A FAST DISCRETE PERIODIC WAVELET TRANSFORM

by

Neil Getz

Memorandum No. UCB/ERL M92/138

22 December 1992

A FAST DISCRETE PERIODIC WAVELET TRANSFORM

by

Neil Getz

Memorandum No. UCB/ERL M92/138

22 December 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Fast Discrete Periodic Wavelet Transform[*]

Neil Getz

Department of Electrical Engineering and Computer Sciences

and

Electronics Research Laboratory

University of California

Berkeley, California 94720

Abstract

We extend the discrete wavelet transform (DWT) to functions on the discrete circle to create a fast and complete discrete periodic wavelet transform (DPWT) for bounded periodic sequences. In so doing we also solve the problem of non-invertibility that arises in the application of the DWT to finite dimensional sequences as well as provide the proper theoretical setting for previous incomplete solutions to the invertibility problem. We show how and prove that the same filter coefficients used with the DWT to create orthonormal wavelets on compact support in $l^\infty(\mathbb{Z})$ may be incorporated through the DPWT to create an orthonormal basis of discrete periodic wavelets. By exploiting transform symmetry and periodicity we arrive at easily implementable, fast, and recursive synthesis and analysis algorithms. We include Matlab functions for DPWT experimentation.

---

[*]This work was supported in part by NASA under grant NAG 2-243 and by NSF under grant IRI 90-14490.

Contents

I. Introduction

The discrete wavelet transform (DWT) provides a means of decomposing sequences of real numbers in a basis of compactly supported orthonormal sequences each of which is related by being a scaled and shifted version of a single function. As such it provides the possibility of efficiently representing those features of a class of sequences localized in both position and scale. *Compactly* supported wavelet bases, like complex exponential bases, carry the significant advantage that fast, numerically stable algorithms exist for sequence analysis (decomposition into the wavelet basis coordinates) and synthesis (reconstruction from the coordinates in the wavelet basis).

The Discrete Wavelet Transform

The DWT has found application in acoustical analysis, image processing, and data compression. It holds promise for use in edge detection, finite element analysis, and, in particular, for optimization where the high computational overhead of many optimization algorithms strongly motivates a search for more efficient representations of useful classes of functions.

The standard form of the DWT analysis algorithm applied to a real valued sequence $f \in l^2(\mathbb{Z})$ is defined recursively by the *level p decomposition*

$$\begin{cases} f^{p-1} = Lf^p \\ w^{p-1} = Hf^p \end{cases} \tag{1}$$

where $f^p = f$ and L and H are respectively $l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$ low-pass and decimate and high-pass and decimate linear filter operators. The decimation is usually, and throughout this paper, by a factor of 2 (*Decimation by* 2 of a sequence x produces a sequence y defined by $y_n = x_{2n}, n \in \mathbb{Z}$). The *inverse* discrete wavelet transform (IDWT) creates sequences $f^p, p \in \mathbb{Z}$ from the set of wavelet coordinate vectors $\{w^k\}$, $k \in \mathbb{Z}$, with $w^k \in l^2(\mathbb{Z})$, and is defined recursively by the *level p reconstruction*

$$f^p = L^* f^{p-1} + H^* w^{p-1} \tag{2}$$

where L^* and H^* denote the adjoints of the maps L and H with respect to the standard inner product on $l^2(\mathbb{Z})$. We will sometimes refer to both the forward and inverse algorithm pair as the DWT. (We discuss the difference between *level* and the more standard *scale* below.)

Let $L^2(\mathbb{R})$ denote the square integrable functions over the real numbers. The *wavelet basis of* $L^2(\mathbb{R})$ is implicit in the DWT. The basis functions are indexed by level and shift. The basis function corresponding to level q and shift n is the limit as p goes to infinity, when such a limit exists, of the interpolated sequences f^p produced by the inverse DWT with $w_k^q = \delta_{k,n}$, $k \in \mathbb{Z}$, and all other w_k^p zero. The convergence of these limits and the particulars of the interpolation scheme are discussed thoroughly in [Daub1] and we refer the reader to this reference for the details. We will be more concerned with sequence spaces and will describe bases of wavelet sequences below.

In order for the algorithm (1), (2) to correspond to an orthogonal DWT we require that L and H obey the following *perfect reconstruction criteria*:

$$L\,L^* = I \tag{3}$$
$$H\,H^* = I \tag{4}$$
$$L\,H^* = 0 \tag{5}$$
$$L^*L + H^*H = I \tag{6}$$

where I, and 0 refer respectively to the identity operator and zero operator, $l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$.

In practice the DWT (1) is implemented through the sums

$$\begin{cases} f_i^{p-1} = \sum_k f_k^p l_{k-2i} \\ w_i^{p-1} = \sum_k f_k^p h_{k-2i} \end{cases} \tag{7}$$

where l and h are equal length finite impulse response (FIR) filter coefficients and are designed such that (3), (4), (5), and (6) are satisfied. The IDWT (2) is implemented by the sum

$$f_i^p = \sum_k f_k^{p-1} l_{i-2k} + w_k^{p-1} h_{i-2k} \tag{8}$$

By choosing l to have support $Q_N \equiv \{0,1,\ldots,2N-1\}$ and to satisfy (3) one may easily create an h which allows satisfaction of (4) and (5) with

$$h_k = (-1)^k l_{2q-k-1} \tag{9}$$

for some q in $\mathbb{Z}$. In this paper we will always take $q = N$ in (9) so that we will have what will turn out to be the computational convenience of l and h having the same support Q_N. For convenience later we tag l and h by their support size and will refer to them as l^N and h^N. With each l^N, h^N pair we will associate the finite impulse response FIR filters L^N and H^N having l_{-k}^N and h_{-k}^N, respectively, as their impulse responses. Since L and H are FIR we may relax the requirements on f^p and w^p to being bounded sequences in $l^\infty(\mathbb{Z})$ rather than square summable sequences $l^2(\mathbb{Z})$. However, we will continue to consider l^N and h^N to be drawn from $l^2(\mathbb{Z})$.

In terms of l^N and h^N, and using the form (7) of the DWT, criteria (3), (4), and (5) become

$$\sum_n l_{n-2k}^N \, l_{n-2l}^N = \delta_{k,l} \tag{10}$$

$$\sum_n h_{n-2k}^N \, h_{n-2l}^N = \delta_{k,l} \tag{11}$$

$$\sum_n l_{n-2k}^N \, h_{n-2l}^N = 0 \tag{12}$$

where $\delta_{k,l}$ denotes the Kronecker delta. Note that (11) and (12) follow from (10) with relation (9). Criterion (10) says that l^N is orthogonal to shifted versions of itself when that shift is by an even integer. Criterion (11) says the same for h^N. Criterion (12) says that l^N and h^N are orthogonal as are all of their even shifts.

There are additional conditions on l^N which assure that (6) is true and that the corresponding *continuous* wavelet bases of $L^2(\mathbb{R})$, where the correspondence is through interpolation, have some degree of regularity. Maximum regularity for a particular choice of N is guaranteed by choosing the FIR low-pass filter associated with l^N to have N zeros at -1. This regularity then increases as N increases. For an in depth study of these conditions and a chart of l^N's for $N \leq 10$ we refer the reader to [Daub1]. Here we will always assume that l^N is such that (10), (11), (12), and (6) are satisfied.

If l^N corresponds to the impulse response of an FIR half-band low-pass filter, then the reversed impulse response h^N represents what we will refer to as its *conjugate* FIR half-band high-pass filter, meaning that $I - L^*L = H^*H$. The halfbandedness of the filters allows, according to the Nyquist sampling theorem, a reduction of the sampling rate by 2. This filter-and-decimate process is then repeated on the decimated output of the low-pass filter.

Shortcomings of the DWT

The discrete wavelet transform (7) is not without its problems. Small shifts in a sequence may yield large changes in the wavelet transform of that sequence as discussed in [SFAH]. The shift problem will not be covered here except to say that useful sequence domains exist where shifted versions of the same sequence are not considered to be *close,* so that some degree of coordinate instability with respect to shift is not a problem. Examples abound in the analysis of spectra. We also note that the shift problem decreases as N increases.

Another problem with the DWT (7), one that we will address here, may be seen as being due to windowing or filter truncation as the operators L and H are restricted to finite dimensional subspaces of $l^\infty(\mathbb{Z})$. The nature of this problem will be illustrated and clarified in section III. We will show that windowing results in non-orthogonal wavelets and lack of invertibility of the DWT. The problem arises from the fact that the shift operator and its adjoint are inverses on $l^\infty(\mathbb{Z})$ but not on finite dimensional subspaces of $l^\infty(\mathbb{Z})$. It is amplified by the recursive nature of the DWT. We will show that these errors may be entirely eliminated by casting finite length sequences as single periods of periodic sequences where the natural shift operator is a rotation and is thus unitary on finite dimensional subspaces of $l^\infty(\mathcal{Z})$.

Mathematical Context of Discrete Periodic Wavelets

We will take a finite dimensional point of view and will consider subspaces of $l^\infty(\mathcal{Z})$ sequences f^p and w^p on support $\mathcal{Z}_p \equiv \{0, 1, \ldots, 2^p - 1\}$ and their periodic counterparts $\tilde{f}^p$ and $\tilde{w}^p$ in copies of $\mathcal{P}_p$, the sequences over the discrete circle of period 2^p. Let $\mathcal{S}$ be a subset of $\mathcal{Z}$. By support $\mathcal{S}$ we mean that if sequence s is in $l^q(\mathcal{S})$, $q \in \{1, 2, \ldots, \infty\}$, then $s_k = 0$ for all k in the complement of $\mathcal{S}$ with respect to $\mathcal{Z}$. We define the relation of $\tilde{f}^p$ to f^p and $\tilde{w}^p$ to w^p by $\tilde{f}^p = f^p_{i \bmod 2^p}$, $\tilde{w}^p = w^p_{i \bmod 2^p}$, $i \in \mathcal{Z}$. From now on f^p and w^p will refer to real valued sequences in $l^\infty(\mathcal{Z}_p) \subset l^\infty(\mathcal{Z})$. More specifically, we consider $\tilde{f}^p$ and $\tilde{w}^p$ to be vectors in copies of $\mathcal{P}_p$, namely $\mathcal{F}_p$ and $\mathcal{W}_p$ respectively. We will often suppress the specificity of $\mathcal{F}_p$ and $\mathcal{W}_p$ by referring to both as $\mathcal{P}_p$. Sometimes we will refer to the vector representations of f^p and w^p as $\mathbf{f}^p = \begin{bmatrix} f_0^p & f_1^p & \cdots & f_{2^p-1}^p \end{bmatrix}$ and $\mathbf{w}^p = \begin{bmatrix} w_0^p & w_1^p & \cdots & w_{2^p-1}^p \end{bmatrix}$, both 2^p-tuples in $\mathcal{R}^{2^p}$. Also we may refer to the matrix representations of our linear operators in bold face, e.g. $\mathbf{L}_p$ and $\mathbf{H}_p$. Often, when context will prevent a sacrifice of clarity, we will not distinguish abstract vectors and operators from their coordinate representations. We denote the restrictions of L and H to domains $l^\infty(\mathcal{Z}_p)$ by L_p and H_p which have codomains $l^\infty(\mathcal{Z}_{p-1})$. We will also continue to refer to L_p and H_p when we do not wish to refer to an explicit N, and L and H when we wish to generalize beyond a particular N or p. Later, in defining the DPWT, we will use tildes to distinguish between the DPWT filters and their DWT counterparts.

Given the notation and distinctions of the previous paragraph we rephrase the wavelet decomposition and reconstruction algorithms as

$$\begin{cases} f^{p-1} = L_p f^p \\ w^{p-1} = H_p f^p \end{cases} \tag{13}$$

and

$$f^p = L_p^* f^{p-1} + H_p^* w^{p-1} \tag{14}$$

where we now index the filters by level p. Criteria (3), (5), (4), and (6) become

$$L_p L_p^* = I_{p-1} \tag{15}$$

$$H_pH_p^* = I_{p-1} \tag{16}$$
$$L_pH_p^* = 0_{p-1} \tag{17}$$
$$L_p^*L_p + H_p^*H_p = I_p \tag{18}$$

and an orthogonal wavelet decomposition must satisfy (15), (16), (17), and (18) at all decomposition and reconstruction levels p. Thus L_p and H_p are surjective (onto) and L_p^* and H_p^* are injective (one-to-one). Note that (18) along with (13) and (14) yield

$$\begin{aligned} f^p &= L_p^*f^{p-1} + H_p^*w^{p-1} \\ &= L_p^*\left(L_pf^p\right) + H_p^*\left(H_pf^p\right) \\ &= \left(L_p^*L_p + H_p^*H_p\right)f^p \\ &= f^p \end{aligned}$$

As a consequence of our finite dimensional point of view the recursion (13) bottoms out at the level 1 decomposition

$$\begin{cases} f^0 = L_1f^1 \\ w^0 = H_1f^1 \end{cases}$$

This is in contrast to the $l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$ DWT which does not bottom out since each decomposition product is of infinite length. Of course we may choose to stop the decomposition at any non-negative level of interest. The IDWT (14), on the other hand, may proceed indefinitely. We will sometimes refer to f^0 as the *residue* or as w^{-1}, whereas w^0 is the level 0 wavelet coordinate.

Wavelet Bases

The basis of orthonormal wavelets for $l^\infty(\mathbb{Z}_p)$ corresponding to the filter sequence $[L_k]$, $k \in \{1, 2, \ldots, p\}$, which, via (9), uniquely determines a conjugate filter sequence $[H_k]$, $k \in \{1, 2, \ldots, p\}$, may be indexed by level and shift. Level ranges from 0 to p-1 and shift, at each level k, ranges from 0 to $2^k - 1$. Let 0_m denote the zero vector in $l^\infty(\mathbb{Z}_m)$. To construct the level k, shift q wavelet, $[\psi_n^{k,q,p}]$, $n \in \mathbb{Z}$, at *base level p* one sets $w_n^k = \delta_{n,q}$, $n \in \mathbb{Z}$, with $f^k = 0_k$, and $w^m = 0_m$, for $m \in \{k+1, \ldots, p-1\}$ and performs the inverse DWT to get

$$\psi^{k,q,p} = L_p^* L_{p-1}^* \cdots L_{k+2}^* H_{k+1}^* w^k$$

To complete the basis for $l^\infty(\mathbb{Z}_p)$ we also include what we will call the level -1 wavelet or residue wavelet at base level p,

$$\psi^{-1,p} = L_p^* L_{p-1}^* \cdots L_1^* f^0$$

Basis element $\psi^{k,q,p}$ is then a real valued discrete function in $l^\infty(\mathbb{Z}_p)$.

Scale versus Level

As we have already begun to do we will index the stages of forward and inverse wavelet transforms by p and refer to p as the *level:* e.g., The products of the *level p* decomposition are level $p-1$ sequences f^{p-1} and w^{p-1} in the $p-1$ dimensional spaces $\mathcal{F}_{p-1}$ and $\mathcal{W}_{p-1}$. The use of *level* is in contrast to the usual practice in the wavelet literature of referring to wavelet basis functions and the coordinates of analyzed functions as being at particular *scales*. We may convert between *scale* and *level* as follows: If the original subject of analysis is f^p, then f^p is the scale 0 or *base level p* sequence and w^q corresponds to the scale $p-q$ wavelet coordinates. We see that scale increases as level decreases, and, in the domain of finite length sequences, scale bottoms out at p. Scale may be negative as a result of applying the inverse transform past scale 0, just as level may become greater than the base level. Using level instead of scale provides a more convenient handle on the dimensions of operator domains and codomains. In the $l^2(\mathbb{Z}) \rightarrow l^2(\mathbb{Z})$ DWT this does not matter. For the DPWT it does.

Tree Structure of the Wavelet Transform

The analysis and synthesis algorithms for our finite length sequences have the tree-like structure illustrated, after Strang [Strang], in figures 1.a and 1.b,

$$\begin{tikzcd}
f^p \arrow[r, "L_p"] \arrow[dr, "H_p"'] & f^{p-1} \arrow[r, "L_{p-1}"] \arrow[dr, "H_{p-1}"'] & f^{p-2} & \cdots & f^2 \arrow[r, "L_2"] \arrow[dr, "H_2"'] & f^1 \arrow[r, "L_1"] \arrow[dr, "H_1"'] & f^0 \\
 & w^{p-1} & w^{p-2} & & & w^1 & w^0
\end{tikzcd}$$

Fig. 1.a. Tree diagram for discrete wavelet transform decomposition.

$$\begin{tikzcd}
f^p & f^{p-1} \arrow[l, "L_p^*"'] & f^{p-2} \arrow[l, "L_{p-1}^*"'] & \cdots & f^2 & f^1 \arrow[l, "L_2^*"'] & f^0 \arrow[l, "L_1^*"'] \\
 & w^{p-1} \arrow[ul, "H_p^*"] & w^{p-2} \arrow[ul, "H_{p-1}^*"] & & & w^1 \arrow[ul, "H_2^*"] & w^0 \arrow[ul, "H_1^*"]
\end{tikzcd}$$

Fig. 1.b. Tree diagram for discrete wavelet transform reconstruction.

where $\mathbf{w}^q \equiv \left[w_0^q, w_2^q, \ldots, w_{2^q-1}^q\right] \in \mathfrak{R}^{2^q}$ and $\mathbf{f}^q \equiv \left[f_0^q, f_2^q, \ldots, f_{2^q-1}^q\right] \in \mathfrak{R}^{2^q}$ with q in the non-negative integers $\mathbf{Z}_+$.

