MATLAB Programs for Generating Orthonormal Wavelets

B.G. SHERLOCK and Y.P. KAKAD
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC 28223
USA

Abstract: - This paper presents MATLAB programs for generating the coefficients of the lowpass analysis filter corresponding to orthonormal wavelet analyses. One of the programs generates the famous Daubechies maxflat wavelets, and a second generates the Daubechies complex symmetric orthonormal wavelets. The remaining two programs generate the space of all orthonormal wavelets in terms of parameterizations whereby the space of wavelets of a given length $2N$ is generated by N parameters. This software should prove useful where it is desired to perform an optimization to obtain the best wavelet for a given application.

Key-Words: - Wavelets, Orthonormal, Software, Parameterization

1 Introduction

Wavelet analysis has been widely used in recent years in a variety of application areas. The reason for this success is that wavelet analysis provides a mixed time-frequency representation that gives a useful compromise between analysis purely in terms of frequency (as exemplified by the Fourier transform) and purely time-domain analysis.

A further advantage is that, unlike the Discrete Fourier Transform, the Discrete Wavelet Transform (DWT) is not unique – the user may choose between an infinite number of different wavelets, each providing its own unique corresponding DWT. This opens up the possibility of performing an optimization procedure in order to obtain the "best" wavelet for any given application according to some given optimality criterion. In order to perform this optimization, it is necessary to have a parameterization of the space of wavelets under consideration. The optimization cost function will then be passed the values of the parameters for each wavelet as the optimization proceeds. One of the authors has successfully used this approach to obtain optimal wavelets for the compression of fingerprint images [7].

The main disadvantage of orthonormal wavelet analysis is that symmetry of real-valued filters cannot be achieved. Symmetry is of great importance because of the linear phase exhibited by such filters, and also because symmetry permits the use of a reflective data extension that substantially reduces edge effects. The usual way of obtaining linear phase filters within the context of a wavelet analysis is to relax the requirement of orthonormality, and this has led to the widespread use of biorthogonal wavelets [1]. Linear phase is obtained at a cost, however: the DWT using biorthogonal wavelets does not preserve energy, i.e. the Parseval relation is no longer valid.

It is possible to obtain linear phase without sacrificing orthonormality if we allow the coefficients of the filter to be complex. In the paper we show how the parameterizations can be used to generate spaces of *symmetric* orthonormal wavelets.

Two further MATLAB programs are included. One of these generates the lowpass filter coefficients for the famous Daubechies maxflat wavelets, and the other for the Daubechies maxflat symmetric complex wavelets.

2 The Space of Orthonormal Wavelets

In this section, software is presented to generate the coefficients for the scaling filters corresponding to all orthonormal wavelets, including wavelets with complex filter coefficients. To do this, a parameterization of the space of orthonormal wavelets is required. We make use of two parameterizations; each of these generates scaling filter coefficents from a given set of parameter values. To generate the space of all orthonormal wavelets with real-valued coefficients, the parameterization of Section 2.1 requires the parameters to be angles in the range 0 to 2π, whereas that of Section 2.2 allows the parameters to have any real values. Both parameterizations produce the space of orthonormal wavelets with complex coefficients if the parameters are given complex values. The major advantage of complex orthonormal wavelets over real orthonormal wavelets is that linear phase can be achieved.

2.1 Generating Wavelets Using Angle Parameters

Let the coefficients of the scaling filter be denoted by $\{h_i\}$, with z-transform $H(z)=\sum_i h_i z^{-i}$. Vaidyanathan [9] proposed the following parameterization of the space of all perfect-reconstruction two-channel filter banks of length 2*M* in terms of *M* angle parameters $\{\theta_0,\theta_1,\ldots,\theta_{M-1}\}$:

$$H(z)=\begin{pmatrix}1 & 0\end{pmatrix}R_0\left(\prod_{i=1}^{M-1}\Lambda R_i\right)\begin{pmatrix}1\\ z^{-1}\end{pmatrix}\text{, where}$$

$$\Lambda=\begin{pmatrix}1 & 0\\ 0 & z^{-1}\end{pmatrix}\text{ and }R_i=\begin{pmatrix}c_i & s_i\\ -s_i & c_i\end{pmatrix}\text{, with}$$

$c_i=\cos\theta_i$ and $s_i=\sin\theta_i$.

