Improving the Goertzel–Blahut algorithm: An example

Sergei Valentinovich Fedorenko

In this preprint, we present an example illustrating the novel method for the discrete Fourier transform (DFT) computation based on the Goertzel–Blahut algorithm introduced in the paper "Improving the Goertzel–Blahut algorithm" (IEEE Signal Processing Letters, vol. 23, no. 6, pp. 824–827, 2016).

EXAMPLE

Consider the DFT of length $n = 15$ over the field $GF(2^4)$. The finite field $GF(2^4)$ is defined by an element α, which is a root of the primitive polynomial $x^4 + x + 1$. Let us take the primitive element α as a transform kernel. The binary conjugacy classes of $GF(2^m)$ are (α^0), $(\alpha^1, \alpha^2, \alpha^4, \alpha^8)$, $(\alpha^3, \alpha^6, \alpha^{12}, \alpha^9)$, $(\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11})$, (α^5, α^{10}).

A. *Goertzel–Blahut algorithm*

The first step of the Goertzel–Blahut algorithm is

$$\begin{array}{rcrlcr}
f(x) & = & (x+1)q_0(x) & +r_0(x), & r_0(x) = & r_{0,0} \\
f(x) & = & (x^4+x+1)q_1(x) & +r_1(x), & r_1(x) = & r_{3,1}x^3 + r_{2,1}x^2 + r_{1,1}x + r_{0,1} \\
f(x) & = & (x^4+x^3+x^2+x+1)q_2(x) & +r_2(x), & r_2(x) = & r_{3,2}x^3 + r_{2,2}x^2 + r_{1,2}x + r_{0,2} \\
f(x) & = & (x^4+x^3+1)q_3(x) & +r_3(x), & r_3(x) = & r_{3,3}x^3 + r_{2,3}x^2 + r_{1,3}x + r_{0,3} \\
f(x) & = & (x^2+x+1)q_4(x) & +r_4(x), & r_4(x) = & r_{1,4}x + r_{0,4}
\end{array},$$

where

$$\begin{array}{ll}
r_{0,0} & = \sum_{i=0}^{14} f_i \\
\hline
r_{0,1} & = f_0 + f_4 + f_7 + f_8 + f_{10} + f_{12} + f_{13} + f_{14} \\
r_{1,1} & = f_1 + f_4 + f_5 + f_7 + f_9 + f_{10} + f_{11} + f_{12} \\
r_{2,1} & = f_2 + f_5 + f_6 + f_8 + f_{10} + f_{11} + f_{12} + f_{13} \\
r_{3,1} & = f_3 + f_6 + f_7 + f_9 + f_{11} + f_{12} + f_{13} + f_{14} \\
\hline
r_{0,2} & = f_0 + f_4 + f_5 + f_9 + f_{10} + f_{14} \\
r_{1,2} & = f_1 + f_4 + f_6 + f_9 + f_{11} + f_{14} \\
r_{2,2} & = f_2 + f_4 + f_7 + f_9 + f_{12} + f_{14} \\
r_{3,2} & = f_3 + f_4 + f_8 + f_9 + f_{13} + f_{14} \\
\hline
r_{0,3} & = f_0 + f_4 + f_5 + f_6 + f_7 + f_9 + f_{11} + f_{12} \\
r_{1,3} & = f_1 + f_5 + f_6 + f_7 + f_8 + f_{10} + f_{12} + f_{13} \\
r_{2,3} & = f_2 + f_6 + f_7 + f_8 + f_9 + f_{11} + f_{13} + f_{14} \\
r_{3,3} & = f_3 + f_4 + f_5 + f_6 + f_8 + f_{10} + f_{11} + f_{14} \\
\hline
r_{0,4} & = f_0 + f_2 + f_3 + f_5 + f_6 + f_8 + f_9 + f_{11} + f_{12} + f_{14} \\
r_{1,4} & = f_1 + f_2 + f_4 + f_5 + f_7 + f_8 + f_{10} + f_{11} + f_{13} + f_{14}.
\end{array}$$

