# HOW TO INTERPOLATE IN ARITHMETIC TRANSFORM ALGORITHMS

*R.J. de Sobral Cintra*

Department of Electronics and Systems
Federal University of Pernambuco
rjsc@ee.ufpe.br

*H.M. de Oliveira*

Department of Electronics and Systems
Federal University of Pernambuco
hmo@npd.ufpe.br

## ABSTRACT

In this paper, we propose a unified theory for arithmetic transform of a variety of discrete trigonometric transforms. The main contribution of this work is the elucidation of the interpolation process required in arithmetic transforms. We show that the interpolation method determines the transform to be computed. Several kernels were examined and asymptotic interpolation formulae were derived. Using the arithmetic transform theory, we also introduce a new algorithm for computing the discrete Hartley transform.

## 1. INTRODUCTION

Originally conceived in 1903 by the the German mathematician Ernst H. Bruns, Arithmetic transform theory has arrived to Engineering framework 85 years later, when Donald W. Tufts and G. Sadasiv — independently — rediscovered an algorithm similar to Bruns' method. They called it "*Arithmetic Fourier Transform*" (AFT). Arithmetic transform method has an important advantage over other procedures: it requires only additions operations (except from multiplications by scale factors) [1]. In fact, as the theory is based on Möbius function there will be only trivial multiplications, i.e., multiplications by $\{-1, 0, 1\}$. However, Tufts-Sadasiv algorithm had a serious constraint: it could only calculate the Fourier series coefficient of even periodic signals [2].

In 1990, Tufts-Sadasiv algorithm was revisited and its restrictions were removed by Irving Reed, Donald W. Tufts *et al.* [1]. Now one could use arithmetic transform method to evaluate all Fourier coefficients (even and odd) of an arbitrary periodic function.

Further improvements were made in 1992. Irving Reed, Ming-Tang Shih and co-workers refined the previous AFT algorithm and proposed the "*Simplified AFT*". This new version of AFT could handle with both even and odd Fourier series coefficient, just as its predecessor. Moreover, the algorithm description was made clearer and more symmetric [3]. But the surprising point is that this last version of AFT (Reed-Shih) is identical to the very first one analysis derived by Bruns back in 1903.

Searching the literature, we did not found any mention about a possible "arithmetic Hartley transform" to compute the discrete Hartley transform (DHT). So we start to devise how it would be formulated.

DHT is the discrete version of the symmetric integral transform created by R. V. L. Hartley in 1942. Besides its numerical appropriateness [4], the DHT has proved along the years to be a important tool with several applications, such as biomedical image compression, OFDM/CDMA systems and ADSL transceivers [5].

Now it is our goal to derive an arithmetic method for discrete Hartley transform. As we seek for this new procedure for the DHT evaluation, a general theory was sketched and a family of arithmetic transforms was identified. More than that — and that is the most important point — we found a new interpretation for the the interpolation role in the arithmetic theory.

In this paper, we begin locating recent applications of Hartley transform, then we derive an arithmetic method to evaluate discrete Hartley transform. Then it is mathematically shown that interpolation has a key role in arithmetic transforms, determining the transform. Also, ideal and non-ideal interpolation are investigated. Finally, an arithmetic transform generalization is suggested.

## 2. THE ARITHMETIC HARTLEY TRANSFORM

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

$$V_k \triangleq \frac{1}{N} \sum_{i=0}^{N-1} v_i \cdot \operatorname{cas}\left(\frac{2\pi k i}{N}\right), \qquad k = 0, 1, \ldots, N-1, \tag{1}$$

where $\operatorname{cas} t \triangleq \cos t + \sin t$ is Hartley's "cosine and sine" kernel.

