

USING OF DISCRETE ORTHOGONAL TRANSFORMS FOR CONVOLUTION

Anna Ušáková* — Jana Kotuliaková* — Michal Zajac**

The convolution is a mathematical way of combining two functions to form a third one. It is the single most important technique in digital signal processing because it relates two signals of interest: input signal, output signal and the impulse response of the transmitting system. The convolution is not used only to compute the output of the system, when the impulse response is known, but also in the image signal processing, *eg*, for the interpolation of 2-D signals, in speech processing, *eg*, homomorphic modelling of the speech, noise reduction in the speech, but also in image processing, etc. The evaluation of the convolution in the time domain is very complicated, especially for the large lengths of the input signal or impulse response. So we try to find another way how to compute convolution. Complicated computation of the convolution in the time domain is transformed into trivial multiplication in the domain of some orthogonal transforms (*eg*, Fourier transform), for another OT's domain this evaluation is more complicated. In this paper we present a convolution formula for some selected discrete orthogonal transforms (DOT).

Keywords: convolution, orthogonal transform, discrete cosine transform, discrete Hartley transform, discrete sine transform, discrete Fourier transform

1 INTRODUCTION

Convolution is a formal mathematical operation, just as multiplication, addition, and integration. Convolution is used in the mathematics of many fields, such as probability and statistics. In linear systems, convolution is used to describe the relationship between three signals of interest: input signal, impulse response and output signal, see Fig. 1. Mathematically, we can express convolution using the following equation

$$y(n) = x(n) * h(n) = \sum_{m=0}^{D_y-1} x(m) \cdot h(n-m) \quad (1)$$

where $x(n)$ is D_x length input signal, $h(n)$ is D_h length

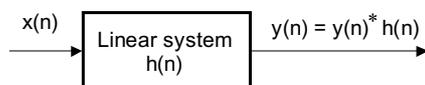


Fig. 1. Model of linear system using convolution

impulse response, $y(n)$ is D_y length output signal. The length of the output signal D_y is equal to the length of the input signal plus the length of the impulse response minus one. Convolution is a mathematical operation used in signal processing, homomorphic signal processing and image signal processing (*eg*, in image interpolation). Computing

of the convolution in the time domain is often replaced by computing of a more simple operation in the domain of a discrete orthogonal transform.

2 DISCRETE GONIOMETRIC TRANSFORMS

2.1 Discrete Fourier transform (DFT)

Nowadays, the most preferred domain is the domain derived by Fourier transform. Convolution of two sequences is transformed by discrete Fourier transform into simple multiplication of the DFTs of these two sequences. In other words, a customary way of performing convolution numerically is to take the discrete Fourier transform of each of the two given sequences and then to do complex multiplication, element by element, in the Fourier domain. Discrete Fourier transform is defined by [1] [2] [7] [10] as

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) \cdot W^{nk} \quad (2)$$

and inverse discrete Fourier transform is defined as:

$$x(n) = \sum_{k=0}^{N-1} X(k) \cdot W^{-nk} \quad (3)$$

where $k = 0, 1, \dots, N-1$, $n = 0, 1, \dots, N-1$, $W = \exp(-j \cdot \frac{2\pi}{N})$ is the system of the complex base functions ($j = \sqrt{-1}$).

For the discrete Fourier transform there applies the following convolution theorem [7]: If $y(n) = x(n) * h(n)$,

* Department of Telecommunications ** Department of Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia,

Anna.Usakova@ktl.elf.stuba.sk, jkotul@ktl.elf.stuba.sk, zajac@kmat.elf.stuba.sk,

$x(n)$ and $h(n)$ are periodic sequences with period N , and holds $x(n) \xrightarrow{\text{DFT}} X(k)$, $h(n) \xrightarrow{\text{DFT}} H(k)$, then

$$y(n) = \text{IDFT}\{X(k) \cdot Y(k)\}. \quad (4)$$

Realize that DFT algorithm is based on periodic signals and on the output there is information about one period of the spectrum. By analogy, the output of the IDFT is one period of the generated periodic signal. So, if we want to compute non-periodic convolution using DFT and IDFT, we have to choose the length of the convolution N that equals at least to the length of linear convolution. It means that before DFT we extend the input sequences with zeros to achieve the needed length.