Sherlock and Monro [6] derived from this recursive formulae expressing the coefficients $\{h_i^{(k+1)}\}$ for a filter of length 2(*k*+1) in terms of the coefficients $\{h_i^{(k)}\}$ for a filter of length 2*k*. They obtained, for the even numbered filter coefficients $\{h_{2i}\}$:

$$\begin{cases}h_0^{(k+1)} &= c_k h_0^{(k)}\\ h_{2i}^{(k+1)} &= c_k h_{2i}^{(k)}-s_k h_{2i-1}^{(k)}\quad \text{for } i=1,\ 2,\ \ldots,\ k-1\\ h_{2k}^{(k+1)} &= -s_k h_{2k-1}^{(k)}\end{cases}$$

with $h_0^{(1)}=c_0$ and $h_1^{(1)}=s_0$. The corresponding formula for the odd numbered filter coefficients $\{h_{2i+1}\}$ is:

$$\begin{cases}h_1^{(k+1)} &= s_k h_0^{(k)}\\ h_{2i+1}^{(k+1)} &= s_k h_{2i}^{(k)}+c_k h_{2i-1}^{(k)}\quad \text{for } i=1,2,\ldots,k-1\\ h_{2k+1}^{(k+1)} &= c_k h_{2k-1}^{(k)}\end{cases}$$

The recurrence of the above equations can be elegantly expressed in the form of the flow diagram of Figure 1. In this diagram a filter of any desired even length 2*M* can be produced by choosing *M* parameters θ_i (with $\sum_i \theta_i=\frac{\pi}{4}$ if regularity is desired) and passing through the flow diagram, terminating after *M*-1 stages of processing.

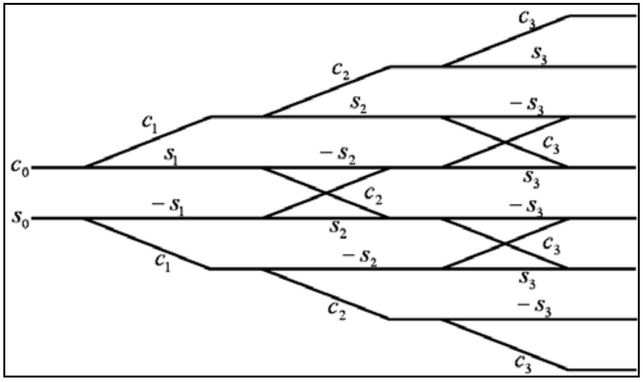

Figure 1. Flow diagram illustrating generation of orthonormal wavelets from angle parameters.

An efficient implementation in MATLAB is given in Figure 2.

```
function h = orthogen(alpha);
% Constructs an array, h(1...N) of lowpass
% orthonormal FIR filter coefficients for any
% even N>=2. The input array, alpha(1...N/2)
% gives N/2 free parameters that are angles
% in radians. If the angles sum to pi/4 the
% filter corresponds to a regular wavelet.
   N  = 2*length(alpha);
   h  = zeros(1, N);
   lo = N/2;
   hi = lo + 1;
   h(lo) = cos(alpha(1));
   h(hi) = sin(alpha(1));
   nstages = N/2;

   for stage = 1 : nstages-1
      c = cos(alpha(stage+1));
      s = sin(alpha(stage+1));
      h(lo-1) = c*h(lo);
      h(lo)   = s*h(lo);
      h(hi+1) = c*h(hi);
      h(hi)   = -s*h(hi);
      nbutterflies = stage-1;
      butterflybase = lo+1;
      for butterfly = 1 : nbutterflies
         hlo = h(butterflybase);
         hhi = h(butterflybase+1);
         h(butterflybase)  = c*hhi - s*hlo;
         h(butterflybase+1)= s*hhi + c*hlo;
         butterflybase = butterflybase + 2;
      end;
      lo = lo - 1;
      hi = hi + 1;
   end;
```

Figure 2. MATLAB function for generating orthonormal wavelets from angle parameters.

This parameterization is, however, a redundant representation because of inherent symmetries in our parameter space. It is shown in [6] that the filter coefficients will be unchanged if any even number of the θ_i are changed by π. The space can therefore be fully covered by choosing θ_0 from the interval [0,2π) and θ_1 through θ_{M-1} from [0,π).

When generating wavelets, however, regularity, equivalent to $\sum_{i=0}^{M-1}\theta_i=\frac{\pi}{4}$, is also required. With this additional constraint, it is shown in [6] that the space of all orthonormal wavelets of length 2*M* is covered by choosing *M*-1 parameters θ_i from the interval [0,π), and calculating the final one from the regularity constraint.