**Lemma 1 (Fundamental Property)** *The function* $\operatorname{cas}(\cdot)$ *satisfies*

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

□

In order to design a fast algorithm for the DHT evaluation, let us define averages $S_k$ of the time-domain elements by

$$S_k \triangleq \frac{1}{k} \sum_{m=0}^{k-1} v_{m\frac{N}{k}}, \qquad k = 1, \ldots, N-1. \tag{3}$$

It is interesting to note that this definition requires fractional index sampling (!). As mentioned before, this fact makes further considerations impractical, 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 $v_{m\frac{N}{k}}$ at Equation 3 yields:

$$S_k = \frac{1}{k} \sum_{k'=0}^{N-1} V_{k'} \sum_{m=0}^{k-1} \operatorname{cas}\left(2\pi m \frac{k'}{k}\right). \tag{4}$$

From Lemma 1 above, it follows that:

$$S_k = \frac{1}{k} \sum_{k'=0}^{N-1} V_{k'} \sum_{m=0}^{k-1} \operatorname{cas}\left(2\pi m (\frac{k'}{k})\right) = \sum_{s=0}^{\lfloor (N-1)/k \rfloor} V_{sk}. \tag{5}$$

For simplicity and without loss of generality, consider a signal $\mathbf{v}$ with zero mean value, i.e., $\frac{1}{N}\sum_{i=0}^{N-1} v_i = 0$. 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 [1].

**Theorem 1 (Möbius Inversion Formula for Finite Series)** *Let $n$ be a positive integer number and $f_n$ a non-null sequence for $1 \leq n \leq N$ and null for $n > N$. If $g_n = \sum_{k=1}^{\lfloor N/n \rfloor} f_{kn}$, then $f_n = \sum_{m=1}^{\lfloor N/n \rfloor} \mu(m) g_{mn}$, where $\lfloor \cdot \rfloor$ is the floor function.* □

According to Theorem 1, we can state the following result.

**Theorem 2 (Reed** *et al.***)** *If*

$$S_k = \sum_{m=1}^{\lfloor (N-1)/k \rfloor} V_{sk}, \quad 1 \leq k \leq N-1, \tag{6}$$

*then*

$$V_k = \sum_{l=1}^{\lfloor (N-1)/k \rfloor} \mu(l) S_{kl}, \tag{7}$$

*where $\mu(\cdot)$ is Möbius function.* □

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

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

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

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

$$v_i = \sum_{l=1}^{\lfloor (N-1)/i \rfloor} \mu(l) \sigma_{il}, \tag{8}$$

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

A careful examination of the above makes us to come to a truly remarkable point: the original Arithmetic *Fourier* Transform has identical equations to those we have just derived for a *Hartley* transform. Just compare Equations found in [2] and Equation 6. An important question arises: if the equations are the same, which spectrum is being evaluated? Fourier or Hartley spectrum? Clear understanding the underlying mechanisms of arithmetic theory will be possible in the next section. Once more we beg the reader to put this "philosophical" question aside for a while and carry on our developments.

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

Usual arithmetic theory deals with spectrum approximations via zero- or first-order interpolation [6, 3, 1]. The analysis presented in this work allows a more encompassing perception of the interpolation mechanisms and gives mathematical tools for establishing validation constraints to such interpolation process.

At this point, we have established the arithmetic transform formula as seen on Equation 7. The arithmetic transform algorithm can be summarized in four major steps:

1. Index generation, i.e., to find out the indexes of necessary samples ($m\frac{N}{k}$).
2. Fractional index samples handling, which requires an interpolation.
3. Computation of the 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 this work we are concerned with step two. In the sequel, we will derive a mathematical method which fully explains the real importance of this algorithm step.

Now let us examine the fractional indexes and derive a method to evaluate them. We will address to the deepest nature of the arithmetic transform: the interpolation process.

## 3. INTERPOLATION

### 3.1. 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 non-integer value $r$, $r \notin \mathbb{N}$, can be computed by

$$\begin{aligned}
v_r &= \sum_{k=0}^{N-1} V_k \operatorname{cas}\left(\frac{2\pi k r}{N}\right) \\
&= \frac{1}{N} \sum_{i=0}^{N-1} v_i \sum_{k=0}^{N-1} \operatorname{cas}\left(\frac{2\pi k i}{N}\right) \operatorname{cas}\left(\frac{2\pi k r}{N}\right).
\end{aligned} \tag{9}$$

Defining the *Hartley weighting function* by

$$w_i(r) \triangleq \frac{1}{N} \sum_{k=0}^{N-1} \operatorname{cas}\left(\frac{2\pi k i}{N}\right) \operatorname{cas}\left(\frac{2\pi k r}{N}\right), \tag{10}$$