The second step of the Goertzel–Blahut algorithm is

$$\begin{array}{lllll}
F_0 & = f(\alpha^0) & = r_{0,0} & & \\ \hline
F_1 & = f(\alpha^1) & = r_1(\alpha^1) & = & r_{3,1}\alpha^3 + r_{2,1}\alpha^2 + r_{1,1}\alpha^1 + r_{0,1} \\
F_2 & = f(\alpha^2) & = r_1(\alpha^2) & = & r_{3,1}\alpha^6 + r_{2,1}\alpha^4 + r_{1,1}\alpha^2 + r_{0,1} \\
F_4 & = f(\alpha^4) & = r_1(\alpha^4) & = & r_{3,1}\alpha^{12} + r_{2,1}\alpha^8 + r_{1,1}\alpha^4 + r_{0,1} \\
F_8 & = f(\alpha^8) & = r_1(\alpha^8) & = & r_{3,1}\alpha^9 + r_{2,1}\alpha^1 + r_{1,1}\alpha^8 + r_{0,1} \\ \hline
F_3 & = f(\alpha^3) & = r_2(\alpha^3) & = & r_{3,2}\alpha^9 + r_{2,2}\alpha^6 + r_{1,2}\alpha^3 + r_{0,2} \\
F_6 & = f(\alpha^6) & = r_2(\alpha^6) & = & r_{3,2}\alpha^3 + r_{2,2}\alpha^{12} + r_{1,2}\alpha^6 + r_{0,2} \\
F_{12} & = f(\alpha^{12}) & = r_2(\alpha^{12}) & = & r_{3,2}\alpha^6 + r_{2,2}\alpha^9 + r_{1,2}\alpha^{12} + r_{0,2} \\
F_9 & = f(\alpha^9) & = r_2(\alpha^9) & = & r_{3,2}\alpha^{12} + r_{2,2}\alpha^3 + r_{1,2}\alpha^9 + r_{0,2} \\ \hline
F_7 & = f(\alpha^7) & = r_3(\alpha^7) & = & r_{3,3}\alpha^6 + r_{2,3}\alpha^{14} + r_{1,3}\alpha^7 + r_{0,3} \\
F_{14} & = f(\alpha^{14}) & = r_3(\alpha^{14}) & = & r_{3,3}\alpha^{12} + r_{2,3}\alpha^{13} + r_{1,3}\alpha^{14} + r_{0,3} \\
F_{13} & = f(\alpha^{13}) & = r_3(\alpha^{13}) & = & r_{3,3}\alpha^9 + r_{2,3}\alpha^{11} + r_{1,3}\alpha^{13} + r_{0,3} \\
F_{11} & = f(\alpha^{11}) & = r_3(\alpha^{11}) & = & r_{3,3}\alpha^3 + r_{2,3}\alpha^7 + r_{1,3}\alpha^{11} + r_{0,3} \\ \hline
F_5 & = f(\alpha^5) & = r_4(\alpha^5) & = & r_{1,4}\alpha^5 + r_{0,4} \\
F_{10} & = f(\alpha^{10}) & = r_4(\alpha^{10}) & = & r_{1,4}\alpha^{10} + r_{0,4}
\end{array}$$

or

$$F_0 = (1)\,(r_{0,0}) = V_0\,(r_{0,0}),$$

$$\begin{pmatrix} F_1 \\ F_2 \\ F_4 \\ F_8 \end{pmatrix} = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} \\ 1 & \alpha^8 & \alpha^1 & \alpha^9 \end{pmatrix} \begin{pmatrix} r_{0,1} \\ r_{1,1} \\ r_{2,1} \\ r_{3,1} \end{pmatrix} = V_1 \begin{pmatrix} r_{0,1} \\ r_{1,1} \\ r_{2,1} \\ r_{3,1} \end{pmatrix},$$

$$\begin{pmatrix} F_3 \\ F_6 \\ F_{12} \\ F_9 \end{pmatrix} = \begin{pmatrix} 1 & \alpha^3 & \alpha^6 & \alpha^9 \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 \\ 1 & \alpha^{12} & \alpha^9 & \alpha^6 \\ 1 & \alpha^9 & \alpha^3 & \alpha^{12} \end{pmatrix} \begin{pmatrix} r_{0,2} \\ r_{1,2} \\ r_{2,2} \\ r_{3,2} \end{pmatrix} = V_2 \begin{pmatrix} r_{0,2} \\ r_{1,2} \\ r_{2,2} \\ r_{3,2} \end{pmatrix},$$

$$\begin{pmatrix} F_7 \\ F_{14} \\ F_{13} \\ F_{11} \end{pmatrix} = \begin{pmatrix} 1 & \alpha^7 & \alpha^{14} & \alpha^6 \\ 1 & \alpha^{14} & \alpha^{13} & \alpha^{12} \\ 1 & \alpha^{13} & \alpha^{11} & \alpha^9 \\ 1 & \alpha^{11} & \alpha^7 & \alpha^3 \end{pmatrix} \begin{pmatrix} r_{0,3} \\ r_{1,3} \\ r_{2,3} \\ r_{3,3} \end{pmatrix} = V_3 \begin{pmatrix} r_{0,3} \\ r_{1,3} \\ r_{2,3} \\ r_{3,3} \end{pmatrix},$$

$$\begin{pmatrix} F_5 \\ F_{10} \end{pmatrix} = \begin{pmatrix} 1 & \alpha^5 \\ 1 & \alpha^{10} \end{pmatrix} \begin{pmatrix} r_{0,4} \\ r_{1,4} \end{pmatrix} = V_4 \begin{pmatrix} r_{0,4} \\ r_{1,4} \end{pmatrix}.$$

