On Fast Finite Field Hartley Transform Algorithms

Authors

- R. M. Campello de Souza, ricardo@npd.ufpe.br
- R. G. F. Távora, rota58@hotmail.com
- D. Silva, danilo_s@uol.com.br
- H. M. de Oliveira, hmo@npd.ufpe.br

Affiliation
CODEC - Communications Research Group

Address
Departamento de Eletrônica e Sistemas, CTG - UFPE,
C.P. 7800, 50711 - 970, Recife - PE , Brasil

Phone +55 81 32718210

Fax +55 81 32718215

Contacting Author R. M. Campello de Souza.

On Fast Finite Field Hartley Transform Algorithms

R. M. Campello de Souza, R. G. F. Távora, D. Silva, H.M. de Oliveira

CODEC- Communications Research Group
Departamento de Eletrônica e Sistemas
C.P. 7800, 50.711-970 Recife - PE BRAZIL
{Ricardo, hmo}@npd.ufpe.br

Abstract New fast algorithms over finite fields are investigated which compute the Hartley transform. These algorithms are derived from properties of finite field trigonometry. Additive and multiplicative complexities are calculated in each case.

1 Introduction

Discrete Transforms defined over finite fields are powerful tools in Electrical Engineering, particularly in Digital Signal Processing. The Finite Field Fourier Transform (FFFT) has applications in many fields, including Digital Signal Processing [1] and Error-Control Codes [2]. Another interesting example is the Finite Field Hartley Transform (FFHT), an involutionary (self-inverse) transform introduced by Campello et al. [3][6]. Recent promising applications of discrete transforms concern the use of FFHT to design digital multiplex systems, efficient multiple access systems [4] and multilevel spread spectrum sequences [5]. In this paper we derive fast algorithms, such as Cooley-Tukey radix-2 with frequency decimation, Rader-Brenner, Cooley-Tukey radix-4 with time decimation, split radix, Winograd and Prime Factor, to compute the FFHT. The set $G(q)$ of Gaussian integers over $GF(q)$ plays an important role in this analysis. This set defines a structure $GI(q)$, which is a finite field isomorphic to $GF(q^2)$ [3].

Definition 1.1 *Let ζ be an element of $GI(q)$ with multiplicative order N, where $q = p^r$, p is a prime, $p \equiv 3 \pmod 4$. The trigonometric functions sin, cos, cas are defined by:*

$$\sin(i) = \frac{\zeta^i - \zeta^{-i}}{2j},\ \cos(i) = \frac{\zeta^i + \zeta^{-i}}{2},$$
$$\textit{and}\ \ \mathrm{cas}(i) = \sin(i) + \cos(i).$$

Definition 1.2 *Let $v = \{v_0, v_1, \ldots, v_N\}$ be a vector of $GF(q)$-valued components, $q = p^r$, $p \equiv 3 \pmod 4$. The Finite Field Hartley Transform (FFHT) of v is the vector $V = \{V_0, V_1, \ldots, V_N\}$, with components $V_k \in GI(q^m)$ given by $V_k = \sum_{i=0}^{N-1} v_i cas(ik)$, where ζ is an element of multiplicative order N over $GI(q^m)$.*

The inverse FFHT is given by the following [6]

Theorem 1.1 *The vector $v = \{v_0, v_1, \ldots, v_N\}$ can be derived from its FFHT according to: $v_i = \frac{1}{N \pmod p} \sum_{k=0}^{N-1} V_k cas(ik)$.*

2 A Prime Factor fast FFHT

Let V be an N-blocklength Hartley transform. The argument ζ of cas has order N over $GF(p)$, $\zeta^N = 1 \pmod p$. Supposing that N has two prime factors, N' and N'', indexes can be redefined so as to break an one-dimensional FFHT of length N into a two-dimensional FFHT in the same way as performed in the classical Fourier transform. Letting $i' = i \pmod{N'}$ and $i'' = i \pmod{N''}$, it follows by the Chinese Remainder Theorem that $i = i'N''n'' + i''N'n' \pmod N$, where

$$N'n' \equiv 1 \pmod{N''}, \quad \text{and} \quad N''n'' \equiv 1 \pmod{N'}.$$

Defining

$$k' \equiv n''k \pmod{N'}, \text{ and } k'' \equiv n'k \pmod{N''},$$

therefore $k \equiv N''k' + N'k'' \pmod N$. The product nk can be written in terms of

$$nk \equiv (i'N''n'' + i''N'n')(N''k' + N'k'') \pmod N,$$

that is,

$$nk \equiv (i'n''k'N''N'' + i''n'k''N'N') \pmod N$$

since $n'n'' = N$. Therefore, the spectral component V_k can be calculated according to

$$V(k', k'') = \sum_{i'=0}^{N'} \sum_{i''=0}^{N''} v(i', i'') \,\mathrm{cas}(k'i'n''N''N'' + \\ + i''n'k''N'N').$$

