A SHORT SURVEY ON ARITHMETIC TRANSFORMS AND THE ARITHMETIC HARTLEY TRANSFORM

Renato José de Sobral Cintra and Hélio Magalhães de Oliveira

Abstract - Arithmetic transforms (AT) offer a method for computing trigonometric transforms without multiplications. We survey on the subject throughout this paper, commenting the evolution of techniques and algorithms for the AT. Arithmetic Fourier transform is examined and a new AT named Arithmetic Hartley transform is introduced for computing the discrete Hartley transform.

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

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 effort. Even today, computations that involve multiplication are much time consuming when compared to addition or subtraction. Table 1 shows the clock count of some mathematical operations of Intel Pentium™ processor. Observe that multiplications and divisions can be by far more demanding than additions, for instance. Sine and cosine functions are also shown.

This fact stimulated the research on discrete transform algorithms that minimize the number of multiplications. An example of this is the Bhatnagar’s algorithm [1a], which uses Ramanujan numbers to eliminate multiplications (however, the choice of blocklength is rather limited). Parallel to this, approximation procedures have been proposed. Approximation procedures perform the trade-off accuracy vs. computational effort. The Dee-Jeoti algorithm [2a] and the Rounded Hartley Transform [3a] are good examples of approximate approaches.

Arithmetic transform methods are algorithms that require only addition operations (except for multiplications by scale factors) in other to perform spectral analysis. As it will be shown throughout this paper, the theory of arithmetic transform is essentially based on Möbius function theorems, which offers only trivial multiplications, i.e., multiplications by \{-1, 0, 1\}. In addition to the computational attractiveness, arithmetic transforms are naturally suited for parallel processing and VLSI design.

The very beginning of research on arithmetic transforms dates back to 1903 when the German mathematician Ernst Heinrich Bruns1 published the Grundlinien des wissenschaftlichen Rechnens [3], the fundamental work in this field. This technique remained unnoticed even among mathematicians. Forty-two years later, in Baltimore, U.S.A., the Hungarian Aurel Freidrich Wintner2 published a monograph entitled An Arithmetical Approach to Ordinary Fourier Series. This monograph presents an arithmetic method using Möbius function to calculate the Fourier series of even periodic functions.

Since then, this theory entered again in hibernation, which persisted for 43 years, one year longer than the last one. Not before 1988, when Dr. Donald W. Tufts and Dr. Angaraih G. Sadasiv, independently, had reinvented Wintner’s arithmetical procedure, the theory was once again awoken.

In their quest for implementing 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 already knew about Wintner’s mono-

---

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Clock count</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>1–3</td>
</tr>
<tr>
<td>sub</td>
<td>1–3</td>
</tr>
<tr>
<td>fadd</td>
<td>1–7</td>
</tr>
<tr>
<td>fsup</td>
<td>1–7</td>
</tr>
<tr>
<td>mul (unsigned)</td>
<td>10–11</td>
</tr>
<tr>
<td>mul (signed)</td>
<td>10–11</td>
</tr>
<tr>
<td>div (unsigned)</td>
<td>17–41</td>
</tr>
<tr>
<td>div (signed)</td>
<td>22–46</td>
</tr>
<tr>
<td>fdiv</td>
<td>39</td>
</tr>
<tr>
<td>sin, cos</td>
<td>17–137</td>
</tr>
</tbody>
</table>

---

1 Bruns (1848-1919) got a doctorate in 1871 under supervision of Weierstrass and Kummer.
2 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.
graph and helped Tufts in many discussions. Particularly Dr. Shisha put them in touch with Wintner’s work. In 1988 The Arithmetic Fourier Transform by Tufts and Sadasiv was published in IEEE Acoustic, Speech, and Signal Processing Magazine [6].

Another breakthrough came in early 1990s when Emeritus Professor Dr. Irving S. Reed entered in scene. Although Dr. Reed is better recognised for his work on coding theory — since he is the principal originator of the widely used Reed-Muller (1954) and Reed-Solomon (1964) codes — his interests are 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 endowed with two fundamental contributions [15, 26]. The first one presented a reformulated version of Tufts-Sadasiv approach making the AFT capable to encompass a larger class of signals. After [15], the AFT algorithm would be able to compute Fourier series of odd periodic functions too.

