International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

A STUDY ON FAST DISCRETE WALSH TRANSFORMS FOR DIGITAL SYSTEM CONTROL

Joon-Hoon Park,
Dept. of Control and Instrumentation Engineering, Korea National University of Transportation, Korea
jhpark@ut.ac.kr

Abstract

This paper presents a method for solving the problem of analysis of digital control systems using Walsh functions and fast discrete transforms. Walsh functions were completed from the incomplete orthogonal function of Rademacher in 1923. Walsh functions and its fast discrete transforms have useful analog-digital properties and are currently being used in a variety of engineering applications, which include image processing, digital filtering, signal processing and digital control systems. The fast discrete Walsh transforms are more useful than the general discrete Walsh transforms because of its calculation advantages. The algorithm adopted in this paper is that of analysis the system parameters by converting the differential equation into a simple algebraic equation. The proposed method is supported by examples for demonstration fast and convenient capabilities.

Keyword: Walsh functions, Walsh transform, fast discrete Walsh transforms

Introduction

The Rademacher functions are a set of square waves for $t \in [0, 1)$, of unit height and repetition rate equal to 2^m, which can be generated by a BCD counter. The Walsh functions constitute a complete set of two values orthonormal functions $\Phi_k(t), k=0,1,2,\ldots n\text{-}1, n=2^m$ in the interval (0, 1), and they can be defined in the several equivalent ways. The set of Walsh functions is generally classified into three groups. The three types of Walsh orderings are a) Walsh ordering, b) Paley ordering and c) Hadamard ordering.[1] First, Walsh ordering is originally employed by Walsh. We can denote Walsh functions belonging to this set by

$$S_w=\{Wal_w(i, t), i=0, 1, \ldots, N\text{-}1\} \tag{1.1}$$

where $N=2^n$, $n=1, 2, 3\ldots\ldots$
The subscript w means Walsh ordering, and i denote the i-th member of S_w. The Cal and Sal functions corresponding to $Wal_w(i, t)$ are denotes as

$$Cal(s_i,t)=Wal_w(i,t),\ i \text{ even}$$

$$Sal(s_i,t)=Wal_w(i,t),\ i \text{ odd} \tag{1.2}$$

Second, the Paley ordering is dyadic type functions. Walsh functions are elements of the dyadic group and can be ordered using the Gray code. This Paley ordering of Walsh functions is denoted as

$$S_p=\{Wal_p(i, t), i=0, 1, \ldots, N\text{-}1\}$$

$$Wal_p(i,t)=Wal_w(i_q,t) \tag{1.3}$$

where i_q is the Gray code to binary conversion. The subscript p means Paley ordering. Third, Hadamard ordering can be denoted by

$$S_h=\{Wal_h(i, t), i=0, 1, \ldots, N\text{-}1\}$$

$$Wal_h(i,t)=Wal_w(i_b,t) \tag{1.4}$$

where i_b is the bit reversal of i. The subscript h means Hadamard ordering. For the purpose of illustration, Table 1 is the results of evaluation for N=8 and the table shows relationship between the Walsh ordering and Hadamard ordering Walsh functions.

Table 1. Relationship between Walsh, Paley and Hadamard ordering

i	Paley to Walsh Ordering	Hadamard to Walsh Ordering
0	$Wal_p(0,t)=Wal_w(0,t)$	$Wal_h(0,t)=Wal_w(0,t)$
1	$Wal_p(1,t)=Wal_w(1,t)$	$Wal_h(1,t)=Wal_w(7,t)$
2	$Wal_p(2,t)=Wal_w(3,t)$	$Wal_h(2,t)=Wal_w(3,t)$
3	$Wal_p(3,t)=Wal_w(2,t)$	$Wal_h(3,t)=Wal_w(4,t)$
4	$Wal_p(4,t)=Wal_w(7,t)$	$Wal_h(4,t)=Wal_w(1,t)$
5	$Wal_p(5,t)=Wal_w(6,t)$	$Wal_h(5,t)=Wal_w(6,t)$
6	$Wal_p(6,t)=Wal_w(4,t)$	$Wal_h(6,t)=Wal_w(2,t)$
7	$Wal_p(7,t)=Wal_w(5,t)$	$Wal_h(7,t)=Wal_w(5,t)$