We will call

$$\begin{aligned}
\hat{\mathbf{f}}^p &\equiv \left[f^0, w^0, w^1, \ldots, w^{p-2}, w^{p-1}\right] \\
&= \left[f_0^0, w_0^0, w_0^1, w_1^1, \ldots, w_0^{p-2}, w_1^{p-2}, \ldots, w_{2^{p-2}-1}^{p-2}, w_0^{p-1}, w_2^{p-1}, \ldots, w_{2^{p-1}-1}^{p-1}\right]
\end{aligned}$$

the *complete* wavelet transform of f^p. Note that $\hat{\mathbf{f}}^p$ has $1 + 2^0 + 2^1 + \cdots + 2^{p-1} = 2^p$ coordinates, precisely as many wavelet coordinates as there are coordinates of f^p. In those instances where we do not require the complete wavelet transform of f^p will refer to

$\hat{\mathbf{f}}^{q,p} = \left[f^q, w^q, w^{q+1}, \ldots, w^{p-2}, w^{p-1}\right]$ as the *partial* wavelet transform of f^p *down to level q*. Note that $\hat{\mathbf{f}}^{q,p}$ also has 2^p coordinates.

Overview

After section II summarizing the symbols to be used in our exposition, we demonstrate the non-invertibility problem in section III with a detailed example. In section IV we derive the DPWT from the DWT. Section V explores, via an example, the representational symmetries of our algorithm that lead to the fast implementations of section IV. We follow with section VI where we prove that the same l^N and h^N used in the creation of filters for the $l^2(\mathbb{Z})$ DWT may be used to derive appropriate filters for the DPWT and, consequently, to create an orthonormal wavelet basis for $\mathcal{P}_p$ or $l^\infty\left(\mathbb{Z}_p\right)$ for any $p \in \mathbb{Z}_+$. In the discussion of section VI we point out some features and caveats relating to use of the DPWT. Section VI is followed by conclusions, a list of references, and a set of Matlab [Matlab] functions that will allow the interested reader to experiment with the DPWT and IDPWT.

Occasionally we will be intentionally verbose, presenting expressions in a number of different but equivalent forms, including matrix forms, that could be stated more tersely with sums. In doing so it is our intention to afford the reader a window into the structure of the DPWT, its operator components, and the problems associated with the DWT.

Previous Work

Orthogonal wavelets have been developed by Meyer [Meyer], Mallat [Mallat], and Daubechies [Daub1] among others. Rioul and Duhamel [RD] have studied the efficiency of implementations of the DWT though they focus on non-orthogonal wavelets and do not discuss the issues we address here. Continuous wavelets on the circle have been studied by Holschneider [Hols]. Kovacevic and Vetterli [KV] generalize the DWT to a nonseparable multidimensional form and Vetterli and Herley connect wavelet transforms with perfect reconstruction filter banks in [VH]. We draw mainly on the theoretical foundations of wavelet theory as presented in [Daub1] and [Mallat].

In [Daub2] Daubechies discusses, in the context of the characterization of functional spaces, the topic of wavelets for $L^1([0,1])$, the space of absolutely integrable functions over the real interval $[0,1]$ and introduces *periodized wavelets* each of which is constructed as a sum of copies of a periodically shifted continuous wavelet having suitable decay. The present paper may be regarded as an extension of Daubechies construction to an easily implementable and fast algorithmic form. The finite dimensional arguments, proofs, and derivations presented here will, we believe, be considerably more accessible to a wide range of readers.

Attempts at a remedy for the non-invertibility of the DWT have appeared before in [SE] as well as [PTVF]. We will briefly describe the methods of these authors in section III where we demonstrate the nature of the non-invertibility.

Contributions of this Paper

Our main result is the extension of the DWT to discrete periodic sequences to create a discrete periodic wavelet transform (DPWT), an inverse transform (IDPWT), and an associated basis of discrete periodic wavelets. Figure 2 shows an example of the periodic discrete wavelets produced for $N = 4$ and base level 7. More examples will be shown in the discussion of section VI.

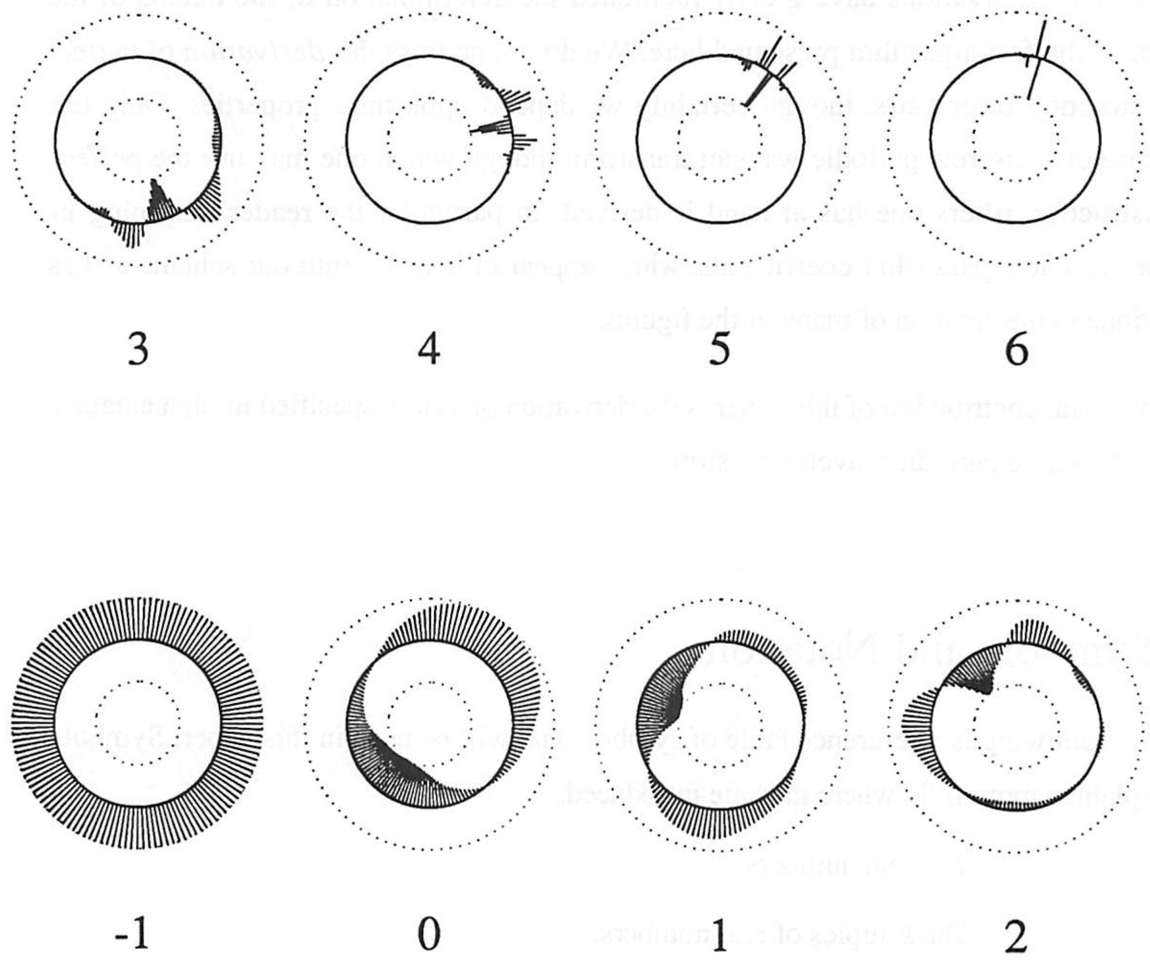

Figure 2. Periodic wavelets on the discrete circle. These were obtained by applying the IDPWT to unit impulse wavelet coordinates at the levels indicated below each graph. These are length 2^7 and arise from Daubechies filters of length 8 ($N = 4$). They have been linearly scaled so that their maxima and minima fall on -1 or 1. The inner and outer dotted circles represent -1 and 1 respectively, and the solid lined middle circle represents 0. Their lack of symmetry has been proven necessary in [Daub1].

Unlike many others we avoid frequency domain derivations in favor of remaining in the shift (time or space) and scale domain. In addition to providing the reader with an alternative point of view to that normally appearing in the signal processing and wavelet literature, the shift domain derivations have greatly facilitated the determination of the details of the design of the fast algorithm presented here. We do not address the *derivation* of perfect reconstruction filter pairs, though certainly we depend upon their properties. Only the structure of a discrete periodic wavelet transform though which one may use the perfect reconstruction filters one has at hand is derived. In particular the reader may plug in Daubechies' low pass filter coefficients, which appear in [Daub1] into our scheme as has been done in the creation of many of the figures.

The main contribution of this paper is the derivation of a fully specified implementation of a fast discrete periodic wavelet transform.

II. Symbols and Notation

The following is a reference table of symbols that will be used in this paper. Symbols are explained more fully where they are introduced.

$\mathcal{R}$	The real numbers.
$\mathcal{R}^k$	The k-tuples of real numbers.
$\mathcal{Z}$	The integers.
$\mathcal{Z}_+$	The non-negative integers.
$\mathcal{Z}_p$	$\{0, 1, 2, ..., 2^p - 1\}$.
$\mathbf{Q}_N$	$\{0, 1, ..., 2N - 1\}$.
$l^q(\mathcal{S})$	Real valued sequences x satisfying $\left(\sum_{k\in\mathcal{S}} \lvert x_k\rvert^q\right)^{1/q} < \infty$, e.g. $l^2(\mathbf{Q}_N)$.
$l^\infty(\mathcal{S})$	Real valued bounded sequences with support set $\mathcal{S}$, e.g. $l^\infty(\mathcal{Z}_p)$.
$\mathcal{F}_p, \mathcal{W}_p, \mathcal{P}_p$	Subspaces of $l^\infty(\mathcal{Z})$ consisting of periodic sequences with period 2^p. Vector space $\mathcal{F}_p$ holds sequences $\tilde{f}^p$, $\mathcal{W}_p$ holds periodic wavelet coefficients $\tilde{w}^p$, and $\mathcal{P}_p$ is used to represent either $\mathcal{F}_p$ or $\mathcal{W}_p$.
$\mathcal{S}_1 \to \mathcal{S}_2$	Indicates the domain, $\mathcal{S}_1$, and the codomain, $\mathcal{S}_2$, of a map.
$x \mapsto y$	Indicates action of a mapping on an object x of the map's domain, with

	y as the image of x through the mapping and in the map's codomain.
$I_p,\ O_p$	The identity and zero operators, respectively, $l^\infty(\mathbb{Z}_p) \to l^\infty(\mathbb{Z}_p)$.
0_p	The zero vector in $l^\infty(\mathbb{Z}_p)$. (The italicized O_p distinguishes the operator O_p from the vector 0_p.)
$\langle \cdot,\cdot \rangle_p$	The real valued inner product on $\mathcal{P}_p$, i.e., for $x, y \in \mathcal{P}_p$, $\langle x, y\rangle_p \equiv \sum_{i=0}^{2^p-1} x_i\, y_i$.
$\langle \cdot,\cdot \rangle$	The standard real inner product on $l^2(\mathbb{Z})$, i.e., for $x, y \in l^2(\mathbb{Z})$, $\langle x, y\rangle \equiv \sum_{i=-\infty}^{\infty} x_i\, y_i$.
$w_n^p, \tilde{w}_n^p$	The nth wavelet coefficient at scale p, non-periodic and periodic respectively.
$f_n^p, \tilde{f}_n^p$	The nth sequence element at scale p, non-periodic and periodic respectively.
$\hat{\mathbf{f}}^p$	The complete wavelet transform of f^p.
$\hat{\mathbf{f}}^{q,p}$	The partial wavelet transform of f^p down to scale q.
L_p^N, H_p^N	Low-pass and decimate and high-pass and decimate, respectively, filter operators $l^\infty(\mathbb{Z}_p) \to l^\infty(\mathbb{Z}_{p-1})$ associated with sequences $l^N \in l^2(\mathbb{Z})$.
$\tilde{L}_p^N, \tilde{H}_p^N$	Low-pass and decimate and high-pass and decimate, respectively, filter operators $\mathcal{P}_p \to \mathcal{P}_{p-1}$ associated with sequences $l^N \in l^2(\mathbb{Z})$.
S^k	Right shift by k operator $l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$, $y_n \mapsto y_{n-k}$.
S_N^k	Right shift by k operator $l^2(\mathbf{Q}_N) \to l^2(\mathbf{Q}_N)$, $x_n \mapsto 0$ if $n < k$, and $x_n \mapsto x_{n-k}$ if $n \geq k$, $n-k \in \mathbf{Q}_N$.
R_p^k	Right shift operator $\mathcal{P}_p \to \mathcal{P}_p$.
$@_N^p$	Wrap $@^p : l^2(\mathbb{Z}) \to \mathcal{P}_p$, $y_n \mapsto \sum_{k=-\infty}^{\infty} y_{i+2^p k}$, $i \in \mathbb{Z}_p$
π_N	The natural projection operator $l^2(\mathbb{Z}) \to l^2(\mathbf{Q}_N)$
N	One-half the number of non-zero elements of $l^N, h^N \in l^2(\mathbf{Q}_N) \subset l^2(\mathbb{Z})$.
l^N, h^N	Filter sequences in $l^2(\mathbf{Q}_N)$.
l_n^N, h_n^N	nth coefficient of l^N, h^N.
A	A linear operator.

A^*	Adjoint of A.
$(A)^k$	$A \circ A \circ A \circ \cdots \circ A$, k times, where $\circ$ denotes map composition.
$\mathbf{A}$	Matrix realization of linear operator A.
$\mathbf{A}^*$	Transpose of $\mathbf{A}$ (we will be using only real valued matrices and standard inner products).
$(\mathbf{A})^k$	Matrix $\mathbf{A}$ to the k^{th} power.
$[\mathbf{A}]_{i,j}$	The row i column j element of matrix $\mathbf{A}$.
$[x]_i$	The i^{th} element of vector or sequence x.
$\{G\}_i$	The i^{th} element of an indexed set G.
$(g)^k$	The k^{th} power of scalar g. We use parentheses for powers to avoid conflict with other superscripts. In practice we will only need squares, i.e. $k = 2$.
$\tilde{s}$	The periodic counterpart of a finite length sequence s defined by $\tilde{s}_k \equiv s^p_{k \bmod 2^p}$, $k \in \mathcal{Z}$.
$\delta_{i,j}$	The Kronecker delta, equal to 1 if $i = j$ and 0 otherwise.
$a \bmod b$	The positive remainder after division of a by b, e.g. 6 mod 4 = 2, -1 mod 4 = 3.
range(A)	The range space of the linear operator $A : \mathcal{U} \to \mathcal{V}$, i.e. the set $\{Ax \mid x \in \mathcal{U}\}$.
null(A)	The null space of the linear operator $A : \mathcal{U} \to \mathcal{V}$, i.e. the set $\{x \in \mathcal{U} \mid Ax = 0\}$ where 0 is the zero element in the vector space $\mathcal{U}$.
$\lceil q \rceil$	The *ceiling* of real number q, meaning the least integer greater than or equal to q.
$\forall$	*for all*
$\ni$	*such that*
$\exists$	*there exists*

III. DWT Non-invertibility on Finite Length Sequences

In this section we illustrate the non-invertibility problem of the DWT (7) with an example. Though this problem has been patched to some extent by others using techniques described later in this section, we believe, and hold the existing patch techniques as evidence, that the nature of the problem has not been well understood. This section is offered as a remedy to any such misunderstanding.

Symbolic Demonstration

We will use a 4 tap filter for the low-pass-and-decimate filter associated with the transform (7) and its conjugate high-pass-and-decimate filter. We will show that each of the four criteria (15), (16), (17), and (18) break down to some extent. Consider a real valued length 8 vector $f = f^3 \in l^\infty(\mathbf{Z}_3)$. The first step in the decomposition algorithm, when restricted to $l^\infty(\mathbf{Z}_3)$ is specified as

$$w_k^2 = \sum_{i=0}^{7} f_i^3 h_{i-2k}^2 \qquad f_k^2 = \sum_{i=0}^{7} f_i^3 l_{i-2k}^2$$

or in operator form $w^2 = H_3^2 f^3$ and $f^2 = L_3^2 f^3$. For a particular choice of phase relationship with $q = N$ in (9) the values of h^2 and l^2 are related by $h_0^2 = l_3^2$, $h_1^2 = -l_2^2$, $h_2^2 = l_1^2$, and $h_3^2 = -l_0^2$ where both l^2 and h^2 have support $\{0, 1, 2, 3\}$. This gives matrix representations of L_3^2 the form

$$\mathbf{L}_3^2 = \begin{bmatrix} l_0^2 & l_1^2 & l_2^2 & l_3^2 & & & & \\ & & l_0^2 & l_1^2 & l_2^2 & l_3^2 & & \\ & & & & l_0^2 & l_1^2 & l_2^2 & l_3^2 \\ & & & & & & l_0^2 & l_1^2 \end{bmatrix}$$

and $\mathbf{H}_3^2$ has the same form as $\mathbf{L}_3^2$ but with l_k^2 replaced by h_k^2. Blanks in $\mathbf{L}_3^2$ should be understood to represent zeros. Note that each successive row of $\mathbf{L}_3^2$ and $\mathbf{H}_3^2$ are related by right shifts by 2 and that the last row displays truncation of l^2 and h^2 due to the restriction of the domains of L_3^2 and H_3^2 to the subspace $l^\infty(\mathbf{Z}_3)$. For surjective operators L_3^2 and H_3^2 we require, according to (18), that

$$L_3^{2^*}L_3^2 + H_3^{2^*}H_3^2 = I_3$$

Note, however, that the upper left 2 by 2 block of $\mathbf{L}_3^{2^*}\mathbf{L}_3^2 + \mathbf{H}_3^{2^*}\mathbf{H}_3^2$ is actually

$$\begin{bmatrix} (l_0^2)^2 + (l_3^2)^2 & l_0^2 l_1^2 - l_2^2 l_3^2 \\ l_0^2 l_1^2 - l_2^2 l_3^2 & (l_1^2)^2 + (l_2^2)^2 \end{bmatrix} \neq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We could achieve equality with $l_0 = l_1 = l_2 = l_3 = 0$ but then we would not have $LL^* = I$. *At every step p of the analysis algorithm we will have the same upper left 2 by 2 block of* $\mathbf{L}_p^{2^*}\mathbf{L}_p^2 + \mathbf{H}_p^{2^*}\mathbf{H}_p^2$. We may chose to modify the phase relationship of l^2 and h^2 but this will only move the distortion elsewhere in $\mathbf{L}_p^{2^*}\mathbf{L}_p^2 + \mathbf{H}_p^{2^*}\mathbf{H}_p^2$. Will we have the same problem for all *N?* The answer is that for every N greater than 1 when the level p is such that $2N-2 \leq 2^p$ the size of the distortion block will be $2N-2$ by $2N-2$. In the course of the wavelet decomposition when p becomes small enough to make $2^p \leq 2N-2$, the size of the distortion block will be the entire matrix $\mathbf{L}_p^{N^*}\mathbf{L}_p^N + \mathbf{H}_p^{N^*}\mathbf{H}_p^N$ and the analysis algorithm breaks down completely. The distortion may be thought of as being due to the truncation of the shifted versions of l^N and h^N in the last $2N-2$ rows of $\mathbf{L}_p^N$ and $\mathbf{H}_p^N$, an inescapable result, for $N > 1$, of the restriction of the domains of L_p^N and H_p^N, as defined by the DWT (7), to the finite dimensional subspace of $l^\infty(\mathbf{Z}_p)$ sequences. We note that when $N = 1$ no distortion occurs at any level. This is the Haar wavelet case, and this is the *only* case where the decomposition algorithm (13) may be completed to the $p = 0$ stage with complete invertibility through the synthesis algorithm (14).

