VENTURING INTO THE WALSH DOMAIN WITH A LABVIEW BASED COMMUNICATION SYSTEMS TOOLKIT

Tyler Anderson[1], Murat Tanyel[2]

[1]Geneva College, Beaver Falls, Pennsylvania: Email: tbanders@geneva.edu
[2]Geneva College, Beaver Falls, Pennsylvania: Email: mtanyel@geneva.edu

1. ABSTRACT

In a recent publication, we reported on the utilization of a LabVIEW based Communication Systems Toolkit in the investigation of a novel idea for envelope detection of amplitude modulated (AM) signals (Wright, Tanyel 2005). In this paper, we will describe our efforts to develop the toolkit further and our investigations into the Walsh domain with the help of our latest developments. In communications theory, the Fourier Transform is the traditional frequency domain analysis tool and as such, the current toolkit is based exclusively on this technique. However, other transforms do exist and in the course of study, we became interested in the Walsh Transform. We developed virtual instruments (VIs) to compute the Walsh transform, generated the Walsh basis functions and modified LabVIEW's natural ordered fast Walsh Transform (FWT) routine to provide Walsh ordered Walsh transforms. In this paper, we investigate the Walsh domain and compare it to the Fourier domain using these newly developed VIs. We demonstrate similarities and differences between FFT and FWT. We then propose a new arrangement of the FWT sequency plots that will correspond to the magnitude spectrum plots obtained by the FFT.

2. INTRODUCTION

This paper follows recent papers that describe a simulation toolkit for communication systems (Tanyel, Nguru 2002), its reception by students at two different institutions (Tanyel, Linder 2004) and its utilization in undergraduate student research (Wright, Tanyel 2005). In those papers we stated that in the absence of hardware that would reinforce the theoretical presentation, computer simulations of the systems described in class are the next available tools to bring these concepts to life. Those papers also describe the particular class environment and the process in which the software development tool, namely LabVIEW, was chosen. Although MATLAB is the standard software tool employed in the areas of signals and systems, as evidenced by the proliferation of books devoted to MATLAB based exercises in those subjects, the choice of the software tool is justified in (Viss, Tanyel 2001), (Tanyel, Nguru 2002) and (Tanyel, Adams 2002).

This paper will report on the results of a term project carried out in ELE 402, *Introduction to Communications Engineering* class. In ELE 402, Fourier series expansion is presented in the context of orthogonal series representation of signals and noise. We define orthogonal functions over an interval, discuss how an arbitrary waveform may be expanded in a series of these orthogonal functions and present the various forms of the Fourier series as a particular type of orthogonal series whose basis functions are sinusoids or complex exponential functions. We mention, in passing, that there are other sets of orthogonal functions that may be employed to expand our waveform functions. In the Fall '05 offering of ELE 402, we mentioned Walsh transforms in this context and one student decided to incorporate Walsh Transforms into the toolkit to provide an alternative example to orthogonal series representation of signals. This paper will describe how Walsh transforms were incorporated into the Communication Systems Toolkit and how the toolkit was used to investigate Walsh transforms in the term project. Section 3 will provide a background for Walsh Transforms and section 4 will describe the new routines added to the toolkit. Section 5 will present our explorations in the Walsh domain using these new tools. We will then conclude with a discussion.

3. BACKGROUND ON WALSH TRANSFORMS

Frequency domain analysis of linear channels is the fundamental tool of communications engineering. Frequency domain analysis provides many advantages over time domain analysis, several of which are as follows (Stern, *et al* 2004):

1) convolution becomes multiplication,
2) complicated signals are reduced to a simple summation of similar functions,
3) channel bandwidth—an important aspect in communication systems design—is easily determined,
4) noise is simple to identify, and
5) Parseval's energy theorem allows for the normalized energy spectrum to be the same in both the frequency and time domain.

Traditionally, the transform of choice has been the Fourier transform, with efficient applications in the FFT. The advantages of this transform are that its basis functions are sinusoidal and phase information can be easily determined. Another useful, though less well known technique is the Walsh transform, with efficient application in the FWT. The FWT is inherently faster than the FFT because its computation requires only addition and subtraction (Tadokoro, *et al* 1978). Also, because its basis functions are square waves, the FWT often requires fewer basis functions than the Fourier transform for abruptly changing signals. The Walsh transform is fundamentally analogous to the Fourier transform in many respects and can be thought of as a "square" version of the Fourier transform.

