Fast invertible nonlinear transforms for intelligent OFDM TCS

V. Labunets[1], N. Ostheimer[2]

[1]Ural State Forest Engineering University, Sibirskiy trakt 37, Ekaterinburg, Russia, 620100
[2]Capricat LLC, Pompano Beach, Florida, US

Abstract. A unified mathematical form of invertible nonlinear transforms has been introduced in the form of fast transform (like to fast Fourier transform). The main goal of this article is to show that fast Fourier transforms (FFT) can be both non-linearized and generalized. The non-linearization and the generalization of FFTs are based on a set of recursive rules, which generate nonlinear transforms with a fast algorithm. For each rule, simple relations give the number of elementary nonlinear operations required by the fast algorithm. The resulting scheme is formed by three stages. The first stage contains so-called basis 2x2 nonlinear transforms (**BNLT**). The second step is based on sparse nonlinear transforms (**SNLT**), which are direct sums (combinations) of **BNLT**s. The third stage produces fast nonlinear transform (**FNLT**) in the form of finite number multiplicative superposition of **SNLT**s. The framework developed allows the introduction of generalized transforms, which include all common fast transforms. The reported architecture generalizes both linear and non-linear fast transforms, which can be considered as a formal framework for generalized signal processing. This approach leads to a number of new linear and nonlinear transforms of potential interest, for example, for OFDM and CDMA telecommunication systems.We propose a novel modulation technique based on nonlinear transforms. The proposed modulation scheme could be used directly instead of a conventional orthogonal frequency division multiplexing (OFDM) transmitter, resulting in a system possessing all benefits of OFDM along with reduced peak-to-average power ratio (PAPR).

1. Introduction

Non-linear signal processing has emerged in the last years as a specific target for signal processing tools, being a direct consequence of the degree of saturation produced in the linear signal processing field. At the beginning, non-linear filtering and modelling were under the scope of the filtering theory based on generalized nonlinear convolution integral, such as the Volterra model [1,2] or the G-functional Wiener model [3-5]. Concerning non-linear filtering, nonlinear signal processing based on aggregation operators was developed and proven useful to opening up new signal and image processing applications [6-14]. More recent non-liner transforms are those approached in a neural network framework, particularly the so-called radial basis functions successfully applied in modern communication receivers [15-17].

In this paper, we present a unified view of discrete non-linear transforms (DNLT) with a fast algorithm. A discrete non-linear N -dimension transform is characterized by N non-linear processing scalar-valued functions $\{f_k(\cdot_0,\cdot_1,\ldots,\cdot_{N-1})\}_{k=0}^{N-1}$. They form a *non-linear transformation* (NLT) $\mathbf{NF}:|\mathbf{x}\rangle \to |\mathbf{y}\rangle$.

Here, a length-N signal column-vector $|\mathbf{x}\rangle = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}$ is transformed by non-linear transform $\mathbf{F}$ as

$$|\mathbf{y}\rangle = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \mathbf{NLT}(|\mathbf{x}\rangle) = \begin{bmatrix} f_0(x_0,x_1,\ldots,x_{N-1}) \\ f_1(x_0,x_1,\ldots,x_{N-1}) \\ \vdots \\ f_{N-1}(x_0,x_1,\ldots,x_{N-1}) \end{bmatrix} = \begin{bmatrix} f_0(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ f_1(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ \vdots \\ f_{N-1}(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \end{bmatrix} \circ \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix},$$

to produce a new column-vector $|\mathbf{y}\rangle$. Concisely, we denote the NLT by the matrix-like operator

$$\mathbf{NLT} \circ |\mathbf{x}\rangle = \begin{bmatrix} f_0(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ f_1(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ \vdots \\ f_{N-1}(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \end{bmatrix} \circ \begin{bmatrix} {}_{0\leftarrow} x_0 \\ \hline {}_{1\leftarrow} x_1 \\ \vdots \\ \hline {}_{N-1\leftarrow} x_{N-1} \end{bmatrix} = \begin{bmatrix} f_0(x_0,x_1,\ldots,x_{N-1}) \\ f_1(x_0,x_1,\ldots,x_{N-1}) \\ \vdots \\ f_{N-1}(x_0,x_1,\ldots,x_{N-1}) \end{bmatrix},$$

where $\cdot_k$ is the place for variable ${}_{k\leftarrow} x_k$ $(k = 0,1,\ldots,N-1)$. For signal samples ${}_{l\leftarrow} x_k$ we use two indexes l and k. The first index $l \leftarrow$ is address for operand x_k and the second index k is the time number in time series $x_0, x_1,\ldots, x_{N-1}$. Samples can be rearranged in side column vector $|\mathbf{x}\rangle$. But functions $\{f_k(\cdot_0,\cdot_1,\ldots,\cdot_{N-1})\}_{k=0}^{N-1}$ take samples from the appropriate places of $|\mathbf{x}\rangle$ ignoring the time number. For example,

$$\mathbf{NLT} \circ |\mathbf{x}\rangle = \begin{bmatrix} f_0(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ f_1(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \\ \vdots \\ f_{N-1}(\cdot_0,\cdot_1,\ldots,\cdot_{N-1}) \end{bmatrix} \circ \begin{bmatrix} {}_{0\leftarrow} x_{N-1} \\ \hline {}_{1\leftarrow} x_0 \\ \vdots \\ \hline {}_{N-1\leftarrow} x_{N-2} \end{bmatrix} = \begin{bmatrix} f_0(x_{N-1},x_0,\ldots,x_{N-2}) \\ f_1(x_{N-1},x_0,\ldots,x_{N-2}) \\ \vdots \\ f_{N-1}(x_{N-1},x_0,\ldots,x_{N-2}) \end{bmatrix}.$$

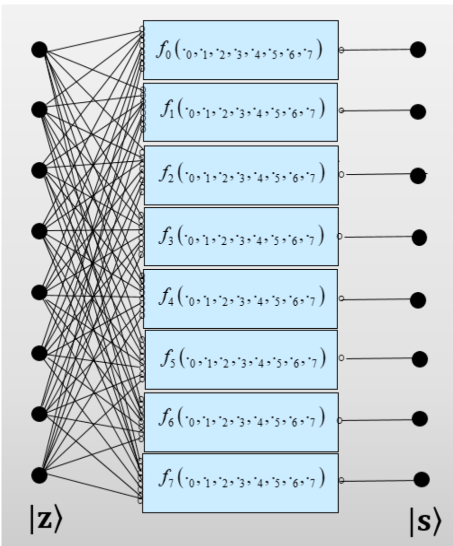

Figure 1. Non-linear signal transformation (NST).

We see that signal samples $x_0,x_1,\ldots,x_{N-1}$ are combined in N non-linear processing scalar-valued functions $\{f_k(\cdot_0,\cdot_1,\ldots,\cdot_{N-1})\}_{k=0}^{N-1}$ (see figure 1), to form samples $y_0,y_1,\ldots,y_{N-1}$ of a new signal $|\mathbf{y}\rangle$. The components $y_0, y_1, \ldots, y_{N-1}$ of the signal $|\mathbf{y}\rangle$ represent non-linear interactions between different time samples $x_0, x_1, \ldots, x_{N-1}$ of the initial signal $|\mathbf{x}\rangle$.

In 1957 Andrei Kolmogorov [18] solved the 13th problem of the collection that Hilbert provided as the mathematical problems for the 20th century. He proved the existence of a solution, based on functions of a single variable, of the problem of finding a continuous function mapping N inputs and outputs. If $|\mathbf{x}\rangle$ and $|\mathbf{y}\rangle$ are the input and output vectors of N components, the existence theorem specifies that the processor which maps $|\mathbf{x}\rangle$ in $|\mathbf{y}\rangle$ is formed by two stages. The first stage has a mathematical representation given by

$$\bar{x}_q = \sum_{n=0}^{N-1} \Psi_{qn}(x_n) = \Psi_{q,0}(x_0) + \Psi_{q,1}(x_1) + \ldots + \Psi_{q,N-1}(x_{N-1}), \quad q = 0,1,\ldots,2N \tag{1}$$

for $(2N+1)N$ functions $\{\Psi_{qn}(\cdot_n)\}_{q=0,n=0}^{2N,N-1}$. They are continuous and also standard, i.e., they are independent of the choice of the function f. This means that in n-dimensional (n-D) space we introduce $2N+1$ special (and very exotic) coordinates $\{\bar{x}_q\}_{q=0}^{2N}$. They are functions of the form (1).

The second stage contains $(2N+1)N$ functions $\{h_{kq}(\cdot_q)\}_{k=0,q=0}^{N-1,2N}$ that produce the N-component of the output

$$y_k = f_k(x_0, x_1, \ldots, x_{N-1}) = \sum_{q=0}^{2N} h_{kq}(\bar{x}_q) = \sum_{q=0}^{2N} h_{kq}\left(\sum_{n=0}^{N-1} \Psi_{qn}(x_n)\right), \quad k = 0,1,..,N-1, \tag{2}$$

or

$$\begin{cases} y_0 = f_0(x_0, x_1, \ldots, x_{N-1}) = \sum_{q=0}^{2N} h_{0,q}\left[\sum_{n=0}^{N-1} \Psi_{qn}(x_n)\right], \\ y_1 = f_1(x_0, x_1, \ldots, x_{N-1}) = \sum_{q=0}^{2N} h_{1,q}\left[\sum_{n=0}^{N-1} \Psi_{qn}(x_n)\right], \\ \ldots \\ y_{N-1} = f_{N-1}(x_0, x_1, \ldots, x_{N-1}) = \sum_{q=0}^{2N} h_{N-1,q}\left[\sum_{n=0}^{N-1} \Psi_{qn}(x_n)\right], \end{cases} \tag{3}$$

where the functions $\{{}^q\chi(\cdot)\}_{q=1}^{L}$ are continuous. We see, that every continuous coordinate function $f_k(x_0, x_1, \ldots, x_{N-1})$ is represented as the sum of continuous functions of new individual coordinates. Thus, in accordance with the Kolmogorov theorem, all continuous functions of many variables can be obtained from continuous functions of one variable using linear operations and superposition.

We can presented (2) and (3) in matrix-like form:

$$|\mathbf{y}\rangle = \mathbf{NLT} \circ |\mathbf{x}\rangle = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} f_0(x_0, x_1, \ldots, x_{N-1}) \\ f_1(x_0, x_1, \ldots, x_{N-1}) \\ \vdots \\ f_{N-1}(x_0, x_1, \ldots, x_{N-1}) \end{bmatrix} =$$

$$= \begin{bmatrix} h_{0,0}(\cdot_0) & h_{0,1}(\cdot_1) & \ldots & h_{0,q}(\cdot_q) & \ldots & h_{0,2N}(\cdot_{2N}) \\ h_{1,0}(\cdot_0) & h_{1,1}(\cdot_1) & \ldots & h_{1,q}(\cdot_q) & \ldots & h_{1,2N}(\cdot_{2N}) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ h_{N-1,0}(\cdot_0) & h_{N-1,1}(\cdot_1) & \ldots & h_{N-1,q}(\cdot_q) & \ldots & h_{N-1,2N}(\cdot_{2N}) \end{bmatrix} \circ \begin{bmatrix} \Psi_{0,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) & \ldots & \Psi_{0,N-1}(\cdot_{N-1}) \\ \Psi_{1,0}(\cdot_0) & \Psi_{1,1}(\cdot_1) & \ldots & \Psi_{1,N-1}(\cdot_{N-1}) \\ \vdots & \vdots & \vdots & \vdots \\ \Psi_{q,0}(\cdot_0) & \Psi_{q,1}(\cdot_1) & \ldots & \Psi_{q,N-1}(\cdot_{N-1}) \\ \vdots & \vdots & \vdots & \vdots \\ \Psi_{2N,0}(\cdot_0) & \Psi_{2N,1}(\cdot_1) & \ldots & \Psi_{2N,N-1}(\cdot_{N-1}) \end{bmatrix} \circ \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} =$$

$$= \mathbf{H} \circ \mathbf{\Psi} \circ |\mathbf{x}\rangle,$$

where $\mathbf{H} := \left[h_{k,q}(\cdot_q) \right]_{k=0,q=0}^{N-1,2N}$, $\mathbf{\Psi} := \left[\Psi_{q,n}(\cdot_q) \right]_{q=0,k=0}^{2N,N-1}$ and $\circ$ is the symbol of superposition. In particular, a non-linear (2×2)-transform $\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} f_0(x_0,x_1) \\ f_1(x_0,x_1) \end{bmatrix}$ can be represented as (the Arnold theorem)

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} f_0(x_0,x_1) \\ f_1(x_0,x_1) \end{bmatrix} = \begin{bmatrix} \chi_{0,0}(\cdot_0) & \chi_{0,1}(\cdot_1) & \chi_{0,2}(\cdot_2) & \chi_{0,3}(\cdot_3) & \chi_{0,4}(\cdot_4) \\ \chi_{1,0}(\cdot_0) & \chi_{1,1}(\cdot_1) & \chi_{1,2}(\cdot_2) & \chi_{1,3}(\cdot_3) & \chi_{1,4}(\cdot_4) \end{bmatrix} \circ \begin{bmatrix} \Psi_{0,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) \\ \Psi_{1,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) \\ \Psi_{2,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) \\ \Psi_{3,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) \\ \Psi_{4,0}(\cdot_0) & \Psi_{0,1}(\cdot_1) \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}.$$

This theorem is not only elegant, but also simple. Its proof is elementarily presented in a brilliant article by V.I. Arnold [19].