The other three criteria (15), (16), and (17), are also violated. In particular the 4,4 entries of the appropriate matrices are

$$\left[\mathbf{L}_3^2\mathbf{L}_3^{2^*}\right]_{4,4} = (l_0^2)^2 + (l_1^2)^2 \neq 1$$

$$\left[\mathbf{H}_3^2\mathbf{H}_3^{2^*}\right]_{4,4} = (l_2^2)^2 + (l_3^2)^2 \neq 1$$

$$\left[\mathbf{L}_3^2\mathbf{H}_3^{2*}\right]_{4,4} = l_1^2 l_2^2 + l_0^2 l_3^2 \neq 0$$

where inequality follows from the fact that for equality we would need l^2 to be identically 0.

As a preventative measure we may choose to stop the wavelet decomposition at a level p_s such that 2^{p_s} is much greater than $2N-2$ (so that only a small part of L^*L+H^*H is distorted, but this is obviously too severe a restriction in cases where we wish to analyze or filter the sequence at scales p_s and below. We may apply zero-padding but this results in reduced computational efficiency of the algorithm.

Numerical Demonstration

We graphically and numerically demonstrate the affect of this distortion on a sequence consisting of 2^7 ones with $N = 4$. First we perform a complete DWT and then reconstruct the sequence from the resulting wavelet coefficients. We will call the original sequence of 16 ones f^4 and the successive results of the decomposition will be referred to as f^3, f^2, f^1, and f^0. We will denote the successive results of the inverse decomposition applied to the wavelet transform of f^4 by $\bar{f}^0$, $\bar{f}^1$, $\bar{f}^2$, $\bar{f}^3$, and $\bar{f}^4$. Thus the decomposition chain is

$$\begin{cases} f^3 = L_4^4 f^4 \\ w^3 = H_4^4 f^4 \end{cases} \quad \begin{cases} f^2 = L_3^4 f^3 \\ w^2 = H_3^4 f^3 \end{cases} \quad \begin{cases} f^1 = L_2^4 f^2 \\ w^1 = H_2^4 f^2 \end{cases} \quad \begin{cases} f^0 = L_1^4 f^1 \\ w^0 = H_1^4 f^1 \end{cases}$$

and the reconstruction chain is, with $\bar{f}^0 = f^0$

$$\begin{aligned} \bar{f}^1 &= L_1^{4*}\bar{f}^0 + H_1^{4*}w^0 \\ \bar{f}^2 &= L_2^{4*}\bar{f}^1 + H_2^{4*}w^1 \\ \bar{f}^3 &= L_3^{4*}\bar{f}^2 + H_3^{4*}w^2 \\ \bar{f}^4 &= L_4^{4*}\bar{f}^3 + H_4^{4*}w^3 \end{aligned}$$

Figure 3 shows the inverse transform result $\bar{f}^4$.

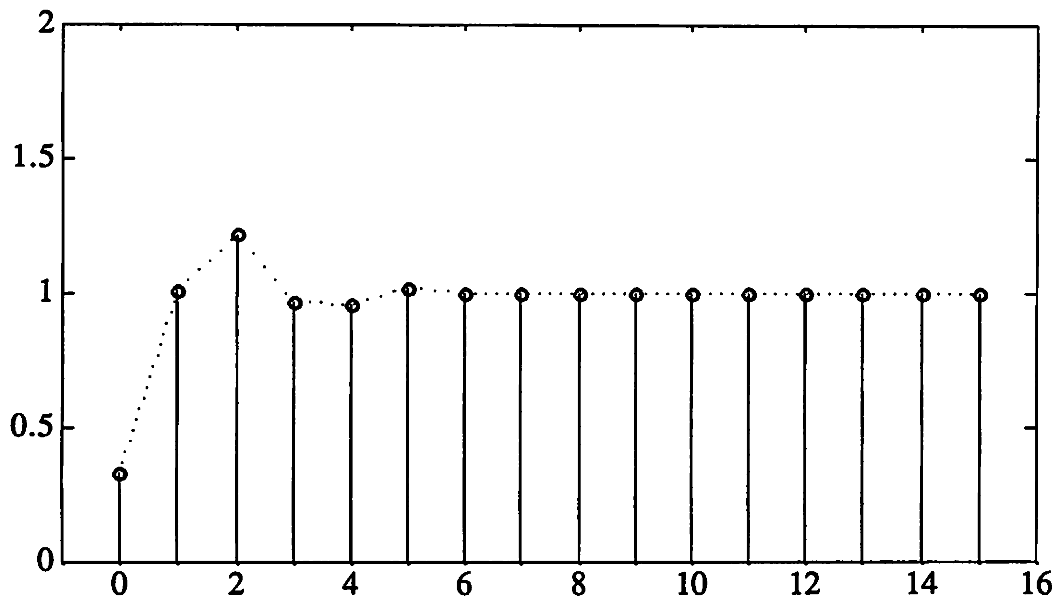

Figure 3: $\bar{f}^4$, the result of performing standard discrete wavelet transform followed by an inverse discrete wavelet transform on f^4, a sequence of 16 ones. The $N = 4$ Daubechies coefficients were used in both the decomposition and the reconstruction. The resulting distortion can be seen in the first 6 elements of $\bar{f}^4$.

The distortion in the first 6 elements of $\bar{f}^4$ is readily apparent. We note that if we had started with a much longer sequence that the distortion region would still have the same size of support and would consequently be much smaller in relation to the entire signal length. The following figures 4.a through 4.d reveal, however, that at small p the support of the distortion is a large part or all of the sequence.

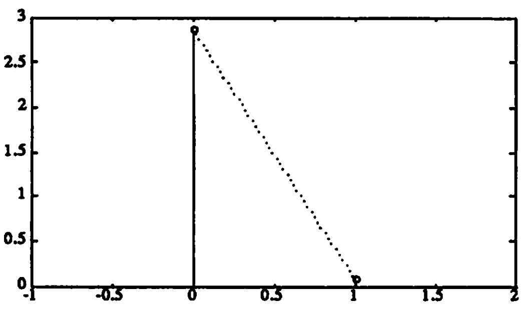

Figure 4.a. $f^1 - \tilde{f}^1$, the error at level 1.

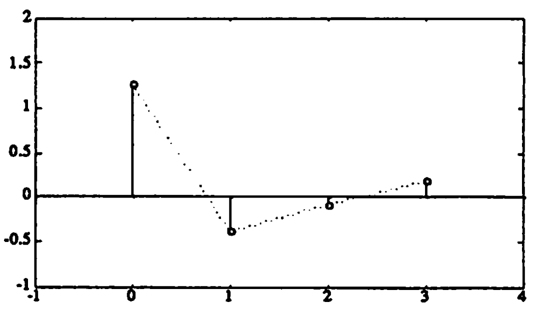

Figure 4.b. $f^2 - \tilde{f}^2$, the error at level 2.

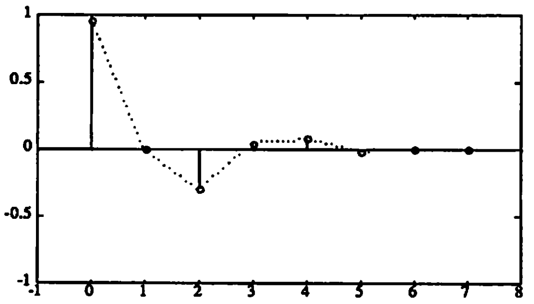

Figure 4.c. $f^3 - \tilde{f}^3$, the error at level 3.

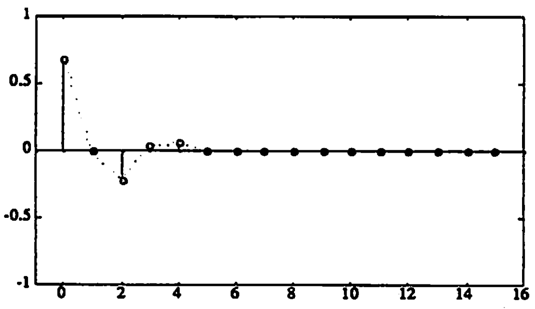

Figure 4.d. $f^4 - \tilde{f}^4$, the error at the top level, 4.

Figure 4.a shows the difference between the decomposition product f^1 and the reconstruction product $\tilde{f}^1$. If the DWT were truly invertible on sequences of length 2 we would expect $\tilde{f}^1 = f^1$. This is seen to not be so. Figures 4.b, 4.c, and 4.d reveal the level and distribution of the reconstruction errors back up to the base level 4.

The non-invertibility problem of the DWT has been approached, within the context of subband coding of images, by Smith and Eddins in [SE] using circular convolution. Unlike Smith and Eddins, however, we derive an explicit realization that does not involve periodically extending sequences at the level of algorithm realization. We do not address, however, the robustness of the DPWT with respect to quantization errors as is done in [SE] and we refer the reader to that paper for such a discussion.

Another attempt at solving the non-invertibility problem of the DWT is shown in

[PTVF]. Those authors show a matrix realization of the DWT in which filter coefficients are wrapped around the filter matrix so that, for instance, those coefficients shifted off of the right end of the bottom row of the filter operator matrix are shifted back into the row at the left end of the same row. This solves the invertibility problem down to a level p where $2N = 2^p$. The authors stop the algorithm there. By coming to an understanding of the nature of the partial solution yielded by the wrap-around of coefficients we will see the true context of this partial fix and will be able to extend it to its ultimate goal of perfect invertibility down to level 0.

We will show in the sequel that the errors demonstrated in the examples above may be eliminated by applying the discrete periodic wavelet transform, which we derive in the next section, to the $l^\infty(\mathbb{Z}_p)$ sequence. A finite length sequence may then be cast as a single period of a periodic sequence. If the sequence is not truly periodic, e.g. a piece of a longer non-periodic sequence, there is some cost to the casting with respect to interpretation of the wavelet decomposition. This cost is discussed in section VI.

IV. A Discrete Periodic Wavelet Transform

In this section we will work from the standard form of the DWT (7) to produce the DPWT and IDPWT. Later, in section VI we will introduce operators which will lend more transparency to the forms of the DPWT derived below.

We retain the algorithmic structure symbolized by the tree diagrams of figures 1.a and 1.b, but we restrict the domains and codomains of the filters operators. We identify f^p, which we consider to be an element of $l^\infty(\mathbb{Z}_p)$, with its periodic counterpart defined by $\tilde{f}_k^p \equiv f_{k \bmod 2^p}^p$, $k \in \mathbb{Z}$, which we consider to be an element of the subspace $\mathcal{F}_p$ of $l^\infty(\mathbb{Z})$ periodic sequences of period 2^p. Likewise, we identify our wavelet coefficient vector w_k^p, $k \in \mathbb{Z}_p$, which we also consider to be an element of $l^\infty(\mathbb{Z}_p)$, with $\tilde{w}_k^p \equiv w_{k \bmod 2^p}^p$, $k \in \mathbb{Z}$ in another vector space of period 2^p $l^\infty(\mathbb{Z})$ sequences, $\mathcal{W}_p$. At the same time we leave our filters in their original non-periodic form l_k^N, h_k^N, $k \in \mathbb{Z}$ where the support of l_k^N and h_k^N is $Q_N \equiv \{0, 1, \ldots, 2N-1\}$, i.e. $l^N, h^N \in l^2(Q_N)$.

Applying the $l^\infty(\mathbb{Z}) \to l^\infty(\mathbb{Z})$ DWT (7) to $\tilde{f}_k^p$ gives

$$\begin{cases} \tilde{f}_i^{p-1} = \sum_{k=-\infty}^{\infty} l_k^N \tilde{f}_{k+2i}^p, & i \in \mathbb{Z} \\ \tilde{w}_i^{p-1} = \sum_{k=-\infty}^{\infty} h_k^N \tilde{f}_{k+2i}^p, & i \in \mathbb{Z} \end{cases} \tag{19}$$

where we have non-periodic operators operating on periodic sequences. We note that the decomposition products are also periodic, but with half the period of $\tilde{f}^p$.

We may take advantage of the compact support of the filters as well as the identification between f_k^p and $\tilde{f}_k^p$, and w_k^p and $\tilde{w}_k^p$ to obtain the decomposition form

$$\begin{cases} \tilde{f}_i^{p-1} = \sum_{k \in \mathbb{Z}_p} \tilde{f}_k^p \sum_{m=-\infty}^{\infty} l_{k-2i+2^p m}^N, & i \in \mathbb{Z} \\ \tilde{w}_i^{p-1} = \sum_{k \in \mathbb{Z}_p} \tilde{f}_k^p \sum_{m=-\infty}^{\infty} h_{k-2i+2^p m}^N, & i \in \mathbb{Z} \end{cases} \tag{20}$$

(If the equivalence between (20) and (19) is not immediately clear it will be made so by observation of the DPWT matrix forms in section V.) Note that for those decomposition stages p such that $2^p \geq 2N$, i.e. such that the period of $\tilde{f}^p$ is greater than or equal to the length of the index N filter, we have

$$\sum_{m=-\infty}^{\infty} l_{k-2i+2^p m}^N = l_{k-2i}^N$$
$$\sum_{m=-\infty}^{\infty} h_{k-2i+2^p m}^N = h_{k-2i}^N$$

Consequently, if we identify $\tilde{f}^p$ with its counterpart $f^p \in l^\infty(\mathbb{Z}_p)$, the first $2^{p-1} - (2N-2)$ samples of f^{p-1} and w^{p-1} from transform (20) are identical to the corresponding transform products of the standard DWT (7) applied to f^p. Thus, for the case of $N = 1$, corresponding to the Haar wavelet, the transform (20) is identical to (7) at all stages of the decomposition.

Exploiting the compact support of l^N and h^N and defining
$\tilde{l}_p^N$ and $\tilde{h}_p^N$, vectors in $\mathcal{P}_p$ by

$$\tilde{l}^N_{p,i} \equiv \sum_{k=0}^{\lfloor (2N-1)/2^p \rfloor} l^N_{i \bmod 2^p + 2^p k}, \quad i \in \mathbb{Z} \tag{21}$$

$$\tilde{h}^N_{p,i} \equiv \sum_{k=0}^{\lfloor (2N-1)/2^p \rfloor} h^N_{i \bmod 2^p + 2^p k}, \quad i \in \mathbb{Z} \tag{22}$$

we may express what we will call, in conjunction with (21) and (22), the discrete periodic wavelet transform:

$$\begin{cases} \tilde{f}^{p-1}_i = \sum_{k \in \mathbb{Z}_p} \tilde{f}^p_k \, \tilde{l}^N_{p,k-2i}, & i \in \mathbb{Z} \\ \tilde{w}^{p-1}_i = \sum_{k \in \mathbb{Z}_p} \tilde{f}^p_k \, \tilde{h}^N_{p,k-2i}, & i \in \mathbb{Z} \end{cases} \tag{23}$$

Note that when $2^p \geq 2N$ the sums on l^N and h^N in expressions (21) and (22) have no affect in that there is only one filter coefficient summed. Indeed, for $2^p \geq 2N$, $l^N_i = \tilde{l}^N_i$ and $h^N_i = \tilde{h}^N_i$ for all $i \in \mathbb{Z}_p$. The sums (21) and (22) are most easily thought of as being the result of applying a linear operator, which we call a *wrap*, to l^N and h^N. Wraps will be illustrated and explained in section VI.

Utilizing the isomorphism between $l^\infty(\mathbb{Z}_p)$ and $\mathcal{P}_p$ we may express the DPWT by

$$\begin{cases} f^{p-1}_i = \sum_{k \in \mathbb{Z}_p} f^p_k \, \tilde{l}^N_{p,k-2i}, & i \in \mathbb{Z}_{p-1} \\ w^{p-1}_i = \sum_{k \in \mathbb{Z}_p} f^p_k \, \tilde{h}^N_{p,k-2i}, & i \in \mathbb{Z}_{p-1} \end{cases} \tag{24}$$

which we refer to as the $l^\infty(\mathbb{Z}_p)$ form of the DPWT. We see that once $\tilde{l}^N_p$ and $\tilde{h}^N_p$ are computed from (21) and (22) that the DPWT is formed by taking the inner product of a single period of $\tilde{f}^p$ with even rotations of $\tilde{l}^N_p$ and $\tilde{h}^N_p$. It will be shown that unlike the DWT applied to $l^\infty(\mathbb{Z}_p)$, the DPWT is perfectly invertible.