2.2 Discrete Hartley transform (DHT)

Discrete Hartley transform is defined by [10] [3]

$$X_{Ha}(k) = N^{-1} \sum_0^{N-1} x(n) \text{cas}(2\pi kn/N). \quad (5)$$

Inverse discrete Hartley transform is given by

$$x(n) = \sum_{k=0}^{N-1} X_{Ha}(k) \text{cas}(2\pi kn/N) \quad (6)$$

where $\text{cas}(\alpha) = \cos(\alpha) + \sin(\alpha)$.

The convolution theorem obeyed by the DHT is as follows [3]. If $y(n)$ is the convolution of $x(n)$ with $h(n)$, *ie*,

$$y(n) = x(n) * h(n) = \sum_{m=0}^{D_x-1} x(m) \cdot h(n-m), \quad (7)$$

then

$$Y_{Ha}(k) = X_{Ha}(k)H_{Ha}^e(k) + X_{Ha}(-k)H_{Ha}^o(k) \quad (8)$$

where $Y_{Ha}(k)$, $X_{Ha}(k)$ and $H_{Ha}(k)$ are the DHTs of $y(n)$, $x(n)$ and $h(n)$, respectively, and $H_{Ha}(k) = H_{Ha}^e(k) + H_{Ha}^o(k)$, the sum of its even and odd parts. The even and odd parts of the Hartley transform are given by (9) and (10)

$$H_{Ha}^e = \frac{H_{Ha}(k) + H_{Ha}(-k)}{2} = \frac{1}{N} \sum_{n=0}^{N-1} h(n) \cos \frac{2\pi kn}{N}, \quad (9)$$

$$H_{Ha}^o = \frac{H_{Ha}(k) - H_{Ha}(-k)}{2} = \frac{1}{N} \sum_{n=0}^{N-1} h(n) \sin \frac{2\pi kn}{N}. \quad (10)$$

To interpret $H_{Ha}(-k)$, or any other $h()$ or $H()$ where the argument falls outside the range 0 to $N-1$, add or subtract multiples of N as needed. Thus $H_{Ha}(-1)$ is interpreted as $H_{Ha}(N-1)$. According to the reversal theorem, $h(-n)$ transforms into $H_{Ha}(-k)$.

Substituting equations (9) and (10) for even and odd part of the Hartley transform into (8), DHT of the convolution can be expressed as:

$$Y_{Ha}(k) = \frac{1}{2} \left[X_{Ha}(k)H_{Ha}(k) + X_{Ha}(-k)H_{Ha}(k) + X_{Ha}(k)H_{Ha}(-k) - X_{Ha}(-k)H_{Ha}(-k) \right]. \quad (11)$$

The use of the Hartley transform for convolution will require only two real multiplications per element (DFT require four real multiplications per element to do complex multiplications per element).

However, it often happens, especially in image processing but also in digital filtering in general, that one of the convoluting functions, say, $h(n)$, is even. In that case H_{Ha}^o (*ie*, the odd part of the DHT's of the $h(n)$) is zero, and the convolution theorem therefore simplifies to $Y_{Ha}(k) = X_{Ha}(k) \cdot H_{Ha}(k)$. In view of this simplified form of the theorem, one needs only to take the DHTs of the two data sequences, multiply together term by term the resulting two real sequences, and take one more DHT.

2.3 Discrete cosine and sine transform

We differentiate four types of bases of discrete cosine transform (DCT), so we know DCT-I, DCT-II, DCT-III and DCT-IV [8], [9], [10]. The definitions of their bases are as follow:

DCT-I

$$U_{C-I}(n, k) = \sqrt{2} \cdot \left\{ c_k \cdot c_n \cdot \cos \frac{kn\pi}{N} \right\} \quad (12)$$

where $n, k = 0, 1, \dots, N$

DCT-II

$$U_{C-II}(n, k) = \sqrt{2} \cdot \left\{ c_k \cdot \cos \frac{(2n+1)k\pi}{2N} \right\} \quad (13)$$

DCT-III

$$U_{C-III}(n, k) = \sqrt{2} \cdot \left\{ c_n \cdot \cos \frac{(2k+1)n\pi}{2N} \right\} \quad (14)$$

DCT-IV

$$U_{C-IV}(n, k) = \sqrt{2} \cdot \left\{ \cos \frac{(2k+1)(2n+1)\pi}{4N} \right\} \quad (15)$$

where for (13)–(15) $n, k = 0, 1, \dots, N-1$.