Remark 1. In [20] Gashkov points out that circuit synthesis of continuous functions might be important for the analysis of artificial neural systems modelling the human brain. In 1987, R. Hecht-Nielsen noticed that Kolmogorov's theorem has an interpretation in terms of neural networks [21].

The search of a fast non-linear transforms is of paramount importance in the design of a suitable non-linear methods in terms of the digital signal processing. In [22] the authors related the mapping theorem of Kolmogorov and FFT for design FNLT. Precisely, it is this theorem, which motivates this work. Starting from the usual architecture of FFT, a new architecture for non-linear transforms formed by three processing stages is introduced. A closed formulation for the design of such an architecture is also presented. The architecture is described in detail. The objective is to find a procedure for fast nonlinear processing from only short input data.

The paper are organized as follows. Section 2 begins an overview of unified approaches to fast Discrete Orthogonal Transforms (DOTs) such as the well-known Fast Fourier Transform (FFT), Fast Walsh Transform (FWT), Fast Haar Transform (FHT) and similar algorithms for other transforms. Section 3 introduces a fast Fourier-like representations for non-linear transforms. The resulting algorithms are formed by three stages. The first stage contains so-called basis 2x2 non-linear transforms (**BNLT**). The second step is based on sparse non-linear transforms (**SNLT**), which are direct sums (combinations) of **BNLT**s. The third stage produces fast non-linear transforms (**FNLTs**) in the form of finite number multiplicative superposition of **SNLT**s. These stages form a new unified approach based on introduction of new families of non-linear invertible fast transforms having a unified structure. This approach allows not only to generalize many well-known fast DOTs but also to synthesize an infinite number of new linear and non-linear transforms that can be adapted to given application. In Section 4 the non-linear Kronecker "product" is briefly described in order to support the new architecture of fast non-linear transforms. .New **FNLTs** generalize both linear and non-linear fast transforms, which can be considered as a formal framework for generalized signal processing. Later on, in Section 5, **t**he explicitly invertible basis (2×2)-transforms $\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} f_0(x_0,x_1) \\ f_1(x_0,x_1) \end{bmatrix}$ are presented as elementary building blocks for designing of **FNLTs.** Finally, in Section 6, we propose a novel modulation technique based on fast non-linear transforms. The proposed modulation scheme could be used directly instead of a conventional orthogonal frequency division multiplexing (OFDM) transmitter, resulting in a system possessing all benefits of OFDM along with reduced peak-to-average power ratio (PAPR).

2. A Little bit of history, and a formulation of a problem

For some specific transforms of interest such as the Fourier, Walsh-Hadamard transforms, a fast algorithm has been found, which requires fewer elementary operations. In 1965 Cooley and Tukey [23] developed an algorithm for accelerating the calculation of the DFT which in some ways revolutionized the numerical computation of linear (primarily orthogonal land unitary) transforms. This algorithm, known as the Fast Fourier Transform (FFT) radix-2. Parallel to the development of FFT, there has been a development of similar algorithms for other transforms: Discrete Walsh transform (DWT) [24,25], Discrete Hartley Transform (DHtT) [26,27], Discrete Haar Transform (DHT) [28,29]. Some workers have considered the definition of fast algorithms for generalized transforms and we mention the works by Andrews, et al. [30-32], [Rao, et al. [33,34] and Labunets [35-44] (for so-called many-parameter transforms, wavelet and packets). A unified set of algorithms have been developed by Demuth [45], Labunets [46] and [47].

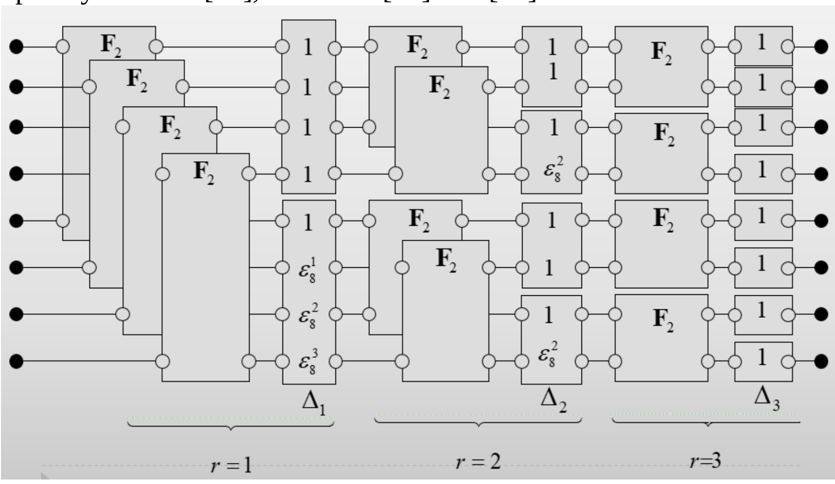

Figure 2. Fast Fourier transform ($N = 8$).

Fast Fourier transform is the following iteration procedure (see figure 2):

$$\mathbf{FT} = \frac{1}{\sqrt{2^n}} \prod_{r=1}^{\rightarrow n} \left(\left[I_{2^{n-r}} \otimes \left(I_{2^{r-1}} \oplus \Delta_{2^{r-1}}\left(\varepsilon^{\cdot 2^{n-r}}\right)\right)\right] \cdot \left[I_{2^{n-r}} \otimes \mathbf{F}_2 \otimes I_{2^{r-1}} \right] \right) =$$

$$= \frac{1}{\sqrt{2^n}} \prod_{r=1}^{\rightarrow n} \left(\left[I_{2^{n-r}} \otimes \left(I_{2^{r-1}} \oplus \Delta_{2^{r-1}}\right)\left(\mathbf{F}_2 \otimes I_{2^{r-1}}\right)\right]\right) = \frac{1}{\sqrt{2^n}} \prod_{r=1}^{\rightarrow n} \left(\left[I_{2^{n-r}} \otimes \left[\begin{array}{c:c} I_{2^{r-1}} & \\ \hdashline & \Delta_{2^{r-1}} \end{array}\right] \left[\begin{array}{c:c} I_{2^{r-1}} & I_{2^{r-1}} \\ \hdashline I_{2^{r-1}} & -I_{2^{r-1}} \end{array}\right] \right] \right) = \quad (4)$$

$$= \frac{1}{\sqrt{2^n}} \prod_{r=1}^{\rightarrow n} \left(\left[I_{2^{n-r}} \otimes \left[\begin{array}{c:c} I_{2^{r-1}} & I_{2^{r-1}} \\ \hdashline \Delta_{2^{r-1}} I_{2^{r-1}} & -\Delta_{2^{r-1}} I_{2^{r-1}} \end{array}\right] \right] \right) = \frac{1}{\sqrt{2^n}} {}^1\mathbf{S}\mathrm{T} \cdot {}^2\mathbf{S}\mathrm{T} \cdot \ldots \cdot {}^n\mathbf{S}\mathrm{T},$$

where ${}^r\mathbf{S}\mathrm{T} = \left[I_{2^{n-r}} \otimes \left[\begin{array}{c:c} I_{2^{r-1}} & I_{2^{r-1}} \\ \hdashline \Delta_{2^{r-1}} I_{2^{r-1}} & -\Delta_{2^{r-1}} I_{2^{r-1}} \end{array}\right] \right]$ and $\Delta_{2^{r-1}}\left(\varepsilon^{2^{n-r}}\right) = \mathbf{Diag}_{2^{r-1}}\left(1, \varepsilon^{2^{n-r}\cdot 1}, \varepsilon^{2^{n-r}\cdot 2}, \ldots, \varepsilon^{2^{n-r}\cdot(2^{r-1}-1)}\right)$.

For each iteration $r = 1, 2, \ldots, n$ we introduce digital representation for a number $p_r \in \{0, 1, \ldots, 2^{n-1}-1\}$:

$p_r = p_r(s_r, b_r) = 2^{r-1}\left\lfloor \frac{p}{2^{r-1}} \right\rfloor + p(\mathrm{mod}\, 2^{r-1}) = 2^{r-1}s_r + b_r$, where $s_r = \left\lfloor \frac{p}{2^{r-1}} \right\rfloor$, $b_r = p(\mathrm{mod}\, 2^{r-1})$. Then

$$\left[\begin{array}{c:c} I_{2^{r-1}} & I_{2^{r-1}} \\ \hdashline \Delta_{2^{r-1}} I_{2^{r-1}} & -\Delta_{2^{r-1}} I_{2^{r-1}} \end{array}\right] = \bigoplus_{b_r=0}^{2^{r-1}-1} \left[\begin{array}{ccccc|ccccc} \ddots & & & & & \ddots & & & & \\ & \ddots & & & & & \ddots & & & \\ & & 1 & & & & & 1 & & \\ & & & \ddots & & & & & \ddots & \\ & & & & \ddots & & & & & \ddots \\ \hline \ddots & & & & & \ddots & & & & \\ & \ddots & & & & & \ddots & & & \\ & & \varepsilon^{2^{n-r}\cdot b_r} & & & & & -\varepsilon^{2^{n-r}\cdot b_r} & & \\ & & & \ddots & & & & & \ddots & \\ & & & & \ddots & & & & & \ddots \end{array}\right] = \left[\bigoplus_{b_r=0}^{2^{r-1}-1} \mathbf{Butt}_{b_r+2^{r-1}}^{b_r} \right],$$

and

$$\left[I_{2^{n-r}} \otimes \begin{bmatrix} I_{2^{r-1}} & I_{2^{r-1}} \\ \Delta_{2^{r-1}} I_{2^{r-1}} & -\Delta_{2^{r-1}} I_{2^{r-1}} \end{bmatrix} \right] = \left[I_{2^{n-r}} \otimes \left[\bigoplus_{b_r=0}^{2^{r-1}-1} \mathbf{Butt}^{b_r}_{b_r+2^{r-1}} \right] \right] = \left[\bigoplus_{s_r=0}^{2^{n-r}-1} \bigoplus_{b_r=0}^{2^{r-1}-1} {}^{s_r 2^{r-1}+b_r}\mathbf{Butt}^{s_r 2^r+b_r}_{s_r 2^r+b_r+2^{r-1}} \right] =$$

$$= \left[\bigoplus_{p_r=0}^{2^{n-1}-1} {}^{p_r(s_r,b_r)}\mathbf{Butt}^{i_r(s_r,b_r)}_{i_r(s_r,b_r)+2^{r-1}} \right] := {}^{r}\mathbf{ST},$$

where $p_r = s_r 2^{r-1} + b_r$ $i_r = s_r 2^r + b_r$ and the butterfly block ${}^{p_r(s_r,b_r)}\mathbf{Butt}^{i_r(s_r,b_r)}_{i_r(s_r,b_r)+2^{r-1}}$ operates on 2D space $\mathbf{V}_2^{p_r}\left\{e_{i_r}, e_{i_r+2^{r-1}}\right\}$ spanned on i_r-th and $(i_r + 2^{r-1})$ orts $e_{i_r}, e_{i_r+2^{r-1}}$. Here ${}^{r}\mathbf{ST} := \left[\bigoplus_{p_r=0}^{2^{n-1}-1} {}^{p_r(s_r,b_r)}\mathbf{Butt}^{i_r(s_r,b_r)}_{i_r(s_r,b_r)+2^{r-1}} \right]$ is the radix-2 *sparse transform* (**ST**). Hence,

$$\mathbf{FT} = \frac{1}{\sqrt{2^n}} \prod_{r=1}^{n} {}^{r}\mathbf{ST} = \frac{1}{\sqrt{2^n}} \prod_{r=1}^{n} \left(\left[I_{2^{n-r}} \otimes \left(I_{2^{r-1}} \oplus \Delta_{2^{r-1}}\left(\varepsilon^{\cdot 2^{n-r}}\right) \right) \right] \cdot \left[I_{2^{n-r}} \otimes \mathbf{F}_2 \otimes I_{2^{r-1}} \right] \right) =$$
$$= \frac{1}{\sqrt{2^n}} \prod_{r=1}^{n} \left[\bigoplus_{s_r=0}^{2^{n-r}-1} \bigoplus_{b_r=0}^{2^{r-1}-1} {}^{s_r 2^{r-1}+b_r}\mathbf{Butt}^{s_r 2^r+b_r}_{s_r 2^r+b_r+2^{r-1}} \right] = \frac{1}{\sqrt{2^n}} \prod_{r=1}^{n} \left[\bigoplus_{p_r=0}^{2^{n-1}-1} {}^{p_r(s_r,b_r)}\mathbf{Butt}^{i_r(s_r,b_r)}_{i_r(s_r,b_r)+2^{r-1}} \right]. \quad (5)$$