Walsh Functions

One of the earliest works in which discrete orthogonal transforms including Walsh functions were applied to the analysis processing of digital control systems and speech

signals. And interest has grown in the possibility of using orthogonal transforms as a means of reducing the bit rate

ISSN:2319-7900

necessary. By sampling the Walsh functions shown in figure 1, we can obtain the *(8x8)* matrix. In general an *(NxN)* matrix would be obtained. We denote such matrices by HAD_m, since they can be obtained by reordering the row of a class of matrices called Hadamard matrices.[2]

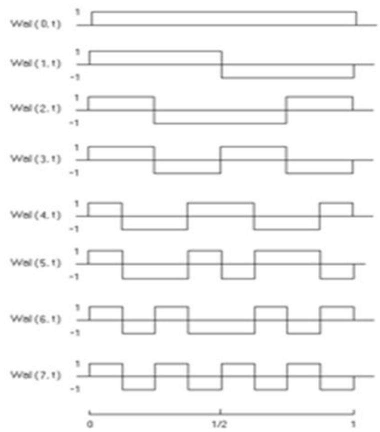

Figure 1. Walsh ordering continuous WF with *N=8*

Higher order matrices, restricted to having powers of two, can be obtained from the recursive relationship. Where ⊗ denotes the direct or Kronecker product and m is a power of two.

$$HAD_m = HAD_{\frac{m}{2}} \otimes HAD_2 \tag{2.1}$$

$$HAD_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{2.2}$$

Discrete Walsh Transforms

The discrete Walsh transforms has found applications in many areas, including signal processing, pattern recognition and digital control systems. Every function *f(t)* which is integrable is capable of being represented by Walsh series defined over the open interval (0, 1) as,

$$f(t) = \sum_{i=0}^{m-1} F_i \, Pal(i,t) \tag{3.1}$$

where coefficients are given by,

$$F_i = \int_0^1 f(t) Pal(i,t) dt \tag{3.2}$$

From (3.1) and (3.2), coefficient of discrete Walsh functions f_n^* can be written as,

$$f_n^* = m \int_{\frac{n}{m}}^{\frac{(n+1)}{m}} f(t)dt \; \cong \; \frac{1}{2} f\left(\frac{n}{m}\right) + f\left(\frac{n+1}{m}\right) \tag{3.3}$$

For numerical use it is convenient to consider a discrete series of N terms set up by sampling the continuous function at N equally spaced points over the open interval (0, 1). The transform and its inverse given by equation (3.1) and (3.2) may be obtained by matrix multiplication using the digital computer.[3] We can also write discrete function, Walsh functions and its coefficients using vector as follows respectively.

$$f^* = [f_0^*, f_1^*, f_2^*, \dots \dots f_{m-1}^*]^T \tag{3.4}$$

$$Pal_i = [pal_{i0}, pal_{i1}, pal_{i2}, \dots \dots pal_{i(m-1)}]^T \tag{3.5}$$

$$F = [F_0, F_1, F_2, \dots \dots F_{m-1}]^T \tag{3.6}$$

Thus we obtain discrete Walsh transforms as,

$$f^* = Pal^T F = PalF \tag{3.7}$$

$$F = \frac{1}{m} Pal f^* \tag{3.8}$$

Where *Pal* is orthogonal function *(mxm)* matrix which consists with elements pal_{ij}.

$$Pal = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{3.9}$$

The basis for efficient implementation of the transformations discussed here is the high degree of redundancy present in the transform matrix. If this redundancy can be removed by using matrix factorization then the efficiency of transformation will be improved.