The Goertzel–Blahut algorithm in matrix form is

$$\pi \boldsymbol{F} = VR\boldsymbol{f}$$

or

$$\left(\begin{array}{c} F_0 \\ \hline F_1 \\ F_2 \\ F_4 \\ F_8 \\ \hline F_3 \\ F_6 \\ F_{12} \\ F_9 \\ \hline F_7 \\ F_{14} \\ F_{13} \\ F_{11} \\ \hline F_5 \\ F_{10} \end{array}\right) = \left(\begin{array}{c|cccc|cccc|cccc|cc} \alpha^0 & & & & & & & & & & & & & & \\ \hline & \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & & & & & & & & & & \\ & \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & & & & & & & & & & \\ & \alpha^0 & \alpha^4 & \alpha^8 & \alpha^{12} & & & & & & & & & & \\ & \alpha^0 & \alpha^8 & \alpha^1 & \alpha^9 & & & & & & & & & & \\ \hline & & & & & \alpha^0 & \alpha^3 & \alpha^6 & \alpha^9 & & & & & & \\ & & & & & \alpha^0 & \alpha^6 & \alpha^{12} & \alpha^3 & & & & & & \\ & & & & & \alpha^0 & \alpha^{12} & \alpha^9 & \alpha^6 & & & & & & \\ & & & & & \alpha^0 & \alpha^9 & \alpha^3 & \alpha^{12} & & & & & & \\ \hline & & & & & & & & & \alpha^0 & \alpha^7 & \alpha^{14} & \alpha^6 & & \\ & & & & & & & & & \alpha^0 & \alpha^{14} & \alpha^{13} & \alpha^{12} & & \\ & & & & & & & & & \alpha^0 & \alpha^{13} & \alpha^{11} & \alpha^9 & & \\ & & & & & & & & & \alpha^0 & \alpha^{11} & \alpha^7 & \alpha^3 & & \\ \hline & & & & & & & & & & & & & \alpha^0 & \alpha^5 \\ & & & & & & & & & & & & & \alpha^0 & \alpha^{10} \end{array}\right)$$

$$\times \left(\begin{array}{ccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right) \left(\begin{array}{c} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{array}\right).$$

B. Novel method based on the Goertzel–Blahut algorithm

The Moore–Vandermonde matrix V_1 factorization [13] is

$$V_1 = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} \\ 1 & \alpha^8 & \alpha^1 & \alpha^9 \end{pmatrix} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & \alpha & 0 \\ 0 & 1 & 0 & \alpha^2 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

$$\times \left(\begin{array}{cc|cc} 1 & \alpha^5 & 0 & 0 \\ 1 & \alpha^{10} & 0 & 0 \\ \hline 0 & 0 & 1 & \alpha^5 \\ 0 & 0 & 1 & \alpha^{10} \end{array}\right) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \alpha^0 & \alpha^5 & \alpha^1 & \alpha^6 \\ \alpha^0 & \alpha^{10} & \alpha^2 & \alpha^{12} \\ \alpha^0 & \alpha^5 & \alpha^4 & \alpha^9 \\ \alpha^0 & \alpha^{10} & \alpha^8 & \alpha^3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = S_1 P_1,$$

where P_1 is the matrix of preadditions.

From Lemma 1 [SPL16] it follows that

$$\begin{array}{lll} V_0 = & & 1 \\ V_1 = & & S_1 P_1 \\ V_2 = & V_1 N_{2,1} = & S_1 P_1 N_{2,1} \\ V_3 = & V_1 N_{3,1} = & S_1 P_1 N_{3,1} \end{array},$$

where $N_{2,1}$ and $N_{3,1}$ are the basis transformation matrices:

$$N_{2,1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \qquad N_{3,1} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

The Moore–Vandermonde matrix V_4 factorization is very simple:

$$V_4 = \begin{pmatrix} 1 & \alpha^5 \\ 1 & \alpha^{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & \alpha^5 \\ 0 & 1 \end{pmatrix}.$$

Further, we obtain

$$V = \begin{pmatrix} V_0 & & & & \\ & V_1 & & & \\ & & V_2 & & \\ & & & V_3 & \\ & & & & V_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & & & & \\ & S_1 & & & \\ & & S_1 & & \\ & & & S_1 & \\ & & & & V_4 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & P_1 & & & \\ & & P_1 & & \\ & & & P_1 & \\ & & & & I_2 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & I_4 & & & \\ & & N_{2,1} & & \\ & & & N_{3,1} & \\ & & & & I_2 \end{pmatrix} = SPN,$$

where I_2 is the 2×2 identity matrix and I_4 is the 4×4 identity matrix.