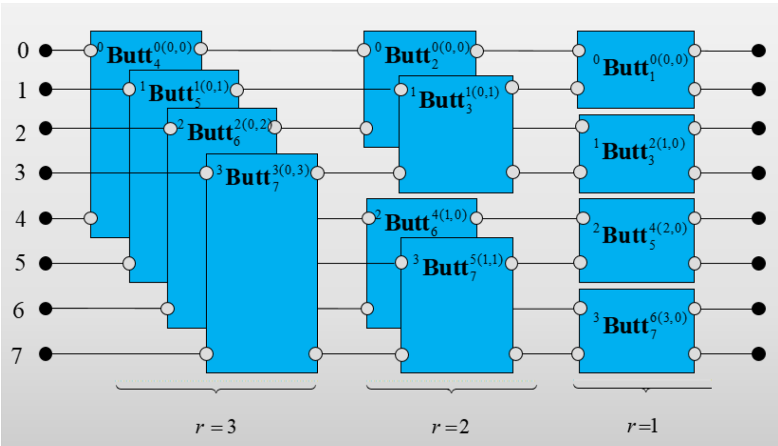

Figure 3. 8-point Fast Fourier transform (and interconnections between butterflies).

We see the basic element of the FFT is a number of operations on complex data, called the butterfly-block. Each butterfly has two inputs and two outputs (figure 3). The values at the inputs are called a_{i_r} and $b_{i_r+2^{r-1}}$, the values at the outputs are called c_{i_r} and $d_{i_r+2^{r-1}}$. Complex number $\varepsilon^{2^{n-r}\cdot b_r}$ (twiddle factor) is the weight factor, and is different for each butterfly. The c_{i_r} and $d_{i_r+2^{r-1}}$ values are computed according to the equations (see figure 4):

$${}^{p_r}\mathbf{Butt}^{i_r}_{i_r+2^{r-1}} \Rightarrow \begin{cases} c_{i_r} &= \left(a_{i_r} + b_{i_r+2^{r-1}}\right), \\ d_{i_r+2^{r-1}} &= \left(a_{i_r} - b_{i_r+2^{r-1}}\right)\varepsilon^{2^{n-r}\cdot b_r}. \end{cases}$$

Every butterfly consists of one complex addition, one complex subtraction and one complex multiplication. In this case, each butterfly in effect executes a two-point FFT.

By combining the butterfly operations in a suitable manner, a 2N point FFT is created. We see that the butterfly representation of the FFT algorithm is an elegant representation, showing the data-flow and the operations on the data in a graphical manner. In the butterfly representation of the FFT, the operations are shown as blocks, and the lines connecting the butterflies represent the data-flow between the blocks.

In this paper, we introduce two new structures for nonlinear signal processing. Both structures are based on a n-step decomposition of DFT (where $n = \log_2 N$). Our approach is synthetic and is based on the following observation: a few types of fast unitary matrices of small order generate recursively fast unitary transforms of arbitrary order. In this paper, we attempt to generalize the fast linear transformations to various non-linear cases based on parallel/serial concatenation elementary nonlinear 2x2-transforms (non-linear butterfly maps). Using this framework we shall define a large family of FUT and derive simply a number of old and new results about the FFT algorithms, other known transforms and establish structural properties between transforms.

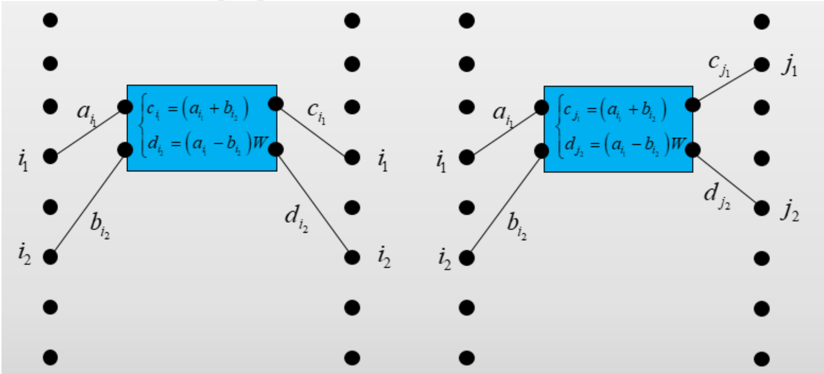

Figure 4. Butterfly in-place and nonin-place operations.

3. Fast nonlinear transforms

We shall present two elementary rules which generate a new non-linear $(N \times N)$-transforms from some basis non-linear (2×2)-transforms. These rules will then be used in a constructive and systematic fashion to generate non-linear transforms. For each rule we relate the set of elementary basis non-linear (2×2)-transforms for the new $(N \times N)$-transform.

Let $\mathbf{V}_N$ be a N-dimension space spanned on basis $\{e_0, e_1, \ldots, e_{N-1}\}$, *i.e.*

$$\mathbf{V}_N = \mathbf{V}_N\{e_0, e_1, \ldots, e_{N-1}\} = \mathbf{Span}\{e_0, e_1, \ldots, e_{N-1}\}.$$

Here $N = 2^n$. We divide the space $\mathbf{V}^N$ on $M = N/2 = 2^{n-1}$ 2-D spaces

$$\mathbf{V}_N = \mathbf{V}_2^0\{e_{i_0^0}, e_{i_1^0}\} \oplus \mathbf{V}_2^1\{e_{i_0^1}, e_{i_1^1}\} \oplus \ldots \oplus \mathbf{V}_2^{M-1}\{e_{i_0^{M-1}}, e_{i_1^{M-1}}\}$$

There are $Q = \prod_{k=0}^{2^{n-1}-1} C_{N-2k}^2 = \frac{N!}{2^{N/2}}$ similar partitions. Further, we introduce the following elementary non-linear (2×2)-transforms, called *basis transform* (**BT**):

$$\mathbf{BT}_2 : \mathbf{V}_2\{e_{i_0}, e_{i_1}\} \to \mathbf{V}_2\{e_{i_0}, e_{i_1}\}, \quad \overline{\mathbf{BT}}_2 : \mathbf{V}_2\{e_{i_0}, e_{i_1}\} \to \mathbf{V}_2\{e_{j_0}, e_{j_1}\}.$$

Each of them is described by two *arbitrary* functions $g_0(\cdot_{i_0}, \cdot_{i_1}),\ g_1(\cdot_{i_0}, \cdot_{i_1})$:

$$\mathbf{BT}_2 \Leftrightarrow \begin{cases} y_{i_0} = {}_{i_0}\mathbf{g}(x_{i_0}, x_{i_1}), \\ y_{i_1} = {}_{i_1}\mathbf{h}(x_{i_0}, x_{i_1}), \end{cases} \quad \overline{\mathbf{BT}}_2 \Leftrightarrow \begin{cases} y_{j_0} = {}_{j_0}\mathbf{g}(x_{i_0}, x_{i_1}), \\ y_{j_1} = {}_{j_1}\mathbf{h}(x_{i_0}, x_{i_1}). \end{cases}$$

We will denote them in two ways:

$$\begin{bmatrix} y_{i_0} \\ y_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow}\mathbf{g}(\cdot_{i_0}, \cdot_{i_1}) \\ {}_{i_1\leftarrow}\mathbf{h}(\cdot_{i_0}, \cdot_{i_1}) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow}\mathbf{g}(x_{i_0}, x_{i_1}) \\ {}_{i_1\leftarrow}\mathbf{h}(x_{i_0}, x_{i_1}) \end{bmatrix}, \quad \begin{bmatrix} y_{j_0} \\ y_{j_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow}\mathbf{g}(\cdot_{i_0}, \cdot_{i_1}) \\ {}_{j_1\leftarrow}\mathbf{h}(\cdot_{i_0}, \cdot_{i_1}) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow}\mathbf{g}(x_{i_0}, x_{i_1}) \\ {}_{j_1\leftarrow}\mathbf{h}(x_{i_0}, x_{i_1}) \end{bmatrix},$$

or as

$$\begin{bmatrix} y_{i_0} \\ y_{i_1} \end{bmatrix} = \left[\mathbf{g}, \mathbf{h} \middle| i_0, i_1\right] \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow}\mathbf{g}(x_{i_0}, x_{i_1}) \\ {}_{i_1\leftarrow}\mathbf{h}(x_{i_0}, x_{i_1}) \end{bmatrix}, \quad \begin{bmatrix} y_{j_0} \\ y_{j_1} \end{bmatrix} = \left[j_0, j_1 \middle| \mathbf{g}, \mathbf{h} \middle| i_0, i_1\right] \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow}\mathbf{g}(x_{i_0}, x_{i_1}) \\ {}_{j_1\leftarrow}\mathbf{h}(x_{i_0}, x_{i_1}) \end{bmatrix}.$$

where $\mathbf{BT}_2 := \left[g, h \middle| i_0, i_1\right] = \begin{bmatrix} {}_{i_0\leftarrow}g(\cdot_{i_0}, \cdot_{i_1}) \\ {}_{i_1\leftarrow}h(\cdot_{i_0}, \cdot_{i_1}) \end{bmatrix}$ and $\overline{\mathbf{BT}}_2 := \left[j_0, j_1 \middle| \mathbf{g}, \mathbf{h} \middle| i_0, i_1\right] = \begin{bmatrix} {}_{j_0\leftarrow}\mathbf{g}(\cdot_{i_0}, \cdot_{i_1}) \\ {}_{j_1\leftarrow}\mathbf{h}(\cdot_{i_0}, \cdot_{i_1}) \end{bmatrix}$.

Sometimes we will use weights for variables and “amplitudes” for non-linear functions:

$$\begin{bmatrix} y_{i_0} \\ y_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{i_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow} a_0 \mathbf{g}(w_{00} x_{i_0}, w_{01} x_{i_1}) \\ {}_{i_1\leftarrow} a_1 \mathbf{h}(w_{10} x_{i_0}, w_{11} x_{i_1}) \end{bmatrix},$$

$$\begin{bmatrix} y_{j_0} \\ y_{j_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{j_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow} a_0 \mathbf{g}(w_{00} x_{i_0}, w_{01} x_{i_1}) \\ {}_{j_1\leftarrow} a_1 \mathbf{h}(w_{10} x_{i_0}, w_{11} x_{i_1}) \end{bmatrix}.$$

We will denote them by the following ways

$$\begin{bmatrix} y_{i_0} \\ y_{i_1} \end{bmatrix} = \left[a_0 \mathbf{g}\left(w_{00}, w_{01}\right), a_1 \mathbf{h}\left(w_{10}, w_{11}\right) \middle| i_0, i_1 \right] \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} :=$$

$$:= \begin{bmatrix} {}_{i_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{i_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{i_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(x_{i_0}), w_{01}(x_{i_1})\right) \\ {}_{i_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(x_{i_0}), w_{11}(x_{i_1})\right) \end{bmatrix},$$

$$\begin{bmatrix} y_{j_0} \\ y_{j_1} \end{bmatrix} = \left[j_0, j_1 \middle| a_0 \mathbf{g}\left(w_{00}, w_{01}\right), a_1 \mathbf{h}\left(w_{10}, w_{11}\right) \middle| i_0, i_1 \right] \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} :=$$

$$:= \begin{bmatrix} {}_{j_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{j_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix} \circ \begin{bmatrix} x_{i_0} \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} {}_{j_0\leftarrow} a_0 \mathbf{g}(w_{00} x_{i_0}, w_{01} x_{i_1}) \\ {}_{j_1\leftarrow} a_1 \mathbf{h}(w_{10} x_{i_0}, w_{11} x_{i_1}) \end{bmatrix},$$

where

$$\mathbf{BT}_2 := \left[a_0 \mathbf{g}\left(w_{00}, w_{01}\right), a_1 \mathbf{h}\left(w_{10}, w_{11}\right) \middle| i_0, i_1 \right] = \begin{bmatrix} {}_{i_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{i_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix},$$

$$\overline{\mathbf{BT}}_2 := \left[j_0, j_1 \middle| a_0 \mathbf{g}\left(w_{00}, w_{01}\right), a_1 \mathbf{h}\left(w_{10}, w_{11}\right) \middle| i_0, i_1 \right] = \begin{bmatrix} {}_{j_0\leftarrow} a_0 \mathbf{g}\left(w_{00}(\cdot_{i_0}), w_{01}(\cdot_{i_1})\right) \\ {}_{j_1\leftarrow} a_1 \mathbf{h}\left(w_{10}(\cdot_{i_0}), w_{11}(\cdot_{i_1})\right) \end{bmatrix}.$$