Another crucial slash on the subject was done by Dr. Reed and collaborators in 1992 with the publication of A VLSI Architecture for Simplified Arithmetic Fourier Transform Algorithm in the IEEE Transactions on ASSP [26]. For the sake of truth, it is worthy to mention that this paper was originally presented at the International Conference on Application Specific Array Processors held in Princeton. However, the 1992 publication reached a vastly larger audience, since it was published in a major journal. This work represents an enhancement of the last proposed algorithm. This version could not only handle with even and odd functions, but the algorithm was re-designed to have a more balanced and computationally efficient performance. As a matter of fact, Reed et al. proved that this 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 issues were directed to the number of sampling required to by the algorithm. But later studies showed that these difficulties could be overcome [11].

The conversion of the standard 1-D AFT to a 2-D version was just a matter of time. Many variants were proposed but they follow the same guide lines of the 1-D case [8, 12, 20, 39, 43, 44]. Further research was carried out in order to implement the AFT in different ways. An alternative implementation [32] proposes a “Möbius-function-free AFT”, which — as suggested — removes the Möbius function from the algorithm. Iterative [30] and adaptive approaches [16] were also examined. However, the most popular presentations of the AFT are those found in [15, 18]. Although the main and original motivation of the arithmetic algorithm was the computation of the Fourier Transform, further generalizations were carried out and this arithmetic approach was used to calculate other transforms. Dr. Luc Knockaert of Department of Information Technology at Ghent University, Belgium, found very interesting generalizations on the Bruns procedure, 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) [47, 49]. These works constitute an effort to make arithmetical procedure applicable in the computation of trigonometrical transforms, other than Fourier transform. In particular the AHT computes the discrete Hartley\(^3\) transform: the real, symmetric, Fourier-like discrete transform defined in 1983 by Professor Emeritus 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 chips made possible AFT to leave theoretical constructs and rapidly reached in practice. 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. Implementations were proposed in [14, 17–19, 21–24, 26, 27, 29, 31, 36, 43].

Initial applications of the AFT took place in several areas. To name a few, one could remark pattern matching techniques [28], measurement and instrumentation [37, 41], auxiliary tool for computation of \(z\)-transform [33, 34] and imaging [13].

This paper is organized in two parts. In section 2, we outline the evolution of the Arithmetic Fourier Transform. In section 3, we summarize the major results of the Arithmetic Hartley. Interpolation issues are addressed and many points of the AFT are clarified, particularly the zero-order approximation.

\(^3\)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 \(f(t) \rightarrow F(\nu)\) by
\[
F(\nu) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)(\cos(\nu t) + \sin(\nu t))dt
\]

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

Throughout this section, we present the three major breakthroughs of the Arithmetic Fourier Transform technique. We briefly survey on the AFT via the algorithm devised by Tufts, Sadasiv, Reed et alii. The focus of our discussion is the theoretical groundwork of the AFT.

In the following, we start to describe the algorithms, it is convenient to call attention to two important results frequently used hereafter. However, some preliminaries are required before starting the description of the algorithms. In the following, $k_1|k_2$ 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.

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

If

\[f_n = \sum_{m=1}^{\lceil N/n \rceil} \mu(m)g_{mn},\]  \hspace{1cm} (6)

then

\[g_n = \sum_{k=1}^{\lfloor N/n \rfloor} f_{kn}.\]

This is the finite version of the Möbius inversion formula [4a]. A proof of this can be found in [15].

2.1 TUFTS-SADASIV APPROACH

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

\[v(t) = \sum_{k=1}^{\infty} v_k(t).\]  \hspace{1cm} (7)

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

\[v_k(t) = a_k \cdot \cos(2\pi kt),\]  \hspace{1cm} (8)

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

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

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

\[S_n(t) \triangleq \frac{1}{n} \sum_{m=0}^{n-1} v \left( t - \frac{m}{n} \right),\]  \hspace{1cm} (9)