The algorithm (24) may be made more concise by realizing that $\tilde{l}^N_{p,k-2i}$ and $\tilde{h}^N_{p,k-2i}$ are nonzero only when $0 \leq k - 2i \leq \min(2N-1, 2^p - 1)$. In other words we may restrict the summation index integer k to $k \in \{2i, \ldots, 2i + \min(2N-1, 2^p-1)\}$. Letting $n = k - 2i$, and defining the integer subset

$$\mathbf{S}_N^p(i) \equiv \left\{2i,\, 2i+1, \ldots, 2i+\min\left(2N, 2^p\right)-1\right\} \bmod 2^p$$

the DPWT may then be rephrased in a computationally economic form as

$$\text{DPWT:} \quad \begin{cases} f_i^{p-1} = \displaystyle\sum_{n \in \mathbf{S}_N^p(i)} f_n^p\, \tilde{l}^N_{p,(n-2i) \bmod 2^p}, & i \in \mathbf{Z}_{p-1} \\ w_i^{p-1} = \displaystyle\sum_{n \in \mathbf{S}_N^p(i)} f_n^p\, \tilde{h}^N_{p,(n-2i) \bmod 2^p}, & i \in \mathbf{Z}_{p-1} \end{cases} \tag{25}$$

which we refer to as the $l^\infty\left(\mathbf{Z}_p\right)$ form of the DPWT.

We will refer to the DPWT versions of L and H by $\tilde{L}$ and $\tilde{H}$. As with the $l^2(\mathbf{Z}) \to l^2(\mathbf{Z})$ DWT the inverse DPWT is obtained by utilizing the adjoints of $\tilde{L}$ and $\tilde{H}$. That this actually works for the $\tilde{L}^N$ and $\tilde{H}^N$ corresponding to $\tilde{l}^N$ and $\tilde{h}^N$ when it did not work for the DWT applied $l^2\left(\mathbf{Z}_p\right) \to l^2\left(\mathbf{Z}_p\right)$ requires proof. In section VI we will prove that the DPWT and IDPWT are indeed an orthonormal wavelet transform pair.

The $l^\infty\left(\mathbf{Z}_p\right)$ form of the IDPWT is

$$f_i^p = \sum_{k \in \mathbf{Z}_{p-1}} \tilde{l}^N_{p,i-2k}\, f_k^{p-1} + \sum_{k \in \mathbf{Z}_{p-1}} \tilde{h}^N_{p,i-2k}\, w_k^{p-1}, \quad i \in \mathbf{Z}_p \tag{26}$$

Since $\tilde{l}^N_{p,i-2k}$ and $\tilde{h}^N_{p,i-2k}$ are non-zero only when $0 \le i-2k \le \min\left(2N-1, 2^p-1\right)$ we may further restrict the range of the summation index integer k in (26) to

$$\mathbf{T}_N^p(i) \equiv \left\{\left\lceil \frac{i+1}{2} \right\rceil - 1, \left\lceil \frac{i+1}{2} \right\rceil - 2, \ldots, \left\lceil \frac{i+1}{2} \right\rceil - \min\left(N, 2^{p-1}\right)\right\} \bmod 2^p$$

where $\lceil x \rceil$ denotes the smallest integer larger than x. Consequently the IDPWT may be rephrased in a computationally economical manner as

$$\text{IDPWT:} \quad f_i^p = \sum_{k \in \mathbf{T}_N^p(i)} \tilde{l}^N_{p,(i-2k) \bmod 2^p}\, f_k^{p-1} + \tilde{h}^N_{p,(i-2k) \bmod 2^p}\, w_k^{p-1}, \quad i \in \mathbf{Z}_p \tag{27}$$

Complexity

The transform pair (25) and (27) is fast. For each level p such that $2^p \geq 2N$ the number of arithmetic operations needed to obtain each element of the transform products is $2N$ multiplications and $2N$ - 1 additions. When p is such that $2^p < 2N$ there are 2^p multiplications and $2^p - 1$ additions needed for each element of the transform products. The number of elements of the transform products is halved at each level. Consequently, assuming that the sequence to be analyzed is of length 2^p, the number of operations for a complete DPWT decomposition using $l(\mathbf{Q}_N)$ filters is

$$\begin{aligned} &2N \times \left(2^{p+1} - 2^{k+1}\right) + 2^{k+1} - 1 \quad \textit{multiplications} \\ &(2N-1) \times \left(2^{p+1} - 2^{k+1}\right) + 2^{k+1} - 1 - k \quad \textit{additions} \end{aligned}$$

where k is the largest integer such that $2^k < 2N$. Thus the number of multiplications in the complete DPWT of a length n sequence is bounded above by $4Nn$. Compared to a fast Fourier transform (FFT) the DPWT is faster by approximately a factor of $\log_2 n$. In addition the DPWT does not require the use of complex numbers.

In the next section matrices will afford insight into the structure of the DPWT.

V. Demonstration of the DPWT

Like Strang [Strang] with the DWT we believe that the DPWT is best illustrated in matrix form. Consequently we arrange the filter coefficients into a matrix realization of the filter operators. We illustrate our algorithm with the simplest possible example that displays all its important symmetries. Sequence f^3 will be decomposed into its wavelet coefficients down to level 0 and its residue $f^0 = w^{-1}$. Those wavelet coefficients will be gathered together to form the wavelet transform vector $\hat{\mathbf{f}}^3$. Then we will reconstruct the original f^3 starting from the wavelet coefficients.

Analysis Tree

We will assume that l^N and h^N satisfy (10), (11), and (9). We perform the DPWT and the IDPWT in their $l^\infty(Z_p)$ forms. We may choose to think of f^p as being the coordinate representation of $\tilde{f}^p$. We start with a sequence f^3 of length 8. We operate on the left of the sequence with a matrix realization of the low-pass and decimate operator $\tilde{L}_3^3$ to obtain $\mathbf{f}^2 = \tilde{\mathbf{L}}_3^3\mathbf{f}^3$.

$$\mathbf{f}^2 = \begin{bmatrix} f_0^2 \\ f_1^2 \\ f_2^2 \\ f_3^2 \end{bmatrix} = \begin{bmatrix} l_0^3 & l_1^3 & l_2^3 & l_3^3 & l_4^3 & l_5^3 & & \\ & & l_0^3 & l_1^3 & l_2^3 & l_3^3 & l_4^3 & l_5^3 \\ l_4^3 & l_5^3 & & & l_0^3 & l_1^3 & l_2^3 & l_3^3 \\ l_2^3 & l_3^3 & l_4^3 & l_5^3 & & & l_0^3 & l_1^3 \end{bmatrix} \begin{bmatrix} f_0^3 \\ f_1^3 \\ f_2^3 \\ f_3^3 \\ f_4^3 \\ f_5^3 \\ f_6^3 \\ f_7^3 \end{bmatrix} \tag{28}$$

Note the right circular shift by two of the low-pass filter coefficients in the matrix as well as the wrapping around of the filter coefficients on the bottom two rows.

The symbol geometry of the matrix forms of the high-pass-and-decimate operation are identical to that of the low-pass-and-decimate cases. Accordingly we obtain $\mathbf{w}^2 = \tilde{\mathbf{H}}_3^3\mathbf{f}^3$, the coordinate representation of the wavelet coefficients of $\tilde{f}^3$ at scale 2, by replacing f_q^k with w_q^k, and l_q^3 with h_q^3 in (28). We have created this convenience by placing l^N and h^N on common support. If we had placed h^N on different support than l^N the forms of the matrices $\tilde{\mathbf{H}}_3^3$ and $\tilde{\mathbf{L}}_3^3$ would have been related, row for row, by a common and even circular shift.

As mentioned in section III the wrap-around of the coefficients shown in the matrix of (28) has appeared in [PTVF] and elsewhere. According to the technique shown in [PTVF] the decomposition stops at this stage (28). By our understanding of the DPWT as gleaned from the last section we may now continue the decomposition.

In the next step of the algorithm we obtain $\mathbf{f}^1 = \tilde{\mathbf{L}}_2^3\mathbf{f}^2$ and $\mathbf{w}^1 = \tilde{\mathbf{H}}_2^3\mathbf{f}^2$. At this stage our filter matrices take on widths less than the lengths of the filter coefficient vectors, i.e.

$p = 2$, and $n = 3$. are such that $2^p < 2N$.

$$\mathbf{f}^1 = \begin{bmatrix} f_0^1 \\ f_1^1 \end{bmatrix} = \begin{bmatrix} l_0^3 + l_4^3 & l_1^3 + l_5^3 & l_2^3 & l_3^3 \\ l_2^3 & l_3^3 & l_0^3 + l_4^3 & l_1^3 + l_5^3 \end{bmatrix} \begin{bmatrix} f_0^2 \\ f_1^2 \\ f_2^2 \\ f_3^2 \end{bmatrix} \tag{29}$$

and the corresponding w equation is obtained by replacing f's and l's with identically indexed w's and h's. Here we see the effects of the sums (21) and (22) defining $\tilde{l}^N$ and $\tilde{h}^N$. Similar to the previous step (28), in (29) we see again a circular right shift by 2 of the filter coefficients and a circular left shift by 2 of the scale 2 sequence coefficients in the matrix representations.

Now, in the final analysis step we form $\mathbf{f}^0 = \tilde{\mathbf{L}}_1^3\mathbf{f}^1$ and $\mathbf{w}^0 = \tilde{\mathbf{H}}_1^3\mathbf{f}^1$ from $\mathbf{f}^1$.

$$\mathbf{f}^0 = \begin{bmatrix} f_0^0 \end{bmatrix} = \begin{bmatrix} l_0^3 + l_2^3 + l_4^3 & l_1^3 + l_3^3 + l_5^3 \end{bmatrix} \begin{bmatrix} f_0^1 \\ f_1^1 \end{bmatrix} \tag{30}$$

$$\mathbf{w}^0 = \begin{bmatrix} w_{0.}^0 \end{bmatrix} = \begin{bmatrix} h_0^3 + h_2^3 + h_4^3 & h_1^3 + h_3^3 + h_5^3 \end{bmatrix} \begin{bmatrix} f_0^1 \\ f_1^1 \end{bmatrix} \tag{31}$$

Note that, for instance,

$$\begin{aligned} \tilde{\mathbf{L}}_1^3\tilde{\mathbf{L}}_1^{3*} &= \begin{bmatrix} l_0^3 + l_2^3 + l_4^3 & l_1^3 + l_3^3 + l_5^3 \end{bmatrix} \begin{bmatrix} l_0^3 + l_2^3 + l_4^3 \\ l_1^3 + l_3^3 + l_5^3 \end{bmatrix} \\ &= \sum_n l_n^3\, l_n^3 + 2\left(\sum_n l_n^3\, l_{n-2}^3 + \sum_n l_n^3\, l_{n-4}^3 \right) = 1 \end{aligned}$$

$$\begin{aligned} \tilde{\mathbf{H}}_1^3\tilde{\mathbf{H}}_1^{3*} &= \begin{bmatrix} h_0^3 + h_2^3 + h_4^3 & h_1^3 + h_3^3 + h_5^3 \end{bmatrix} \begin{bmatrix} h_0^3 + h_2^3 + h_4^3 \\ h_1^3 + h_3^3 + h_5^3 \end{bmatrix} = 1 \\ &= \sum_n l_n^3\, l_n^3 + 2\left(\sum_n l_n^3\, l_{n-2}^3 + \sum_n l_n^3\, l_{n-4}^3 \right) = 1 \end{aligned}$$

and

$$\begin{aligned}\tilde{\mathbf{L}}_1^3\tilde{\mathbf{H}}_1^{3*} &= \begin{bmatrix} l_0^3+l_2^3+l_4^3 & l_1^3+l_3^3+l_5^3\end{bmatrix}\begin{bmatrix} h_0^3+h_2^3+h_4^3 \\ h_1^3+h_3^3+h_5^3\end{bmatrix} \\ &= \sum_n l_n^3\, h_n^3 + \sum_n l_{n-2}^3\, h_n^3 + \sum_n l_n^3\, h_{n-2}^3 \\ &\quad + \sum_n l_n^3\, h_{n-4}^3 + \sum_n l_{n-4}^3\, h_n^3 = 0\end{aligned}$$

as well as

$$\begin{aligned}\tilde{\mathbf{L}}_1^{3*}\tilde{\mathbf{L}}_1^3 + \tilde{\mathbf{H}}_1^{3*}\tilde{\mathbf{H}}_1^3 &= \begin{bmatrix} l_0^3+l_2^3+l_4^3 \\ l_1^3+l_3^3+l_5^3\end{bmatrix}\begin{bmatrix} l_0^3+l_2^3+l_4^3 & l_1^3+l_3^3+l_5^3\end{bmatrix} \\ &\quad + \begin{bmatrix} l_5^3+l_3^3+l_1^3 \\ -l_4^3-l_2^3-l_0^3\end{bmatrix}\begin{bmatrix} l_5^3+l_3^3+l_1^3 & -l_4^3-l_2^3-l_0^3\end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22}\end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22}\end{bmatrix}\end{aligned}$$

where

$$\begin{aligned} a_{11} &= (l_0^3)^2 + (l_2^3)^2 + (l_4^3)^2 + 2(l_0^3 l_2^3 + l_0^3 l_4^3 + l_2^3 l_4^3) \\ a_{12} = a_{21} &= l_0^3 l_1^3 + l_1^3 l_2^3 + l_0^3 l_3^3 + l_2^3 l_3^3 + l_1^3 l_4^3 + l_3^3 l_4^3 + l_0^3 l_5^3 + l_2^3 l_5^3 + l_4^3 l_5^3 \\ a_{22} &= (l_1^3)^2 + (l_3^3)^2 + (l_5^3)^2 + 2(l_1^3 l_3^3 + l_1^3 l_5^3 + l_3^3 l_5^3) \\ b_{11} &= a_{22} \\ b_{12} = b_{21} &= -a_{12} \\ b_{22} &= a_{11}\end{aligned}$$

and therefore

$$\tilde{\mathbf{L}}_1^{3*}\tilde{\mathbf{L}}_1^3 + \tilde{\mathbf{H}}_1^{3*}\tilde{\mathbf{H}}_1^3 = \mathbf{I}_1$$

It may be verified in a similar manner that the pairs $\tilde{L}_2$, $\tilde{H}_2$, and $\tilde{L}_3$, $\tilde{H}_3$ also satisfy criteria (15), (16), (17), and (18). The wavelet transform of f^3 is then $\hat{\mathbf{f}}^3 = [\mathbf{w}^{-1}, \mathbf{w}^0, \mathbf{w}^1, \mathbf{w}^2] = [f_0^0, w_0^0, w_0^1, w_1^1, w_0^2, w_1^2, w_2^2, w_3^2]$ and this completes the analysis algorithm.

Synthesis Tree

The synthesis portion of the algorithm starts from $\hat{\mathbf{f}}^3$ and proceeds as follows: First we use $\mathbf{w}^{-1} = f_0^0$ and $\mathbf{w}^0$ to obtain $\mathbf{f}^1 = \tilde{\mathbf{L}}_1^{3*}\mathbf{f}^0 + \tilde{\mathbf{H}}_1^{3*}\mathbf{w}^0$.

$$\mathbf{f}^1 = \begin{bmatrix} f_0^1 \\ f_1^1 \end{bmatrix} = \begin{bmatrix} l_0^3 + l_2^3 + l_4^3 \\ l_1^3 + l_3^3 + l_5^3 \end{bmatrix} \begin{bmatrix} f_0^0 \end{bmatrix} + \begin{bmatrix} h_0^3 + h_2^3 + h_4^3 \\ h_1^3 + h_3^3 + h_5^3 \end{bmatrix} \begin{bmatrix} w_0^0 \end{bmatrix} \tag{32}$$

The filter matrices in (32) are the transposes of the level 1 filter matrices in the level 1 analysis step (30). Similarly we obtain $\mathbf{f}^2 = \tilde{\mathbf{L}}_2^{3*}\mathbf{f}^1 + \tilde{\mathbf{H}}_2^{3*}\mathbf{w}^1$:

$$\mathbf{f}^2 = \begin{bmatrix} f_0^2 \\ f_1^2 \\ f_2^2 \\ f_3^2 \end{bmatrix} = \begin{bmatrix} l_0^3 + l_4^3 & l_2^3 \\ l_1^3 + l_5^3 & l_3^3 \\ l_2^3 & l_0^3 + l_4^3 \\ l_3^3 & l_1^3 + l_5^3 \end{bmatrix} \begin{bmatrix} f_0^1 \\ f_1^1 \end{bmatrix} + \begin{bmatrix} h_0^3 + h_4^3 & h_2^3 \\ h_1^3 + h_5^3 & h_3^3 \\ h_2^3 & h_0^3 + h_4^3 \\ h_3^3 & h_1^3 + h_5^3 \end{bmatrix} \begin{bmatrix} w_0^1 \\ w_1^1 \end{bmatrix}$$

and $\mathbf{f}^3 = \tilde{\mathbf{L}}_3^{3*}\mathbf{f}^2 + \tilde{\mathbf{H}}_3^{3*}\mathbf{w}^2$ with

$$\mathbf{f}^3 = \begin{bmatrix} f_0^3 \\ f_1^3 \\ f_2^3 \\ f_3^3 \\ f_4^3 \\ f_5^3 \\ f_6^3 \\ f_7^3 \end{bmatrix} = \begin{bmatrix} l_0^3 & & l_4^3 & l_2^3 \\ l_1^3 & & l_5^3 & l_3^3 \\ l_2^3 & l_0^3 & & l_4^3 \\ l_3^3 & l_1^3 & & l_5^3 \\ l_4^3 & l_2^3 & l_0^3 & \\ l_5^3 & l_3^3 & l_1^3 & \\ & l_4^3 & l_2^3 & l_0^3 \\ & l_5^3 & l_3^3 & l_1^3 \end{bmatrix} \begin{bmatrix} f_0^2 \\ f_1^2 \\ f_2^2 \\ f_3^2 \end{bmatrix} + \begin{bmatrix} h_0^3 & & h_4^3 & h_2^3 \\ h_1^3 & & h_5^3 & h_3^3 \\ h_2^3 & h_0^3 & & h_4^3 \\ h_3^3 & h_1^3 & & h_5^3 \\ h_4^3 & h_2^3 & h_0^3 & \\ h_5^3 & h_3^3 & h_1^3 & \\ & h_4^3 & h_2^3 & h_0^3 \\ & h_5^3 & h_3^3 & h_1^3 \end{bmatrix} \begin{bmatrix} w_0^2 \\ w_1^2 \\ w_2^2 \\ w_3^2 \end{bmatrix}$$

This completes our example. Note that the synthesis algorithm may be continued to interpolate f^3 to higher levels by letting $w^p = 0_p$ for all $p > 3$.