In particular, if

$$\begin{bmatrix} a_0 \mathbf{g}\left(w_{00}(\cdot), w_{01}(\cdot)\right) \\ a_1 \mathbf{h}\left(w_{10}(\cdot), w_{11}(\cdot)\right) \end{bmatrix} = \begin{bmatrix} a_0 \mathbf{Add}\left(w_{00}(\cdot), w_{01}(\cdot)\right) \\ a_1 \mathbf{Sub}\left(w_{10}(\cdot), w_{11}(\cdot)\right) \end{bmatrix} = \begin{bmatrix} a_0 \left(w_{00}(\cdot) + w_{01}(\cdot)\right) \\ a_1 \left(w_{10}(\cdot) - w_{11}(\cdot)\right) \end{bmatrix},$$

then it is a linear matrix transform

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_0 \mathbf{Add}\left(w_{00}(\cdot), w_{01}(\cdot)\right) \\ a_1 \mathbf{Sub}\left(w_{10}(\cdot), w_{11}(\cdot)\right) \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} a_0 \left(w_{00} x_0 + w_{01} x_1\right) \\ a_1 \left(w_{10} x_0 - w_{11} x_1\right) \end{bmatrix} = \begin{bmatrix} a_0 & \\ & a_1 \end{bmatrix} \begin{bmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}.$$

The analysis of FFT (5) has been done by factorization of the matrix $\mathbf{FT}$ into a set of largely sparse matrices, each expressing a stage of computation. The general formulas for nonlinear transforms below have been derived for this interconnection scheme.

An arbitrary $S = M - 1 = 2^{n-1} - 1$ basis transforms generate in $\mathbf{V}_N$ some nonlinear radix-2 *sparse non-linear transform* (**ST**)

$$\mathbf{ST} = \begin{array}{|c|} \hline \mathbf{BT}_2^0 : \mathbf{V}_2^0\{e_{i_0^0}, e_{i_1^0}\} \to \mathbf{V}_2^0\{e_{i_0^0}, e_{i_1^0}\} \\ \hline \mathbf{BT}_2^1 : \mathbf{V}_2^1\{e_{i_0^1}, e_{i_1^1}\} \to \mathbf{V}_2^1\{e_{i_0^1}, e_{i_1^1}\} \\ \hline \vdots \\ \hline \mathbf{BT}_2^S : \mathbf{V}_2^S\{e_{i_0^S}, e_{i_1^S}\} \to \mathbf{V}_2^S\{e_{i_0^S}, e_{i_1^S}\} \\ \hline \end{array} = \begin{array}{|c|} \hline \left[\mathbf{g}^0, \mathbf{h}^0 \middle| i_0^0, i_1^0\right] \\ \hline \left[\mathbf{g}^1, \mathbf{h}^1 \middle| i_0^1, i_1^1\right] \\ \hline \vdots \\ \hline \left[\mathbf{g}^S, \mathbf{h}^S \middle| i_0^S, i_1^S\right] \\ \hline \end{array} = \bigoplus_{p=0}^{S} \mathbf{BT}_2^p (i_0^p, i_1^p) = \bigoplus_{p=0}^{S} \left[\mathbf{g}^p, \mathbf{h}^p \middle| i_0^p, i_1^p\right], \quad (6)$$

for basis transforms of the type $\mathbf{BT}_2$ and

$$\overline{\mathbf{ST}}=\begin{array}{|c|}\hline \overline{\mathbf{BT}}{}^{0}_{2}:\mathbf{V}^{0}_{2}\{e_{i^0_0},e_{i^0_1}\}\to\mathbf{V}^{0}_{2}\{e_{j^0_0},e_{j^0_1}\}\\ \hline \overline{\mathbf{BT}}{}^{1}_{2}:\mathbf{V}^{1}_{2}\{e_{i^1_0},e_{i^1_1}\}\to\mathbf{V}^{1}_{2}\{e_{j^1_0},e_{j^1_1}\}\\ \hline \vdots\\ \hline \overline{\mathbf{BT}}{}^{S}_{2}:\mathbf{V}^{S}_{2}\{e_{i^S_0},e_{i^S_1}\}\to\mathbf{V}^{S}_{2}\{e_{j^S_0},e_{j^S_1}\}\\ \hline\end{array}=\begin{array}{|c|}\hline \left[j^0_0,j^0_1\middle|\mathbf{g}^0,\mathbf{h}^0\middle|i^0_0,i^0_1\right]\\ \hline \left[j^1_0,j^1_1\middle|\mathbf{g}^1,\mathbf{h}^1\middle|i^1_0,i^1_1\right]\\ \hline \vdots\\ \hline \left[j^S_0,j^S_1\middle|\mathbf{g}^S,\mathbf{h}^S\middle|i^S_0,i^S_1\right]\\ \hline\end{array}=\bigoplus_{p=0}^{S}\overline{\mathbf{BT}}{}^{p}_{2}(j^p_0,j^p_1/i^p_0,i^p_1)=\bigoplus_{p=0}^{S}\left[j^p_0,j^p_1\middle|\mathbf{g}^p,\mathbf{h}^p\middle|i^p_0,i^p_1\right],\qquad(7)$$

for basis transforms of the type $\overline{\mathbf{BT}}$. Each sparse transform contains S basis transforms.

The sequence $\{i^0_0,i^0_1;i^1_0,i^1_1;\ldots;i^{M-1}_0,i^{M-1}_1\}$ forms addressing scheme of all basis transforms $\{\mathbf{BT}^r_2\}_{r=0}^{S}$ for the so-called in-place radix-2 FNLT and two sequences $\left\{\begin{matrix}i^0_0,\ i^0_1;\ i^1_0,\ i^1_1;\ldots;i^{M-1}_0,\ i^{M-1}_1\\ j^0_0,\ j^0_1;\ j^1_0,\ j^1_1;\ldots;j^{M-1}_0,\ j^{M-1}_1\end{matrix}\right\}$ form two addressing schemes of all basis transforms $\{\overline{\mathbf{BT}}{}^r_2\}_{r=0}^{S}$ for the so-called nonin-place radix-2 FNLT.

Let $\mathrm{A}=\{{}^r\mathbf{ST}\}_{r=0}^{Q-1}$ and $\mathrm{B}=\{{}^r\overline{\mathbf{ST}}\}_{r=0}^{Q-1}$ be two full sets of the non-linear radix-2 “sparse” transforms ${}^r\mathbf{ST}:\mathbf{V}^N_{\text{in}}\to\mathbf{V}^N_{\text{out}}$, ${}^r\overline{\mathbf{ST}}:\mathbf{V}^N_{\text{in}}\to\mathbf{V}^N_{\text{out}}$ of the forms (6) and (7), respectively.

Definition 1. The following $\circ$-products

$$\mathbf{FNLT}=\prod_{r=1}^{L}{}^r\mathbf{ST}=\prod_{r=1}^{L}\left[\bigoplus_{p=0}^{S}{}^r\mathbf{BT}^p_2({}^ri^p_0,{}^ri^p_1)\right]=\prod_{r=1}^{L}\left[\bigoplus_{p=0}^{S}\left[{}^r\mathbf{g}^p,{}^r\mathbf{h}^p\middle|{}^ri^p_0,{}^ri^p_1\right]\right],\qquad(8)$$

$$\overline{\mathbf{FNLT}}=\prod_{r=1}^{L}{}^r\overline{\mathbf{ST}}=\prod_{r=1}^{L}\left[\bigoplus_{p=0}^{S}{}^r\overline{\mathbf{BT}}{}^p_2({}^rj^p_0,{}^rj^p_1/{}^ri^p_0,{}^ri^p_1)\right]=\prod_{k=1}^{L}\left[\bigoplus_{p=0}^{S}\left[{}^rj^p_0,{}^rj^p_1\middle|{}^r\mathbf{g}^p,{}^r\mathbf{h}^p\middle|{}^ri^p_0,{}^ri^p_1\right]\right]\qquad(9)$$

are called the in-place and nonin-place radix-2 FNLTs of stage L respectively. For classical orthogonal (and unitary) transforms $L=\log_2 N=n$. In details, these expressions have the following forms:

$$\mathbf{FNLT}=\prod_{r=1}^{L}{}^r\mathbf{ST}={}^L\mathbf{ST}\circ\ldots\circ{}^2\mathbf{ST}\circ{}^1\mathbf{ST}=$$

$$=\begin{array}{|c|}\hline {}^L\mathbf{BT}^0_2({}^Li^0_0,{}^Li^0_1)\\ \hline {}^L\mathbf{BT}^1_2({}^Li^1_0,{}^Li^1_1)\\ \hline \vdots\\ \hline {}^L\mathbf{BT}^S_2({}^Li^S_0,{}^Li^S_1)\\ \hline\end{array}\circ\ldots\circ\begin{array}{|c|}\hline {}^2\mathbf{BT}^0_2({}^2i^0_0,{}^2i^0_1)\\ \hline {}^2\mathbf{BT}^1_2({}^2i^1_0,{}^2i^1_1)\\ \hline \vdots\\ \hline {}^2\mathbf{BT}^S_2({}^2i^S_0,{}^2i^S_1)\\ \hline\end{array}\circ\begin{array}{|c|}\hline {}^1\mathbf{BT}^0_2({}^1i^0_0,{}^1i^0_1)\\ \hline {}^1\mathbf{BT}^1_2({}^1i^1_0,{}^1i^1_1)\\ \hline \vdots\\ \hline {}^1\mathbf{BT}^S_2({}^1i^S_0,{}^1i^S_1)\\ \hline\end{array}=\qquad(10)$$

$$=\begin{array}{|c|}\hline \left[{}^L\mathbf{g}^0,{}^L\mathbf{h}^0\middle|{}^Li^0_0,{}^Li^0_1\right]\\ \hline \left[{}^L\mathbf{g}^1,{}^L\mathbf{h}^1\middle|{}^Li^1_0,{}^Li^1_1\right]\\ \hline \vdots\\ \hline \left[{}^L\mathbf{g}^S,{}^L\mathbf{h}^S\middle|{}^Li^S_0,{}^Li^S_1\right]\\ \hline\end{array}\circ\ldots\circ\begin{array}{|c|}\hline \left[{}^2\mathbf{g}^0,{}^2\mathbf{h}^0\middle|{}^2i^0_0,{}^2i^0_1\right]\\ \hline \left[{}^2\mathbf{g}^1,{}^2\mathbf{h}^1\middle|{}^2i^1_0,{}^2i^1_1\right]\\ \hline \vdots\\ \hline \left[{}^2\mathbf{g}^S,{}^2\mathbf{h}^S\middle|{}^2i^S_0,{}^2i^S_1\right]\\ \hline\end{array}\circ\begin{array}{|c|}\hline \left[{}^1\mathbf{g}^0,{}^1\mathbf{h}^0\middle|{}^1i^0_0,{}^1i^0_1\right]\\ \hline \left[{}^1\mathbf{g}^1,{}^1\mathbf{h}^1\middle|{}^1i^1_0,{}^1i^1_1\right]\\ \hline \vdots\\ \hline \left[{}^1\mathbf{g}^S,{}^1\mathbf{h}^S\middle|{}^1i^S_0,{}^1i^S_1\right]\\ \hline\end{array},$$

$$\overline{\mathbf{FNLT}}=\prod_{r=1}^{L}{}^r\overline{\mathbf{ST}}={}^L\overline{\mathbf{ST}}\circ\ldots\circ{}^1\overline{\mathbf{ST}}\circ{}^0\overline{\mathbf{ST}}=$$

