hartley transform of a particular signal computed by the plain definition of the DHT and by the arithmetic transform method using  $m = 2$ . The resemblance is evident.

## 4. CONCLUSIONS

This paper supplied a short survey on arithmetic transforms. The arithmetic Fourier transform is explained and its three main 1-D versions were described. Furthermore, some comments on implementation, challenging points, and advantages of the arithmetic transforms were discussed via simple examples.

In this paper, the introduction of the AHT emphasized the key point of the arithmetic transforms: the interpolation process. It was shown that the fundamental equations of the arithmetic transform algorithms were essentially the same (regardless the kernel). This property could open path to the implementation of "universal transformers". In this type of construction, the circuitry would be the same for several transforms, except the interpolation module. A different interpolation module would reflect different transform (Fourier, Hartley, Cosine). Finally, this paper could be taken as a starting point to those who want to investigate arithmetic transforms.

## ACKNOWLEDGMENTS

The first author would like to thank Emeritus Professor Dr. Irving S. Reed, University of Southern California, for kindly sending him a copy of Chin-Chi Hsu's Ph.D. thesis [33].

## REFERENCES

- [1] P. Harzer. Über eine von Herrn Tschebyschef angegebene Interpolationsmethode. *Astron. Nach.*, 115, 1886.
- [2] R. Radau. Études sur les formules d'interpolation. *Bull. Astronomique*, 8, 1891.
- [3] Ernst Heinrich Bruns. *Grundlinien des wissenschaftlichen Rechnens*. B. G. Teubner Verlag, Leipzig, Germany, 1903.