Fast Discrete Walsh Transforms

A procedure for obtaining a fast discrete Walsh transform algorithm is described in this chapter. This follows the well-known Cooley-Tukey algorithm used for fast Fourier transformation and has similar limitations. We can obtain coefficient and discrete value of natural ordering fast discrete Walsh transform(FDWT) as follows,

$$F = \frac{1}{m} FDWT_m f_n^* \tag{4.1}$$

$$FDWT_m = FDWT_{\frac{m}{2}} \otimes FDWT_2$$

$$= \begin{bmatrix} FDWT_{\frac{m}{2}} & FDWT_{\frac{m}{2}} \\ FDWT_{\frac{m}{2}} & FDWT_{\frac{m}{2}} \end{bmatrix} \tag{4.2}$$

where $FDWT_0=1$, $m=2^k$ $(k=1, 2, 3,\dots)$
The calculation of the fast discrete Walsh transform is carried out in a series of stage, one stage for each power of 2 for m. [4] For example, in the case when $m=8$, we get the following equation using (4.1) and (4.2).

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ \vdots \\ \vdots \\ F_7 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} FDWT_4 & FDWT_4 \\ FDWT_4 & -FDWT_4 \end{bmatrix}\begin{bmatrix} f_0^* \\ f_1^* \\ f_2^* \\ f_3^* \\ \vdots \\ \vdots \\ f_7^* \end{bmatrix} \tag{4.3}$$

And we can divide equation (4.3) into two parts as follows,

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \frac{1}{8} FDWT_4 \begin{bmatrix} f_0^* \\ f_1^* \\ f_2^* \\ f_3^* \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \frac{1}{8} FDWT_4 \begin{bmatrix} f_4^* \\ f_5^* \\ f_6^* \\ f_7^* \end{bmatrix} \tag{4.5}$$

$$f_1^*(k) = f_k^* + f_{(k+4)}^*, \; k=0, 1, 2, 3 \tag{4.6}$$

$$f_1^*(k) = f_{(k+4)}^* - f_k^*, \; k=4, 5, 6,7$$

As the same method, we can get equation (4.7) and (.4.8) from (4.3).

$$\begin{bmatrix} F_0 \\ F_1 \\ \dots \\ F_2 \\ F_3 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} FDWT_2 & FDWT_2 \\ FDWT_2 & -FDWT_2 \end{bmatrix}\begin{bmatrix} f_1^*(0) \\ f_1^*(1) \\ \dots \\ f_1^*(2) \\ f_1^*(3) \end{bmatrix} \tag{4.7}$$

$$\begin{bmatrix} F_4 \\ F_5 \\ \dots \\ F_6 \\ F_7 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} FDWT_2 & FDWT_2 \\ FDWT_2 & -FDWT_2 \end{bmatrix}\begin{bmatrix} f_1^*(4) \\ f_1^*(5) \\ \dots \\ f_1^*(6) \\ f_1^*(7) \end{bmatrix} \tag{4.8}$$

A derivation of a fast discrete Walsh transform algorithm having sequency-ordered coefficients is most conveniently

obtained from the continued product. Computation for the fast discrete transform algorithm can be described conveniently by means of signal flow diagram. Figure 2 is signal flow graph of fast discrete Walsh transforms and figure 3 is signal flow graph of calculation algorithm.

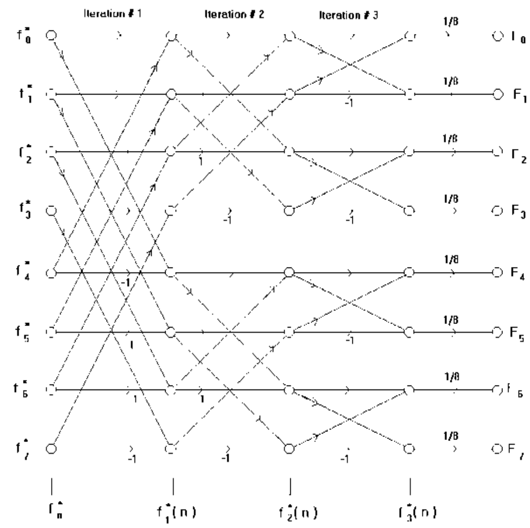

Figure 2. Signal flow graph of fast discrete Walsh transforms

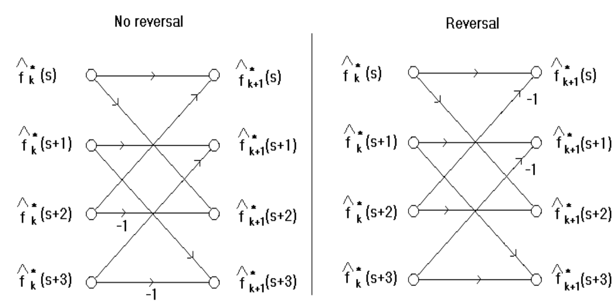

Figure 3. Signal flow graph for reversal and no reversal

Simulation and Conclusions

To illustrate the theoretical results we shall give two numerical examples.