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

After an application of Equations 7 and 8 into 9, we find:

\[S_n(t) = \frac{1}{n} \sum_{m=0}^{n-1} v \left( t - \frac{m}{n} \right) = \frac{1}{n} \sum_{k=0}^{\infty} a_k \cos \left( 2\pi kt - 2\pi \frac{km}{n} \right),\]

\[= \frac{1}{n} \sum_{k=1}^{\infty} a_k \cos \left( 2\pi \frac{km}{n} \right) \left( \frac{\cos(2\pi kt) \cos \left( 2\pi \frac{km}{n} \right)}{\cos(2\pi \frac{km}{n})} - 1 \right) \]

\[= \frac{1}{n} \sum_{k=1}^{\infty} a_k \cos(2\pi kt) \cdot \begin{cases} n & \text{if } n | k, \\ 0 & \text{otherwise} \end{cases} \]

\[= \sum_{n=1}^{\infty} v_k(t) = \sum_{m=1}^{\infty} v_{mn}(t), \quad n = 1, \ldots, N.\]  \hspace{1cm} (11)

Proceeding that way, we have just expressed the $n$th average 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 11 may possibly be nonnull.

Our task is now to invert Equation 11. Doing so, the harmonics are expressed in terms of the averages, which are derived from the samples of the signal $v(t)$. This inversion can be accomplished by invoking the Möbius inversion formula.
Theorem 2.2 The harmonics of \( v(t) \) can be obtained by:

\[
v_k(t) = \sum_{m=1}^{\infty} \mu(m) S_{mk}(t), \quad \forall k = 1, \ldots, N.
\]

Proof: To obtain this result a little manipulation is needed. Start substituting Equation 11 in Equation 12. Therefore, we find

\[
\sum_{m=1}^{\infty} \mu(m) S_{mk}(t) = \sum_{m=1}^{\infty} \mu(m) \sum_{n=1}^{\infty} v_{kmn}(t).
\]

Now it is the tricky part of the proof. Some further effort leads to the following:

\[
\sum_{m=1}^{\infty} \mu(m) \sum_{n=1}^{\infty} v_{kmn}(t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \mu(m) v_{kmn}(t)
\]

\[
= \sum_{j=1}^{\infty} \bar{v}_j(t) \left( \sum_{m=1}^{\infty} \mu(m) \right).
\]

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 are worthy to be highlighted [6]:

- This initial version of the AFT has a strong constraint: it can only handle even signal.
- All computations are performed using only additions (except for few multiplications due to scaling).
- The algorithm architecture is suitable for parallel processing, since each average is computed independently from the others.
- The AT theory is 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 the former one proposed by Tufts-Sadasiv. The main constraint of the previous procedure (handling only a generalization of the former one proposed by Tufts-Sadasiv) was removed. This opened path for the computation of any Fourier series coefficient of an arbitrary periodic function.

Take a real \( T \)-periodic function \( v(t) \), whose Fourier series is finite with \( N \) terms, given by

\[
v(t) = a_0 + \sum_{n=1}^{N} a_n \cos \left( \frac{2\pi nT}{T} \right) + \sum_{n=1}^{N} b_n \sin \left( \frac{2\pi nT}{T} \right),
\]

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

\[
\bar{v}(t) = v(t) - a_0
\]

\[
= \sum_{n=1}^{N} a_n \cos \left( \frac{2\pi nT}{T} \right) + \sum_{n=1}^{N} b_n \sin \left( \frac{2\pi nT}{T} \right).
\]

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

\[
\bar{v}(t + \alpha T) = \sum_{n=1}^{N} a_n \cos \left( \frac{2\pi nT}{T} + \alpha \right) + \sum_{n=1}^{N} b_n \sin \left( \frac{2\pi nT}{T} + \alpha \right)
\]

\[
= \sum_{n=1}^{N} c_n(\alpha) \cos \left( \frac{2\pi nT}{T} \right) + \sum_{n=1}^{N} d_n(\alpha) \sin \left( \frac{2\pi nT}{T} \right),
\]

where \(-1 < \alpha < 1\)

\[
c_n(\alpha) = a_n \cos(2\pi n\alpha) + b_n \sin(2\pi n\alpha),
\]

\[
d_n(\alpha) = -a_n \sin(2\pi n\alpha) + b_n \cos(2\pi n\alpha).
\]

After such preliminaries, the formula for the \( n \)th average (Tufts-Sadasiv) is updated in the next definition.