Similarly, there are four types of discrete sine transform (DST) — DST-I, DST-II, DST-III and DST-IV. They have their bases defined by following equations:

DST-I

$$U_{S-I}(n, k) = \sqrt{2} \cdot \left\{ \sin \frac{kn\pi}{N} \right\} \quad (16)$$

$n, k = 1, 2, \dots, N - 1$
DST-II

$$U_{S-II}(n, k) = \sqrt{2} \cdot \left\{ c_k \cdot \sin \frac{(2n - 1)k\pi}{2N} \right\} \quad (17)$$

$n, k = 1, 2, \dots, N - 1$
DST-III

$$U_{S-III}(n, k) = \sqrt{2} \cdot \left\{ c_n \cdot \sin \frac{(2k - 1)n\pi}{2N} \right\} \quad (18)$$

$n, k = 1, 2, \dots, N - 1$
DST-IV

$$U_{S-IV}(n, k) = \sqrt{2} \cdot \left\{ \sin \frac{(2k + 1)(2n + 1)\pi}{4N} \right\} \quad (19)$$

$n, k = 0, 1, \dots, N - 1$

For all discrete cosine and sine transforms parameter c_p is given by:

$$c_p = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } p = 0 \text{ or } N, \\ 1 & \text{otherwise.} \end{cases} \quad (20)$$

The convolution theorem for each of the discrete sine and cosine transforms is shown in Table 1 ($X^T(k)$ is the T's of $x(n)$ multiplied by $1/c_n$).

3 DISCRETE TRANSFORMS WITH RECTANGULAR FUNCTIONS

3.1 Walsh-Hadamard transform (WHT)

It is well known that for some orthogonal transforms the image of the convolution of a pair of functions is equal to the product of their transforms. In order to obtain an analogous rule for Walsh-Hadamard transform the convolution should be replaced by the dyadic convolution. Recall that if

$$m = \sum_{i=0}^{\infty} m_i 2^i, \quad n = \sum_{i=0}^{\infty} n_i 2^i$$

are binary expansions of non-negative integers m, n then their dyadic sum is

$$m \oplus n = \sum_{i=0}^{\infty} |m_i - n_i| 2^i. \quad (21)$$

Denote by $R^{m \times n}$ the space of all $m \times n$ matrices with real entries.

Let $N_m = 2^m$, $m = 0, 1, 2, \dots$. Then a (column) vector $y = x * h \in R^{N_m \times 1}$ is called dyadic convolution of $x, h \in R^{N_m \times 1}$ if its n -th entry is

$$y(n) = \frac{1}{N_m} \sum_{d=0}^{N_m-1} x(d)h(n \oplus d). \quad (22)$$

where $n = 0, 1, \dots, N_m - 1$.

The Walsh-Hadamard transform of $x \in R^{2^m-1}$ is given by [1]:

$$X_h(k) = \frac{1}{2^m} W_{2^m} x(n) \quad (23)$$

and inverse Walsh-Hadamard transform is defined as

$$x(n) = W_{2^m} X_h(k), \quad (24)$$

where W_{2^m} is the $2^m \times 2^m$ Walsh-Hadamard transform matrix.

The theorem for dyadic convolution using Hadamard-ordered Walsh-Hadamard transform (WHT) $_h$ is: Let m be a non-negative integer, $N_m = 2^m$ and $x, y \in R^{N_m \times 1}$. Then holds

$$y(n) = x(n) * h(n) \xrightarrow{(WHT)_h} Y_h(k) = X_h(k)H_h(k) \quad (25)$$

$n, k = 0, 1, \dots, 2^m - 1$, ie, Y_h is coordinate wise multiplication of X_h and H_h .