From the sum property $\mathrm{cas}(a + b) = \mathrm{cas}(a)\,\mathrm{cas}(b) - 2\sin(a)\sin(b)$,

$$V(k', k'') = \sum_{i'=0}^{N'} \sum_{i''=0}^{N''} v(i', i'')[\mathrm{cas}(k'i'n''N''N'') \\ \mathrm{cas}(i''n'k''N'N') - 2\sin(k'i'n''N''N'')\sin(i''n'k''N'N')]. \quad (1)$$

Defining cas'(.) and cas''(.), which have argument $\zeta' = \zeta^{n''N''N''}$ and $\zeta'' = \zeta^{n'N'N'}$, the order of ζ'' is N' since $N'' \mid N$ and $\gcd(n''N'', N') = 1$. Also, $ord(\zeta') = N''$

by the same reasoning. Let now $\sin'(.)$ and $\sin''(.)$ be the trigonometric functions with argument ζ' and ζ'', respectively. Eqn(1) can be rewritten as:

$$V(k',k'') = \sum_{i'=0}^{N'} \sum_{i''=0}^{N''} v(i',i'')[\mathrm{cas}'(k'i')\,\mathrm{cas}''(i''k'') - \\ -2\sin'(k'i')\sin''(i''k'')].$$

The above expression is not a two-dimensional transform due to the second term in the double summation. However, computing first the one-dimensional FFHT on i'', such a summation can be evaluated as a two-dimensional FFHT. Defining $\widehat{v}$ as

$$\widehat{v}(k',i'') = \tfrac{1}{2}[v(k',i'') + v(N'-k',i'') + \\ +v(k',N''-i'') - v(N'-k',N''-i'')], \\ 0 \le k' \le N'-1, 0 \le k'' \le N''-1,$$

another one-dimensional FFHT can be then carried out on i', resulting

$$V(k',k'') = \sum_{i'=0}^{N'} \widehat{v}(i',k'')\,\mathrm{cas}'(k'i').$$

This can be verified by a direct substitution of the expression for $\widehat{v}(i',k'')$ into $V(k',k'')$. An amount of $2N$ additions is needed in such a derivation. Thus, the total complexity is the same as computing the FFFT by the Prime Factor algorithm and then evaluating the FFHT. Another interesting version of such an algorithm to compute the DHT was introduced by W.S.Siu et al. [8], where the $2N$ additions are embedded into the inner transform. Let $M(N)$ (respectively, $A(N)$) be the number of multiplications (respectively, additions) required to compute an N-FFHT. The algorithm complexity for computing an N-FFHT, $N = N'N''$, satisfy $M(N'N") = N"M(N') + N'M(N")$ and $A(N'N") = N"A(N') + N'A(N") + 2N$. The number of additions is greater than the one of the FFFT. The multiplicative complexity can be reduced by using the Kronecker product [9][p. 247] [7].

3 A Cooley-Tukey-radix-2-like fast FFHT

Let x be a sequence of length N with elements over $GF(p)$. Its Hartley transform over $GI(p^m)$ is:

$$H(k) = \sum_{i=0}^{N-1} x(i)\,\mathrm{cas}(ki). \tag{2}$$

If $N = KL$, indexes can be redefined according to

$$i = i_1 + Li_2,\ k = k_1 + Kk_2.$$

Applying this index redefinition into eqn(2), it follows that

$$H(k_1+Kk_2) = \\ = \sum_{i_2=0}^{K-1}\sum_{i_1=0}^{L-1} x(i_1+Li_2)\,\mathrm{cas}((k_1+Kk_2)(i_1+Li_2)) = \\ = \sum_{i_2=0}^{K-1}\sum_{i_1=0}^{L-1} x(i_1+Li_2)\,\mathrm{cas}(k_1i_1 + Kk_2i_1 + k_1Li_2). \tag{3}$$