- [4] Aurel Freidrich Wintner. An Arithmetical Approach to Ordinary Fourier Series. Monograph, 1947.
- [5] Donald W. Tufts and Angaraih G. Sadasiv. Arithmetic Fourier Transform and Adaptive Delta Modulation: A Symbiosis for High Speed Computation. In *Proceedings of The International Society for Optical Engineering (SPIE)*, pages 168–178, January 1988.
- [6] Donald W. Tufts and Angaraih G. Sadasiv. The Arithmetic Fourier Transform. *IEEE ASSP Magazine*, pages 13–17, January 1988. IEEE Catalog Number: 0740-7467/88/0100-0013 © IEEE.
- [7] J. L. Schiff and W. J. Walker. A Sampling Theorem and Wintner's Results on Fourier Coefficients. *Journal of Mathematical Analysis and Applications*, 133(2), August 1988.
- [8] Irving S. Reed, Y. Y. Choi, and X. Yu. Practical Algorithm for Computing the 2-D Arithmetic Fourier Transform. In *Proceedings of The International Society for Optical Engineering (SPIE)*, pages 54–61, January 1989.
- [9] G. Faye Bartels-Boudreaux, Donald W. Tufts, P. Dhir, Angaraih G. Sadasiv, and G. Fischer. Analysis of Errors in the Computation of Fourier Coefficients Using the Arithmetic Fourier Transform (AFT) and Summation by Parts (SBP). In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing – ICASSP'89*, pages 1011–1014, Glasgow, U.K., May 1989. IEEE Catalog Number: CH2673-2/89/0000-1011 © IEEE.
- [10] Nazif Tepedelenlioglu. A Note on the Computational Complexity of the Arithmetic Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1146–1147, July 1989. IEEE Catalog Number: 0096-3518/89/0700-1146 © IEEE.
- [11] Donald W. Tufts. Comments on “A Note on the Computational Complexity of the Arithmetic Fourier Transform”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1147–1148, July 1989. IEEE Catalog Number: 0096-3518/89/0700-1147 © IEEE.
- [12] Y. Y. Choi. *Algorithms for Computing the 2-D Arithmetic Fourier Transform*. PhD thesis, Department of Electrical Engineering – Systems, University of Southern California, Los Angeles, CA, 1989.
- [13] Donald W. Tufts, Z. Fan, and Z. Cao. Image Processing and the Arithmetic Fourier Transform. *SPIE High Speed Computing II*, 1058:46–53, 1989.
- [14] G. Fischer, Donald W. Tufts, and Angaraih G. Sadasiv. *VLSI Implementation of the Arithmetic Fourier Transform (AFT): A New Approach to High Speed Communication for Signal Processing*, chapter VLSI Signal Processing III, pages 264–275. IEEE Press, New York, 1989. R. Broderon and H. Moscovitz, Eds.
- [15] Irving S. Reed, Donald W. Tufts, Xiaoli Yu, T.K. Truong, Ming-Tang Shih, and Xiaowei Yin. Fourier Analysis and Signal Processing by Use of the Möbius Inversion Formula. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(3):458–470, March 1990. IEEE Catalog Number: 0096-3518/90/0300-0458 © IEEE.
- [16] Weiping Li. Fourier Analysis Using Adaptive AFT. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing – ICASSP'90*, pages 1523–1526, Albuquerque, U.S.A., April 1990. IEEE Catalog Number: CH2847-2/90/0000-1523 © IEEE.
- [17] G. Fischer, Donald W. Tufts, and R. Unnikrishnan. VLSI Implementation of the Arithmetic Fourier Transform. In *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, pages 800–803, August 1990. IEEE Catalog Number: CH2785-7/90/0000-0800 © IEEE.
- [18] Irving S. Reed, Ming-Tang Shih, E. Hendon, T. K. Truong, and Donald W. Tufts. A VLSI Architecture for Simplified Arithmetic Fourier Transform Algorithm. In *International Conference on Application Specific Array Processors, Special-Purpose Systems*, pages 542–553, Princeton, U.S.A., September 1990. IEEE Catalog Number: CH2920-7/90/0000/0542 © IEEE, ISBN: 0-8186-9089-5.
- [19] Heonchul Park and Viktor K. Prassana. VLSI Architectures for Computing the Arithmetic Fourier Transform. Technical report, Department of Electrical Engineering – Systems, University of Southern California, December 1990.
- [20] Y. Y. Choi, Irving S. Reed, and Ming-Tang Shih. The New Arithmetical Approach to Fourier Series Analysis for a 2-D Signal. In *Proceedings of the International Conference on Acoustic, Speech, and Signal Processing – ICASSP'90*, pages 1997–2000, New Mexico, 1990. IEEE Catalog Number: CH2847-2/90/0000/1997 © IEEE.
- [21] N. Wigley and G. A. Jullien. Sampling Reduction for the Arithmetic Fourier Transform. In *Proceedings of 32nd Midwest Symposium on Circuits and Systems*, pages 841–844, 1990. IEEE Catalog Number: CH2785-4/90/0000-841 © IEEE.
- [22] Heonchul Park and Viktor K. Prassana. VLSI Architectures for Computing the Arithmetic Fourier Transform. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing – ICASSP'91*, pages 1029–1032, May 1991. IEEE Catalog Number: CH2977-7/91/0000-1029 © IEEE.
- [23] Heonchul Park and Viktor K. Prassana. Modular VLSI Architectures for Computing Arithmetic Fourier Transform. Technical report, IRIS #273, Department of Electrical Engineering – Systems, University of Southern California, August 1991.
- [24] Heonchul Park and Viktor K. Prassana. Fixed Size Array Architectures for Computing Arithmetic Fourier Transform. In *Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems and Computers*, pages 85–89, November 1991. IEEE Catalog Number: 1058-6393/91 © IEEE.
- [25] V. K. Ananthashayana. Comments on “Fourier Analysis and Signal Processing by Use of the Möbius Inversion Formula”. *IEEE Transactions on Signal Processing*, 40(3):676, March 1992. IEEE Catalog Number: 1053-587X/92 © IEEE.
- [26] Irving S. Reed, Ming-Tang Shih, T. K. Truong, E. Hendon, and Donald W. Tufts. A VLSI Architecture for Simplified Arithmetic Fourier Transform Algorithm. *IEEE Transactions on Signal Processing*, 40(5):1122–1133, May 1992.
- [27] N. Wigley and G. A. Jullien. On Implementing the Arithmetic Fourier Transform. *IEEE Transactions on Signal Processing*, 40(9):2233–2242, September 1992.
- [28] A. Abo Zaid, A. El-Mahdy, A. O. Attia, and M. M. Selim. A High Speed Classifier Using the Arithmetic Fourier Transform. In *Proceedings of the 35th Midwest Symposium on Circuits and Systems*, pages 36–39, 1992. IEEE Catalog Number: 0-7803-0510-8/92 © IEEE.
- [29] Brian T. Kelley and Vijay K. Madiseti. Efficient VLSI Architectures for the Arithmetic Fourier Transform. *IEEE Transactions on Signal Processing*, 41(1):365–384, January 1993.
- [30] Donald W. Tufts and Haiguang Chen. Iterative Realization of the Arithmetic Fourier Transform. *IEEE Transactions on Signal Processing*, 41(1):152–161, January 1993. IEEE Catalog Number: 1053-587X/93 © IEEE.
- [31] Heonchul Park and Viktor K. Prassana. Modular VLSI Architectures for Computing Arithmetic Fourier Transform. *IEEE Transactions on Signal Processing*, 41(6):2236–2246, June 1993. IEEE Catalog Number: 1053-587X/93 © IEEE.
- [32] Frederick P. Lovine and Sawasd Tantaratana. Some Alternate Realizations of the Arithmetic Fourier Transform. In

- Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 310–314, Pacific Grove, November 1993. IEEE Catalog Number: 1058-6393/93 © IEEE, ISBN: 0-8186-4120-7.
- [33] Chin-Chi Hsu. *Use of Number Theory and Modern Algebra in the Reed-Solomon Code and the Arithmetic Fourier Transform*. PhD thesis, Department of Electrical Engineering – Systems, University of Southern California, Los Angeles, CA, August 1994.
- [34] Chin-Chi Hsu, Irving S. Reed, and T. K. Truong. Inverse Z-Transform by Möbius Inversion and Error Bounds of Aliasing in Sampling. *IEEE Transactions on Signal Processing*, 42(10):2823–2831, October 1994.
- [35] Luc Knockaert. A Generalized Möbius Transform and Arithmetic Fourier Transforms. *IEEE Transactions on Signal Processing*, 42(11):2967–2971, November 1994.
- [36] Vincenzo Di Lecce and Andrea Guerriero. A FT Processor Based in Short AFT Module. In *Proceedings of International Symposium on Applied Informatics, IASTED*, Innsbruck, February 1995.
- [37] Gregorio Andria, Vincenzo Di Lecce, and Mario Savino. Application of the AFT Technique for Low-Cost and Accurate Measurements. In *8th Mediterranean Electrotechnical Conference – MELECON'96*, pages 1347–1350, Bari, Italy, May 1996. IEEE Catalog Number: 96CH35884 © IEEE, ISBN: 0-7803-31-09-5/96.
- [38] Luc Knockaert. A Generalized Möbius Transform, Arithmetic Fourier Transforms, and Primitive Roots. *IEEE Transactions on Signal Processing*, 44(5):1307–1310, May 1996.
- [39] Qian Huisheng and Li Ping. 2-D Arithmetic Fourier Transform Algorithm. In *Proceedings of 3rd International Conference on Signal Processing – ICSP'96*, pages 147–150, October 1996. ISBN: 0-7803-2912-0.
- [40] Qian Huisheng, Li Ping, and Zhou Feng. On Arithmetic Cosine Transform Algorithm. In *Proceedings of 3rd International Conference on Signal Processing – ICSP'96*, pages 143–146, October 1996. IEEE Catalog Number: 96TH8116 © IEEE, ISBN: 0-7803-2912-0.
- [41] Gregorio Andria, Vincenzo Di Lecce, and Andrea Guerriero. An AFT-based Virtual Instrument for Low-cost Spectrum Analysis. In *Proceedings of IMCT'96*, Brussels, 1996.
- [42] Vincenzo Di Lecce and Andrea Guerriero. Spectral Estimation by AFT Computation. *Digital Signal Processing*, 6(24):213–223, 1996.
- [43] Valery G. Atlas, Dmitry G. Atlas, and Evgeny I. Bovbel. 2-D Arithmetic Fourier Transform Using the Bruns Method. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 44(6):546–551, June 1997.
- [44] Xin-Jin Ge, Nan-Xian Chen, and Zhao-Dou Chen. Efficient Algorithm for 2-D Arithmetic Fourier Transform. *IEEE Transactions on Signal Processing*, 45(8):2136–2140, August 1997.
- [45] Zhang Xianchao, Wan Yingyu, and Chen Guoliang. A New Approach for Implementing the Arithmetic Fourier Transform (AFT). In *Proceedings of the Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, pages 633–634, Beijing, China, May 2000. ISBN: 0-7695-0589-2/00.
- [46] Zhang Xianchao, Huang Liusheng, and Chen Guoliang. A New Approach for Computing the Discrete Fourier Transform of Arbitrary Length. In *Proceedings of ICSP2000*, pages 81–84, August 2000. ISBN: 0-7803-5747-7/00.
- [47] Renato José de Sobral Cintra and Hélio Magalhães de Oliveira. How to Interpolate in Arithmetic Transform Algorithms. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing – ICASSP'02*, Orlando, Florida, U.S.A., May 2002.
- [48] Renato José de Sobral Cintra and Hélio Magalhães de Oliveira. Interpolating in Arithmetic Transform Algorithms. In *6th WSEAS CSCC Multiconference, 2nd WSEAS International Conference on Signal Processing and Computational Geometry And Vision*, Crete Island, Greece, July 2002.