In matrix form, the DFT algorithm can be written as

$$\pi \boldsymbol{F} = VR\boldsymbol{f} = S(PNR)\boldsymbol{f}$$

or

$$\left(\begin{array}{c} F_0 \\ \hline F_1 \\ F_2 \\ F_4 \\ F_8 \\ \hline F_3 \\ F_6 \\ F_{12} \\ F_9 \\ \hline F_7 \\ F_{14} \\ F_{13} \\ F_{11} \\ \hline F_5 \\ F_{10} \end{array}\right) = \left(\begin{array}{c|cccc|cccc|cccc|cc} \alpha^0 & & & & & & & & & & & & & & \\ \hline & \alpha^0 & \alpha^5 & \alpha^1 & \alpha^6 & & & & & & & & & & \\ & \alpha^0 & \alpha^{10} & \alpha^2 & \alpha^{12} & & & & & & & & & & \\ & \alpha^0 & \alpha^5 & \alpha^4 & \alpha^9 & & & & & & & & & & \\ & \alpha^0 & \alpha^{10} & \alpha^8 & \alpha^3 & & & & & & & & & & \\ \hline & & & & & \alpha^0 & \alpha^5 & \alpha^1 & \alpha^6 & & & & & & \\ & & & & & \alpha^0 & \alpha^{10} & \alpha^2 & \alpha^{12} & & & & & & \\ & & & & & \alpha^0 & \alpha^5 & \alpha^4 & \alpha^9 & & & & & & \\ & & & & & \alpha^0 & \alpha^{10} & \alpha^8 & \alpha^3 & & & & & & \\ \hline & & & & & & & & & \alpha^0 & \alpha^5 & \alpha^1 & \alpha^6 & & \\ & & & & & & & & & \alpha^0 & \alpha^{10} & \alpha^2 & \alpha^{12} & & \\ & & & & & & & & & \alpha^0 & \alpha^5 & \alpha^4 & \alpha^9 & & \\ & & & & & & & & & \alpha^0 & \alpha^{10} & \alpha^8 & \alpha^3 & & \\ \hline & & & & & & & & & & & & & \alpha^0 & \alpha^5 \\ & & & & & & & & & & & & & \alpha^0 & \alpha^{10} \end{array}\right)$$

$$\times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{pmatrix}.$$

C. Complexity of the 15-point DFT computation

The novel method of the 15-point DFT computation based on the Goertzel–Blahut algorithm consists of two steps:

1) multiplication of the binary matrix PNR by the vector $\boldsymbol{f}$ (using the heuristic algorithm [19]): 44 additions;

2a) triple multiplication by the matrix S_1: $3 \times 4 = 12$ multiplications and $3 \times 8 = 24$ additions;

2b) multiplication by the matrix V_4: 1 multiplication and 2 additions.

The complexity of this method is 13 multiplications and 70 additions.

The complexity of several methods of the 15-point DFT computation is shown in Table II.

TABLE II
COMPLEXITY OF THE 15-POINT DFT COMPUTATION

method	multiplications	additions
Cyclotomic algorithm [18]	16	77
Cyclotomic algorithm with common subexpression elimination algorithm [7]	16	74
Novel method [SPL16]	13	70

REFERENCES

[SPL16] S. V. Fedorenko, "Improving the Goertzel–Blahut algorithm," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 824–827, 2016.

[7] N. Chen and Z. Yan, "Cyclotomic FFTs with reduced additive complexities based on a novel common subexpression elimination algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1010–1020, 2009.

[13] S. V. Fedorenko, "Normalized cyclic convolution: The case of even length," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5307–5317, 2015.

[18] P. V. Trifonov and S. V. Fedorenko, "A method for fast computation of the Fourier transform over a finite field," *Problems of Information Transmission*, vol. 39, no. 3, pp. 231–238, 2003. Translation of *Problemy Peredachi Informatsii*.

[19] P. Trifonov, "Matrix-vector multiplication via erasure decoding," Proceedings of the XI international symposium on problems of redundancy in information and control systems at St.Petersburg, Russia, pp. 104–108, July 2007.
[Online]. Available: http://dcn.ftk.spbstu.ru/∼petert/papers/mo.pdf