Supposing N even a frequency decimation can be obtained by letting $L = \frac{N}{2}$ and $K = 2$

$$H(k_1+2k_2) = \sum_{i_1=0}^{\frac{N}{2}-1}\sum_{i_2=0}^{1} x(i_1+\tfrac{N}{2}i_2) \\ \mathrm{cas}(k_1i_1 + 2k_2i_1 + k_1\tfrac{N}{2}i_2) = \\ = \sum_{i_1=0}^{\frac{N}{2}-1}[x(i_1)\,\mathrm{cas}(k_1i_1+2k_2i_1) + \\ +x(i_1+\tfrac{N}{2})\,\mathrm{cas}(k_1i_1+2k_2i_1+k_1\tfrac{N}{2})]. \tag{4}$$

At $k_1 = 0$,

$$H(2k_2) = \\ = \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1)\,\mathrm{cas}(2k_2i_1) + x(i_1+\tfrac{N}{2})\,\mathrm{cas}(2k_2i_1)\right] = \\ = \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1)+x(i_1+\tfrac{N}{2})\right]\mathrm{cas}(2k_2i_1).$$

At $k_1 = 1$,

$$H(2k_2+1) = \sum_{i_1=0}^{\frac{N}{2}-1}[x(i_1)\,\mathrm{cas}(i_1+2k_2i_1) + \\ +x(i_1+\tfrac{N}{2})\,\mathrm{cas}(i_1+2k_2i_1+\tfrac{N}{2})]. \tag{5}$$

Applying the property $\mathrm{cas}(a+N/2) = -\mathrm{cas}(a)$ into (5) leads to

$$H(2k_2+1) = \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1) - x(i_1+\frac{N}{2})\right]\mathrm{cas}(i_1+2k_2i_1).$$

But $\mathrm{cas}(i_1 + 2k_2i_1) = \cos(i_1)\,\mathrm{cas}(2k_2i_1) + \sin(i_1)\,\mathrm{cas}(-2k_2i_1)$, so that

$$H(2k_2+1) = \\ = \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1) - x(i_1+\tfrac{N}{2})\right]\cos(i_1)\,\mathrm{cas}(2k_2i_1) + \\ + \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1) - x(i_1+\tfrac{N}{2})\right]\sin(i_1)\,\mathrm{cas}(-2k_2i_1). \tag{6}$$

Letting now $i_1' = \frac{N}{2} - i_1$, the second summation above becomes

$$\sum_{i_1'=1}^{\frac{N}{2}}\left[x(\frac{N}{2}-i_1') - x(-i_1')\right]\sin(\frac{N}{2}-i_1')\,\mathrm{cas}(-2k_2(\frac{N}{2}-i_1')).$$

But $\sin(\frac{N}{2} - i_1')=\sin(i_1')$ and $\mathrm{cas}(-2k_2(\frac{N}{2} - i_1')) = \mathrm{cas}(2k_2i_1')$. Then the expression above becomes

$$\sum_{i_1'=1}^{\frac{N}{2}}\left[x(\frac{N}{2}-i_1') - x(-i_1')\right]\sin(i_1')\,\mathrm{cas}(2k_2i_1')). \tag{7}$$

At $i_1' = \frac{N}{2}$, $[x(0) - x(\frac{N}{2})]\sin(\frac{N}{2})\,\mathrm{cas}(k_2N) = 0$, and at $i_1' = 0$, $[x(\frac{N}{2}) - x(0)]\sin(0)\,\mathrm{cas}(0) = 0$. The index $i_1\prime$ in eqn(7) can be adjusted

$$\sum_{i_1'=0}^{\frac{N}{2}-1}\left[x(\frac{N}{2}-i_1') - x(-i_1')\right]\sin(i_1')\,\mathrm{cas}(2k_2i_1').$$

Putting this into eqn(6) and gathering summations

$$H(2k_2+1) = \\ = \sum_{i_1=0}^{\frac{N}{2}-1}\left[x(i_1) - x(i_1+\tfrac{N}{2})\right]\cos(i_1)\,\mathrm{cas}(2k_2i_1) + \\ + \sum_{i_1=0}^{\frac{N}{2}-1}[x(\tfrac{N}{2}-i_1) - \\ -x(N-i_1)]\sin(i_1)\,\mathrm{cas}(2k_2i_1).$$