Example 1:

We assume that $f(t)=t$, $[0, 1)$ and let us find the discrete coefficients of Walsh transform.[5]
Then we can define f_n^* as follows,

$$f_0^* = 0.125, f_1^* = 0.375, f_2^* = 0.625, f_3^* = 0.875 \tag{5.1}$$

Table 2. Walsh Transform coefficients for a simple sine waveform with N=32

0.663	0.063	0.000	0.000	-0.263	0.025
-0.052	-0.006	0.000	0.000	-0.126	0.013
-0.013	-0.002	0.000	0.000	0.006	0.000
-0.025	-0.002	0.000	0.000	-0.062	0.006

ISSN:2319-7900

Example 2:

Now consider the function $f(t) = \frac{1}{2}t^2$ and let us define the fast discrete coefficients of Walsh transforms. Then we can define f_n^* as follows,

$$f_n^* = [f_0^*, f_1^*, f_2^*, \dots \dots f_7^*] \quad (5.2)$$

$$f_0^* = 0.0039, f_1^* = 0.0195, f_2^* = 0.0508, f_3^* = 0.0977$$
$$f_4^* = 0.1602, f_5^* = 0.2383, f_6^* = 0.3320, f_7^* = 0.4414$$

$$F = [F_0, F_1, F_2, \dots \dots F_7]$$
$$= [0.1638, -0.0313, -0.0625, 0.0078, -0.1250, 0.0156, 0.0313, 0.0000] \quad (5.3)$$

$$f_1^*(0) = 0.1641 \quad f_2^*(0) = 0.5469$$
$$f_1^*(1) = 0.2578 \quad f_2^*(1) = 0.7969$$
$$f_1^*(2) = 0.3828 \quad f_2^*(2) = -0.2188$$
$$f_1^*(3) = 0.5391 \quad f_2^*(3) = -0.2813$$
$$f_1^*(4) = -0.1563 \quad f_2^*(4) = -0.4375$$
$$f_1^*(5) = -0.2188 \quad f_2^*(5) = -0.5625$$
$$f_1^*(6) = -0.2813 \quad f_2^*(6) = 0.1250$$
$$f_1^*(7) = -0.3438 \quad f_2^*(7) = 0.1250 \quad (5.4)$$

We have seen that the Walsh functions and its fast discrete transforms can be used as an approximating set of functions to find a finite series expansion of systems and analysis digital control systems. By applying suggested discrete and fast discrete transforms for a systems analysis, we show the numerical result that is simple and useful. The procedure is extended to the case of digital control system design.

Acknowledgments

"The research was supported by a grant from the Academic Research Program of Korea National University of Transportation in 2013."

References

[1] M.G.Singh, "Dynamical Hierarchical Control, Automatica", Vol. 18, No.4, pp. 501-503, 1983.

[2] Chyi Hwang and Yen-Ping Shih, "On the Operational matrices of Block Pulse Functions", Int., J. Systems Sci., Vol. 17, No. 10, pp. 1489-1498, 1986

[3] Jian-Min Zhu and Yong-Zai Lu, "Hierarchical Optimal Control for Distributed Parameter Systems via Block Pulse Operator", Int. J. Control, Vol. 48, No. 2, pp. 685-703, 1988.

[4] J. H. Park, "Trajectory Optimization for Large Scale Systems via Block Pulse Functions and Transformations", Int. J. of Control and Automation, Vol.5, No.4, pp. 39-48, 2013.

[5] J. H. Park and R. D. Oh, "A Study on Discrete WALSH Transforms for Digital System Controls", AICT2013, vol. 26, pp. 167-172, 2013.

Biography

Joon-Hoon Park, Ph. D.

Since 1991, he is currently Professor of Dept. of Control and Instrumentations Eng. of KNUT in Korea. He is interested in optimal control theory, system design and its applications