Definition 2.3 A \( n \)th average is given by

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

where \(-1 < \alpha < 1\).

It will be shown in the sequel how to compute the Fourier coefficients \( a_n \) and \( b_n \) from \( c_n(\alpha) \). Let us before derive an expression for \( c_n(\alpha) \) in terms of the averages.

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

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

Proof: Substituting the result of Equation 17 into Equation 20, we obtain:

\[
S_n(\alpha) = \sum_{k=1}^{N} c_k(\alpha) \frac{1}{n} \sum_{m=0}^{n-1} \cos \left( \frac{2\pi km}{n} \right)
\]

\[
+ \sum_{k=1}^{N} d_k(\alpha) \frac{1}{n} \sum_{m=0}^{n-1} \sin \left( \frac{2\pi km}{n} \right).
\]

A direct application of Lemma 2.1 takes us to

\[
S_n(\alpha) = \sum_{l=1}^{\lfloor N/n \rfloor} c_n(\alpha).
\]

The theorem is then proved invoking the Möbius inversion formula for finite series Theorem 2.1.

We are now in a position to state the following result.

Theorem 2.4 (Reed-Tufts) The Fourier series coefficients \( a_n \), \( b_n \) are computed by

\[
a_n = c_n(0),
\]

\[
b_n = (-1)^m c_n \left( \frac{1}{2^{k+2}} \right) n = 1, \ldots, N,
\]

where \( k \) and \( m \) are determined by the factorization \( n = 2^k (2m + 1) \).
**Proof:** For $\alpha = 0$, using Equation 18, it is straightforward to show that $a_n = c_n(0)$.

For $\alpha = \frac{1}{2}\pi$ and $n = 2^k(2m+1)$, we have two subcases: $m$ even or odd.

- For $m = 2q$, we have that $n = 2^k(4q+1)$. Therefore,
  \[ 2\pi n\alpha = 2\pi \frac{2^k(4q+1)}{2^{k+2}} = 2\pi q + \frac{\pi}{2}. \tag{26} \]
  Consequently, substituting this quantity in Equation 18, yields
  \[ c_n \left( \frac{1}{2^{k+2}} \right) = a_n \cos \left( 2\pi q + \frac{\pi}{2} \right) + b_n \sin \left( 2\pi q + \frac{\pi}{2} \right) = b_n. \tag{27} \]

- For $m = 2q+1$, we have that $n = 2^k(4q+3)$. It follows that
  \[ 2\pi n\alpha = 2\pi \frac{2^k(4q+3)}{2^{k+2}} = 2\pi q + \frac{3\pi}{2}. \tag{28} \]
  Again invoking the Equation 18, we derive the following expression.
  \[ c_n \left( \frac{1}{2^{k+2}} \right) = a_n \cos \left( 2\pi q + \frac{3\pi}{2} \right) + b_n \sin \left( 2\pi q + \frac{3\pi}{2} \right) = -b_n. \tag{29} \]

Joining these two subcases, one can easily verify that
  \[ b_n = (-1)^m c_n \left( \frac{1}{2^{k+2}} \right). \tag{30} \]

The multiplicative and additive complexities of this algorithm are given by [15]
  \[
  M_R(N) = \frac{3}{2}N, \tag{31} \\
  A_R(N) = \frac{3}{8}N^2, \tag{32}
  \]
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 Reed-Tufts one. Surprisingly, in this new method, the averages are re-defined in accordance with 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
  \[ B_{2n}(\alpha) = \frac{1}{2n} \sum_{m=0}^{2n-1} (-1)^m \cdot v \left( \frac{m}{2n} T + \alpha T \right). \tag{33} \]

Invoking the definition of $c_n$, applying Theorem 2.3 and Definition 2.3, one can derive the following.

**Theorem 2.5** The coefficients $c_n(\alpha)$ are given by the Möbius inversion formula for finite series as
  \[ c_n(\alpha) = \sum_{t=1,3,5,...} \mu(t) \cdot B_{2nt}(\alpha). \tag{34} \]

**Proof:** See [26].

We are almost ready to compute the Fourier series coefficients. We already possess a relation between the signal samples and the Bruns alternating averages and an expression connecting the $c_n$ coefficients and the Bruns alternating averages.