The multiplicative and additive complexity, respectively $M(N)$ and $A(N)$, satisfy $M(N) = 2M(\frac{N}{2}) + N$ and $A(N) = 2A(\frac{N}{2}) + \frac{3N}{2}$. Consequently, the algorithm achieves the same multiplicative complexity as the FFFT, despite presenting a greater additive complexity compared to FFFT. Reducing the number of multiplication by means of a scheme equivalent to a complex multiplication, the multiplicative and additive complexities are $M(N) = N\log_2(N) - 3N + 4$ and $A(N) = \frac{3N\log_2 N - 3N + 4}{2}$.

4 A Rader-Brenner-like fast FFHT

The Rader-Brenner algorithm allows to compute eqn(5). The blocklength is N, and $L = \frac{N}{2}$.

$$H(2k+1) = \sum_{i=0}^{L-1}\{[x(i) - x(L+i)]\cos(i) + \\ +[x(L-i) - x(N-i)]\sin(i)\}\,\mathrm{cas}(ik),\ 0 \le k \le L-1.$$

In order to do so, an auxiliary sequence a_i is defined by

$$a_i = \begin{cases} 0, & i = 0 \\ \frac{x(L-i)-x(N-i)}{-2\sin(i)}, & i = 1,\ldots,L-1. \end{cases}$$

Such a sequence a_i is well-defined since $\sin(i) \neq 0$, for $i = 1,\ldots,L-1$, and the values of $\sin(i)$ have inverse once they belong to $GF(p^2)$. In contrast with the DHT, the accuracy of computing a term of the sequence remains unchanged over a finite field. The FFHT of a_i , $A(k) = FFHT(a_i)$, for $k = 0,\ldots,L-1$, is

$$A(k) = \sum_{i=0}^{L-1} a_i\,\mathrm{cas}(2ik) = \\ = \sum_{i=1}^{L-1} \frac{x(L-i)-x(N-i)}{-2\sin(i)}\,\mathrm{cas}(2ik),$$

so that

$$A(k+1) = \sum_{i=0}^{L-1} a_i\,\mathrm{cas}\left(2i(k+1)\right), k = 0,\ldots,L-1.$$

Applying the $cas(.)$ property

$$\mathrm{cas}(2i(k+1)) = \cos(2i)\,\mathrm{cas}(2ik) + \sin(2i)\,\mathrm{cas}(-2ik)$$

into the expression of $A(k+1)$, it follows that

$$A(k+1) = \sum_{i=0}^{L-1} a_i\,[\cos(2i)\,\mathrm{cas}(2ik) + \\ +\sin(2i)\,\mathrm{cas}(-2ik)] = \sum_{i=0}^{L-1} a_i\cos(2i)\,\mathrm{cas}(2ik) + \\ +\sum_{i=0}^{L-1} a_i\sin(2i)\,\mathrm{cas}(-2ik).$$

By the change of variables $n = L - i$,

$$a_i = a_{L-n} = \begin{cases} 0, & n = L \\ \frac{x(n)-x(n+L)}{-2\sin(n)}, & n = 1,\ldots,L-1, \end{cases}$$
$$\sin(2i) = \sin\left(2(L-n)\right) = -\sin(2n),$$
$$\mathrm{cas}(-2ik) = \mathrm{cas}\left(-2k(L-n)\right) = \mathrm{cas}(2kn),$$

the second summation becomes:

$$\sum_{n=1}^{L-1}[x(n) - x(n+L)]\frac{\sin(2n)}{2\sin(n)}\,\mathrm{cas}(2kn).$$

Replacing this into the expression of $A(k+1)$,

$$A(k+1) = \sum_{i=1}^{L-1}\{[x(i) - x(i+L)]\frac{\sin(2i)}{2\sin(i)} + \\ +[x(L-i) - x(N-i)]\frac{\cos(2i)}{-2\sin(i)}\}\,\mathrm{cas}(2ki).$$