The theorem holds also for any other order of Walsh functions (ie Walsh, Paley and Cal-Sal ordering) since the corresponding transform of $x \in R^{N_m \times 1}$ can be obtain from its Hadamard transform X_h by a fixed permutation of components of X_h .

The theorem for the cyclic convolution using (WHT) $_h$ is derived from the cyclic shift theorem, which says that if $X_h(k)$ is (WHT) $_h$'s of sequence $x(n)$, and $x_l(n)$ is obtained by cyclic shift by l step, then

$$x_l(n) \xrightarrow{(WHT)_h} A(m)^l X_h(k), \quad (26)$$

where $A(m)$ is called similarity transformation and equals to

$$A(m) = \frac{1}{2^m} W_{2^m} M(m) W_{2^m}$$

$$M(m) = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \quad (27)$$

Then for cyclic convolution defined by (1) applies

$$y(N_m/2 - 1 - j) = y_j(N_m/2 - 1),$$

$$y(N_m - 1 - j) = y_j(N_m - 1),$$

for $j = 0, 1, \dots, N_m/2 - 1$

$$\text{and } y_j(n) = (IWHT)_h \{ X_h(k) H_{jh}(k) \}, \quad (28)$$

where $X_h(k) = (WHT)_h x(n)$,

$$H_{jh}(k) = A(m)^j \cdot H_h(k),$$

$$H_h(k) = (WHT)_h \{ h(n) \},$$

$n, k = 0, 1, \dots, N_m - 1$.

Table 1. Relation between convolution and DOT

Type of DOT	Convolution	DOT's of the convolution
DFT	$y(n) = x(n) * h(n)$	$Y_F(k) = X_F(k) \cdot H_F(k)$
DHT	$y(n) = x(n) * h(n)$	$Y_{Ha}(k) = X_{Ha}(k)H_{Ha}^e(k) + X_{Ha}(-k)H_{Ha}^o(k)$
DCT-I	$y(n)/c_n = [x(n) * h(n)]/c_n$	$Y_{C-I}(k) = 1/c_k \cdot X'_{C-I}(k) \cdot H'_{C-I}(k) - c_k \cdot X_{S-I}(k) \cdot H_{S-I}(k)$
DCT-II	$y(n) = x(n) * h(n)$	$Y_{C-II}(k) = 1/c_k \cdot X_{C-II}(k) \cdot H'_{C-I}(k) - X_{S-II}(k) \cdot H_{S-I}(k)$
DCT-III	$y(n)/c_n = [x(n) * h(n)]/c_n$	$Y_{C-III}(k) = X'_{C-III}(k) \cdot H'_{C-III}(k) - X'_{S-III}(k) \cdot H'_{S-III}(k)$
DCT-IV	$y(n) = x(n) * h(n)$	$Y_{C-IV}(k) = X_{C-IV}(k) \cdot H'_{C-III}(k) - X_{S-IV}(k) \cdot H'_{S-III}(k)$
DST-I	$y(n) = x(n) * h(n)$	$Y_{S-I}(k) = 1/c_k \cdot \{X_{S-I}(k) \cdot H'_{C-I}(k) + X'_{C-I}(k) \cdot H_{S-I}(k)\}$
DST-II	$y(n) = x(n) * h(n)$	$Y_{S-II}(k) = 1/c_k \cdot X_{S-II}(k) \cdot H'_{C-I}(k) + X_{C-II}(k) \cdot H_{S-I}(k)$
DST-III	$y(n) = x(n) * h(n)$	$Y_{S-III}(k) = X'_{S-III}(k) \cdot H'_{C-III}(k) + X'_{C-III}(k) \cdot H'_{S-III}(k)$
DST-IV	$y(n) = x(n) * h(n)$	$Y_{S-IV}(k) = X_{S-IV}(k) \cdot H'_{C-III}(k) + X_{C-IV}(k) \cdot H'_{S-III}(k)$
WHT	$y(n) = \text{dyadic } x(n) * h(n)$	$Y_h = X_h \cdot H_h$