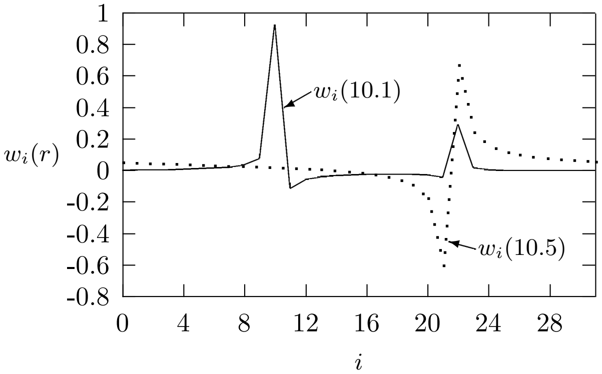

**Fig. 1**. Hartley weighting functions used to interpolate $v_{10.1}$ and $v_{10.5}$ ($N = 32$ blocklength).

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

$$v_r \triangleq \sum_{i=0}^{N-1} w_i(r) \cdot v_i. \tag{11}$$

The same way for each transform there is a kernel associated, for each kernel there is weighting function associated. Consequently, a different interpolation process for each weighting function is required. This is the way our equations were the same. The difference form 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. As matter of fact, $|w_i(r)| \leq 1$. In the cases where $r$ is an integer number, it follows from the orthogonality properties of $\mathrm{cas}(\cdot)$ function that $w_r(r) = 1$ 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 $\mathrm{Sa}(\cdot)$, $\mathrm{Sa}(x) \triangleq \frac{\sin x}{x}$.

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

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

*Sine Kernel*

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

*Hartley Kernel*

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

□

With this proof, we complete the mathematical description of the algorithm. At this point, 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: $V_k = \sum_{i=0}^{N-1} v_i \,\mathrm{cas}\left(\frac{2\pi}{N}ki\right)$.

To exemplify, in Figure 1 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.

## 3.2. Non Ideal Interpolation

According to the index generation ($m\frac{N}{k}$), we find that the number of points $R$ which will require some kind of interpolation is upper bounded by $R \leq \sum_{d\nmid N} d - 1$. This sum represents the number of samples with fractional index. So, this approach is attractive for large non-prime blocklenght $N$ with great number of factors, because it will require a smaller number of interpolations.

Our idea is to find simpler formulae for weighting functions, constrained to large blocklegth condition. Rather than using exact weighting function formulae, 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*

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

*Sine Kernel*

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

*Hartley Kernel*

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

□

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

$$\hat{w}_i(r) \approx \frac{\mathrm{Sa}(2\pi(i-r))}{2} - \mathcal{H}il\Big\{\,\mathrm{Sa}(2\pi(i+r))\Big\}, \tag{12}$$

where $\mathcal{H}il$ denotes the Hilbert transform.

### *3.2.1. Zero-order Interpolation*

Zero-order interpolation is done by rounding the fractional index. The estimated (interpolated) signal $\hat{v}_j$ will be found by a simple rounding, i.e., $\hat{v}_j = v_{[j]}$, 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:

$$\begin{aligned} \hat{w}_i(r) &\approx 0 \quad \forall\, i \neq [r], N-[r], \\ \hat{w}_{[r]}(r) &\approx \frac{\mathrm{Sa}\left(2\pi([r]-r)\right)}{2} \approx \frac{1}{2}, \\ \hat{w}_{N-[r]}(r) &\approx \frac{\mathrm{Sa}\left(2\pi([r]-r)\right)}{2} \approx \frac{1}{2}. \end{aligned} \tag{13}$$

An essential observation is that $v_r \triangleq \sum_{i=0}^{N-1} w_i(r) \cdot v_i$, because it yields

$$\begin{aligned} \hat{v}_r &\approx w_{[r]}(r)v_{[r]} + w_{N-[r]}(r)v_{N-[r]} \\ &\approx \frac{1}{2}v_{[r]} + \frac{1}{2}v_{N-[r]}. \end{aligned} \tag{14}$$

Thus, for even signal ($v_k = v_{N-k}$) we have that the approximated value of the interpolated sample is roughly given by $\hat{v}_r \approx v_{[r]}$.