$$=\begin{array}{|c|}\hline {}^L\mathbf{BT}^0_2({}^Lj^0_0,{}^Lj^0_1/{}^Li^0_0,{}^Li^0_1)\\ \hline {}^L\mathbf{BT}^1_2({}^Lj^1_0,{}^Lj^1_1/{}^Li^1_0,{}^Li^1_1)\\ \hline \vdots\\ \hline {}^L\mathbf{BT}^S_2({}^Lj^S_0,{}^Lj^S_1/{}^Li^S_0,{}^Li^S_1)\\ \hline\end{array}\circ\ldots\circ\begin{array}{|c|}\hline {}^2\mathbf{BT}^0_2({}^2j^0_0,{}^2j^0_1/{}^2i^0_0,{}^2i^0_1)\\ \hline {}^2\mathbf{BT}^1_2({}^2j^1_0,{}^2j^1_1/{}^2i^1_0,{}^2i^1_1)\\ \hline \vdots\\ \hline {}^2\mathbf{BT}^S_2({}^2j^S_0,{}^2j^S_1/{}^2i^S_0,{}^2i^S_1)\\ \hline\end{array}\circ\begin{array}{|c|}\hline {}^1\mathbf{BT}^0_2({}^1j^0_0,{}^1j^0_1/{}^1i^0_0,{}^1i^0_1)\\ \hline {}^1\mathbf{BT}^1_2({}^1j^1_0,{}^1j^1_1/{}^1i^1_0,{}^1i^1_1)\\ \hline \vdots\\ \hline {}^1\mathbf{BT}^S_2({}^1j^S_0,{}^1j^S_1/{}^1i^S_0,{}^1i^S_1)\\ \hline\end{array}=\qquad(11)$$

$$=\begin{array}{|c|}\hline \left[{}^Lj^0_0,{}^Lj^0_1\middle|{}^L\mathbf{g}^0,{}^L\mathbf{h}^0\middle|{}^Li^0_0,{}^Li^0_1\right]\\ \hline \left[{}^Lj^1_0,{}^Lj^1_1\middle|{}^L\mathbf{g}^1,{}^L\mathbf{h}^1\middle|{}^Li^1_0,{}^Li^1_1\right]\\ \hline \vdots\\ \hline \left[{}^Lj^S_0,{}^Lj^S_1\middle|{}^L\mathbf{g}^S,{}^L\mathbf{h}^S\middle|{}^Li^S_0,{}^Li^S_1\right]\\ \hline\end{array}\circ\ldots\circ\begin{array}{|c|}\hline \left[{}^2j^0_0,{}^2j^0_1\middle|{}^2\mathbf{g}^0,{}^2\mathbf{h}^0\middle|{}^2i^0_0,{}^2i^0_1\right]\\ \hline \left[{}^2j^1_0,{}^2j^1_1\middle|{}^2\mathbf{g}^1,{}^2\mathbf{h}^1\middle|{}^2i^1_0,{}^2i^1_1\right]\\ \hline \vdots\\ \hline \left[{}^2j^S_0,{}^2j^S_1\middle|{}^2\mathbf{g}^S,{}^2\mathbf{h}^S\middle|{}^2i^S_0,{}^2i^S_1\right]\\ \hline\end{array}\circ\begin{array}{|c|}\hline \left[{}^1j^0_0,{}^1j^0_1\middle|{}^1\mathbf{g}^0,{}^1\mathbf{h}^0\middle|{}^1i^0_0,{}^1i^0_1\right]\\ \hline \left[{}^1j^1_0,{}^1j^1_1\middle|{}^1\mathbf{g}^1,{}^1\mathbf{h}^1\middle|{}^1i^1_0,{}^1i^1_1\right]\\ \hline \vdots\\ \hline \left[{}^1j^S_0,{}^1j^S_1\middle|{}^1\mathbf{g}^S,{}^1\mathbf{h}^S\middle|{}^1i^S_0,{}^1i^S_1\right]\\ \hline\end{array}.$$

Each fast non-linear transform (see figure 5) contains $L(S+1)=LN/2$ basis transforms and depends on LN non-linear basis functions $\{{}^r\mathbf{g}^p, {}^r\mathbf{h}^p\}_{r=1,p=0}^{r=L,p=S}$. If all $\left[{}^r\mathbf{g}^p, {}^r\mathbf{h}^p\right]$ has the following form

$$\left[{}^r a_0^p \cdot {}^r\mathbf{g}^p\left({}^r w_{00}^p(\cdot), {}^r w_{01}^p(\cdot)\right), {}^r a_1^p \cdot {}^r\mathbf{h}^p\left({}^r w_{10}^p(\cdot), {}^r w_{11}^p(\cdot)\right)\right],$$

then each fast non-linear transform depends on LN non-linear basis functions $\{{}^r\mathbf{g}^p, {}^r\mathbf{h}^p\}_{r=1,p=0}^{r=L,p=S}$, LN coefficients $\{{}^r a_0^p, {}^r a_1^p\}_{r=1,p=0}^{r=L,p=S}$ and $2LN$ coefficients $\{{}^r w_{00}^p, w_{01}^p, {}^r w_{10}^p, {}^r w_{11}^p\}_{r=1,p=0}^{r=L,p=S}$.

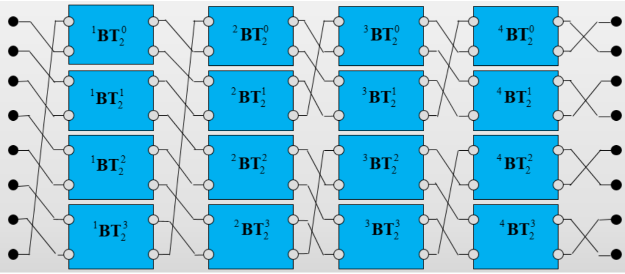

Figure 5. Fast non-linear transform contains $LN/2=16$ basis transforms and depends on $LN=32$ non-linear basis functions.

There are several address schemes in digital signal processing. For example,

$$1)\begin{cases} {}^r i_0^p = p_r, \\ {}^r i_1^p = p_r + 2^{r-1}, \end{cases} \qquad 2)\begin{cases} {}^r j_0^p = 2p, & {}^r i_0^p = p_r, \\ {}^r j_1^p = 2p+1 & {}^r i_1^p = p_r + 2^{r-1}, \end{cases}$$

where $p_r = 2^r \left\lfloor \dfrac{p}{2^{r-1}} \right\rfloor + p(\mathrm{mod}\, 2^{r-1})$. For both schemes we have the following fast nonlinear transforms:

$$\mathbf{FNLT} = \prod_{r=1}^{L}\prod_{p=0}^{S}\left[\left[{}^r\mathbf{g}^p, {}^r\mathbf{h}^p \middle| p_r, p_r + 2^{r-1}\right]\right], \tag{12}$$

$$\overline{\mathbf{FNLT}} = \prod_{r=1}^{L}\prod_{p=0}^{S}\left[\left[2p, 2p+1 \middle| {}^r\mathbf{g}^p, {}^r\mathbf{h}^p \middle| p_r, p_r + 2^{r-1}\right]\right]. \tag{13}$$

It is easy to see that nonin-place radix-2 FNLT (9) can be represented in one of the following two standard forms

$$\overline{\mathbf{FNLT}} = \prod_{r=1}^{L}\left[\bigoplus_{p=0}^{S}\left[{}^r j_0^p, {}^r j_1^p \middle| {}^r\mathbf{g}^p, {}^r\mathbf{h}^p \middle| 2p, 2p+1\right]\right], \quad \overline{\mathbf{FNLT}} = \prod_{r=1}^{L}\left[\bigoplus_{p=0}^{S}\left[2p, 2p+1 \middle| {}^r\mathbf{g}^p, {}^r\mathbf{h}^p \middle| {}^r i_0^p, {}^r i_1^p\right]\right]. \tag{14}$$

In the first of ones we use standard input ordering and in the second – standard output ordering. All three forms (10)-(11) and (14) allow a general universal form:

$$\mathbf{FNLT} = \mathbf{P}_{out} \circ \prod_{r=1}^{L}\left({}^r\mathbf{ST} \circ {}^r\mathbf{P}\right) = \mathbf{P}_{out} \circ \left({}^L\mathbf{ST} \circ {}^L\mathbf{P}\right) \circ \ldots \circ \left({}^2\mathbf{ST} \circ {}^2\mathbf{P}\right) \circ \left({}^1\mathbf{ST} \circ {}^1\mathbf{P}\right) =$$

$$= \mathbf{P}_{out} \circ \prod_{r=1}^{L}\left(\left[\bigoplus_{p=0}^{S} {}^r\mathbf{BT}_2^p\right] \circ {}^r\mathbf{P}\right) = \prod_{r=1}^{L}\left(\left[\bigoplus_{p=0}^{S}\left[{}^r\mathbf{g}^p, {}^r\mathbf{h}^p\right]\right] \circ {}^r\mathbf{P}\right) =$$

$$= \mathbf{P}_{out} \circ \left(\begin{bmatrix} {}^L\mathbf{BT}_2^0 \\ {}^L\mathbf{BT}_2^1 \\ \vdots \\ {}^L\mathbf{BT}_2^S \end{bmatrix} \circ {}^L\mathbf{P}\right) \circ \ldots \circ \left(\begin{bmatrix} {}^2\mathbf{BT}_2^0 \\ {}^2\mathbf{BT}_2^1 \\ \vdots \\ {}^2\mathbf{BT}_2^S \end{bmatrix} \circ {}^2\mathbf{P}\right) \circ \left(\begin{bmatrix} {}^1\mathbf{BT}_2^0 \\ {}^1\mathbf{BT}_2^1 \\ \vdots \\ {}^1\mathbf{BT}_2^S \end{bmatrix} \circ {}^1\mathbf{P}\right) =$$

$$= \mathbf{P}_{out} \circ \left(\begin{bmatrix} \left[{}^L\mathbf{g}^0, {}^L\mathbf{h}^0\right] \\ \left[{}^L\mathbf{g}^1, {}^L\mathbf{h}^1\right] \\ \vdots \\ \left[{}^L\mathbf{g}^S, {}^L\mathbf{h}^S\right] \end{bmatrix} \circ {}^L\mathbf{P}\right) \circ \ldots \circ \left(\begin{bmatrix} \left[{}^2\mathbf{g}^0, {}^2\mathbf{h}^0\right] \\ \left[{}^2\mathbf{g}^1, {}^2\mathbf{h}^1\right] \\ \vdots \\ \left[{}^2\mathbf{g}^S, {}^2\mathbf{h}^S\right] \end{bmatrix} \circ {}^2\mathbf{P}\right) \circ \left(\begin{bmatrix} \left[{}^1\mathbf{g}^0, {}^1\mathbf{h}^0\right] \\ \left[{}^1\mathbf{g}^1, {}^1\mathbf{h}^1\right] \\ \vdots \\ \left[{}^1\mathbf{g}^S, {}^1\mathbf{h}^S\right] \end{bmatrix} \circ {}^1\mathbf{P}\right),$$

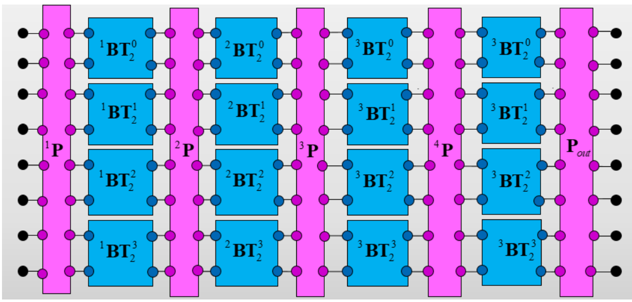

Figure 6. General universal form of **FNLT,** where basis transforms have standard input and output ordering.

where we use basis transforms with standard input and output ordering (see figure 6):

$$ {}^{r}\mathbf{BT}_2^{p} = \left[{}^{r}\mathbf{g}^{p}, {}^{r}\mathbf{h}^{p} \right] := \left[2p, 2p+1 \middle| {}^{r}\mathbf{g}^{p}, {}^{r}\mathbf{h}^{p} \middle| 2p, 2p+1 \right], $$

and where $\mathbf{P}_{out}, {}^{1}\mathbf{P} \equiv \mathbf{P}_{inp}$ are output and input permutation matrices.

4. The explicitly invertible transformation

In this section, we introduce first of all the simplest version of the *explicitly invertible* transformation: it consists of a change of variables, involving 2 *arbitrary* functions $\mathbf{g}(\cdot,\cdot), \mathbf{h}(\cdot,\cdot)$, from 2 quantities x_1, x_2 to 2 quantities y_1, y_2 and vice versa. It reads as follows $\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} \mathbf{g}(\cdot_0,\cdot_1) \\ \mathbf{h}(\cdot_0,\cdot_1) \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$.