3.1 Background Theory of the Fourier Transform and the FFT

The Fourier transform uses sinusoids as a complete set of orthogonal basis functions to provide spectral representation of time domain signals. For a given time domain signal, $s(t)$, the Fourier transform $S(f)$ is defined as

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-j2pft}dt \qquad (1).$$

For N samples of a signal, $x[k]$, the discrete Fourier transform $X[n]$ is defined as

$$X[n] = \sum_{k=0}^{N-1} x[k]e^{-j\left(\frac{2p}{N}\right)nk} \text{ where } n, k = \{0, 1, \ldots, N\text{-}1\} \;\; (2).$$

The actual frequency is $f = nf_s /N$, where f_s is the sampling frequency of the system. Note that the magnitude spectrum will be symmetric about $N/2$, and the negative spectrum can be inferred from symmetry. The FFT is simply the application of fast algorithms that reduce the number of computations when the number of samples is a power of 2.

3.2 The Walsh Transform

Walsh functions are a complete, orthogonal set of square-wave functions defined over the unit interval. These functions form the basis for the Walsh transform - the analogue to the Fourier transform for abruptly changing signals. Various orderings of the Walsh functions exist, including Walsh (or sequency) ordered Walsh functions, Hadamard (or natural) ordered Walsh functions, and Paley (or dyadic) ordered Walsh functions. These functions are correspondingly denoted by $wal_w(k,t)$, $wal_h(k,t)$, $wal_P(k,t)$, where $k = \{0,1\ldots\}$ represents the order of the function (Elliot, *et al* 1982).

For any ordering of the Walsh functions, Walsh-Hadamard matrices can be generated to transform the discrete time signal, $\mathbf{x}^T = \{x(0),x(1),\ldots x(N\text{-}1)\}$ of size N, into the discrete Walsh domain, $\mathbf{X}^T=\{X(0),X(1),\ldots X(N\text{-}1)\}$. Any row of a Walsh-Hadamard matrix can be generated by sampling the corresponding Walsh function N times.

3.3 Generation of the Natural Ordered Walsh-Hadamard Matrices

The natural ordered N x N Walsh-Hadamard matrix, $[H_w(n)]$, to transform a data set of length $N = 2^n$, is defined by the Kronecker Product (Elliot, *et al* 1982):

$$[H_w(n)] = \bigotimes_{i=1}^{n} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (3).$$

For example, $[H_w(2)]$ is found by,

$$[H_w(2)] = \bigotimes_{i=1}^{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1\bullet\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1\bullet\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ 1\bullet\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & 1\bullet\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad (4)$$

3.4 Generation of the Sequency Ordered Walsh-Hadamard Matrices

The natural ordered Walsh transform is the easiest to compute using fast transform routines and LabVIEW 7.1 has the predefined function, “Walsh Hadamard.vi” that computes the natural ordered FWT for a given input sequence. However, the sequency ordered Walsh transform provides the most direct analogy to the Fourier transform, because it is ordered by increasing sequency. Sequency is the binary analogue of frequency, and is defined as

$$\text{sequency} = \begin{cases} \dfrac{1}{2}(z.c.), & \text{for even } z.c. \\ \dfrac{1}{2}(z.c.+1), & \text{for odd } z.c. \end{cases} \qquad \mathbf{(5)}$$

where *z.c.* is the number of zero crossings in a unit interval (Tadokoro, *et al* 1978).