In section VI we will formalize some mathematical notions that generalize and afford mathematical insight into those symbol symmetries that are readily apparent in the above matrices. During a reading of section VI the reader may wish to refer back to this section and compare its symbol symmetries to the symmetries of the operators that will be found there.

VI. Filter Coefficients for the DPWT

Here we prove, after some mathematical preliminaries, that if $l^N \in l^2(\mathbf{Q}_N)$ can be used to create a $l^\infty(\mathbf{Z}) \to l^\infty(\mathbf{Z})$ DWT (7) then it can also be used, according to (21), (22), (24), and (26) of section IV to create a DPWT.

This section may be thought of as providing an alternative to the derivation of the DPWT in section IV where we now start from the filter vectors $\tilde{l}_p^N$ (21) and $\tilde{h}_p^N$ (22) and prove that in the context of (25) they produce a wavelet transform for and a basis of $l^\infty(\mathbf{Z})$.

Mathematical Preliminaries

We first remind the reader of a few properties of the Kronecker delta $\delta_{a,b}$ which for our purposes will be defined as follows: Let $\mathcal{A}$ and $\mathbf{B}$ be subsets of $\mathbf{Z}$. Let $\mathbf{R}$ denote the real numbers. For $a \in \mathcal{A}$ and $b \in \mathbf{B}$, if $a = b$ then $\delta_{a,b} = 1 \in \mathbf{R}$, and if $a \neq b$ then $\delta_{a,b} = 0 \in \mathbf{R}$. Alternatively the delta may be thought of as an indicator function for the event that its arguments are equal. We note the following useful properties of $\delta_{a,b}$:

$$\delta_{a,b} = \delta_{b,a} \quad \textit{commutativity of arguments} \quad (33)$$

$$\begin{gathered}\forall g : \mathcal{A} \to \mathbf{R}, \forall a, b \in \mathcal{A} \subset \mathbf{R}, \\ \sum_{b \in \mathcal{A}} \delta_{a,b}\, g(a) = g(b)\end{gathered} \quad \textit{sifting property} \quad (34)$$

We point out some consequences of these properties that will prove useful later in this section:

Fact: Noting that $a \bmod p = (b + c) \bmod k$ if and only if $(a - c) \bmod k = b \bmod k$ gives

$$\delta_{a \bmod k, (b+c) \bmod k} = \delta_{(a-c) \bmod k, b \bmod k} \quad (35)$$

■

Fact: Let $\mathcal{A}$ and $\mathbf{C}$ be subsets of $\mathbf{Z}$. For any $a, b \in \mathcal{A}$ and any $c \in \mathbf{C}$,

$$\sum_{b\in\mathcal{A}} \delta_{a,b}\,\delta_{b,c} = \delta_{a,c} \tag{36}$$

■

Fact: Let $\mathcal{A}$, $\mathbb{B}$, and $\mathbb{C}$ be subsets of $\mathbb{Z}$ with $\mathcal{A} \subset \mathbb{B}$. Let *f* be an arbitrary function $f:\mathbb{B} \to \mathbb{Z}$. Then for any $a \in \mathcal{A}$ and any $c \in \mathbb{C}$

$$\sum_{i\in\mathbb{B}} \delta_{a,i}\,\delta_{f(i),c} = \delta_{f(a),c} \tag{37}$$

■

Wraps

As above we let $\mathbf{P}_p$ denote the subspace of $l^{\infty}(\mathbb{Z})$ consisting of periodic functions of period 2^p and $\mathbb{Z}_p$ denotes $\{0, 1, 2, ..., 2^p\text{-}1\}$. We define a mapping which we call a *wrap*,

$$\begin{aligned} &@^p : l^1(\mathbb{Z}) \to \mathbf{P}_p \\ &y_n,\ n\in\mathbb{Z} \mapsto \sum_{k=-\infty}^{\infty} y_{i \bmod 2^p + 2^p k},\quad i\in\mathbb{Z} \end{aligned} \tag{38}$$

The sum in (38) will converge for all $l^1(\mathbb{Z})$ sequences y, where $l^1(\mathbb{Z})$ denotes the space of all absolutely summable sequences.

The action of the wrap $@^p$ may be visualized by imagining taking an $l^1(\mathbb{Z})$ sequence s, wrapping it into a right handed helix of circumference 2^p, summing all entries along the length of the helix corresponding to a common circumference coordinate, and mapping that sum to the element of $@^p s$ with the same circumference coordinate.

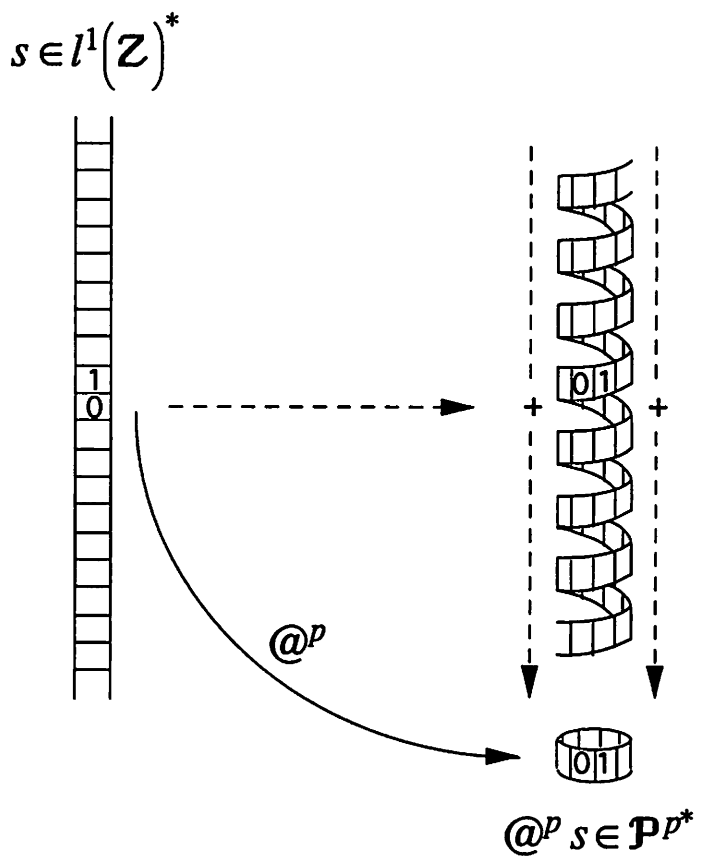

Figure 5. The wrap $@^p$ acts on an infinite length sequence s in $l^1(\mathbb{Z})$ to create a periodic sequence in $\mathcal{P}_p \subset l^\infty(\mathbb{Z})$. The 0 and 1 in the diagram indicate corresponding indices and orientation in the infinite length sequence, the helix, and the periodic sequence.

In practice we will restrict $@^p$ to $l^2(\mathrm{Q}_N) \subset l^1(\mathbb{Z})$ consisting of sequences with compact support $\mathrm{Q}_N = \{0, ..., 2N - 1\}$, $N < \infty$. Of course $l^2(\mathrm{Q}_N)$ is then isomorphic to $\mathbb{R}^{2N-1}$ from which we draw its coordinate representation. Note that by $l^2(\mathrm{Q}_N)$ we restrict only the support of $l^2(\mathbb{Z})$ sequences and not the range of their indices. Thus if x is in $l^2(\mathrm{Q}_N)$, then, for example, $x_{2N} = 0, x_{-1} = 0$. We denote the restriction of the wrap to $l^2(\mathrm{Q}_N)$ by

$$\begin{aligned} @_N^p &: l^2(\mathrm{Q}_N) \to \mathcal{P}_p \\ y_n, n \in \mathrm{Q}_N &\mapsto \sum_{k=0}^{\left\lfloor \frac{2N-1}{2^p} \right\rfloor} y_{i \bmod 2^p + 2^p k}, i \in \mathbb{Z} \end{aligned} \tag{39}$$

The restricted wrap $@_N^p$ has a matrix representation

$$\left[@_N^p\right]_{i,j} = \delta_{i,j \bmod 2^p}, \quad i \in \mathcal{Z}_p, j \in \mathbf{Q}_N \tag{40}$$

(Note that in referring to matrices we will consider first rows and columns to have index 0.) A restricted wrap is invertible if and only if $2N \le 2^p$ in which case $\left[@_N^p y\right]_{i \bmod 2^p} = y_i$, $i \in \mathcal{Z}$. If $@_N^p$ is invertible then its left inverse is its adjoint $@_N^{p*}$.

As an example of a wrap with $2N > 2^p$ we let $N = 3$ so that $\mathbf{Q}_N = \{0, 1, \ldots, 5\}$. Choosing $p = 2$ we have as the matrix representation of $@_3^2$ with respect to the standard bases for domain $\mathbf{Q}_3$ and codomain $\mathcal{P}_2$

$$@_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \left[\delta_{i,j \bmod 4}\right], \quad i \in \{0, \ldots, 3\}, j \in \{0, \ldots, 5\}$$

For an example of the case where $2N < 2^p$ we let $N = 2$ and $p = 3$ to get

$$@_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The invertibility of $@_2^3$ shows up as the left invertibility of its matrix representation by its transpose $@_2^{3*}$.

Two Shifts

We will need two shift operators; one for $l^2(\mathbb{Z})$ and one for $\mathcal{P}_p$. For $l^2(\mathbb{Z})$ we have for $k \in \mathbb{Z}$

$$\begin{gathered} S^k : l^2(\mathbb{Z}) \to l^2(\mathbb{Z}) \\ y_n, n \in \mathbb{Z} \mapsto y_{n-k}, n \in \mathbb{Z} \end{gathered}$$

representable by the kernel $\delta_{i-k,j}$ as in

$$\left[S^k y\right]_n = \sum_{j=-\infty}^{\infty} \delta_{n-k,j}\, y_j$$

a right shift by k. We have the following useful properties of S_k:

$$\left(S^k\right)^{-1} = \left(S^k\right)^* = S^{-k} \qquad \textit{unitarity} \quad (41)$$

$$S^k S^j = S^j S^k \qquad \textit{commutativity} \quad (42)$$

$$S^j S^k = S^{j+k} \qquad \textit{group property} \quad (43)$$

and we see that $\left\{S^k\right\}_{k \in \mathbb{Z}}$ is a unitary group under the operation of map composition.

To connect S^k with its finite dimensional restriction we will use the projection operator

$$\begin{gathered} \pi_N : l^2(\mathbb{Z}) \to l^2(\mathrm{Q}_N) \\ x_n, n \in \mathbb{Z} \mapsto \left[\pi_N x\right]_m = \begin{cases} x_m, & m \in \mathrm{Q}_N \\ 0, & m \notin \mathrm{Q}_N, m \in \mathbb{Z} \end{cases} \end{gathered} \qquad (44)$$

Pertaining to the projection operator and the $l^2(\mathbb{Z})$ inner product $\langle \cdot, \cdot \rangle$ we have the following:

Fact: For any x in $l^2(\mathrm{Q}_N)$ and any y in $l^2(\mathbb{Z})$,

$$\langle x, y \rangle = \langle x, \pi_N y \rangle \qquad (45)$$

Proof: $\langle x, y\rangle = \sum_{k=-\infty}^{\infty} x_k y_k = \sum_{k=0}^{2N-1} x_k y_k = \sum_{k=-\infty}^{\infty} x_k[\pi_N y]_k = \langle x, \pi_N y\rangle.$ ■

We use fact (45) immediately to conclude that for any x in $l^2(\mathbf{Q}_N)$ and any y in $l^2(\mathbf{Z})$

$$\langle x, S^k y\rangle = \langle x, \pi_N S^k y\rangle \tag{46}$$

We will refer to the operator $\pi_N S^k$ restricted to $l^2(\mathbf{Q}_N)$ by S_N^k which is $l^2(\mathbf{Q}_N) \to l^2(\mathbf{Q}_N)$. Using the standard basis of $\mathfrak{R}^{2N}$ to represent $l^2(\mathbf{Q}_N)$ we have as the matrix representation of S_N^k,

$$[\mathbf{S}_N^k]_{i,j} = \delta_{i-k,j}, \quad i, j \in \mathbf{Q}_N \tag{47}$$

For example,

$$\mathbf{S}_2^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

It is clear that S_N^k is *nilpotent* for $k \neq 1$ meaning that there exists an $m \in \mathbf{Z}_+$ such that $(S_N^k)^m = 0$, the zero operator.

It is important to note that in general, for arbitrary $k, j \in \mathbf{Z}$, $S_N^j S_N^k \neq S_N^{j+k}$. For an example consider

$$S_N^{-2} S_N^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The reason that the naive application of the DWT to $l^\infty(\mathbf{Z}_p)$ does not work is the fact that $\pi_N S^j S^k \neq S_N^j S_N^k$.

The shift operator on $\mathcal{P}_p$ is circular. For integer k:

$$\begin{aligned} R_p^k &: \mathcal{P}_p \to \mathcal{P}_p \\ x_{n \bmod 2^p}, n \in \mathbb{Z} &\mapsto x_{(n-k) \bmod 2^p}, n \in \mathbb{Z} \end{aligned} \tag{48}$$

It too is a right shift. R_p^k, having finite dimensional domain and codomain, has the matrix representation with respect to the standard basis of $\mathcal{P}_p$

$$\left[\mathbf{R}_p^k\right]_{i,j} = \delta_{(i-k) \bmod 2^p, j}, \quad i,j \in \mathbb{Z}_p \tag{49}$$

Letting k and j be in $\mathbb{Z}$, with p *in* $\mathbb{Z}_+$ we have the following properties

$$\left(R_p^k\right)^{-1} = \left(R_p^k\right)^* = R_p^{-k} \qquad \textit{unitarity} \tag{50}$$
$$R_p^k R_p^j = R_p^j R_p^k \qquad \textit{commutativity} \tag{51}$$
$$R_p^j R_p^k = R_p^{j+k} \qquad \textit{group property} \tag{52}$$
$$\exists \eta \in \mathbb{Z} \ni \left(R_p^k\right)^\eta = R_p^k \tag{53}$$

We see that $\left\{R_p^k\right\}_{k \in \mathbb{Z}}$ is also a unitary group over composition of operators. If n is the smallest η satisfying (53) then R_p^k is said to be *n-potent*. For an example particularly relevant to our current context we have that for any p in $\mathbb{Z}_+$

$$\left(R_p^2\right)^{2^p/2} = I_p$$

For an example of a matrix representation of R_p^k we let $p = 2$ and $k = 1$ to get

$$\mathbf{R}_2^1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Filters for the DPWT

For our DPWT analysis and synthesis algorithms we need two filter operators at each scale p. We will call these linear operators $\tilde{L}_p^N : \mathcal{P}_p \to \mathcal{P}_{p-1}$ and $\tilde{H}_p^N : \mathcal{P}_p \to \mathcal{P}_{p-1}$. In order to have a discrete wavelet transform $\tilde{L}_p^N$ and $\tilde{H}_p^N$ must satisfy criteria (15), (16), (17), and (18) which we repeat here using $\tilde{L}_p^N$ and $\tilde{H}_p^N$.

$$\tilde{L}_p^N \tilde{L}_p^{N*} = I \tag{54}$$

$$\tilde{H}_p^N \tilde{H}_p^{N*} = I \tag{55}$$

$$\tilde{L}_p^N \tilde{H}_p^{N*} = 0 \tag{56}$$

$$\tilde{L}_p^{N*} \tilde{L}_p^N + \tilde{H}_p^{N*} \tilde{H}_p^N = I \tag{57}$$

The adjoint maps $\tilde{L}_p^{N*}$ and $\tilde{H}_p^{N*}$ are defined through the $\mathbf{P}_p$ inner product: For any $p \in \mathbf{Z}_+$ and for any pair of sequences c and d in $\mathbf{P}_p$,

$$\langle c, d \rangle_p \equiv \sum_{k=0}^{2^p-1} c_k \, d_k$$

With this inner product the matrix representations of $\tilde{L}_p^{N*}$ and $\tilde{H}_p^{N*}$ are the transposes of those of $\tilde{L}_p^N$ and $\tilde{H}_p^N$.