If $\mathbf{f}(\cdot_0,\cdot_1) = \mathbf{f}_{00}(\cdot_0) + \mathbf{f}_{01}(\cdot_1)$ and $\mathbf{h}(\cdot_0,\cdot_1) = \mathbf{h}_{10}(\cdot_0) + \mathbf{h}_{11}(\cdot_1)$ then

$$ \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{00}(\cdot_0) + \mathbf{f}_{01}(\cdot_1) \\ \mathbf{h}_{10}(\cdot_0) + \mathbf{h}_{11}(\cdot_1) \end{bmatrix} \circ \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{00}(\cdot_0) & \mathbf{f}_{01}(\cdot_1) \\ \mathbf{h}_{10}(\cdot_0) & \mathbf{h}_{11}(\cdot_1) \end{bmatrix} \circ \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{00}(x_0) + \mathbf{f}_{01}(x_1) \\ \mathbf{h}_{10}(x_0) + \mathbf{h}_{11}(x_1) \end{bmatrix} $$

In particular, we are going to use the following basis transforms

$$ {}_1\mathbf{BT}_2 : \begin{cases} y_1 = x_1 + \mathbf{f}(x_2) = x_1 + \mathbf{f}(x_2), \\ y_2 = x_2 + \mathbf{h}(y_1) = x_2 + \mathbf{h}(x_1 + \mathbf{f}_1(x_2)), \end{cases} \quad {}_2\mathbf{BT}_2 : \begin{cases} y_1 = x_2 + \mathbf{f}(x_1) = \mathbf{f}(x_1) + x_2, \\ y_2 = x_1 + \mathbf{h}(y_1) = x_1 + \mathbf{h}(\mathbf{f}(x_1) + x_2), \end{cases} $$

$$ {}_3\mathbf{BT}_2 : \begin{cases} y_1 = [cx_1 + sx_2] + [c\mathbf{f}(x_2) + s\mathbf{h}(x_1 + \mathbf{f}(x_2))], \\ y_2 = [sx_1 - cx_2] + [s\mathbf{f}(x_2) - c\mathbf{h}(x_1 + \mathbf{f}(x_2))], \end{cases} $$

$$ {}_4\mathbf{BT}_2 : \begin{cases} y_1 = [sx_1 + cx_2] + [c\mathbf{f}(x_1) + s\mathbf{h}(\mathbf{f}(x_1) + x_2)], \\ y_2 = [sx_2 - cx_1] + [s\mathbf{f}(x_1) - c\mathbf{h}(\mathbf{f}(x_1) + x_2)], \end{cases} $$

or in matrix-like form

$$ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_1\mathbf{BT}_2 \circ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}(\cdot_1) + \mathbf{f}(\cdot_2) \\ \mathbf{h}(\cdot_1 + \mathbf{f}(\cdot_2)) + \mathbf{1}(\cdot_2) \end{bmatrix} \circ \begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix} = \begin{bmatrix} x_1 + \mathbf{f}(x_2) \\ x_2 + \mathbf{h}(x_1 + \mathbf{f}(x_2)) \end{bmatrix}, $$

$$ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_2\mathbf{BT}_2 \circ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\cdot_1) + \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_1) + \mathbf{h}(\cdot_2 + \mathbf{f}(\cdot_1)) \end{bmatrix} \circ \begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}(x_1) + x_2 \\ x_1 + \mathbf{h}(\mathbf{f}(x_1) + x_2) \end{bmatrix}, $$

$$ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_3\mathbf{BT}_2 \circ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} [c\mathbf{1}(\cdot_1) + s\mathbf{1}(\cdot_2)] + [c\mathbf{f}(\cdot_2) + s\mathbf{h}(\cdot_1 + \mathbf{f}(\cdot_2))] \\ [s\mathbf{1}(\cdot_1) - c\mathbf{1}(\cdot_2)] + [s\mathbf{f}(\cdot_2) - c\mathbf{h}(\cdot_1 + \mathbf{f}(\cdot_2))] \end{bmatrix} \circ \begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix} = $$

$$ = \begin{bmatrix} [cx_1 + sx_2] + [c\mathbf{f}(x_2) + s\mathbf{h}(x_1 + \mathbf{f}(x_2))] \\ [sx_1 - cx_2] + [s\mathbf{f}(x_2) - c\mathbf{h}(x_1 + \mathbf{f}(x_2))] \end{bmatrix}, $$

$$ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_4\mathbf{BT}_2 \circ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} [s\mathbf{1}(\cdot_1) + c\mathbf{1}(\cdot_2)] + [c\mathbf{f}(\cdot_1) + s\mathbf{h}(\mathbf{f}(\cdot_1) + \cdot_2)] \\ [s\mathbf{1}(\cdot_2) - c\mathbf{1}(\cdot_1)] + [s\mathbf{f}(\cdot_1) - c\mathbf{h}(\mathbf{f}(\cdot_1) + \cdot_2)] \end{bmatrix} \circ \begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix} = $$

$$ = \begin{bmatrix} [sx_1 + cx_2] + [c\mathbf{f}(x_1) + s\mathbf{h}(\mathbf{f}(x_1) + x_2)] \\ [sx_2 - cx_1] + [s\mathbf{f}(x_1) - c\mathbf{h}(\mathbf{f}(x_1) + x_2)] \end{bmatrix}. $$

where $c := \cos\varphi,\ s := \sin\varphi$. It is easy to see that

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_1\mathbf{BT}_2(\mathbf{f},\mathbf{h})\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ \mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} \mathbf{1}(\cdot_1) & \mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ \mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow}\left(x_1 + \mathbf{f}(x_2)\right) \\ {}_{2\leftarrow} x_2 \end{bmatrix} = \begin{bmatrix} x_1 + \mathbf{f}(x_2) \\ x_2 + \mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) \end{bmatrix},$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_2\mathbf{BT}_2(\mathbf{f},\mathbf{h})\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ \mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} \mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_1) & \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} x_2 \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ \mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow}\mathbf{f}(x_1) + x_2 \\ {}_{2\leftarrow} x_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}(x_1) + x_2 \\ x_1 + \mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) \end{bmatrix},$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_3\mathbf{BT}_3(\mathbf{f},\mathbf{h},\varphi)\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}\circ\left({}_1\mathbf{BT}_2(\mathbf{f},\mathbf{h})\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} cx_1 + sx_2 + c\mathbf{f}(x_2) + s\mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) \\ sx_1 - cx_2 + s\mathbf{f}(x_2) - c\mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) \end{bmatrix},$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = {}_4\mathbf{BT}_3(\mathbf{f},\mathbf{h},\varphi)\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}\circ\left({}_2\mathbf{BT}_2(\mathbf{f},\mathbf{h})\circ\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} sx_1 + cx_2 + c\mathbf{f}(x_1) + s\mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) \\ sx_2 - cx_1 + s\mathbf{f}(x_1) - c\mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) \end{bmatrix}.$$

The most remarkable aspect of these transformations is its *explicitly invertible* character. For example,

$$\begin{bmatrix} \mathbf{1}(\cdot_1) & -\mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & -\mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} x_1 + \mathbf{f}(x_2) \\ {}_{2\leftarrow} x_2 + \mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & -\mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} x_1 + \mathbf{f}(x_2) \\ -\mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) + x_2 + \mathbf{h}\left(x_1 + \mathbf{f}(x_2)\right) \end{bmatrix} = \begin{bmatrix} \mathbf{1}(\cdot_1) & -\mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} x_1 + \mathbf{f}(x_2) \\ {}_{2\leftarrow} x_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 + \mathbf{f}(x_2) - \mathbf{f}(x_2) \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

$$\begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_2) & \end{bmatrix}\circ\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_2) & \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} \mathbf{f}(x_1) + x_2 \\ {}_{2\leftarrow} x_1 + \mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) \end{bmatrix}\right) =$$

$$= \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} -\mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) + x_1 + \mathbf{h}\left(\mathbf{f}(x_1) + x_2\right) \\ \mathbf{f}(x_1) + x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} {}_{1\leftarrow} x_1 \\ {}_{2\leftarrow} \mathbf{f}(x_1) + x_2 \end{bmatrix} =$$

$$= \begin{bmatrix} x_1 \\ \mathbf{f}(x_1) + x_2 - \mathbf{f}(x_2) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Hence,

$${}_1\mathbf{BT}_2^{-1} = \begin{bmatrix} \mathbf{1}(\cdot_1) & -\mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\begin{bmatrix} \cdot_1 \\ \cdot_2 \end{bmatrix}\right),$$

$${}_2\mathbf{BT}_2^{-1} = \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ -\mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}\circ\left(\begin{bmatrix} -\mathbf{h}(\cdot_1) & \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_2) & \end{bmatrix}\circ\begin{bmatrix} \cdot_1 \\ \cdot_2 \end{bmatrix}\right),$$

$${}_3\mathbf{BT}_2^{-1} = {}_1\mathbf{BT}_2^{-1}\circ\begin{bmatrix} c\mathbf{1}(\cdot_1) & -s\mathbf{1}(\cdot_2) \\ -s\mathbf{1}(\cdot_1) & -c\mathbf{1}(\cdot_2) \end{bmatrix}, \quad {}_4\mathbf{BT}_2^{-1} = {}_1\mathbf{BT}_2^{-1}\circ\begin{bmatrix} c\mathbf{1}(\cdot_1) & -s\mathbf{1}(\cdot_2) \\ -s\mathbf{1}(\cdot_1) & -c\mathbf{1}(\cdot_2) \end{bmatrix}.$$

Note that both the *direct and inverse transforms* involve *only* the 2 *arbitrary* functions $\mathbf{f},\mathbf{h}$, and *not* their inverses.
Let

$$ {}_0\mathbf{L}_2(\mathbf{f}) = \begin{bmatrix} \mathbf{1}(\cdot_1) & \\ \mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \end{bmatrix}, \quad {}_1\mathbf{L}_2(\mathbf{f}) = \begin{bmatrix} \mathbf{1}(\cdot_1) & \mathbf{f}(\cdot_2) \\ & \mathbf{1}(\cdot_2) \end{bmatrix}, \quad {}_2\mathbf{L}_2(\mathbf{f}) = \begin{bmatrix} & \mathbf{1}(\cdot_1) \\ \mathbf{1}(\cdot_2) & \mathbf{f}(\cdot_1) \end{bmatrix}, \quad {}_3\mathbf{L}_2(\mathbf{f}) = \begin{bmatrix} \mathbf{f}(\cdot_1) & \mathbf{1}(\cdot_2) \\ \mathbf{1}(\cdot_1) & \end{bmatrix}. \quad (15) $$

Obviously,

$$ {}_1\mathbf{BT}_2(\mathbf{f},\mathbf{h}) = {}_0\mathbf{L}_2(\mathbf{h}) \circ \left({}_1\mathbf{L}_2(\mathbf{f}) \circ \right), \quad {}_2\mathbf{BT}_2(\mathbf{f},\mathbf{h}) = {}_0\mathbf{L}_2(\mathbf{h}) \circ \left({}_3\mathbf{L}_2(\mathbf{f}) \circ \right) $$

Using 4 elementary basis transforms (15) we can construct

1) 4^2 different basis transforms with two nonlinear functions

$$ \left\{ {}_{(q_1 q_0)}\mathbf{BT}_2(\mathbf{f}_{q_1},\mathbf{f}_{q_0}) \right\}_{q_1=0,q_0=0}^{q_1=3,q_0=3} = \left\{ {}_{q_1}\mathbf{L}_2(\mathbf{f}_{q_1}) \circ \left({}_{q_0}\mathbf{L}_2(\mathbf{f}_{q_0}) \circ \right) \right\}_{q_1=0,q_0=0}^{q_1=3,q_0=3}, $$

2) 4^3 different basis transforms with two nonlinear functions

$$ \left\{ {}_{(q_2 q_1 q_0)}\mathbf{BT}_2(\mathbf{f}_{q_2},\mathbf{f}_{q_1},\mathbf{f}_{q_0}) \right\}_{q_2=0,q_1=0,q_0=0}^{q_2=3,q_1=3,q_0=3} = \left\{ {}_{q_2}\mathbf{L}_2(\mathbf{f}_{q_2}) \circ \left({}_{q_1}\mathbf{L}_2(\mathbf{f}_{q_1}) \circ \left({}_{q_0}\mathbf{L}_2(\mathbf{f}_{q_0}) \circ \right) \right) \right\}_{q_2=0,q_1=0,q_0=0}^{q_2=3,q_1=3,q_0=3}, $$

and so on. We can continue this process until the m -th step on which it will be built 4^m different basis transforms:

$$ \left\{ {}_{(q_{m-1}\ldots q_1 q_0)}\mathbf{BT}_2(\mathbf{f}_{q_{m-1}},\ldots,\mathbf{f}_{q_1},\mathbf{f}_{q_0}) \right\}_{q_{m-1}=0,\ldots,q_1=0,q_0=0}^{q_{m-1}=3,\ldots,q_1=3,q_0=3} = $$

$$ = \left\{ {}_{q_2}\mathbf{L}_2(\mathbf{f}_{q_2}) \circ \left(\ldots \circ \left({}_{q_1}\mathbf{L}_2(\mathbf{f}_{q_1}) \circ \left({}_{q_0}\mathbf{L}_2(\mathbf{f}_{q_0}) \circ \right) \right) \ldots \right) \right\}_{q_{m-1}=0,\ldots,q_1=0,q_0=0}^{q_{m-1}=3,\ldots,q_1=3,q_0=3}, $$

which involve m arbitrary functions.