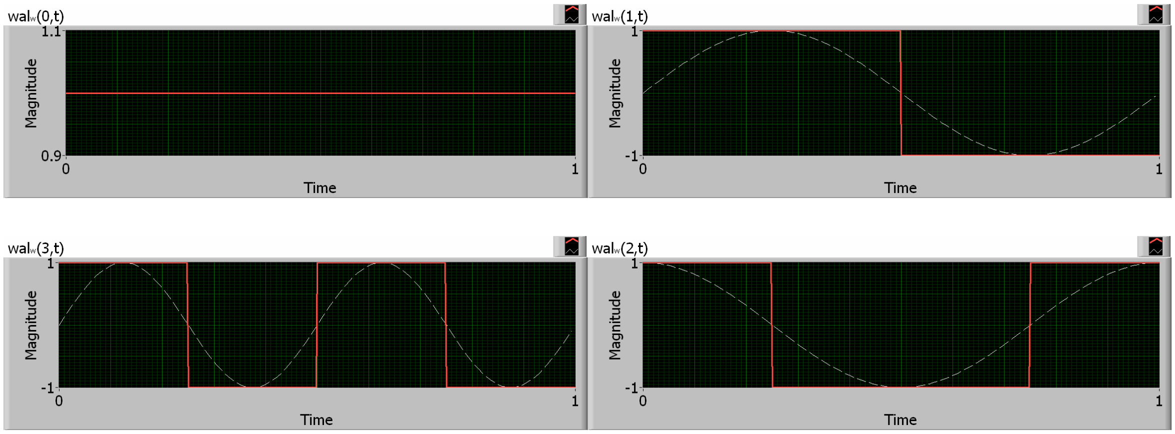

Figure 1: The first four Walsh functions with corresponding Fourier sinusoids superimposed.

For sequency ordered Walsh functions, the terms “sal” (sine-Walsh) and “cal” (cosine-Walsh) are commonly used for the basis functions to emphasize the correlation between the standard Walsh functions and their analogous Fourier component (Elliot *et al* 1982). The first four sequency ordered Walsh functions are plotted in Figure 1 with their corresponding Fourier sinusoids in order to further demonstrate the similarity of this relationship using our own toolkit.

4. Implementation of Tools to Explore Walsh Transforms

The following subroutines (or subVIs) were written in order to explore the Walsh domain with our toolkit:

6) walsh_ordered_walsh_generator.vi,
7) walsh_by_had.vi,
8) gray_code_invert.vi,
9) walsh_direct.vi.

walsh_ordered_walsh_generator.vi generates the basis functions, “sal” (sine-Walsh) and “cal” (cosine-Walsh). The plots in figure 1 were obtained through this VI.

walsh_by_had.vi employs two subVIs to compute the sequency-ordered FWT. The first is LabVIEW’s predefined natural ordered FWT. The second is “gray_code_invert.vi.”

gray_code_invert.vi converts the indices of components in the natural ordered FWT to the indices of the sequency ordered FWT. The algorithm for this conversion is as follows:

Step 1: Convert the natural ordered index into binary representation.

Step 2: Truncate the binary index such that the number of bits is $L = log_2(N)$, where N is the sample size. This is necessary because LabVIEW only performs decimal to binary conversions into bytes, words, and long words.

Step 3: Bit reverse the truncated binary index.

Step 4: Perform inverse gray code to binary conversion on the reversed truncated binary index.

Inverse gray code to binary conversion can be inferred from the forward gray code to binary conversion algorithm given in (Elliot *et al* 1982), and is performed as follows:

For the L-bit binary number representation of the index $(i)_{10} = (i_{L-1}, i_{L-2},...i_2, i_1, i_0)_2$, the inverse gray code to binary conversion is $(k)_{10} = (k_{L-1}, k_{L-2}...k_2, k_1, k_0)_2$, where
$k_{L-1} = i_{L-1},\ k_{L-2} = i_{L-1} \oplus i_{L-2},\ k_{L-3} = i_{L-2} \oplus i_{L-3}...$

walsh_direct.vi computes the sequency-ordered Walsh Transform using the algorithm described in sections 3.3 and 3.4, employing equation (3) directly.

5. Explorations in the Walsh Domain

After the tools to explore the Walsh domain were developed, we performed experiments to investigate this new domain. We will report on the results of three experiments here:

1) Comparison of the number of nonzero components of FWT and FFT for some typical waveforms,
2) Comparison of FWT and FFT for a stream of random bits,
3) Explorations of "phase" in the Walsh domain.

The first two experiments were designed to validate our expectations of FWT and FFTs. The third experiment was designed to investigate the nature of FWT and revealed results that were undisclosed in our literature search.

5.1 Comparison of the Number of Spectral Components of FWT and FFT

As previously noted, one advantage of the FWT is that spectral representations of abruptly changing signals can often be expressed with fewer components than necessary in the FFT. We compared the number of non-zero components of the FWT with that of the FFT for 512 samples of various standard waveforms. These waveforms were generated using the Communication Toolkit's signal generator subVI. The results are summarized in Table 1. We arbitratily decided that whenever a spectral component had a magnitude less than 1E-6, it was considered to be zero and had our VI count the nonzero components.