On the other hand, $i \neq 0$,

$$\frac{\sin(2i)}{2\sin(i)} = \frac{2\sin(i)\cos(i)}{2\sin(i)} = \cos(i),$$
$$\frac{\cos(2i)}{-2\sin(i)} = \frac{1-2\sin^2(i)}{-2\sin(i)} = \sin(i) - \frac{1}{2\sin(i)},$$

which implies

$$A(k+1) = \sum_{i=1}^{L-1}\{[x(i) - x(i+L)]\cos(i) + \\ +[x(L-i) - x(N-i)]\sin(i)\}\,\mathrm{cas}(2ik) - \\ -\sum_{i=1}^{L-1}\frac{[x(L-i)-x(N-i)]}{2\sin(i)}\,\mathrm{cas}(2ik).$$

Finally,

$$A(k+1) - A(k) = \\ = \sum_{i=1}^{L-1}\{[x(i) - x(i+L)]\cos(i) + \\ +[x(L-i) - x(N-i)]\sin(i)\}\,\mathrm{cas}(2ik).$$

Comparing the above expression with $H(2k+1)$, the following relationship can be derived:

$$H(2k+1) = A(k+1) - A(k) + [x(0) - x(\frac{N}{2})].$$

This approach halves the number of multiplications. The complexity satisfy $M(N) = \frac{N}{2} - 2 + 2M(\frac{N}{2})$ and $A(N) = 2N + 2A(\frac{N}{2})$. Both the Rader-Brenner multiplicative and additive complexity of the FFHT are the same as in the FFFT case. The complexities are given by $M(2^m) = 2^{m-1}(m-3) + 2$ and $A(2^m) = 2^{m-1}(3m-5) + 8$.

5 A Cooley-Tukey-radix-4-like fast FFHT

In the case $4 \mid N$, a time-decimation can be derived by letting $L = 4$ and $K = \frac{N}{4}$ in eqn(3):

$$H(k) = \sum_{i_2=0}^{\frac{N}{4}-1}\sum_{i_1=0}^{3} x(i_1 + 4i_2)\,\mathrm{cas}(k(i_1 + 4i_2)) = \\ = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2)\,\mathrm{cas}(4ki_2) + \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+1) \\ \mathrm{cas}(k(4i_2+1)) + \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+2)\,\mathrm{cas}(k(4i_2+2)) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+3)\,\mathrm{cas}(k(4i_2+3)).$$

It follows that

$$H(k) = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2)\,\mathrm{cas}(4ki_2) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+1)\,\mathrm{cas}(4ki_2)\cos(k) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+1)\,\mathrm{cas}(-4ki_2)\sin(k) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+2)\,\mathrm{cas}(4ki_2)\cos(2k) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+2)\,\mathrm{cas}(-4ki_2)\sin(2k) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+3)\,\mathrm{cas}(4ki_2)\cos(3k) + \\ +\sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+3)\,\mathrm{cas}(-4ki_2)\sin(3k),$$

since $\mathrm{cas}(a+b) = \cos(a)\,\mathrm{cas}(b) + \sin(a)\,\mathrm{cas}(-b)$. Defining

$$H_{4n+1}(k) = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+1)\,\mathrm{cas}(4ki_2),$$
$$H_{4n+2}(k) = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+2)\,\mathrm{cas}(4ki_2),$$
$$H_{4n+3}(k) = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2+3)\,\mathrm{cas}(4ki_2),$$

the Hartley spectral component $H(k)$ can be rewritten as:

$$H(k) = \sum_{i_2=0}^{\frac{N}{4}-1} x(4i_2)\,\mathrm{cas}(4ki_2) + \\ + \cos(k)H_{4n+1}(k) + \sin(k)H_{4n+1}(N-k) + \\ + \cos(2k)H_{4n+2}(k) + \sin(2k)H_{4n+2}(N-k) + \\ + \cos(3k)H_{4n+3}(k) + \sin(3k)H_{4n+3}(N-k).$$

In this case, the multiplicative and the additive complexities satisfy

$$M(N) = 4M(\tfrac{N}{4}) + \tfrac{3N}{2} \text{ and } A(N) = 4A(\tfrac{N}{4}) + 6N.$$

The multiplicative and additive complexities are

$$M(N) = \tfrac{3N}{2}\log_4 N - \tfrac{7N}{3} + \tfrac{10}{3},$$
$$A(N) = \tfrac{11N}{4}\log_4 N - \tfrac{5N}{4} + 2.$$