It remains to derive an expression that relates the Fourier series coefficients ($a_n$ and $b_n$) and the coefficients $c_n$. If we take a look at Equation 18, two conditions are distinguishable:

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

That is the final relation. The following final Theorem can be then stated from Theorem 2.5.

**Theorem 2.6 (Reed-Shih)** The Fourier series coefficients $a_n$ and $b_n$ are computed by
  \[
  a_0 = \frac{1}{T} \int_{0}^{T} v(t)dt, \tag{35} \\
  a_n = \sum_{t=1,3,5,...} \mu(t) B_{2nt}(0), \tag{36} \\
  b_n = \sum_{t=1,3,5,...} \mu(t)(-1)^{\left\lfloor \frac{t}{2} \right\rfloor} B_{2nt} \left( \frac{1}{4nt} \right), \tag{37}
  \]
for $n = 1, \ldots, N$.

The proof of this theorem follows the same way as that of Theorem 2.4.

For a blocklength $N$, the multiplicative and additive complexities of this algorithm are given by
  \[
  M_R(N) = N, \tag{38} \\
  A_R(N) = \frac{1}{2}N^2. \tag{39}
  \]

The AFT algorithm proposed by Reed-Shih presents some advancements over the previous one. The chief points points are listed below.

- The computation of both $a_n$ and $b_n$ account for the computational effort. This makes the algorithm more balanced than Reed-Tufts algorithm.
- The algorithm is naturally suited to a parallel processing implementation.
- The computational complexity is smaller than Reed-Tufts algorithm complexity.

### 2.4 A NAIVE EXAMPLE
In this subsection, we add some comments to a toy example of the Reed-Shih algorithm. Consider a signal \( v(t) \) with period \( T = 1 \) s. Suppose that we are interested in computing the Fourier series coefficients up to the 5th harmonic.

The coefficients \( a_n \) and \( b_n \) of the Fourier series of \( v(t) \) are expressed by

\[
\begin{bmatrix}
    a_1 \\
    a_2 \\
    a_3 \\
    a_4 \\
    a_5 \\
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & -1 & 0 & -1 \\
    0 & 1 & 0 & 0 & 0 \\
    0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
    B_2(0) \\
    B_3(0) \\
    B_4(0) \\
    B_5(0) \\
    B_{10}(0) \\
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
    b_1 \\
    b_2 \\
    b_3 \\
    b_4 \\
    b_5 \\
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 1 & 0 & -1 \\
    0 & 1 & 0 & 0 & 0 \\
    0 & 0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
    B_3(\frac{1}{2}) \\
    B_4(\frac{1}{2}) \\
    B_5(\frac{1}{2}) \\
    B_8(\frac{1}{2}) \\
    B_{10}(\frac{1}{2}) \\
\end{bmatrix}
\]

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 are obtained through similar matrices.

After a more careful thought, we can construct a table relating Bruns alternative averages \( B_n(\alpha) \) with the necessary time samples to compute it.