## APPENDIX

The following references are not primarily concerned with arithmetic transforms, however they are relevant.

- [1a] N. Bhatnagar. A Binary Friendly Algorithm for Computing Discrete Hartley Transform. In *Digital Signal Processing Proceedings*, pages 353–356, Santorini, Greece, July 1997.
- [2a] H. S. Dee and V. Jeoti. Computing DFT using approximate fast Hartley transform. In *Sixth International Symposium on Signal Processing and its Applications*, pages 100–103, Kuala Lumpur, Malaysia, August 2001.
- [3a] V. Dimitrov and R. Baghaie. Computing Discrete Hartley Transform Using Algebraic Integers. In *Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, pages 1351–1355, Pacific Grove, CA, USA, October 1999.
- [4a] R. J. de Sobral Cintra, H. M. de Oliveira, and C. O. Cintra. The Rounded Hartley Transform. In *IEEE International Telecommunications Symposium*, Natal, Brazil, September 2002.
- [5a] D. M. Burton. *Elementary Number Theory*, page 111. McGraw-Hill, 4th edition, 2000.
- [6a] R. N. Bracewell. *The Hartley Transform*. Oxford University Press, 1986.

**Renato José de Sobral Cintra** was born in Recife. He received the B.Sc. and M.Sc. degrees in electrical engineering from *Universidade Federal de Pernambuco* in 1999 and 2001, respectively. Currently he is a doctorate student at the same institution. He spent an year as research associate at the University of Calgary, Canada, working in the area of Biomedical Engineering. His research interests include biomedical signal analysis, fast algorithms, and wavelets. He is member of AMS, IEEE, SIAM, and SBrT.

**Hélio Magalhães de Oliveira** was born in Arcoverde, Pernambuco, in 1959. He received the B.Sc. and M.S.E.E. degrees in electrical engineering from *Universidade Federal de Pernambuco*, in 1980 and 1983. Since 1983 he has been with the Department of Electronics and Systems (DES-UFPE). In 1992 he earned the *Docteur* degree from *École Nationale Supérieure des Télécommunications*, Paris. Prof. de Oliveira was appointed as honored professor by more than 15 undergraduate classes and he authored the book *Análise de Sinais para Engenheiros: Wavelets* (Manole pub., 2004), ISBN 85-204-1624-1, São Paulo. His research topics include communication theory, information theory and digital signal processing. Dr. de Oliveira is member of IEEE and SBrT.