6 A Split-Radix-like fast FFHT

If $4 \mid N$, let $L = \frac{N}{4}$ and $K = 4$. Therefore, eqn(3) can be computed using a base-2 decomposition (considering $k_1 = 0$ and $k_1 = 2$), or a base-4 decomposition (considering $k_1 = 1$ and $k_1 = 3$). Thus, $H(4k_2 + k_1)$, for k_1 even, is given by:

$$H(2k_2) = \sum_{i_1=0}^{\frac{N}{2}-1} \left[x(i_1) + x(i_1 + \frac{N}{2}) \right] \mathrm{cas}(2k_2 i_1).$$

At $k_1 = 1$:

$$H(4k_2+1) = \sum_{i_1=0}^{\frac{N}{4}-1} \sum_{i_2=0}^{3} x(i_1 + \frac{N}{4}i_2)\,\mathrm{cas}(4k_2 i_1 + i_1 + \frac{N}{4}i_2).$$

Expanding the inner-summation,

$$H(4k_2+1) = \sum_{i_1=0}^{\frac{N}{4}-1} [x(i_1)\,\mathrm{cas}(4k_2 i_1 + i_1) + \\ + x(i_1 + \tfrac{N}{4})\,\mathrm{cas}(4k_2 i_1 + i_1 + \tfrac{N}{4}) + \\ + x(i_1 + \tfrac{N}{2})\,\mathrm{cas}(4k_2 i_1 + i_1 + \tfrac{N}{2}) + \\ + x(i_1 + \tfrac{3N}{4})\,\mathrm{cas}(4k_2 i_1 + i_1 + \tfrac{3N}{4})].$$

Since $\mathrm{cas}(a + \frac{N}{2}) = -\mathrm{cas}(a)$, it follows that

$$H(4k_2+1) = \\ = \sum_{i_1=0}^{\frac{N}{4}-1} \{[x(i_1) - x(i_1 + \tfrac{N}{2})]\,\mathrm{cas}(4k_2 i_1 + i_1) + \\ + [x(i_1 + \tfrac{N}{4}) - x(i_1 + \tfrac{3N}{4})]\,\mathrm{cas}(4k_2 i_1 + i_1 + \tfrac{N}{4})\}.$$

However, $\mathrm{cas}(a + \frac{N}{4}) = \mathrm{cas}(-a)$, which implies

$$\sum_{i_1=0}^{\frac{N}{4}-1} [x(i_1 + \tfrac{N}{4}) - x(i_1 + \tfrac{3N}{4})]\,\mathrm{cas}(4k_2 i_1 + i_1 + \tfrac{N}{4}) = \\ = \sum_{i_1=0}^{\frac{N}{4}-1} [x(i_1 + \tfrac{N}{4}) - x(i_1 + \tfrac{3N}{4})]\,\mathrm{cas}(-4k_2 i_1 - i_1).$$

Replacing $\mathrm{cas}(-4k_2 i_1 - i_1) = \cos(i_1)\,\mathrm{cas}(-4k_2 i_1) - \sin(i_1)\,\mathrm{cas}(-4k_2 i_1)$ into the expression of $H(4k_2+1)$,

$$H(4k_2+1) = \sum_{i_1=0}^{\frac{N}{4}-1} \mathrm{cas}(4k_2 i_1)\{[x(i_1) - x(i_1 + \tfrac{N}{2})] \\ \cos(i_1) + [x(i_1 + \tfrac{3N}{4}) - x(i_1 + \tfrac{N}{4})]\sin(i_1)\} + \\ + \sum_{i_1=0}^{\frac{N}{4}-1} \mathrm{cas}(-4k_2 i_1)\{[x(i_1) - x(i_1 + \tfrac{N}{2})]\sin(i_1) - \\ - [x(i_1 + \tfrac{N}{4}) - x(i_1 + \tfrac{3N}{4})]\cos(i_1)\}.$$

Letting now $i_1' = \frac{N}{4} - i_1$,

$$\sin(i_1) = \cos(i_1'), \text{ and } \cos(i_1) = \sin(i_1').$$

Thus,

$$H(4k_2+1) = \sum_{i_1=0}^{\frac{N}{4}-1} \mathrm{cas}(4k_2 i_1)\{[x(\tfrac{N}{4} - i_1) - \\ - x(-i_1 - \tfrac{N}{4}) + x(i_1) - x(i_1 + \tfrac{N}{2})]\cos(i_1) + [-x(-i_1) + \\ + x(-i_1 - \tfrac{N}{2}) + x(i_1 + \tfrac{3N}{4}) - x(i_1 + \tfrac{N}{4})]\sin(i_1)\}.$$