Let $\mathbf{BT}_2(\varphi;\mathbf{f}_{q_{m-1}},\ldots,\mathbf{f}_{q_1},\mathbf{f}_{q_0}) = \begin{bmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{bmatrix} \circ \mathbf{BT}_2(\mathbf{f}_{q_{m-1}},\ldots,\mathbf{f}_{q_1},\mathbf{f}_{q_0})$. Using such basis non-linear transforms, we can construct **FNLT** of the following generalized form:

$$ \mathbf{FNLT}(\varphi_1^0,\varphi_2^0,\ldots,\varphi_{P_\varphi}^0;\mathbf{f}_1^0,\mathbf{f}_2^0,\ldots,\mathbf{f}_{P_f}^0) = \mathbf{P}_{out} \circ \prod_{r=1}^{L} \left(\left[\bigoplus_{p=0}^{S} {}^r\mathbf{BT}_2^p({}^r\varphi^p;{}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p) \right] \circ {}^r\mathbf{P} \right) = $$

$$ = \mathbf{P}_{out} \circ \prod_{r=1}^{L} \left(\begin{array}{|c|} \hline {}^r\mathbf{BT}_2^0({}^r\varphi^p;{}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p) \\ \hline {}^r\mathbf{BT}_2^1({}^r\varphi^p;{}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p) \\ \hline \vdots \\ \hline {}^r\mathbf{BT}_2^S({}^r\varphi^p;{}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p) \\ \hline \end{array} \circ {}^r\mathbf{P} \right). \quad (16) $$

Each fast non-linear transform $\mathbf{FNLT}(\varphi_1^0,\varphi_2^0,\ldots,\varphi_{P_\varphi}^0;\mathbf{f}_1^0,\mathbf{f}_2^0,\ldots,\mathbf{f}_{P_f}^0)P_\varphi$ contains $L(S+1) = LN/2$ basis transforms ${}^r\mathbf{BT}_2^p({}^r\varphi^p;{}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p)$. It depends on $P_\varphi = LN/2$ φ -parameters $\left\{ {}^r\varphi^p \right\}_{r=1,p=0}^{r=L,p=S}$ and on $P_f = q_{m-1}LN/2$ non-linear basis functions $\left\{ {}^r\mathbf{f}_{q_{m-1}}^p,\ldots,{}^r\mathbf{f}_{q_1}^p,{}^r\mathbf{f}_{q_0}^p \right\}_{r=1,p=0}^{r=L,p=S}$.

5. The intelligent secrete nonlinear OFDM-TCS

Most of the data transmission systems nowadays use orthogonal frequency division multiplexing telecommunication system (OFDM-TCS) based on the discrete Fourier transform (DFT). Some versions of it is: digital audio broadcast (DAB), digital video broadcast (DVB), and wireless local area network (WLAN), standards such as IEEE802.11g and long term evolution (LTE and its extension LTE- Advanced, Wi-Fi (IEEE 802.11), worldwide interoperability for microwave ACCESS (WiMAX IEEE 802.16) or ADSL [48]. The concept of using parallel data broadcast by means of frequency division multiplexing (FDM) was printed in mid 60s [49].

The conventional OFDM is a multi-carrier modulation technique that is basic technology having high-speed transmission capability with bandwidth efficiency and robust performance in multipath fading environments. OFDM divides the available spectrum into a number of parallel orthogonal sub-carriers and each sub-carrier is then modulated by a low rate data stream at different carrier frequency. In OFDM system, the modulation and demodulation can be applied easily by means of inverse and direct discrete Fourier transforms (DFT). The conventional OFDM will be denoted by the symbol F_N-**OFDM**. Conventional OFDM-TCS makes use of signal orthogonality of the multiple sub-carriers $e^{j2\pi kn/N}$ (discrete complex exponential harmonics). All sub-carriers $\{\mathbf{subc}_k(n)\}_{k=0}^{N-1} = \{e^{j2\pi kn/N}\}_{k=0}^{N-1}$ form matrix of discrete orthogonal Fourier transform $\mathrm{F}_N = [\mathbf{subc}_k(n)]_{k,n=0}^{N-1} \equiv [e^{j2\pi kn/N}]_{k,n=0}^{N-1}$. At the time, the idea of using the fast algorithm of different orthogonal transforms $\mathrm{U}_N = [\mathbf{subc}_k(n)]_{k,n=0}^{N-1}$ for a software-based implementation of the OFDM's modulator and demodulator, transformed this technique from an attractive. OFDM-TCS, based on arbitrary orthogonal (unitary) transform U_N will be denoted as U_N-**OFDM**. The idea which links F_N-**OFDM** and U_N-**OFDM** is that, in the same manner that the complex exponentials $\{e^{j2\pi kn/N}\}_{k=0}^{N-1}$ are orthogonal to each-other, the members of a family of U_N-sub-carriers $\{\mathbf{subc}_k(n)\}_{k=0}^{N-1}$ (rows of the matrix U_N) will satisfy the same property.

The U_N-**OFDM** reshapes the multi-carrier transmission concept, by using carriers $\{\mathbf{subc}_k(n)\}_{k=0}^{N-1}$ instead of OFDM's complex exponentials $\{e^{j2\pi kn/N}\}_{k=0}^{N-1}$. There are a number of candidates for orthogonal function sets used in the OFDM-TCS: discrete wavelet sub-carriers [50]-[51], Golay complementary sequences [52]-[56], rectangle pulses [50], Walsh functions [57]-[59], pseudo random sequences [60], manyparameter sub-carriers [61-66] based on many-parameter transforms $\mathrm{U}_N(\varphi_1,\varphi_2,\ldots,\varphi_q)$.

Intelligent secrete OFDM TCS can be described as a dynamically reconfigurable TCS that can adaptively regulate its internal parameters as a response to changes in the surrounding environment. One of the most important capacities of Intelligent OFDM systems is their capability to optimally adapt their operating parameters based on observations and previous experiences. There are several possible approaches towards realizing such intelligent capabilities. In this work, we aim to investigate the superiority and practicability of **FNLTs** from the physical layer security (PHY-LS) perspective.

In this work, we propose a simple and effective anti-eavesdropping and anti-jamming Intelligent OFDM system, based on many-parameter fast nonlinear transforms. In our Intelligent-OFDM-TCS we use $\mathbf{FNLT}(\varphi_1,\varphi_2,\ldots,\varphi_{P_\varphi};\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_{P_f})$ instead of DFT F_N.

Each **FNLT** depends on P_φ of independent Jacobi angles $\{\varphi_1,\varphi_2,\ldots,\varphi_{P_f}\}$ and on P_f of independent nonlinear functions $\{\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_{P_f}\}$, which could be changed in dependently of one another. When parameters and non-linear functions are changed, non-linear transform $\mathbf{FNLT}(\varphi_1,\varphi_2,\ldots,\varphi_{P_\varphi};\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_{P_f})$ is changed too. The vector of parameters $(\varphi_1,\varphi_2,\ldots,\varphi_{P_\varphi}) \in \mathbf{Tor}_{P_\varphi}[0,2\pi] = [0,2\pi]^{P_\varphi}$ belongs to the P_φ-D torus. Intelligent OFDM system uses some concrete values of the parameters $\varphi_1 = \varphi_1^0, \varphi_2 = \varphi_2^0, \ldots, \varphi_q = \varphi_{P_\varphi}^0$ and some concrete the set of nonlinear basis functions $\{\mathbf{f}_1^0,\mathbf{f}_2^0,\ldots,\mathbf{f}_{P_f}^0\}$, *i.e.,* it uses a concrete realization of fast non-linear transform $\mathbf{FNLT}(\varphi_1^0,\varphi_2^0,\ldots,\varphi_{P_\varphi}^0;\mathbf{f}_1^0,\mathbf{f}_2^0,\ldots,\mathbf{f}_{P_f}^0)$. The vector $(\varphi_1^0,\varphi_2^0,\ldots,\varphi_q^0)$ and the set $\{\mathbf{f}_1^0,\mathbf{f}_2^0,\ldots,\mathbf{f}_{P_f}^0\}$ forms of the set of all bunches of φ- and f-keys (see figure 7), whose knowing is necessary for entering into the OFDM TCS with the aim of intercepting the confidential information.

Quantity of parameters and basis non-linear function can achieve the values $P_\varphi = P_f \approx 10\ 000$. So, searching the bunch of φ - and f -keys by scanning with the aim of finding the working parameters $(\varphi_1^0, \varphi_2^0, ..., \varphi_q^0)$ and $\{\mathbf{f}_1^0, \mathbf{f}_2^0, ..., \mathbf{f}_{P_f}^0\}$ is very difficult problem for the enemy cyber-means. But if, nevertheless, this key were found by the enemy in an cyber-attack, then the system could change values of the working parameters $(\varphi_1^0, \varphi_2^0, ..., \varphi_q^0)$ and $\{\mathbf{f}_1^0, \mathbf{f}_2^0, ..., \mathbf{f}_{P_f}^0\}$ for rejecting the enemy attack. As a result, the system will counteract against the enemy radio-electronic attacks.

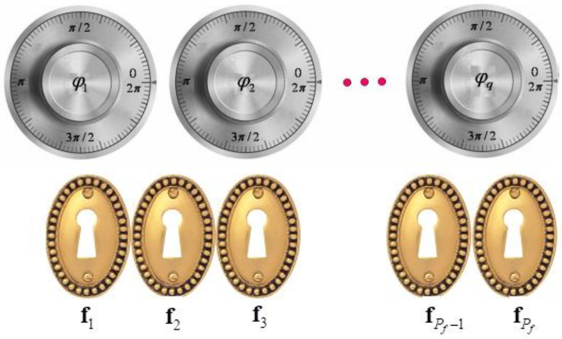

Figure 7. Bunch of φ - and f -keys.

6. Conclusions

In this work we have presented a unified approach to non-linear transforms having a fast algorithm. The use of recursive rules to describe non-linear transforms allows a systematic way to view known orthogonal, unitary and non-linear transforms, to generate new transforms. The framework provided can be used in several other studies and applications of non-linear transforms. We believe that the proposed nonlinear transforms might be useful in some wired and wireless communication applications

7. Acknowledgments

This work was supported by the RFBR grant 19-29-09022\19 and by the Ural State Forest Engineering’s Center of Excellence in «Quantum and Classical Information Technologies for Remote Sensing Systems.

8. References

[1] Rugh, W.J. Nonlinear system theory: the Volterra – Wiener approach // The Johns Hopkins University Press,1981. – 330 p.
[2] Schetzen, M. The Volterra und Winer theorems of nonlinear systems –New York: Wiley, 1980.
[3] Kim, K.I. A digital method of modeling quadratically non-linear systems with a general random input / E.J. Powers // IEEE Trans. Acoust. Speech Signal Process. – 1988. – Vol. 36(11). – P. 1758-1769.
[4] Schetzen, M. Non-linear system modeling based on the Wiener theory // Proc. IEEE. –1981. – Vol. 69(12). – P. 1557-1573.
[5] Pages-Zamora, A. The K-filter: A new architecture to model and design non-linear systems from Kolmogorov's theorem / M.A. Lagunas, M. Nahjar, A. Pehrezneira // Signal Processing. – 1995. – Vol. 44(3). – P. 249-267.
[6] Ostheimer, E. Systematic approach to nonlinear filtering associated with aggregation operators. Part 1. SISO-filters // Procedia Engineering. – 2017. – Vol. 201. – P. 372-384.
[7] Labunets, V. Systematic approach to nonlinear filtering associated with aggregation operators. Part 2. Frechet MIMO-filters / E. Ostheimer // Procedia Engineering. – 2017. – Vol. 201.–P. 385-397.
[8] Ostheimer, E. Fréchet filters for color and hyperspectral images filtering / V. Labunets, D. Komarov, T. Fedorova // Communications in Computer and Information Science. – 2015. – Vol. 542. – P. 57-70.