Table 1: Comparison of the number of nonzero components between the FFT and FWT of various waveforms

Waveform	**FWT Components**	**FFT Components**
Pulse Train	2	257
Impulse Train	512	512
Exp. Ramp	35	171
Square	1	256
Sawtooth	9	171

Triangular	9	14
Sinusoid	20	2

The number of nonzero components of most of the waveforms listed in Table 1 is fewer for the FWT. This is due to the fact that most of these functions have at least one discontinuity in one period. As expected, the Walsh expansions for these functions require fewer components. We also observe that this trend is reversed for the sinusoid: the Walsh expansion requires more components than the Fourier expansion (see Table 1 and Figure 2), again as expected. We should note that the highest sequency component represented is *N*/2, which is analogous to the highest frequency component in FFT being $f_s/2$. It is interesting to see that the FWT and FFT spectral responses to a single impulse require the same number of basis functions. These two spectra are in fact identical, with an equal magnitude of all frequencies and sequencies represented in the spectrum.

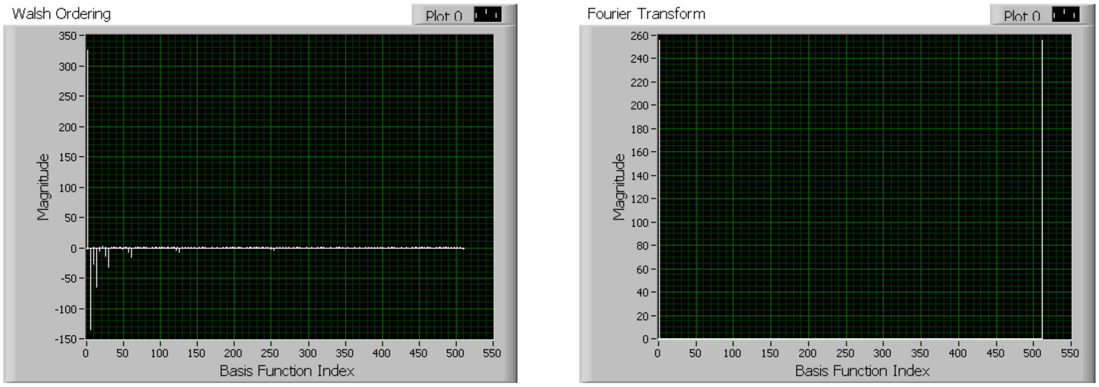

Figure 2: A comparison of the sequency ordered FWT and the FFT for a sinusoidal input.

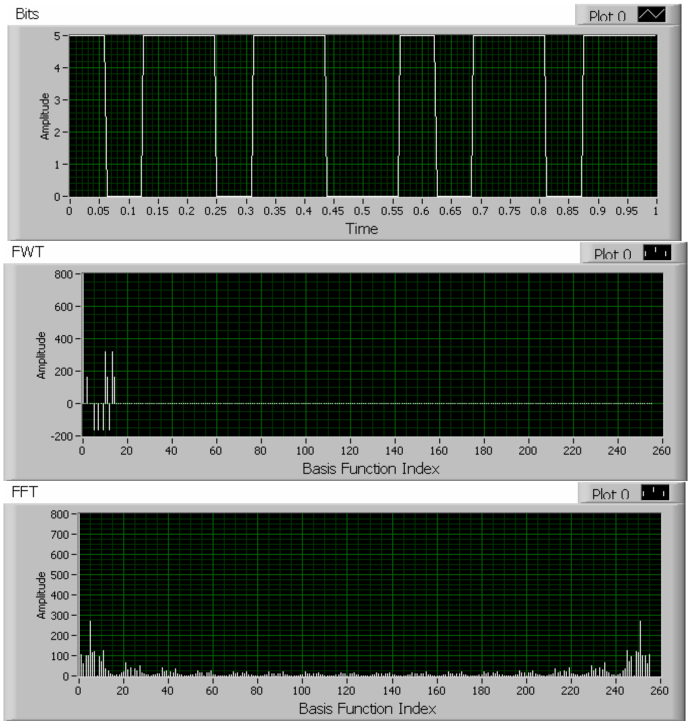

Figure 3: Comparison of FWT and FFT for random eight bit input sequence.

5.2 Comparison of FWT and FFT "Spectra" for a Random Stream of Bits

We expect that random bits are most simply represented by the square wave basis functions of the FWT. We wrote the VI "walsh_random_bits.vi" to demonstrate this effect. This VI allows the user to specify a number of random bits to generate and observe the FWT and FFT for a random bit sequence. The simplicity of the FWT for such a scenario becomes clear, especially for small numbers of bits. One example can be seen in Figure 3. The FWT has fewer components than the FFT.