4 CONCLUSION

As we can see from the equations mentioned in the text above and also from Table 1, only for the discrete Fourier transform holds relation $x(n) * h(n) \xrightarrow{\text{DFT}} X(k) \cdot H(k)$, *ie*, the simple multiplication element by element in the DFTs domain corresponds to the convolution in the time domain. This also holds for discrete Hartley transform, when certain conditions are fulfilled. Discrete Fourier transform requires complex multiplications, which imposes memory and time requirements during the computation. On the other side, discrete Hartley transform requires only real multiplications, but this simple relation for the convolution is not true every time. The evaluation of the WHTs product needs only real multiplications by +1 or 1 (*ie*, additions or subtractions, so it is very simple to evaluate). Enumeration of the convolution in the time domain also corresponds to multiplication in WHT domain, but there we need rearranging of one of the input sequences (rearranging corresponds to multiplication input sequence by matrix A in the WHT domain). The problem of using WHT is rearranging itself because by increasing the length of convolution the number of rearrangements grows.

Acknowledgements

This research was supported by the Grant Agency of the Ministry of Education of the Slovak Republic (VEGA) under No. 1/7625/20 and No. 1/7627/20. Methods and algorithms for digital signal processing, development of protocols, modelling of communication channels and simulation of signal transmission via various communication media.

REFERENCES

- [1] AHMED, N.—RAO, K. R.: Orthogonal Transforms for Digital Signal Processing, Springer-Verlag, Berlin, Heidelberg, New York, 1975.
- [2] BESSLICH, P., W.—TIAN LU: Diskrete Orthogonaltransformationen — Algorithmen und Flussgraphen für Signalverarbeitung, Springer, Heidelberg, 1990.
- [3] BRACEWELL, R. N.: Discrete Hartley transform, JOSA Letters, Vol. 73, No. 12, December 1983.
- [4] BRACEWELL, R. N.: The Fast Hartley Transform, Proceedings of The IEEE, Vol. 72, No. 8, August 1984.
- [5] ELLIOT, D. F.—RAO, K. R.: Fast Transforms, Algorithms, Analyses, Applications, Academic Press, Orlando, London, 1982.
- [6] HENNING, F. H.: Transmission of Information by Orthogonal Functions, Springer Verlag, New York, 1972.
- [7] KOTULIAKOVÁ, J.—ROZINAJ, G. Digital Signal Processing (Číslíkové spracovanie signálov I.): FABER, Bratislava. (in Slovak)
- [8] POULARIKAS, A. D.: The Handbook of Formulas and Tables for Signal Processing, Boca Raton: CRC Press LLC, 1999.
- [9] RAO, K. R.—YIP, P.: Discrete Cosine Transform — Algorithms, Advantages, Applications, Academic Press, Inc., London, 1990.
- [10] ROZINAJ, G.—POLEC, J.—KOTULIAKOVÁ, J.—PODHRADSKÝ, P.—MARČEK, A.—MARCHEVSKÝ, S. et al: Digital Signal Processing II (Číslíkové spracovanie signálov II.), FABER, Bratislava, 1997. (in Slovak)

Received 9 July 2002

Jana Kotuliaková (Doc, Ing, CSc) was born (1939) in Levča, graduated from the Faculty of Electrical Engineering of the Slovak University of Technology in Bratislava in 1962. Since 1964 she has been working at the Department of Telecommunications. She received the CSc (PhD) degree in 1980. Her research interests include digital signal processing, digital filtering.

Michal Zajac (Doc, RNDr, CSc) was born (1951) in Bratislava, graduated in Mathematics from Comenius University in Bratislava in 1975. In 1983 he received the CSc (PhD) degree from the Institute of Mathematics of the Czechoslovak Academy of Sciences in Prague. Since 1991 he has been working at the Department of Mathematics, Faculty of Electrical Engineering and Information Technology of the Slovak University of Technology in Bratislava His research interests include linear algebra and functional analysis, and their applications in engineering.

Anna Ušáková (Ing) was born (1977) in Trnava, graduated from the Faculty of Electrical Engineering and Information Technology of the Slovak University of technology in Bratislava in January 2001. She is a postgraduate student at the Department of Telecommunications FEI STU in Bratislava. Her research interests include orthogonal transforms and digital image processing.