Assuming $k_1 = 3$, a similar analysis can be carried out for $H(4k_2+3)$, leading to

$$H(4k_2+3) = \sum_{i_1=0}^{\frac{N}{4}-1} \mathrm{cas}(4k_2 i_1)\{[x(i_1) - x(i_1 + \tfrac{N}{2}) + \\ + x(-i_1 + \tfrac{3N}{4}) - x(-i_1 + \tfrac{N}{4})]\cos(i_1) + [x(-i_1) - \\ - x(-i_1 + \tfrac{N}{2}) + x(i_1 + \tfrac{3N}{4}) - x(i_1 - \tfrac{N}{4})]\sin(i_1)\}.$$

The multiplicative and additive complexities satisfy $M(N) = M(\frac{N}{2}) + 2M(\frac{N}{4}) + N$ and $A(N) = M(\frac{N}{2}) + 2M(\frac{N}{4}) + \frac{7N}{2}$. Therefore the complexity of the FFHT is the same as in the FFFT case when the split-radix algorithm is used. The multiplicative and the additive complexities, for $N = 2^m$, are

$$M(N) = \tfrac{2N}{3}\log_2 N - \tfrac{19N + (-1)^m}{9} + 3,$$
$$A(N) = \tfrac{4N}{3}\log_2 N - \tfrac{14N + (-1)^m 5}{9} + 3.$$

7 A Winograd-like fast FFHT

Let $\{V\}$ be an $N - FFHT$, N prime, of a sequence $\{v\}$ with elements over $GF(p)$. The dc-term can be computed separately,

$$V_0 = \sum_{i=0}^{N-1} v_i,$$
$$V_k = v_0 + \sum_{i=1}^{N-1} \mathrm{cas}(ik) v_i \quad , k = 1, \ldots, N-1.$$

There is a primitive element λ, which generates a cyclic group containing $\{1, 2, \ldots, N-1\}$, since N is assumed to be prime. All the indexes can be rewritten in terms of powers of λ: $i = \lambda^{r(i)}$ and $k = \lambda^{r(k)}$:

$$V_{\lambda^{r(k)}} = v_0 + \sum_{i=1}^{N-1} \mathrm{cas}(\lambda^{r(i)}\lambda^{r(k)})\, v_i, k = 1, \ldots, N-1. \tag{8}$$

Letting now

$$l = r(k) \pmod{N-1}, l = 1, 2, \ldots, N-1,$$
$$j = -r(i) \pmod{N-1}, j = 1, 2, \ldots, N-1,$$
$$V_l' = V_{\lambda^{r(k)}},\ v_j' = v_{\lambda^{r(i)}} = v_{\lambda^{-j}},$$

equation (8) becomes

$$V'_l = v_0 + \sum_{j=1}^{N-1} \mathrm{cas}(\lambda^{(l-j)}) v'_j, l = 1, 2, \ldots, N-1.$$

Thus, an N-FFHT, N prime, is converted into a convolution of length $N-1$. Clearly,

$$V'_l - V_0 = \sum_{j=1}^{N-1} \left[\mathrm{cas}(\lambda^{(l-j)}) - 1\right] v'_j, l = 1, 2, \ldots, N-1.$$

This equation can be written as a polynomial product, $s(x) = g(x)d(x)$, where

$$\begin{aligned} g(x) &= [\mathrm{cas}(\lambda) - 1] + \left[\mathrm{cas}(\lambda^2) - 1\right] x + \ldots \\ &\quad + \left[\mathrm{cas}(\lambda^{(N-1)}) - 1\right] x^{N-1}, \\ d(x) &= v'_1 + v'_2 x + \ldots + v'_{N-1} x^{N-1}, \\ s(x) &= (V'_1 - V_0) + (V'_2 - V_0)\, x + \\ &\quad + \ldots + \left(V'_{N-1} - V_0\right) x^{N-1}. \end{aligned}$$

The Winograd algorithm can be used so as to compute the convolution required in the Rader algorithm. Example: $N = 5$, $p = 11$, $\zeta = 3$. In this case,

$$\begin{aligned} g(x) &= [\mathrm{cas}(3) - 1] + [\mathrm{cas}(4) - 1]\, x + \\ &\quad + [\mathrm{cas}(5) - 1]\, x^2 + [\mathrm{cas}(4) - 1]\, x^3. \end{aligned}$$