5.3 Explorations of "Phase" in the Walsh Domain

Our literature survey reiterated the correspondence of "sal and "cal" functions of the Walsh transform to the sine and cosine basis functions in the trigonometric Fourier series expansion. We posed the question: "If phase information is conveyed in the ratio of the sine to cosine coefficients in the Fourier domain, is there a similar relation in the Walsh domain?" We investigated the response of the FWT to phase changes in *fft_fwt_phase_compare.vi.* In this VI, coefficients of "sal" and "cal" functions as well as the real and imaginary parts of the FFT are plotted on separate graphs as various standard waveforms are phase shifted. The behavior exhibited further underscores the strong analogy between the FFT and FWT. As a sine wave experiences a positive phase shift, the spectrum of both the real part of the FFT and the cal functions of the FWT increase, while the imaginary part of the FFT and the sal functions of the FWT decrease. However, this behavior reverses after 90 degrees, at which point the odd symmetry of the reflected sine wave begins to be asserted. Similar phase behavior for other waveforms underscores the correspondence of FWT and FFT. This behavior was a discovery for us.

6. DISCUSSION

In Communication Systems class, we discuss orthogonal functions and orthogonal series representation of signals and noise. We then present the Fourier series expansion of periodic signals as an example to this abstract operation. We have now added the tools to provide another example, namely, Walsh transforms to illustrate this concept. We have developed virtual experiments by which a student may investigate Walsh transforms. We have gone through a preliminary investigation ourselves and have verified the similarities and differences between Walsh transforms and Fourier transforms.

We have discovered that under phase shifts of the original periodic function, the cal and sal components of the Walsh transform behave very similarly to the real and imaginary components of the Fourier transform. In our literature search so far, we have not encountered any reference to such behavior. We need to expand our literature search to find out if this behavior has been noted, studied or analyzed.

Future work with the toolkit will not only involve class demonstrations but also facilitate investigations of the Walsh domain. We suspect that a “magnitude spectrum” and a “phase spectrum” in the Walsh domain may be defined from the cal and sal components. In fact, we propose that the square root of the sum of the squares of the cal and sal components at any sequency be defined as the magnitude at that sequency. We intend to use the toolkit to refine our definitions of magnitude and phase in the Walsh domain.

The Communication Systems Toolkit has expanded from a source of demonstrations to a utility to test new ideas. Initially, it was developed for a course without any laboratory. It has also proven useful as a quick trial tool before committing oneself to a more costly hardware set-up (Wright & Tanyel, 2005) and is developing into an undergraduate research tool.

REFERENCES

Elliot, F., Douglas, and Rao, K. Ramamohan. (1982), Fast Transforms Algorithms, Analyses, Applications. p306, 314, 305, 310, 447. Academic Press, Inc., Orlando, Florida.

Stern, P. E., Harold, and Mahmoud, A., Samy (2004), Communication Systems Analysis and Design. pp. 6-8. Pearson Prentice Hall, Upper Saddle River, NJ 07458.

Tadokoro, Yoshiaki, and Higuchi, Tatsuo (1978), “Discrete Fourier Transform Computation via the Walsh Transform.” IEEE Transactions on Acoustics, Speech, and Signal Processing, **VOL. ASSP-26, NO. 3**.

Tanyel, M., Adams, C., (2002), “On the Aesthetics of Computer Aided Tools for Signal Processing”, *Proceedings of 64th Annual ASEE North Midwest Section Meeting*, Madison, WI.

Tanyel, M., Linder, J (2004), “Are Our Toolkits as Exciting as We Think They Are?”, *2004 ASEE Annual Conference and Exposition Proceedings*, Salt Lake City, UT.

Tanyel, M., Nguru, K (2002), “Preparation of a Virtual Toolkit for Communication Systems”, *2002 ASEE Annual Conference and Exposition Proceedings*, Montréal, QC.

Viss, M. and Tanyel, M. (2001), “From Block Diagrams to Graphical Programs in DSP,” *2001 ASEE Annual Conference & Exposition Proceedings*, Albuquerque, NM.

Wright, A., Tanyel, M. (2005),. Simulation of a Proposed Technique of Envelope Detection with a LabVIEW Based Communication Systems Toolkit. *Proceedings of 2005 ASEE North Central Section Meeting*.