We choose to construct our filters as follows: We take two sequences l^N and h^N from $l^2(\mathbf{Q}_N) \subset l^2(\mathbf{Z})$ and wrap them onto $\mathbf{P}_p$ with $@_N^p$. Then we operate on the result with even rotations $R_N^{2i}, i \in \mathbf{Z}_{p-1}$. Thus $\tilde{l}_p^N$ and $\tilde{h}_p^N$ (21), (22) are defined by $\tilde{l}_p^N \equiv @_N^p \, l^N$ and $\tilde{h}_p^N \equiv @_N^p \, h^N$. We form $\tilde{L}_p^N$ with

$$\begin{aligned} \tilde{L}_p^N &: \mathbf{P}_p \to \mathbf{P}_{p-1} \\ & x \mapsto \left[\langle R_N^{2i} @_N^p l^N, x \rangle_p \right]_i, \quad i \in \mathbf{Z}_{p-1} \end{aligned}$$

and $\tilde{H}_p^N$ with

$$\begin{aligned} \tilde{H}_p^N &: \mathbf{P}_p \to \mathbf{P}_{p-1} \\ & x \mapsto \left[\langle R_N^{2i} @_N^p h^N, x \rangle_p \right]_i, \quad i \in \mathbf{Z}_{p-1} \end{aligned}$$

Given the forms of our operators $\tilde{L}_p^N$ and $\tilde{H}_p^N$, satisfaction of (54), (55), and (56) is equivalent to the following requirements on l^N and h^N for $i \in \mathbf{Z}_{p-1}$ and $j \in \mathbf{Z}_{p-1}$:

$$\left\langle R_p^{2i} @_N^p l^N, R_p^{2j} @_N^p l^N \right\rangle_p = \delta_{i,j} \tag{58}$$
$$\left\langle R_p^{2i} @_N^p h^N, R_p^{2j} @_N^p h^N \right\rangle_p = \delta_{i,j} \tag{59}$$
$$\left\langle R_p^{2j} @_N^p h^N, R_p^{2j} @_N^p l^N \right\rangle_p = 0 \tag{60}$$

Satisfaction of (57) follows from satisfaction of these three criteria with $\tilde{L}_p$ and $\tilde{H}_p$ being surjective. Using the definitions of $\tilde{l}_p^N$ and $\tilde{h}_p^N$ as well as the unitarity and group properties of R_p^k, and letting $k = j - i$ we have that $\forall \alpha, \beta \in \mathbf{P}_p$

$$\begin{aligned}\left\langle R_p^{2i} \alpha, R_p^{2j} \beta \right\rangle_p &= \left\langle \alpha, R_p^{2(j-i)} \beta \right\rangle_p \\ &= \left\langle \alpha, R_p^{2k} \beta \right\rangle_p\end{aligned}$$

This yields as equivalent forms of (58), (59), and (60) with $k \in \mathbf{Z}_{p-1}$:

$$\left\langle \tilde{l}_p^N, R_p^{2k} \tilde{l}_p^N \right\rangle_p = \delta_{k,0} \tag{61}$$
$$\left\langle \tilde{h}_p^N, R_p^{2k} \tilde{h}_p^N \right\rangle_p = \delta_{k,0} \tag{62}$$
$$\left\langle \tilde{l}_p^N, R_p^{2k} \tilde{h}_p^N \right\rangle_p = 0 \tag{63}$$

We contrast our maps $\tilde{L}_p^N$ and $\tilde{H}_p^N$ with the standard DWT maps L_p^N and H_p^N,

$$\begin{aligned}L_p^N &: l^2(\mathbf{Z}) \to l^2(\mathbf{Z}) \\ &x \mapsto \left[\left\langle S_N^{2i} l^N, x \right\rangle\right]_i, \quad i \in \mathbf{Z}\end{aligned}$$

$$\begin{aligned}H_p^N &: l^2(\mathbf{Z}) \to l^2(\mathbf{Z}) \\ &x \mapsto \left[\left\langle S_N^{2i} h^N, x \right\rangle\right]_i, \quad i \in \mathbf{Z}\end{aligned}$$

where we point out that unlike $\tilde{L}_p^N$ and $\tilde{H}_p^N$, the domains and codomains of L_p^N and H_p^N are infinite dimensional. Maps L_p^N and H_p^N satisfy criteria (15), (16), (17), and (18). Given the forms of L_p^N and H_p^N, criteria (15), (16), and (17) are equivalent to the following requirements on l^N and h^N for $i \in \mathbf{Z}$ and $j \in \mathbf{Z}$:

$$\langle S^{2i}l^N, S^{2j}l^N\rangle = \delta_{i,j} \tag{64}$$
$$\langle S^{2i}h^N, S^{2j}h^N\rangle = \delta_{i,j} \tag{65}$$
$$\langle S^{2i}h^N, S^{2j}l^N\rangle = 0 \tag{66}$$

Using the definition of the adjoint of a linear map as well as the unitarity of S^k, $k \in \mathbb{Z}$ we have that for any α^N and β^N in $l^2(\mathbf{Q}_N)$ and any integers i and j,

$$\begin{aligned}\langle S^{2i}\alpha^N, S^{2j}\beta^N\rangle &= \langle \alpha^N, (S^{2i})^* S^{2j}\beta^N\rangle \\ &= \langle \alpha^N, S^{-2i}S^{2j}\beta^N\rangle\end{aligned}$$

Applying the group property of S^k gives

$$= \langle \alpha^N, S^{2(j-i)}\beta^N\rangle$$

Fact (46) tells us that

$$= \langle \alpha^N, \pi_N S^{2(j-i)}\beta^N\rangle$$

and the definition of S_N^k gives

$$= \langle \alpha^N, S_N^{2(j-i)}\beta^N\rangle$$

Applying this and letting $k = j - i$ we obtain the alternative forms of (64), (65), (66) for $k \in \mathbb{Z}$:

$$\langle l^N, S_N^{2k}l^N\rangle = \delta_{k,0} \tag{67}$$
$$\langle h^N, S_N^{2k}h^N\rangle = \delta_{k,0} \tag{68}$$
$$\langle h^N, S_N^{2k}l^N\rangle = 0 \tag{69}$$

Thus (15), (16), and (17) are equivalent to (67), (68), and (69).

DPWT Filters from DWT Filters

We will prove that if l^N and h^N satisfy (67), (68), and (69), then they will satisfy (61), (62), and (63), but first we will need two lemmas.

Lemma : Let N and p be in $\mathbb{Z}_+$. Let $@_N^p$, R_p^k, and S_N^k be defined as above with $k \in \mathbb{Z}$. Then

$$@_N^{p*} R_p^k \, @_N^p = \sum_{i=-(2N-1)}^{2N-1} \delta_{(i-k)\bmod 2^p,0} \, S_N^i \tag{70}$$

Proof: By definition of $@_N^p$ (40) and R_p^k (48) we have

$$\left[R_p^k \, @_N^p\right]_{i,n} = \sum_{j \in \mathbf{Q}_N} \delta_{(i-k)\bmod 2^p,j} \, \delta_{j,n \bmod 2^p}, \quad i \in \mathbb{Z}_p, n \in \mathbf{Q}_N$$

Applying fact (36) gives us

$$= \delta_{(i-k)\bmod 2^p, n \bmod 2^p}, \quad i \in \mathbb{Z}_p, n \in \mathbf{Q}_N$$

Substituting this result into $@_N^{p*} R_p^k \, @_N^p$ yields

$$@_N^{p*} R_p^k \, @_N^p = @_N^{p*}\left[\delta_{(i-k)\bmod 2^p, n \bmod 2^p}\right], \quad i \in \mathbb{Z}_p, n \in \mathbf{Q}_N$$

We apply the definition of $@_N^p$ (40) again, along with the commutativity of the arguments of the Kronecker delta (33) to get

$$= \left[\sum_{i \in \mathbb{Z}_p} \delta_{m \bmod 2^p, i} \, \delta_{(i-k)\bmod 2^p, n \bmod 2^p}\right], \quad m, n \in \mathbf{Q}_N$$

and apply fact (35) to get

$$= \left[\sum_{i \in \mathbf{Z}_p} \delta_{m \bmod 2^p, i} \, \delta_{i \bmod 2^p, (n+k) \bmod 2^p} \right], \quad m, n \in \mathbf{Q}_N$$

For i restricted to $\mathbf{Z}_p$ we have that $i \bmod 2^p = i$. Using this and fact (37) we have

$$= \left[\delta_{m \bmod 2^p, (n+k) \bmod 2^p} \right], \quad m, n \in \mathbf{Q}_N$$

Another application of (35) gives

$$= \left[\delta_{(m-n) \bmod 2^p, k \bmod 2^p} \right], \quad m, n \in \mathbf{Q}_N$$

Since m and n are both in $\mathbf{Q}_N = \{0, \ldots, 2N - 1\}$, $m-n$ is in $\{-(2N-1), \ldots, -1, 0, 1, \ldots, 2N-1\}$. Using this we may apply (37) to get

$$= \left[\sum_{i=-(2N-1)}^{2N-1} \delta_{i \bmod 2^p, k \bmod 2^p} \, \delta_{m-n, i} \right], \quad m, n \in \mathbf{Q}_N$$

and another application of fact (35) brings us to

$$= \left[\sum_{i=-(2N-1)}^{2N-1} \delta_{(i-k) \bmod 2^p, 0} \, \delta_{m-n, i} \right], \quad m, n \in \mathbf{Q}_N$$

Now note that the summation is independent of m and n so we may bring the sum outside of the bracket to obtain

$$= \sum_{i=-(2N-1)}^{2N-1} \delta_{(i-k) \bmod 2^p, 0} \left[\delta_{m-n, i} \right], \quad m, n \in \mathbf{Q}_N$$

Finally we recognize from (47) that the m and n dependent term is just S_N^i and we have our proof. ■

From lemma (70) easily follows another

Lemma: For any $\alpha^N, \beta^N \in l^2(\mathbf{Q}_N)$

$$\left\langle @_p^N \alpha^N, R_p^{2k} @_p^N \beta^N \right\rangle_p = \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0} \left\langle \alpha^N, S_N^n \beta^N \right\rangle \qquad (71)$$

Proof: Application of the definition of the adjoint, properties of the projection π_N, and the definition of the inner product gives

$$\begin{aligned} \left\langle @_p^N \alpha^N, R_p^{2k} @_p^N \beta^N \right\rangle_p &= \left\langle \alpha^N, @_N^{p*} R_p^{2k} @_N^p \beta^N \right\rangle \\ &= \left\langle \alpha^N, @_N^{p*} R_p^{2k} @_N^p \pi_N \beta^N \right\rangle \\ &= \left(\alpha^N\right)^* @_N^{p*} R_p^{2k} @_N^p \left(\pi_N \beta^N\right) \end{aligned}$$

We now use lemma (70), some algebra, and the projection property (46) to get

$$\begin{aligned} &= \left(\alpha^N\right)^* \sum_{n=-(2N-1)}^{2N-1} \delta_{(i-2k) \bmod 2^p, 0}\, S_N^n \left(\pi_N \beta^N\right) \\ &= \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0} \left\langle \alpha^N, S_N^n \pi_N \beta^N \right\rangle \\ &= \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0} \left\langle \alpha^N, S^n \beta^N \right\rangle \end{aligned}$$

proving our assertion. ■

Now a straightforward lemma (71) will prove the following theorem:

Theorem: For $\tilde{l}_p^N \equiv @_N^p l^N$, $\tilde{h}_p^N \equiv @_N^p h^N$ with l^N and h^N in $l^2(\mathbf{Q}_N)$,

$$\forall k \in \mathbf{Z}_{p-1}, \left\langle l^N, S_N^{2k} l^N \right\rangle = \delta_{k,0} \;\Rightarrow\; \forall k \in \mathbf{Z}_{p-1}, \left\langle \tilde{l}_p^N, R_p^{2k} \tilde{l}_p^N \right\rangle_p = \delta_{k,0} \qquad (72)$$

$$\forall k \in \mathbf{Z}_{p-1}, \left\langle h^N, S_N^{2k} h^N \right\rangle = \delta_{k,0} \;\Rightarrow\; \forall k \in \mathbf{Z}_{p-1}, \left\langle \tilde{h}^N, R_p^{2k} \tilde{h}^N \right\rangle_p = \delta_{k,0} \qquad (73)$$

$$\forall k \in \mathbf{Z}_{p-1}, \left\langle h^N, S_N^{2k} l^N \right\rangle = 0 \;\Rightarrow\; \forall k \in \mathbf{Z}_{p-1}, \left\langle \tilde{h}^N, R_p^{2k} \tilde{l}_p^N \right\rangle_p = 0 \qquad (74)$$

Proof: We prove (72) by substituting l^N for α^N and β^N in lemma (71) to get for each $k \in \mathbf{Z}_{p-1}$

$$\left\langle @_N^p l^N, R_p^{2k} @_N^p l^N \right\rangle_p = \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0} \left\langle l^N, S_N^n l^N \right\rangle$$
$$= \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0}\, \delta_{n,0}$$
$$= \delta_{(-2k) \bmod 2^p, 0} = \delta_{(-k) \bmod 2^{p-1}, 0}$$

Since i and j are in $\mathbf{Z}_{p-1}$, k is in $\{-(2^{p-1}-1), \ldots, 0, \ldots, 2^{p-1}-1\}$. Thus, $\delta_{-k \bmod 2^{p-1}, 0} = \delta_{k,0}$.

By substitution of h^N for α^N and β^N into the above argument and using (68) we immediately obtain that for each $k \in \mathbf{Z}_{p-1}$

$$\left\langle @_N^p h^N, R_p^{2k} @_N^p h^N \right\rangle_p = \delta_{k,0}$$

proving (73).

Letting $\alpha^N = h^N$ and $\beta^N = l^N$ and applying (69) gives for each $k \in \mathbf{Z}_{p-1}$

$$\left\langle @_N^p h^N, R_p^{2k} @_N^p l^N \right\rangle_p = \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0} \left\langle h^N, S_N^n l^N \right\rangle$$
$$= \sum_{n=-(2N-1)}^{2N-1} \delta_{(n-2k) \bmod 2^p, 0}\, 0$$

so we have that (69) implies (63) proving (74).

This completes our proof. ■

An important consequence of the theorem is that we may use Daubechies' filter coefficients, which satisfy (67), (68), and (69) by construction, to obtain a basis of orthonormal periodic wavelets for $\mathcal{P}_p$ and the associated multiresolution decomposition of $\mathcal{P}_p$.

In the symbolism of this section we restate the DPWT with $\tilde{l}_p^N = @_N^p l^N$ and $\tilde{h}_p^N = @_N^p \tilde{h}^N$ as

$$\begin{cases} f_i^{p-1} = \left\langle R_N^{2i}\, \tilde{l}_p^N, f^p \right\rangle_p \\ w_i^{p-1} = \left\langle R_N^{2i}\, \tilde{h}_p^N, w^p \right\rangle_p \end{cases}, \quad i \in \mathcal{Z}_{p-1}$$

and

$$f^p = \sum_{i=0}^{2^p-1} \left(R_N^{2i}\, \tilde{l}_p^N\right)^* f_i^{p-1} + \left(R_N^{2i}\, \tilde{l}_p^N\right)^* w_i^{p-1}$$

VII. Discussion

Though the DPWT is perfectly invertible in any signal domain, it may be expected to be particularly convenient in the study of discrete and truly periodic functions such as sequences of measurements taken in a spatially closed loop, or strongly non-linear oscillations. Since the DPWT may be stopped at any level, it contains, as a special case, DWT implementations of the type presented in [PTVF]. The DPWT is simply stopped at the smallest level k such that $2^k \geq 2N$.

A consideration in applying the DPWT to non-periodic sequences is that the largest scale wavelet components, corresponding to the lowest level coefficients, wrap around the support of the sequence as if the sequence were periodic. This may be considered to be a problem in some applications though it is easily avoided by simply not taking the DPWT to the lowest levels.

Multidimensional Discrete Periodic Wavelets

The discrete periodic wavelet transform is easily extended to a separable transform in n dimensions. Conceptually the n-dimensional sequence is cast to an n-dimensional discrete torus and the DPWT is applied along each dimension in succession. For example, if the torus is a 2-torus and is represented by a matrix, the DPWT may be applied to first replace each row by its wavelet transform, and then replace each column of the resulting matrix by its wavelet transform. The creation of a non-separable transform from the DPWT is a subtler problem that we will not address.

Edge Effects

Highly irregular sequences give rise to high concentrations of energy in the wavelet coefficients at the higher levels. The small scale wavelets are needed in creating a convergent expansion at the irregularities. A consequence of this is that when the DPWT is applied to a sequence that has a difference between its endpoints that is large with respect to differences between all other pairs of successive points, then a relatively great deal of energy will appear at small scales in locations near the end points at those scales. Consider, however, the following economical and invertible transformation:

Assume that the number of sequence elements along each dimension $k \in \{1, \ldots, n\}$ is 2^{p_k}, where p_k is a positive integer. Let the sequence be represented by $[s(i_1, i_2, \cdots, i_n)]$ $i_k \in \mathbf{Z}_{p_k}$. For convenience define

$$s_k(i) \equiv s(i_1, \cdots, i_{k-1}, i, i_{k+1}, \cdots, i_n), \quad i \in \mathbf{Z}_{p_k}$$

Now, for each dimension k from 1 to n, replace $[s_k(i)]$, $i \in \mathbf{Z}_{p_k}$ by $[\tilde{s}_k(i)]_{i \in \mathbf{Z}_{p_k}}$ defined as

$$\left[s_k(0), s_k(2), \ldots, s_k(2^{p_k-1} - 2), s_k(2^{p_k-1}), s_k(2^{p_k-1} - 1), s_k(2^{p_k-1} - 3), \ldots, s_k(3), s_k(1)\right]$$

If, for instance, $s_k(m)$ for $m \in \{1, 2, 3, 4\}$ are *close* then the ends of $\tilde{s}_k$ will also be close. Thus, if the original multidimensional sequence has a suitable covariance structure in its interior, then the sequence created by this transformation will have a closely related covariance structure when considered as an n-torus, i.e. without endpoints or boundaries. It is doubtful that this rearrangement will provide any advantage in sets of sequences obtained from critically sampled continuous signals. But there are many useful signal domains, e.g. video images, where continuous signal sources are, on average, sampled well above their critical Nyquist rate.

Local Event Filtering

We illustrate the utility that the DPWT shares with the DWT in the selective filtering of local events. The sequence shown in the upper graph is a sequence of 256 samples of an electromyograph signal, all 256 of which are non-zero. The lower graph was obtained by applying the DPWT using a Daubechies filter corresponding to $N = 4$, setting to zero all those wavelet coefficients whose magnitudes were less than 0.24 times the maximum wavelet coefficient magnitude, and then applying the IDPWT to the resulting wavelet transform vector. The lower graph is represented by only 35 non-zero samples. The threshold value of 0.24 was chosen to be just small enough to capture the event centered around sample 8.