**Table 2. Necessary samples for the Bruns alternating averages.**

<table>
<thead>
<tr>
<th>Bruns averages</th>
<th>Sample time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_2(0) )</td>
<td>0, ( \frac{1}{2} )</td>
</tr>
<tr>
<td>( B_4(0) )</td>
<td>0, ( \frac{1}{2} ), ( \frac{3}{4} )</td>
</tr>
<tr>
<td>( B_6(0) )</td>
<td>0, ( \frac{1}{2} ), ( \frac{3}{4} ), ( \frac{5}{6} )</td>
</tr>
<tr>
<td>( B_8(0) )</td>
<td>0, ( \frac{1}{2} ), ( \frac{3}{4} ), ( \frac{5}{6} ), ( \frac{7}{8} )</td>
</tr>
<tr>
<td>( B_{10}(0) )</td>
<td>0, ( \frac{1}{10} ), ( \frac{3}{10} ), ( \frac{5}{10} ), ( \frac{7}{10} ), ( \frac{9}{10} )</td>
</tr>
<tr>
<td>( B_2(\frac{1}{2}) )</td>
<td>( \frac{1}{4} ), ( \frac{3}{4} )</td>
</tr>
<tr>
<td>( B_4(\frac{1}{2}) )</td>
<td>( \frac{1}{8} ), ( \frac{3}{8} ), ( \frac{5}{8} ), ( \frac{7}{8} )</td>
</tr>
<tr>
<td>( B_6(\frac{1}{12}) )</td>
<td>( \frac{1}{12} ), ( \frac{1}{6} ), ( \frac{1}{3} ), ( \frac{5}{12} ), ( \frac{7}{12} ), ( \frac{11}{12} )</td>
</tr>
<tr>
<td>( B_8(\frac{1}{15}) )</td>
<td>( \frac{1}{15} ), ( \frac{2}{15} ), ( \frac{3}{15} ), ( \frac{4}{15} ), ( \frac{5}{15} ), ( \frac{6}{15} ), ( \frac{7}{15} ), ( \frac{8}{15} )</td>
</tr>
<tr>
<td>( B_{10}(\frac{1}{20}) )</td>
<td>( \frac{1}{20} ), ( \frac{2}{20} ), ( \frac{3}{20} ), ( \frac{4}{20} ), ( \frac{5}{20} ), ( \frac{6}{20} ), ( \frac{7}{20} ), ( \frac{8}{20} ), ( \frac{9}{20} ), ( \frac{10}{20} )</td>
</tr>
</tbody>
</table>

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 for this problem is the approximation. An interpolation based on uniform sampling can be used to estimate the sample values required by AFT.

For example, let assume that the 1Hz signal \( v(t) \) were sampled by an clock with period \( T_0 = \frac{1}{10} \) s. Hence, we pick up the following sample points:

\[
\begin{align*}
    v(0), & \quad v\left(\frac{1}{10}\right), & \quad v\left(\frac{2}{10}\right), & \quad v\left(\frac{3}{10}\right), & \quad v\left(\frac{4}{10}\right), \\
    v\left(\frac{5}{10}\right), & \quad v\left(\frac{6}{10}\right), & \quad v\left(\frac{7}{10}\right), & \quad v\left(\frac{8}{10}\right), & \quad v\left(\frac{9}{10}\right).
\end{align*}
\]

For the sake of simplification, let us focus our attention in the computation of \( B_4(0) \). To calculate it, we would need — among other samples — \( v\left(\frac{1}{2}\right) \), for example (see Table 2). This sample is clearly not available.

To overcome this difficulty, a rounding is introduced. Thus, we would use \( v\left(\frac{1}{2}\right) \) when the algorithm calls for \( v\left(\frac{1}{2}\right) \) (\( \frac{10}{2} \) \( / \) \( 10 = 3/10 \)). 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, one should expect to increase sampling. This makes the rounding to introduce smaller errors.

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:

\[
\text{accuracy} \quad \times \quad \text{order of interpolation}.
\]

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

### 3. A NEW ARITHMETIC TRANSFORM

Searching the literature, we did not found any mention about a possible “Arithmetic Hartley Transform” to compute the discrete Hartley transform (DHT).

Besides its numerical appropriateness [5a], the 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.

In this section, we condense the main results of the Arithmetic Hartley Transform (AHT). The definition of the AHT made some aspects of the arithmetic transform clearer, such as the role of interpolation. It is mathematically shown that interpolation plays a key role in arithmetic transforms, determining the transform.

A new approach to arithmetic transform is adopted. Instead of considering a signal \( v(t) \) which was sampled according to a sample period, the AHT is defined from the purely discrete signal. Thus, the starting point of the development
is the discrete transform, not the series expansion, as it was done in the AFT algorithm.

This approach seems to be philosophically appealing, since in a final analysis a discrete transform relates two sets of points, not continuous functions.

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

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

(42)

where \( \text{cas} x \triangleq \cos x + \sin x \) is Hartley’s “cosine and sine” kernel. The inverse discrete Hartley transform is then

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

(43)

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

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

(44)

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

In order to design an arithmetic algorithm for the DHT evaluation, let us define averages \( S_k \) of the time-domain elements by

\[
S_k \triangleq \frac{1}{k} \sum_{m=0}^{k-1} v_m \cdot \text{cas} \left( \frac{2\pi m}{k} \right), \quad k = 1, \ldots, N - 1.
\]  

(45)

It is interesting to note that this definition requires fractional index sampling (!). As mentioned, this fact seems to make further considerations impracticable, since we have only integer index samples. This subtle question will be treated in the sequel. Let us accept these fractional indexes for a while.