By Winograd, embedding the index changes into the pre-addition and post-addition matrices, results in:

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & -1 & -1 & 1 & 0 \\ 1 & 1 & -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_0 & & & & & \\ & B_1 & & & & \\ & & B_2 & & & \\ & & & B_3 & & \\ & & & & B_4 & \\ & & & & & B_5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix},$$

where

$$\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 2 & 0 & -2 & 0 \\ -2 & 2 & 2 & 0 \\ 2 & 2 & -2 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{cas}(3)-1 \\ \mathrm{cas}(4)-1 \\ \mathrm{cas}(5)-1 \\ \mathrm{cas}(4)-1 \end{bmatrix} = \begin{bmatrix} 0+10j \\ 1+8j \\ 1+7j \\ 4+5j \\ 3+3j \end{bmatrix}.$$

In the last example, the elements of the multiplicative matrix belongs to $GI(p)$, once that $\mathrm{cas}(i) \in GI(p)$, in contrast with the DHT case, where the elements are real numbers. A way to assure that all the elements of the matrix lie over $GF(p)$ is to choose $\zeta = a + bj$ where $a^2 + b^2 \equiv 1 \pmod{p}$. There is an advantage in computing the FFHT instead of the FFFT by the Winograd method, because it is not necessary to store complex elements (over $GI(p)$). The complexity to compute the FFFT and the FFHT are the same. When iterating an FFHT for same N, the multiplicative matrix can be previously computed. The constraint $a^2 + b^2 \equiv 1 \pmod{p}$ on ζ implies $N \mid p+1$, since the set of such elements ζ is a cyclic group of order $p+1$. For N prime, this algorithm has the minimal multiplicative complexity, however the additive complexity increases with the length N. For this reason it is suitable for short block length transforms, (e.g., up to $N = 16$).

8 Conclusions

The FFHT seems to have interesting applications in Communication Systems. Real-time successful applications of the FFHT depends on the existence of the so-called fast algorithms. This paper investigates the computation of the FFHT, presenting several fast algorithms.

References

[1] I.S. Reed, T.K. Truong, The use of Finite Fields to Compute Convolutions, IEEE Trans. Info. Theory, IT-21, pp.208-213, Mar. 1975.

[2] R.E. Blahut, Transform Techniques for Error-Control Codes, IBM J. Res. Dev., vol.23, pp.299-315, May, 1979.

[3] R.M. Campello de Souza, H.M. de Oliveira, A. N. Kauffman, A. J. A. Paschoal, Trigonometry in Finite Fields and a New Hartley Transform, Proc. of the IEEE Int. Symp. on Info. Theory, ISIT, Cambridge, MA, August, 1998.

[4] H.M. de Oliveira, R.M. Campello de Souza, and A.N. Kauffman, Efficient Multiplex for Band-limited Channels: Galois Division Multiple Access, Proc. of the Workshop on Coding and Cryptography, WCC 99, pp.235-241, Paris, Jan., 1999.

[5] H.M. de Oliveira, R.M. Campello de Souza, Orthogonal Multilevel Spreading Sequence Design, in: Coding Communications and Broadcasting, P.G. Farrell , M. Darnell, B. Honary, Eds; Research Studies Press, Hertfordshire, UK, 2000, pp. 291-301.

[6] R.M. Campello de Souza, H.M. de Oliveira, A. N. Kauffman, The Complex Finite Field Hartley Transform, in: Coding Communications and Broadcasting, P.G. Farrell , M. Darnell, B. Honary, Eds; Research Studies Press, Hertfordshire, UK, 2000, pp. 267-276.

[7] R. N. Bracewell, O. Buneman, H. Hao, and J. Villasenor, Fast Two-dimensional Hartley Transform, IEEE Proc., vol 74, No. 9, pp. 1282-1283, Sep., 1986.

[8] D.P.-K. Lun, W.-S. Siu, On Prime Factor Mapping for the Discrete Hartley Transform, IEEE Transactions on Signal Processing, vol. 40, No.6, pp. 1399-1451, Jun., 1992.

[9] R.E. Blahut, Fast Algorithms for digital signal processing, Addison-Wesley, 1985.