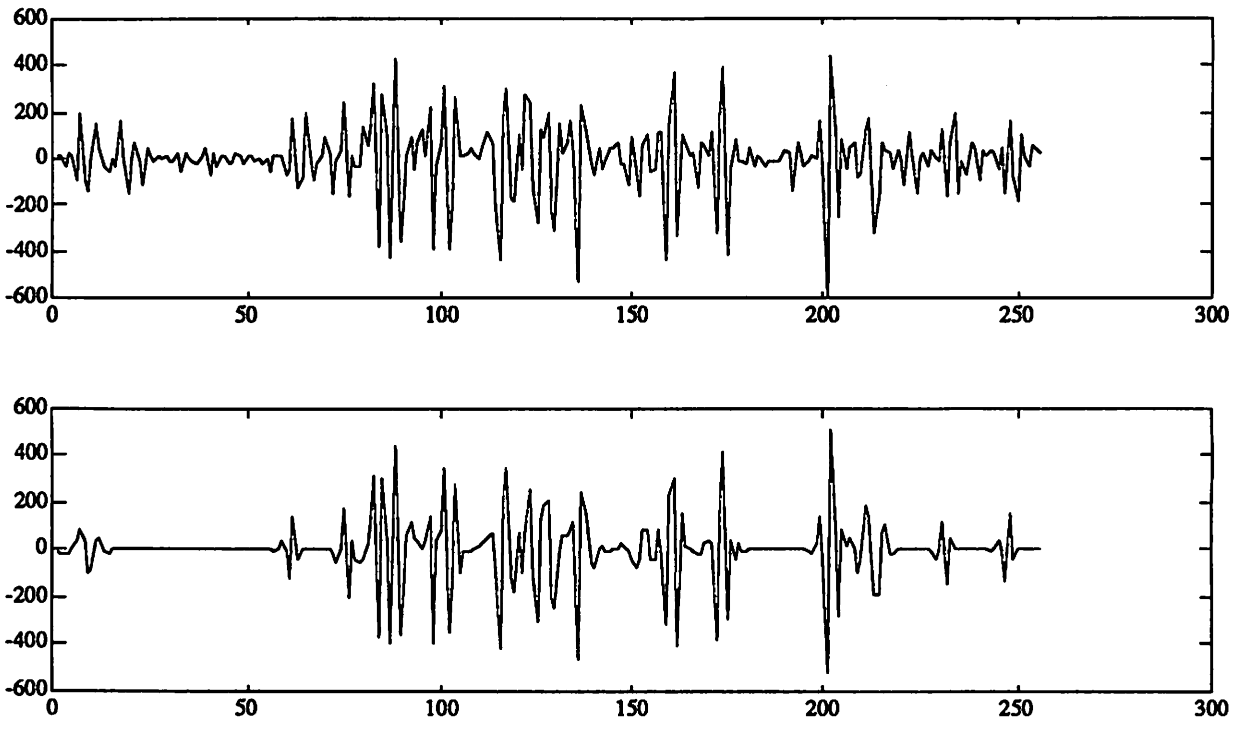

Figure 6: The top graph is a plot of 256 samples of a zero mean EMG signal. The bottom plot was obtained by applying the DPWT to the EMG signal, setting to zero all those wavelet coefficients whose magnitude fell below 0.24 times the maximum wavelet coefficient magnitude, and applying the IDPWT. Daubechies coefficients corresponding to $N = 4$ were used. The bottom plot was obtained using 35 non-zero coefficients.

We see that in the appropriate signal domain, the increase in coding efficiency can be dramatic. Of course the efficacy of this sort of filtering is best evaluated in the context of a specific application. It is clear, however, that the illustrated filtering technique can be of great utility in the reduction of computational load in the analysis of signals with localized features.

More Discrete Periodic Wavelet Bases

In figure 2 we showed the basis set for $\mathcal{P}_7$ corresponding to $N = 4$ that is associated with the DPWT. In figures 7a and 7b we display the basis sets of $\mathcal{P}_7$ corresponding to $N = 1$ and $N = 10$ respectively. We show only those basis elements corresponding to the 0 shift at each level.

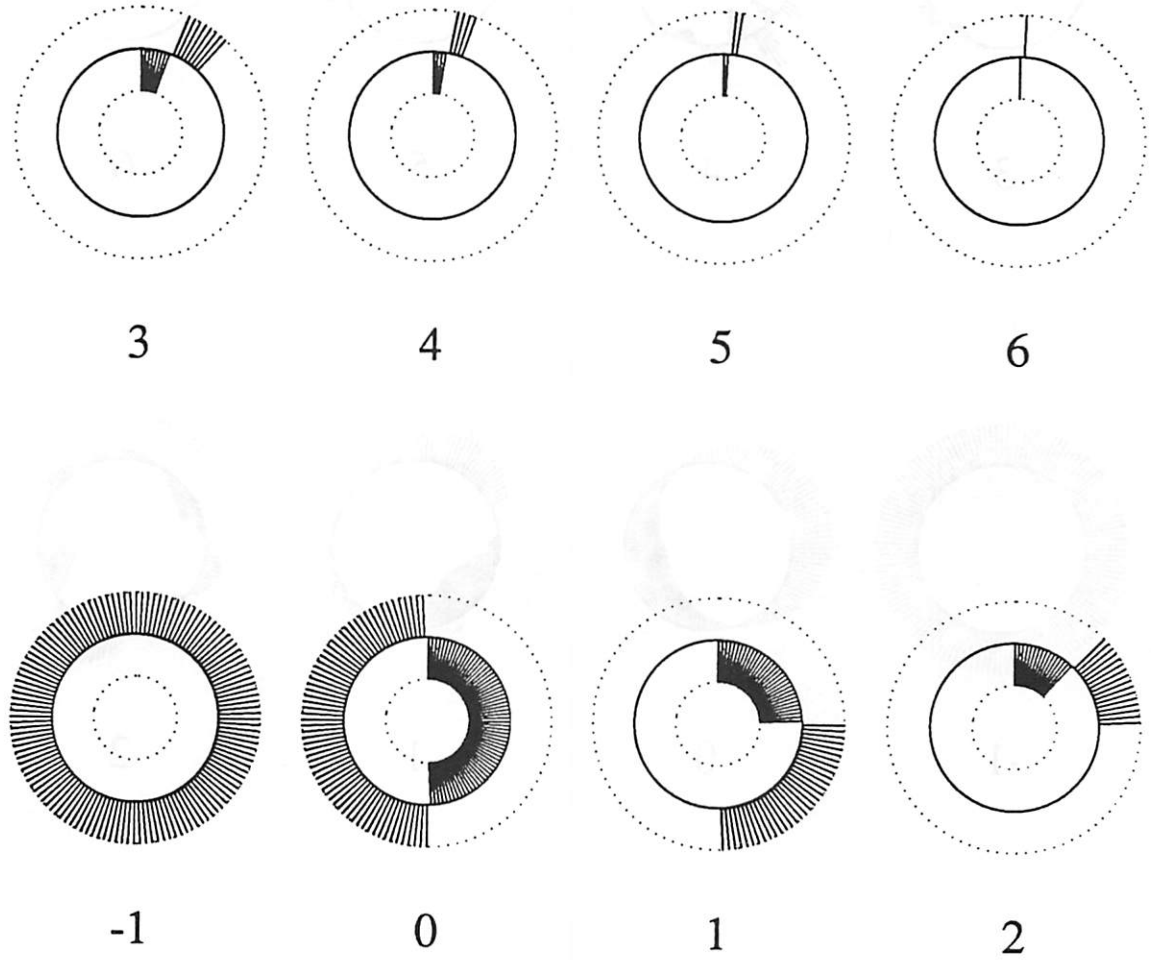

Figure 7a: Orthogonal basis of discrete periodic wavelet basis for periodic sequence of period 2^7 using Daubechies filters corresponding to $N = 1$. This is a Haar basis for the discrete circle. Basis elements have been normalized so that the peak value of each has magnitude 1. Only the shift 0 elements are shown.

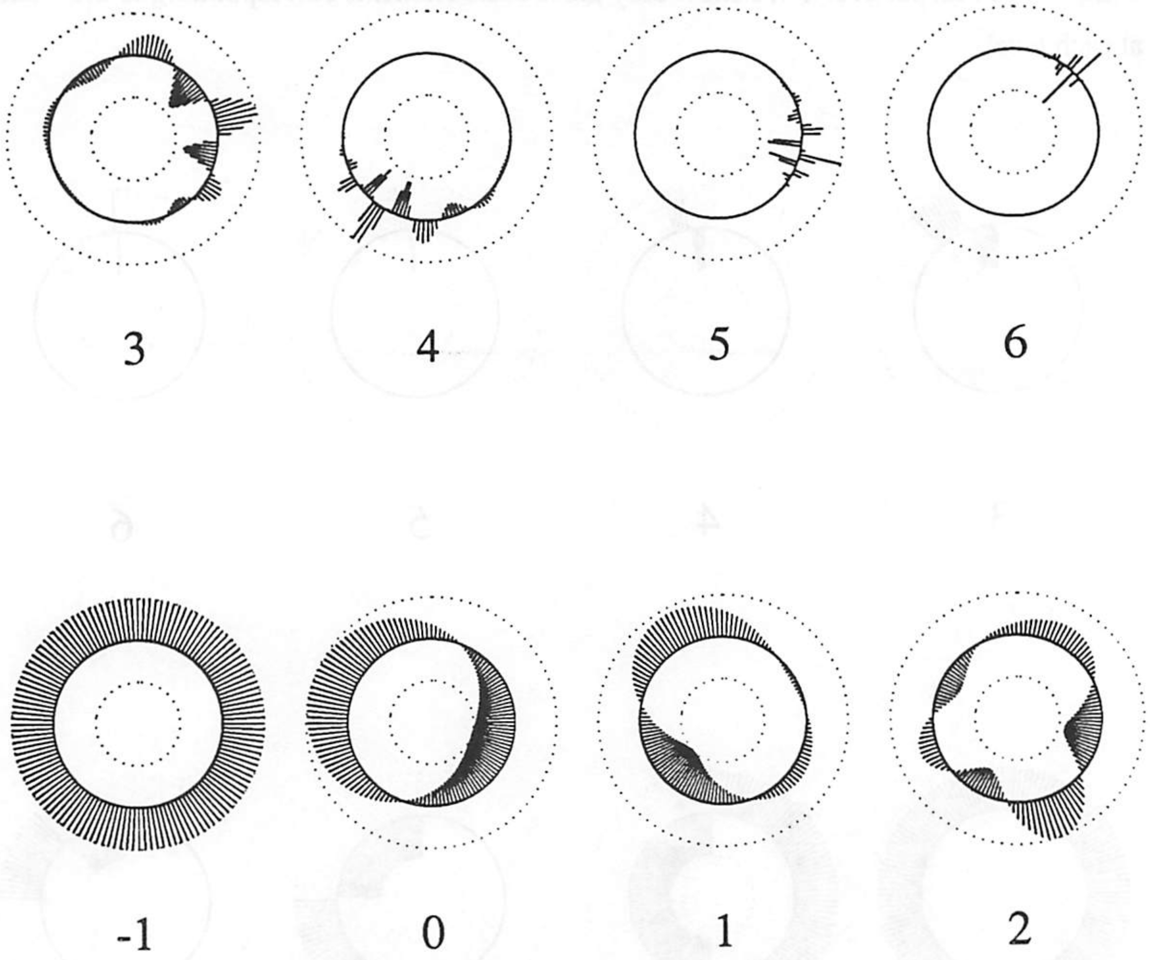

Figure 7b: Orthogonal basis of discrete periodic wavelets for periodic sequences of period 2^7 using Daubechies filters corresponding to N = 10. Only the shift 0 elements are shown.

In every case the basis elements corresponding to f^0 are seen to be the constant function.

Hybrid Discrete Periodic Wavelet Bases

At any level of the DPWT decomposition we are free to switch *N's* with the assurance that we will still produce an orthonormal basis. We call the result a *hybrid DPWT basis* and illustrate such a basis for $\mathcal{P}_7$ in figure 8 where the 0, 1, and 2 basis elements correspond to the Haar, or $N = 1$, wavelets, the 3 and 4 basis elements correspond to $N = 2$ wavelets, and the rest of the basis elements correspond to the $N = 8$ wavelets. Again, we show only the shift 0 elements of each level.

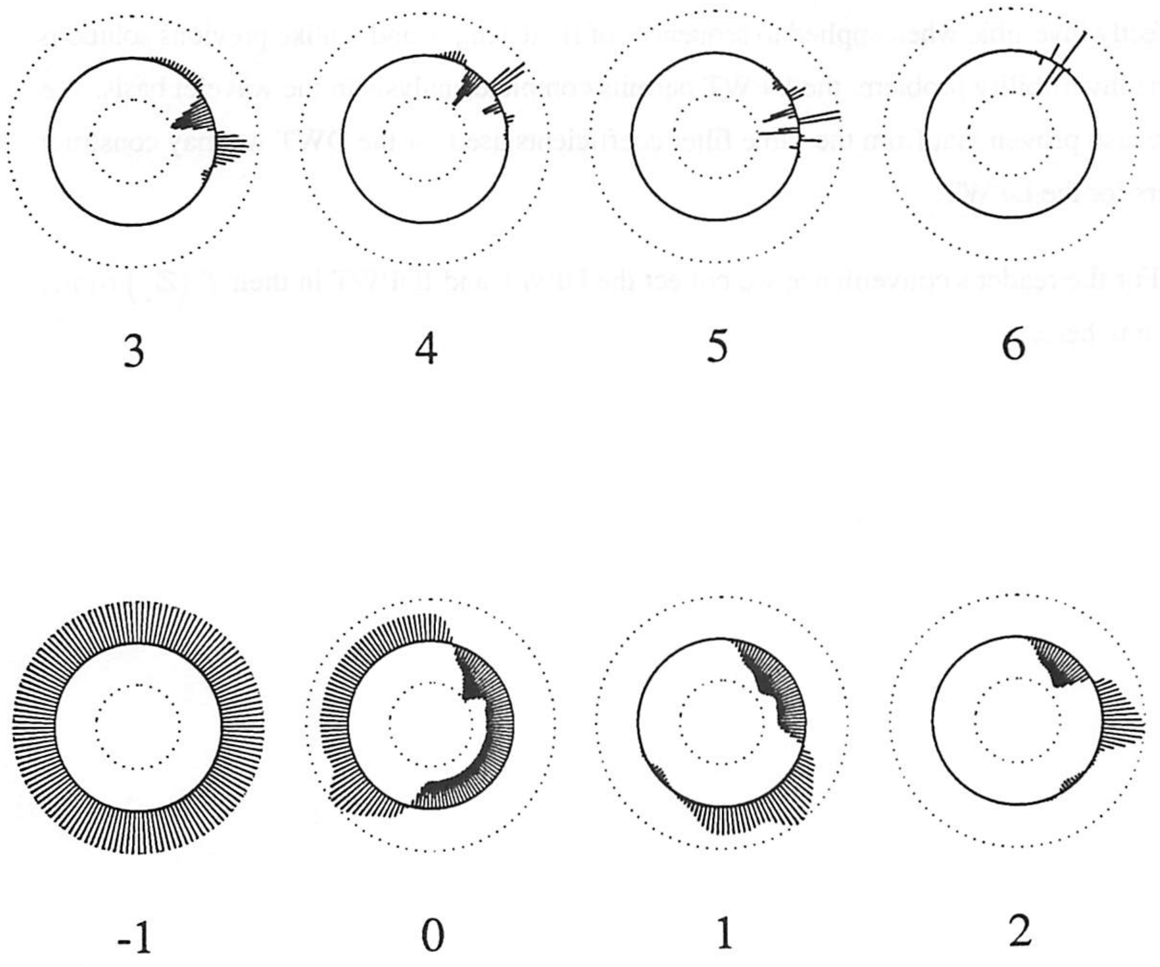

Figure 8: Hybrid DPWT basis for $\mathcal{P}_7$. The 0, 1, and 2 basis elements are $N = 1$ wavelets, the 3 and 4 basis elements are $N = 2$ wavelets, and the rest of the

basis elements are $N = 8$ wavelets. Daubechies coefficients have been used.

Such a hybrid basis may be useful in those situations where one must insist upon compactly supported wavelets at all levels but wishes a higher degree of smoothness where available.

VIII. Conclusions

We have introduced a discrete periodic wavelet transform and have shown that unlike the standard recursive realization of the orthogonal $l^2(\mathbb{Z}) \to l^2(\mathbb{Z})$ DWT, the DPWT is perfectly invertible when applied to sequences of finite length, and, unlike previous solutions to the invertibility problem, the DPWT permits complete analysis in the wavelet basis. We have also proven that from the same filter coefficients used for the DWT we may construct filters for the DPWT.

For the reader's convenience we collect the DPWT and IDPWT in their $l^\infty(\mathbb{Z}_p)$ forms in a box here.

$$\text{DPWT}: \begin{cases} f_i^{p-1} = \sum\limits_{n \in \mathcal{S}_N^p(i)} f_n^p \, \tilde{l}^N_{p,(n-2i) \bmod 2^p}, & i \in \mathbb{Z}_{p-1} \\ w_i^{p-1} = \sum\limits_{n \in \mathcal{S}_N^p(i)} f_n^p \, \tilde{h}^N_{p,(n-2i) \bmod 2^p}, & i \in \mathbb{Z}_{p-1} \end{cases}$$

$$\mathcal{S}_N^p(i) \equiv \{2i, 2i+1, \ldots, 2i + \min(2N, 2^p) - 1\} \bmod 2^p$$

$$\text{IDPWT}: \quad f_i^p = \sum_{k \in \mathcal{T}_N^p(i)} \tilde{l}^N_{p,(i-2k) \bmod 2^p} \, f_k^{p-1} + \tilde{h}^N_{p,(i-2k) \bmod 2^p} \, w_k^{p-1}, \quad i \in \mathbb{Z}_p$$

$$\mathcal{T}_N^p(i) \equiv \left\{ \left\lceil \frac{i+1}{2} \right\rceil - 1, \left\lceil \frac{i+1}{2} \right\rceil - 2, \ldots, \left\lceil \frac{i+1}{2} \right\rceil - \min(N, 2^{p-1}) \right\} \bmod 2^p$$

$$\tilde{l}^N_{p,i} \equiv \sum_{k=0}^{\lfloor (2N-1)/2^p \rfloor} l^N_{i \bmod 2^p + 2^p k}, \quad i \in \mathbb{Z}$$

$$\tilde{h}^N_{p,i} \equiv \sum_{k=0}^{\lfloor (2N-1)/2^p \rfloor} h^N_{i \bmod 2^p + 2^p k}, \quad i \in \mathbb{Z}$$

No one basis set can ever hope to be the *best* basis set in too wide a range of applications. The most that can be hoped for is that expansion of sequences in a domain of interest are easily obtained, meaningful, and brief. It is hoped that the discrete periodic wavelet transform may prove to be such a tool in a domain of interest to the reader.