An application of inverse Hartley transform on \( v_m \) at Equation 45 yields:

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

(46)

From Lemma 3.1 above, it follows that:

\[
S_k = \frac{1}{k} \sum_{k'=0}^{N-1} V_{k'} \cdot \text{cas} \left( \frac{2\pi m}{k} \right) = \sum_{s=0}^{\lfloor (N-1)/k \rfloor} V_{sk}.
\]  

(47)

For simplicity and without loss of generality, consider a signal \( v \) with zero mean value, i.e., \( \frac{1}{N} \sum_{i=0}^{N-1} v_i = 0 \). This consideration has no influence on the values of \( V_k, k \neq 0 \).

Then, the Arithmetic Hartley Transform can be derived by the use of modified Möbius inversion formula for finite series [15]. According to Theorem 2.1, we can state the following result.

**Theorem 3.1 (Reed et al.)** If

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

(48)

then

\[
V_k = \sum_{t=1}^{\lfloor (N-1)/k \rfloor} \mu(t) S_{kt},
\]  

(49)

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

Now we are able to handle with zero mean signals, computing its transform. To illustrate, let us consider an 8-point DHT. Using Möbius inversion formula, the spectral analysis is given by:

\[
V_1 = S_1 - S_2 - S_3 - S_5 + S_6 + S_7,
\]  

\[
V_2 = S_2 - S_3 - S_6,
\]  

\[
V_3 = S_3 - S_6,
\]  

\[
V_4 = S_4,
\]  

\[
V_5 = S_5,
\]  

\[
V_6 = S_6,
\]  

\[
V_7 = S_7.
\]  

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

The above theorem and equations completely specifies how to compute the Hartley spectrum. Additionally, the inverse transform can also be established. The following is straightforward.
3.1 INTERPOLATION

In this section, we will see that the definition of the fractional index components, \( v_r, r \notin \mathbb{N} \) is characterized by an interpolation process from the known components (integer index samples). This section is closed with brief comments on the trade-off between accuracy and computational cost required by interpolation process.

3.1.1 IDEAL INTERPOLATION

What does a fractional index discrete signal component really mean? Let \( \mathbf{v} = (v_0, \ldots, v_{N-1})^T \). The value of \( v_r \) for a noninteger value \( r \notin \mathbb{N} \), can be computed by

\[
v_r = \frac{1}{N} \sum_{k=0}^{N-1} V_k \cos\left(\frac{2\pi kr}{N}\right)
\]

The value of the signal at fractional indexes can be found by an \( N \)-order interpolation expressed by:

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

Each transform kernel is associated with a different weighting function. Consequently, a different interpolation process for each weighting function is required. 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. In the cases where \( r \) is an integer number, it follows from the orthogonality properties of \( \text{cas}(\cdot) \) function that \( w_i(r) = 1 \) and \( w_i(r) = 0 \) \( \forall i \neq r \). As expected, there is no need for interpolation.

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

Proposition 1 An \( N \)-point transform has interpolation weighting functions given by

Cosine Kernel

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

Sine Kernel

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

Hartley Kernel

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

With this proposition, we fulfil the mathematical description of the algorithm. The derived formulae furnish the exact value of the spectral components. The computational complexity of this ideal interpolation implementation is similar to the direct implementation, i.e., computing the transform by its definition: \( V_k = \sum_{i=0}^{N-1} v_i \cos\left(\frac{2\pi}{N} ki\right) \).
To exemplify, in Figure 3 we show two weighting functions used to compute $v_{10.1}$ and $v_{10.5}$ during a Hartley transform. These functions were calculated by closed formulae.

3.1.2 NON IDEAL INTERPOLATION

According to the index generation ($m = N/2$), the number $R$ of points which will require interpolation is upper bounded by $R \leq \sum_{d \mid N} d - 1$. This sum represents the number of samples with fractional index. So, this approach is attractive for large non-prime blocklength $N$ with great number of factors, because it requires a smaller number of interpolations.