[9] Ostheimer, E. Families of Heron digital filters for images filtering / V. Labunets, F. Myasnikov // CEUR Workshop Proceedings. – 2015. – Vol. 1452. – P. 56-63.
[10] Ostheimer, E. New bi-, tri-, and four-lateral filters for color and hyperspectral images filtering / V. Labunets, A. Kurganski, D. Komarov, I. Artemov // Communications in Computer and Information Science. – 2015. – Vol. 542. – P. 102-113.
[11] Labunets, V.G. Filters based on aggregation operators. Part 1. Aggregation operators // International Crimean Conference Microwave and Telecommunication Technology Conference Proceedings–CriMiCo, 2014. – P. 1239-1240.
[12] Labunets, V.G. Filters based on aggregation operators. Part 2. The Kolmogorov filters / D.N. Gainanov, E. Ostheimer // 24th International Crimean Conference Microwave and Telecommunication Technology Conference Proceedings, 2014.– P. 1241-1242.
[13] Labunets, V.G. Filters based on aggregation operators. Part 3. The Heron filters / D.N. Gainanov, A.D. Tarasov, E. Ostheimer // 24th International Crimean Conference Microwave and Telecommunication Technology Conference Proceedings, 2014. – P. 1243-1244.
[14] Labunets, V.G. Filters based on aggregation operators. Part 4. Generalized vector median filters / D.N. Gainanov, R.A. Arslanova, E. Ostheimer // 24th International Crimean Conference Micro-wave and Telecommunication Technology Conference Proceedings. – 2014. – P. 1245-1246.
[15] Biglieri, S. Analysis and compensation of non-linearities in digital transmission systems / S. Barberis, M. Catena // IEEE J. Sel. Areas Commun. – 1988. – Vol. 6(1). – P. 42-51.
[16] Mulgrew, B. Applying radial basis functions // Signal Process. Mag. – 1966. – Vol. 13(2). – P. 50-65.
[17] Wahba, G. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels // Proc. Non-linear Modeling and Forecasting. In: Studies in the Sciences of Complexity – MA: Addison-Wesley, 1992. – Vol. XII. – P. 95-112.
[18] Kolmogorov, A.N. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition // Dokl. Akad. Nauk SSSR. – 1957. – Vol. 114. – P. 369-373.
[19] Arnol'd, V.I. On the representation of continuous functions depending on three variables by superpositions of continuous function depending on two variables // Dokl. Acad. Nauk SSSR. – 1957. – Vol. 114. – P. 679-681.
[20] Gashkov, S.B. On the circuit and formula complexity of Boolean functions // Probl. Kibernet. – 1980. – Vol. 37. – P. 57-118.
[21] Hecht-Nielsen, R. Kolmogorov's mapping neural network existence theorem // IEEE International Conference on Neural Networks. – San Diego: SOS Printing. – 1987. – Vol. 2. – P. 11-14.
[22] Labunets, V. New networks for nonlinear, linear and orthogonal transforms / E. Labunets // V-th Int. Workshop on Parallel Processing by Cellular Automata and Arrays (PARCELLA) – Berlin, 1990. – P. 239-244.
[23] Cooley, J.W. An algorithm for the machine calculation of complex Fourier series / J.W. Tukey // Math. Comp. – 1965. – Vol. 19(4). – P. 297-301.
[24] Whetchel, J.E. The fast Fourier–Hadamard transform and its use in signal representation and classification / D.F. Guinn // IEEE Transactions on Aerospace and Electronic Systems. – 1968. – P. 561-573.
[25] Beauchamp G. Walsh Functions and Their Applications – London, New York: Academic, 1975. – 248 p.
[26] Bracewell, R.N. Fast Hatrley transform. – New York; Oxford University Press, 1986. – 228 p.
[27] Hartley, R.V.L. A more symmetrical Fourier analysis applied to transmission problems // Proc. of the IRE. – 1942. – Vol.30. – P. 142-150.
[28] Fino, B.J. Relations between Haar and Walsh-Hadamard transforms // Proc. IEEE (Lett.). – 1972. – Vol. 60. – P. 647-648.
[29] Labunets, V.G. Generalized Haar transforms/ Multi–valued elements, structures and systems // Kiev: Institute of Cybernetics of Ukraian Academy of Sciences Press, 1983. – P. 46-58.

[30] Andrews, H.C. Computer techniques in image processing – New York: Academic Press, 1970. – 244 p.
[31] Andrews, H.C. Kronecker matrices, computer implementation and generalized spectra / J. Kane // J. Assoc. Comput. Math. – 1970. – Vol. 17. – P. 260-268.
[32] Andrews, H.C. A generalized technique for spectral analysis / K.L. Caspari // IEEE Trans. Computers. – 1970. – Vol. C-19. – P. 16-25.
[33] Ahmed, N. A generalized discrete transform / K.R. Rao, R.B. Schultz // Proc. IEEE. – 1971. – Vol. 59. – P. 1360-1362.
[34] Ahmed, N. Orthogonal Transforms for Digital Signal Processing / N. Ahmed, K.R. Rao – Heidelberg, Berlin, New York: Springer–Verlag, 1975. – 248 p.
[35] Labunets, V.G. Fast multiparameter transforms //Proceedings of Radioelectronics. – 1985. – Vol. 8. – P. 89-109.
[36] Labunets, V.G. Unified approach to fast algorithms of unitary transforms // Multivalued elements, structures and systems – Kiev: Institute of Cybernetics of Ukraian Academy of Sciences Press, 1983. – P. 58-70.
[37] Labunets, V.G. Multi–parameter Golay 2–complementary sequences and transforms / V.P. Chasovskikh, E. Ostheimer // Proceedings of the 4th International Youth Conference on Information technologies and nanotechnologies – Samara: New Technics, 2018. – P. 1013-1022.
[38] Labunets, V.G. Multi-parameter Golay *m*–complementary sequences and transforms / V.P. Chasovskikh, E. Ostheimer // Proceedings of the 4th International Youth Conference on Information technologies and nanotechnologies.–Samara: New Technics, 2018. – P. 1005-1012.
[39] Labunets, V.G. Many–parametric cyclic wavelet transforms. Part 1. The first and second canonical forms / K. Egiazarian, J. Astola, E. Ostheimer // Proceedings of the International TICSP Workshop on Spectral Methods and Multirate Signal Processing –Tampere, Finland: Tampere University Technology, 2007. – P. 111-120.
[40] Labunets, V.G. Many-parametric cyclic wavelet transforms. Part 2. The third and fourth canonical forms / K. Egiazarian, J. Astola, E. Ostheimer // Proceedings of the International TICSP Workshop on Spectral Methods and Multirate Signal Processing – Tampere, Finland: Tampere University Technology, 2007. – P. 121-132.
[41] Labunets, V.G. Fast multi-parametric wavelet transforms and packets for image processing / D.E. Komarov, E. Ostheimer // CEUR Workshop Proceedings. – 2016. – Vol. 1710. – P.134-145.
[42] Labunets, V. Multi-parametric wavelet transforms and packets / D.N. Gainanov, D.A. Berenov // Proceedings of the 11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies – Samara: New Technics, 2013. – Vol. 1. – P. 52-56.
[43] Labunets, V. The best multi-parameter wavelet transforms / V. Labunets, D. Gainanov, A. Berenov // Proceedings of the 11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies – Samara: New Technics, 2013. – Vol. 1. – P. 56-60.
[44] Labunets, V.G. Many-parameter Golay m-complementary sequences and transforms // V.P. Chasovskikh, Ju. G. Smetanin, E. Ostheimer // Computer Optics. – 2018. – Vol. 42(6). – P. 1074-1083.
[45] Demuth, G.L. Algorithms for defining mixed radix FFT flow graphs // IEEE Trans. Acoustics, Speech, and Signal Processing. – 1989. – Vol. 37(9). – P. 1349-1358.
[46] Labunets, V.G. Unified approach to fast transform algorithm // Orthogonal methods in signal processing and system analysis – Sverdlovsk: Urals Polytechnical Institute Press, 1980. – P. 4-14.
[47] Labunets, V.G. Unified approach to fast algorithms of unitary transforms // Multi–valued elements, structures and systems – Kiev: Institute of Cybernetics of Ukrainian Academy of Sciences Press, 1983. – P. 58-70.
[48] Baig, S. Performance comparison of DFT, discrete wavelet packet and wavelet transforms, in an OFDM transceiver for multipath fading channel / M. Fazal-ur-Rehman, M.J. Mughal // IEEE Communication Magazine, 2004. – P. 1-6.

[49] Chang, R.W. Synthesis of band limited orthogonal signals for multichannel data transmission // Bell Syst. Tech. J. –1966. – Vol. 45. – P. 1775-1796.
[50] Wornell, G. Emerging applications of multirate signal processing and wavelets in digital communications // Proc. IEEE. – 1966. – Vol. 84. – P. 586-603.
[51] Sandberg, S.D. Overlapped discrete multitone modulation for high-speed copper wire communications / M.A. Tzannes // IEEE Journal on Sel. Areas in Commun. – 1985. – Vol. 13. – P. 1571-1585.
[52] Halford, K. Complementary code keying for rake-based indoor wireless communication / S. Halford, M. Webster, C. Andren // Proceedings of IEEE International Symposium on Circuits and Systems, 1999. – P. 427-430.
[53] Golay, M.J.E. Multisplit spectroscopy // J. Opt. Soc. Amer. – 1949. – Vol. 39. – P. 437-444.
[54] Golay, M.J.E. Complementary series // IEEE Trans. Inform. Theory. – 1961. – Vol. 7. – P. 82-87.
[55] Davis, J.A. Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes/ J. Jedwab // IEEE Trans. Inform. Theory. – 1999. – Vol. 45. – P. 2397-2417.
[56] Fiedler, F. A framework for the construction of Golay sequences / J. Jedwab, M.G. Parker // IEEE Trans. Inform. Theory. – 2008. – Vol. 54. – P. 3114-3129.
[57] Michailow, N. WHT – GFDM for the next generation of wireless networks / L. Mendes, M. Matthe, I. Festag, A. Fettweis, G. Robust // IEEE Communications Letters. – 2015. – Vol. 19. – P. 106-109.
[58] Xiao, J. Hadamard transform combined with companding transform technique for PAPR reduction in an optical direct-detection OFDM system / J. Yu, X. Li, Q. Tang, H. Chen, F. Li, Z.Cao, L. Chen L // IEEE J. Opt. Commun. Netw. – 2012. – Vol. 4(10). – P. 709-714.
[59] Wang, S. A Walsh-Hadamard coded spectral efficient full frequency diversity OFDM system / S. Zhu, G. Zhang // IEEE Trans. Commun. – 2012. – Vol. 58(1). – P. 28-34.
[60] Wilkinson, T.A. Minimization of the peak to mean envelope power ratio of multicarrier transmission schemes by block coding / A.E. Jones // Proceedings of the IEEE 45th Vehicular Technology Conf. – 1995. – P. 825-829.
[61] Labunets, V.G. Intelligent OFDM telecommunication system. Part 1. Model of complex and quaternion systems / E. Ostheimer // Journal of Physics: Conference Series. – 2019. – Vol.1368. – P. 1-11.
[62] Labunets, V.G. Intelligent OFDM telecommunication system. Part 2. Examples of complex and quaternion many-parameter transforms / E. Ostheimer // Journal of Physics: Conference Series. – 2019. – Vol. 1368. – P. 1-14.
[63] Labunets, V.G. Intelligent OFDM telecommunication system. Part 3.Anti-eavesdropping and anti-jamming properties of system, based on many-parameter wavelet and Golay transforms / D.E. Komarov, V.P. Chasovskikh, Ju. G. Smetanin, E. Ostheimer // Journal of Physics: Conference Series. – 2019. – Vol. 1368. – P. 1-14.
[64] Labunets, V.G. Intelligent OFDM telecommunication system. Part 4.Anti-eavesdropping and anti-jamming properties of the system, based on many-parameter and fractional Fourier transforms/ S.A. Martyugin, V.P. Chasovskikh, Ju. G. Smetanin, E. Ostheimer // Journal of Physics: Conference Series. – 2019. – Vol. 1368. – P. 1-15.
[65] Labunets, V.G. Many-parameter quaternion Fourier transforms for intelligent OFDM telecommunication system /E. Ostheimer // Advances in Artificial Systems for Medicine and Education-III. Part of the Advances in Intelligent Systems and Computing – Switzeland: Springer Nature, 2020. – Vol. 1126. – P. 76-92.
[66] Labunets, V.G. Intelligent OFDM telecommunication systems based on many-parameter complex or quaternion Fourier transforms / E. Ostheimer // Advances in Intelligent Systems, Computer Science and Digital Economics. Part of the Advances in Intelligent Systems and Computing – Switzeland: Springer Nature. – 2020. – Vol. 1127. – P. 129-144.