Acknowledgments

The author is grateful to Charles Desoer, Adam Schwartz, and Shahram Shahruz for their thoughtful suggestions and comments, to Elise Burmeister and Steve Lehman for supplying the EMG signal used for figure 6, to Shankar Sastry for his encouragement and support, and to Avideh Zakhor and Philip Stark for introducing the author to wavelets.

This work was supported in part by NASA under grant NAG 2-243 and by NSF under grant IRI 90-14490.

References

[Daub1] I. Daubechies, "Orthonormal bases of compactly supported wavelets", *Commun. Pure Appl. Math.*, vol. 41, pp. 909-996, 1988.

[Daub2] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadelphia, PA, 1992.

[Hols] M. Holschneider, "Wavelet Analysis on the Circle", *J. Math. Phys.* 31 (1), January 1990.

[KV] J. Kovcacevic and M. Vetterli, "Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for $\mathbb{R}^n$", *IEEE Trans. Information Theory,* vol. 38, no. 2, pp. 533-555, March 1992.

[Matlab] Matlab, The MathWorks, Inc. Cochituate Place, 24 Prime Park Way, Natick, MA 01760.

[Meyer] Y. Meyer, *Ondelettes et Operateurs, Tome I.* Paris: Hermann, 1990

[RD] O. Rioul and P. Duhamel, "Fast Algorithms for Discrete and Continuous Wavelet Transforms", *IEEE Trans. Information Theory,* vol. 38, no. 2, pp. 569-586, March 1992.

[SFAH] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms", *IEEE Trans. Information Theory.,* vol. 38, no. 2, pp. 587-607, March 1992.

[SE] M. J. T. Smith and S. L. Eddins, "Analysis/Synthesis Techniques for Subband Image Coding", *IEEE Trans. on Acoustics, Speech, and Signal Proc.,* v. 38,

n. 8, pp. 1446-1456, August 1990.

[SE] M. J. T. Smith and S. L. Eddins, "Analysis/Synthesis Techniques for Subband Image Coding", *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, v. 38, n. 8, pp. 1446-1456, August 1990.

[Strang] G. T. Strang, "Wavelets and dilation equations: A brief introduction." *SIAM Rev.*, vol. 31, no. 4, pp. 614-627, Dec. 1989.

[Mallat] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Pattern Anal. Machine Intell.*, vol. 11, pp. 674-693, July 1989

[PTVF] W.H. Press, S. A. Teukolsky, W. T. Vetterling, and B. T. Flannery, Numerical Recipes in C : the art of scientific computing, 2nd edition, Cambridge ; New York : Cambridge University Press, 1992.

[VH] M. Vetterli and C. Herley, "Wavelets and filter banks: relationships and new results", *Proc. Int. Conf. Acoust., Speech, Signal Processing,* Albuquerque, NM, 1990, pp. 1723-1726.

Programs

Here we include a number of Matlab programs that will enable the curious reader to experiment with the DPWT and IDPWT. Those familiar with Matlab will note that these programs have not been optimized for speed in that for-loops have been used where they may have been avoided. The code has been designed to approximate pseudocode and correspond clearly with the DWT and DPWT as presented in the main part of the article. This should facilitate the translation of the algorithms to other programming languages or to hardware implementations. The reader who wishes faster, C-language realizations of the algorithms may contact the author.

As an example of how one might use the listed functions the following script is offered: Assume that the function daubcofs(N) returns Daubechies' low-pass filter coefficients of length $2N$. This short script performs the corresponding DPWT to obtain wavelet coefficient vector w, and then applies the IDPWT to w to obtain *ff*. The difference between each element of sequences *f* and *ff* are then calculated to obtain the reconstuction error *er*. The l^∞ and l^2 norm of the error is then calculated and displayed.

```
% Calculate reconstruction error.
rand('normal');
p = 5;
N = 8;
l = daubcofs(N);
f = rand(1,2^p);
w = dpwt(f,l);
ff = idpwt(w,l);
er = ff - f;
max(abs(er)), sum(er.^2)
```

File: dpwt.m

```
function w = dpwt(f, l, level)
% FUNCTION: dpwt
%
% SYNOPSIS: w = dpwt(f, l, [, level])
%
% DESCRIPTION:  This is the discrete periodic wavelet transform
%               using the low-pass filter corresponding to 'l'.
%               The DPWT transform is performed down to level 'level'.
%               The resulting 'w' has the same length as f. If level is
%               not specified, or if level = 0 then the complete DPWT
%               is calculated. Vector 'l' is assumed to consist of
%               the low-pass half of a perfect reconstruction
%               filter pair. The high-pass half 'h' is made from
%               'l'.
%
%               The returned row vector 'w' is w = [f0, w0, w1, ..., wp]
%               where f0 is length 1 and wp is a length 2^p row. These
%               are the wavelet coordinates of 'f'.
%
%               This routine does some argument testing and calculations
%               which are then passed on to the recursive function
%               r_dpwt() which actually implements the dpwt.
%
%               The length of  'f' must be an integer power of 2.
%               Filter sequence 'l' is not checked for satisfaction of
%               perfect reconstruction criteria. It must be of
%               even length.
%
%               For a faster C-language version of dpwt() and idpwt()
%               contact the author via email to
%               getz@robotics.Berkeley.EDU.
%
% NEEDS:        r_dpwt() which needs mod() and lh_tilde()
%
% SEE ALSO:     r_dpwt(), idpwt(), r_idpwt(), lh_tilde()
%
% REFERENCE:    N. Getz, "A Fast Discrete Periodic Wavelet Transform",
%               Electronics Research Laboratory, U.C. Berkeley, 1992.
%

% AUTHOR:       Neil Getz
%
% ORGANIZATION: University of California
%               Department of Electrical Engineering
%               and the Electronics Research Laboratory
%               Cory Hall
%               Berkeley, CA 94720
%
% DATE: 12-11-92
%
p = log(length(f))/log(2);
N = length(l)/2;

% Argument checking.
%
if(nargin < 2 | nargin > 3),
  help dpwt;
  return;
elseif(round(N) ~= N),
  error('Length of arg 2 must be an even integer.')
elseif(p ~= round(p)),
  error('Length of arg 1 must be an integer power of 2.')
elseif(nargin == 3),
  if((level > p)|(level < 0))
    error(['Level (arg 3) must not be greater than', ...
           ' log 2 of length of arg 1.'])
  end
end
%

MATSHIFT = 1;     % Matlab indices start at 1, not 0.

% Conjugate filter calculation
%
h = 1;            % For length.
for k = 0:2*N-1,
  h( MATSHIFT + k ) = (-1)^(MATSHIFT + k) * l(MATSHIFT + 2*N - 1 - k);
end
%

% If level is not specified then go all the way.
if(nargin == 2),
  level = 0;
end
%

% The beef.
w = r_dpwt(f, l, h, level);

% END dpwt().
```

File: idpwt.m

```
function f = idpwt(w,l,level)
% FUNCTION:     idpwt
%
% SYNOPSIS:     f = idpwt(w,l [,level])
%
% DESCRIPTION:  This is the inverse discrete periodic
%               wavelet transform using filter 'l'. The
%               wavelet coefficient vector is assumed
%               to be the product of a DPWT decomposition
%               down to level 'level'. The output
%               vector 'f' is the same size as 'w' whose
%               size should be an integer power of 2. Wavelet
%               coefficient vector w should have structure,
%               w  = [flevel, wlevel, ...w(p-1), wp]
%               where flevel and wlevel are the row vectors
%               that are the level 'level' decomposition
%               products of the DPWT applied to fp.
%
%               This routine does some argument testing and calculations
%               which are then passed on to the recursive function
%               r_idpwt() which actually implements the idpwt.
%
%               The length of 'w' must be an integer power of 2.
%               Filter sequence 'l' is not checked for satisfaction of
%               perfect reconstruction criteria. It must be of even
%               length.
%
%               For a faster C-language version of dpwt() and idpwt()
%               contact the author via email to
%               getz@robotics.Berkeley.EDU.
%
% NEEDS:        r_idpwt() which needs mod() and lh_tilde()
%
% SEE ALSO:     r_idpwt(), dpwt(), r_dpwt(), lh_tilde()
%
% REFERENCE:    N. Getz, "A Fast Discrete Periodic Wavelet Transform",
%               Electronics Research Laboratory, U.C. Berkeley, 1992.
%
% AUTHOR:       Neil Getz
%
% ORGANIZATION: University of California
%               Department of Electrical Engineering
%               and the Electronics Research Laboratory
%               Cory Hall
%               Berkeley, CA 94720
%
% DATE:         12-11-92
%
p = log(length(w))/log(2);
N = length(l)/2;

% Argument checking.
%
if(nargin < 2 | nargin > 3),
  help idpwt;
  return;
elseif(round(N) ~= N),
  error('Length of arg 2 must be an even integer.')
elseif(p ~= round(p)),
  error('Length of arg 1 must be an integer power of 2.')
elseif(nargin == 3),
  if((level > p)|(level < 0))
    error(['Level (arg 3) must not be greater than', ...
           ' log 2 of length of arg 1.'])
  end
end
%

MATSHIFT = 1;      % Matlab indices start at 1, not 0.

% Conjugate filter calculation
%
h = l;      % For length.
for k = 0:2*N-1,
  h( MATSHIFT + k ) = (-1)^( MATSHIFT + k ) * l(MATSHIFT + 2*N - 1 - k);
end
%

% If level is not specified then go all the way.
if (nargin == 2),
  level = 0;
end
%

% The beef
f = r_idpwt(w, l, h, level);

% END idpwt().
```

File: r_dpwt.m

```
function w = r_dpwt(f, l, h, level)
% FUNCTION:     r_dpwt
%
% SYNOPSIS:     w = r_dpwt(f, l, h, level)
%
% DESCRIPTION:  This is the recursive portion of the
%               discrete periodic wavelet transform using
%               filters 'l' and 'h', which are assumed to be
%               a perfect reconstruction filter pair of even
%               length. Decomposition is performed to level
%               'level'. For a description of the functionality
%               of dpwt see the documentation for dpwt() or
%               enter "help dpwt".
%
%               The returned row vector 'w' is w = [f0, w0, w1, ..., wp]
%               where f0 is length 1 and wp is a length 2^p row. These
%               are the wavelet coordinates of 'f'.
%
%               The length of 'f' must be an integer power of 2.
%               Filter sequence 'l' is not checked for satisfaction of
%               perfect reconstruction criteria. It must be of
%               even length.
%
%               For a faster C-language version of dpwt() and idpwt()
%               contact the author via email to
%               getz@robotics.Berkeley.EDU.
%
% NEEDS:        mod(), lh_tilde()
%
% REFERENCE:    N. Getz, "A Fast Discrete Periodic Wavelet Transform",
%               Electronics Research Laboratory, U.C. Berkeley, 1992.
%

% AUTHOR:       Neil Getz
%
% ORGANIZATION: University of California,
%               Department of Electrical Engineering
%               and the Electronics Research Laboratory
%               Cory Hall
%               Berkeley, CA 94720
%
% DATE:         12-11-92
%
```

```
lenf = length(f);

if (lenf == 2^level),
  % Then we've hit bottom.
  w = f;
else,
  % Haven't hit bottom yet.
  MATSHIFT = 1;  % matlab indices start at 1, not 0.

  % Initialization and local constants
  %
  lenfpml = lenf/2;      % Saves a couple of divisions.
  fpml = zeros(1,lenfpml);
  wpml = fpml;   % For size
  Nt2 = length(l);

  % Make DPWT filters l_tilde and h_tilde
  %
  [l_tilde,h_tilde] = lh_tilde(l, h, lenf);
  %

  % The DPWT calculations.
  %
  for i = 0:lenfpml - 1,
    two_i = 2*i;  % Saves a couple of multiplications.
    for n = mod([two_i:(two_i + min(lenf,Nt2) - 1)],lenf),
      dexmod = mod(n - two_i, lenf);
      fpml(MATSHIFT + i) = fpml(MATSHIFT + i) ...
                           + l_tilde(MATSHIFT + dexmod)*f(MATSHIFT + n);
      wpml(MATSHIFT + i) = wpml(MATSHIFT + i) ...
                           + h_tilde(MATSHIFT + dexmod)*f(MATSHIFT + n);
    end
  end
  %

  % The recursion
  %
  w = [r_dpwt(fpml,l,h,level), wpml];
  %

end

% END of r_dpwt().
```

File: r_idpwt.m

```
function fp = r_idpwt(w, l, h, level)
% FUNCTION:     r_idpwt
%
% SYNOPSIS:     fp = r_idpwt(w, l, h, level)
%
% DESCRIPTION:  This is the recursive portion of the inverse
%               discrete periodic wavelet transform using filters
%               'l' and 'h', which are assumed to be a perfect
%               reconstruction filter pair of even length.
%               Decomposition is performed to level 'level'.For a
%               description of the functionality of dpwt see
%               the documentation for dpwt() or enter "help dpwt".
%
%               Wavelet coefficient vector w has structure,
%               w  = [flevel, wlevel, ...w(p-1), wp]
%               where flevel and wlevel are the row vectors
%               that are the level 'level' decomposition
%               products of the DPWT applied to fp. The length
%               of 'w' must be an integer power of 2. Filter
%               sequence 'l' is not checked for satisfaction of
%               perfect reconstruction criteria. It must be of even
%               length.
%
%               For a faster C-language version of dpwt() and idpwt()
%               contact the author via email to
%               getz@robotics.Berkeley.EDU.
%
% NEEDS:        mod(), lh_tilde()
%
% REFERENCE:    N. Getz, "A Fast Discrete Periodic Wavelet Transform",
%               Electronics Research Laboratory, U.C. Berkeley, 1992.
%

% AUTHOR:       Neil Getz
%
% ORGANIZATION: University of California
%               Department of Electrical Engineering
%               and the Electronics Research Laboratory
%               Cory Hall
%               Berkeley, CA 94720
%
% DATE:         12-11-92
%
```

```
lenw = length(w); % lenw is 2^p;

if(lenw == 2^level)
  % Then we've hit bottom.
  fp = w;
else,
  % Haven't hit bottom yet.
  MATSHIFT = 1;  % Matlab indices start at 1, not 0.

  % Initialization
  %
  fp = zeros(1,lenw);        % Place to put result.
  halflenw = lenw/2; % Saves a couple of divisions. This is 2^(p-1)
  N = length(l)/2;

  %
  % DPWT filters l_tilde and h_tilde
  %
  [l_tilde,h_tilde] = lh_tilde(l, h, lenw);
  %

  % The recursion.
  %
  fpml = r_idpwt(                                          ...
           w( (MATSHIFT + 0):(MATSHIFT + halflenw - 1) ), ...
              l, h, level                                  ...
         );
  %

  % The IDPWT calculations
  %
  for i = 0:lenw - 1,
    firstcol = ceil((i+1)/2)-1;
    for k = mod(                                                 ...
              [firstcol:-1:(firstcol - min([halflenw, N]) + 1)], ...
              halflenw                                           ...
            ),
      dexmod = mod(i-2*k,lenw);
      fp(MATSHIFT + i) = fp(MATSHIFT + i) ...
                         + l_tilde(MATSHIFT + dexmod)     ...
                           * fpml(MATSHIFT + k)           ...
                         + h_tilde(MATSHIFT + dexmod)     ...
                           * w(MATSHIFT + halflenw + k    ...
      );
    end
  end
  %
```

```
end

% END of r_idpwt().
```

File: lh_tilde.m

```
function [ l_tilde, h_tilde ] = lh_tilde(l, h, two2p)
% FUNCTION: lh_tilde
%
% SYNOPSIS: [ l_tilde, h_tilde ] = lh_tilde(l, h, two2p)
%
% DESCRIPTION:  Make DPWT filters l_tilde and h_tilde.
%
%               The vectors 'l' and 'h' are assumed to be
%               the values of the finite impulse response
%               low-pass and high-pass filters over their interval
%               of support {0,1,...,length(l)-1}. The filters
%               are not checked to see if they form a perfect
%               reconstruction pair. Nor are they checked for
%               common length though this is assumed.
%
% NOTE:         For more speed in dpwt() and idpwt() these could be
%               precalculated and either passed as function
%               parameters or made globally available.
%
% AUTHOR:       Neil Getz
%
% ORGANIZATION: University of California
%               Department of Electrical Engineering
%               and the Electronics Research Laboratory
%               Cory Hall
%               Berkeley, CA 94720
%
% DATE:         12-11-92

MATSHIFT = 1;  % matlab indices start at 1, not 0.

Nt2 = length(l); % Saves some multiplications.
if(Nt2~=length(h)),
  error('Arg 1 and arg 2 must be the same length.');
end

if ( length(l) <= two2p ),
  % In this case 2N<=2^p, so the first 2N taps
  % of l_tilde and h_tilde are identical to
  % those of l and h, respectively.
  l_tilde = l;
  h_tilde = h;
else,
  % Here we have 2N>2^p and we need to apply a wrap.
  l_tilde = zeros(1, two2p);
  h_tilde = l_tilde; % for size
  for i = 0:min([Nt2,two2p])-1,
    imod2to_p = i - two2p*floor(i/two2p);
    for k = 0:floor((Nt2-1-imod2to_p)/two2p),
      l_tilde( MATSHIFT + i ) ...
        = l_tilde(  MATSHIFT + i )  ...
          + l(  MATSHIFT + imod2to_p + k*two2p);
      h_tilde( MATSHIFT + i ) ...
        = h_tilde( MATSHIFT + i )  ...
          + h( MATSHIFT + imod2to_p + k*two2p);
    end
  end
end
```

File: mod.m

```
function y = mod(x,n)
% FUNCTION:    mod
%
% SYNOPSIS:    y = mod(x,n)
%
% DESCRIPTION: mod(x,n) gives the remainder on division
%              of x by n. The result always has the same
%              sign as n. This function is essentially just
%              an alias that makes the other functions easier
%              to read.
%

% AUTHOR:      Neil Getz
% DATE:        10-17-92
y = x - n*floor(x/n);
```