Our idea is to find simpler formulae for weighting functions, constrained to large blocklength condition. Rather than using exact weighting functions, let us take the limit when $N \to \infty$ and 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**

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

- **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**

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

It is interesting to note that the asymptotic weight for Hartley transform can be written in terms of $\text{Sa}()$’s. Provided that a Hilbert transform is used, the asymptotic weighting function for Hartley kernel is given by

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

or alternatively,

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

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

**Zero-order Interpolation.** Zero-order interpolation is done by rounding the fractional index. The estimated (interpolated) signal $\hat{v}_j$ can be found by a rounding, i.e., $\hat{v}_j = v_{\lfloor j \rfloor}$. 

---

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

**Figure 4.** These curve families represent the values of $w_i(r)$. For this example, $N = 16$, $r = 0.0, 0.1, \ldots, 15$ $i = 0, \ldots, 15$. Observe that $i \in \mathbb{N}$ and $r \in \mathbb{R}$. The maxima values are achieved at $i = 0$ or $i = N/2 = 8$ (central peak) and they correspond to the unity. The value of the local maxima is exactly $1/2$. (a) Weighting profile for the Cosine kernel. Note that each curve is near zero everywhere, except when $i \approx r$ and $i \approx N - r$. (b) Weighting profile for the Hartley kernel.
A Short Survey on Arithmetic Transforms and the Arithmetic Hartley Transform

clearly that it is essentially used in previous work by Tufts, Reed et al.[6, 15, 26].

Equation 53 under such assumptions furnishes:

\[ \hat{w}_r(r) = 0 \quad \forall \ i \neq [r], N - [r], \]
\[ \hat{w}_r(r) \approx \frac{\text{Sa}(2\pi([r] - r))}{2} \approx \frac{1}{2} \]
\[ \hat{w}_{N-r}(r) \approx \frac{\text{Sa}(2\pi([r] - r))}{2} \approx \frac{1}{2} \]

(56)

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:

\[ \hat{v}_r \approx w_r(r)v_r + w_{N-r}(r)v_{N-r} \]
\[ \approx \frac{1}{2} v_r + \frac{1}{2} v_{N-r}. \]

(57)

Thus, for even signal (\(v_k = v_{N-k}\)) we have that 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. Zero-order interpolation is “blind” to odd parts. So, zero-order interpolation is deeply associated with cosine transform. In fact, as show by the set of Equations 56, zero-order interpolation is an (indeed good) approximation to the cosine asymptotic weighting function.

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

Interpolation Order. A simple way to gradually improve the interpolation process is to retain the \(m\) most significant coefficients \(w_i(r)\). For zero-order interpolation, we have clearly that \(m = 1\).

Let us gather the indexes of these \(m < N\) more significant weights in a set \(M_m\). Proceeding this way, a non-ideal interpolation method is to perform the following calculation:

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

(58)

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 definition and by the arithmetic method using \(m = 2\). Note the relatively small blocklength.

4. CONCLUSIONS

This paper supplies a short survey on arithmetic transform. Arithmetic Fourier transform is explained and its three main 1-D versions are described. Some comments on the implementation, challenging points and advantages of the ATs are discussed via a small example...

The purely discrete definition of the AHT led us to arithmetic transforms key point: the interpolation process. We showed that the fundamental equations of the AT algorithms are essentially the same (kernel independent). In addition, we proved that interpolation determines the kind transformation.

This property opens path to the implementation of “universal transformers”. In this kind of construct, the circuitry for different transforms remains unchanged, except for the interpolation module. A different interpolation module would reflect different transform (Fourier, Hartley, Cosine).

This paper can be taken as starting point to those who want to investigate arithmetic transforms.

5. 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


FURTHER REFERENCES

The following references are not primarily concerned with arithmetic transforms. They have been listed apart in order to not contaminating the main bibliography survey.


Renato José de Sobral Cintra was born in Recife. He received the B.Sc. and M.S.E.E degrees in electrical engineering from Universidade Federal de Pernambuco in 1999 and 2001, respectively. Currently he is a doctorate student at the same institution. His research interests include fast algorithms and signal analysis. 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 Analyse de Signaux pour Ingénieurs: 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.