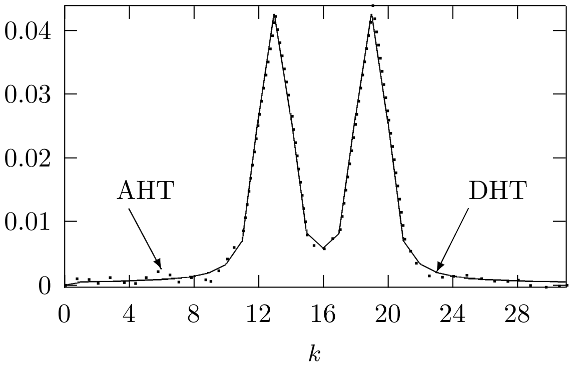

**Fig. 2**. Comparison of Hartley transform of an arbitrary $f(x)$ computed by definition and by arithmetic algorithm. Simulation data: $f(x) = \cos(90\pi x)\left(x - \frac{1}{2}\right)^2$, $N = 32$.

It is straightforward to see that the influence of odd part of the signal is annulled by 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 13, 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.* [1, 2, 3]. Hsu, in his Ph.D. dissertation, derives an analysis of first-order interpolation effect [6].

### *3.2.2. Interpolation Order*

A simple and naive way to control interpolation process is use only the $t$ more significant values of $w_i(r)$. For zero-order interpolation, we have clearly that $t = 1$.

Let us get the indexes of these $t$ more significant weight in a set $T_t$. Proceeding this way, a non-ideal interpolation method is to perform the following calculation:

$$\hat{v}_r = \frac{1}{\eta} \sum_{i \in T_t} w_i(r) \cdot v_i, \qquad (15)$$

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

In Figure 2, we present a 32-point discrete Hartley transform of an arbitrary signal computed by definition and by the arithmetic method using $t = 2$. Note the small blocklength.

## 4. GENERALIZATION

A kernel invariant approach to discrete transforms can be obtained in a simple way.

Consider a discrete transform with $\Psi_N$ as kernel:

$$V_k = \frac{1}{N} \sum_{i=0}^{N-1} v_i \Psi_N(k, i), \qquad k = 0, \ldots, N-1. \qquad (16)$$

For instance we will consider only the following kernels

$$\Psi_N(k, i) \in \left\{ \text{cas}\left(\frac{2\pi k i}{N}\right), \cos\left(\frac{2\pi k i}{N}\right), e^{-j\frac{2\pi k i}{N}} \right\}. \qquad (17)$$

**Lemma 2 (Generalization of Lemma 1)**

$$\frac{1}{k} \sum_{m=0}^{k-1} \Psi_N\left(\frac{k'}{k}, mN\right) = \begin{cases} 1 & \textit{se } k|k', \\ 0 & \textit{otherwise.} \end{cases} \qquad (18)$$

□

The arithmetic transform has the same formulation for all transforms in a certain class. The difference from one transform to another is the interpolation process used. The fractional index samples will be estimated in a difference way for each transform, since the interpolation depends on the kernel.

## 5. CONCLUSION

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

It has not escaped our notice that this property opens path to the implementation of "universal transformers". In this kind of construct, the circuitry for different transforms remains unchanged, except from the interpolation module. A different interpolation module would reflect different transform (Fourier, Hartley, Cosine).

Closed formulae for interpolation of several transforms were derived. For large blocklength, we proposed asymptotic approximations. These considerations made the interpolation formulae very simple.

## 6. ACKNOWLEDGEMENT

The first author would like to acknowledge the support of Prof. Irving S. Reed in calling attention to [6].

## 7. REFERENCES

[1] 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*, vol. 38, pp. 459–470, Mr 1990.

[2] Donald W. Tufts and G. Sadasiv, "The Arithmetic Fourier Transform," *IEEE ASSP Magazine*, pp. 13–17, 1988.

[3] 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*, vol. 40, pp. 1122–1133, My 1992.

[4] Ronald N. Bracewell, *The Hartley Transform*, Oxford, 1986.

[5] C. L. Wang and C. H. Chang, "A Novel DHT-based FFT/IFFT Processor for ADSL Transceivers," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 1, pp. 51–54, 1999.

[6] Chin-Chi Hsu, *Use of Number Theory and Modern Algebra in the Reed-Solomon Code and the Arithmetic Fourier Transform*, Ph.D. thesis, University of Southern California, Ag 1994.