For a filter of length 2*M* corresponding to an orthonormal wavelet, the transformation

$$\theta_0' = \frac{\pi}{2} - \theta_0,$$

$$\theta_i' = \pi - \theta_i, \quad i = 1,2,\ldots,M-2$$

transforms the filter into its time-reverse. Therefore, half of the parameter space corresponds to wavelets which are time-reversals of the other half. A subspace of the parameter space which covers all orthonormal wavelets but excludes their time reversals is given (for length-2*M* wavelets) by choosing one parameter from the interval [0,π/2), *M*-2 parameters [0,π), and calculating the remaining parameter from the regularity constraint.

The parameterization can be extended to produce filters with complex-valued coefficients, simply by allowing the θ_i to be complex. However, the parameterization of Section 2.2 turns out to be better suited to this purpose.

2.2 Generating Wavelets Using Real-Valued Parameters

The parameterization used in this section is based upon the work of Pollen [5] and the subsequent results of Lina [3,4]. As in Section 2.1, let the coefficients of the scaling filter be denoted by $\{h_i\}$, with z-transform $H(z) = \sum_i h_i z^{-i}$. Then the parameterization of the space of all regular perfect-reconstruction two-channel filter banks of length 2*M+2* in terms of *M* parameters $\{\nu_0, \nu_1, \ldots, \nu_{M-1}\}$ is produced as described below.

Define 2-by-2 matrices $G_i(z)$ by

$$G_i(z) = \begin{pmatrix} u_i(z) & w_i(z) \\ -\overline{w_i(z)} & \overline{u_i(z)} \end{pmatrix}, \quad i=0,\ldots,M\text{-}1.$$

where $u_i(z) = 1 + \dfrac{\nu_i \overline{\nu}_i}{1+\nu_i \overline{\nu}_i}(z^{-1} - 1)$

and $w_i(z) = \dfrac{\nu_i}{1+\nu_i \overline{\nu}_i}(z-1)$.

Then form $G(z) = G_0 \overline{G}_1^T G_2 \overline{G}_3^T \ldots G_{M-2} \overline{G}_{M-1}^T$, where the notation $\overline{G}^T$ represents the conjugate transpose. Note that conjugate transposition is applied to every second matrix in the above product.

Finally, the filter coefficients in this parametrization are obtained as

$$H(z) = \tfrac{1}{2}\begin{pmatrix} 1 & -1 \end{pmatrix} G(z^2) \begin{pmatrix} z \\ -1 \end{pmatrix}.$$

Because, just as in Section 2.1, the filter is produced by a successive product of matrices, it is also possible to express the present parameterization in the form of a recurrence. Lina and Mayrand [3] presented recursive formulae expressing the coefficients $\{h_i^{(k+1)}\}$ for a filter of length 2(*k*+1) in terms of the coefficients $\{h_i^{(k)}\}$ for a filter of length 2*k*. They obtained, for the even numbered filter coefficients $\{h_{2i}\}$:

$$h_{2i}^{(k+1)} = \frac{1}{1+\nu_k \overline{\nu}_k}\left(\overline{h}_{2i}^{(k)} - \overline{\nu}_k \overline{h}_{2i-1}^{(k)} + \overline{\nu}_k \overline{h}_{2i+1}^{(k)} + \nu_k \overline{\nu}_k \overline{h}_{2i-2(-1)^k}^{(k)}\right),$$

and for the odd numbered filter coefficients $\{h_{2i+1}\}$:

$$h_{2i+1}^{(k+1)} = \frac{1}{1+\nu_k \overline{\nu}_k}\left(\overline{h}_{2i+1}^{(k)} - \nu_k \overline{h}_{2i}^{(k)} + \nu_k \overline{h}_{2i+2}^{(k)} + \nu_k \overline{\nu}_k \overline{h}_{2i+1+2(-1)^k}^{(k)}\right).$$

The recurrence is started with the Haar length-2 wavelet, $h_0^{(1)} = h_1^{(1)} = \frac{1}{2}$; then regularity of the wavelet is preserved throughout the iterative process irrespective of the choice of ν_i. An advantage over the parameterization in Section 2.1 is that this parameterization does not exhibit the redundancies that occurred in Section 2.1 due to the angular nature of the θ_i parameters. Here, the space of all orthonormal regular wavelets with real coefficients is simply produced by allowing the ν_i to range over all real values.

Furthermore, restricting all ν_i to be pure imaginary generates the space of all symmetric complex orthonormal wavelets. These wavelets are important because they are the only orthonormal

```
 function h=nuorthgen(nus)
% Generates all orthonormal regular wavelets.
% nus is an array of parameters satisfying:
%  nus all real => space of all real
%        orthonormal regular wavelets
%  nus all imaginary => space of all
%        symmetric complex orthonormal
%        regular wavelets
%  nus complex => space of all orthonormal
%        regular wavelets
h0 = 1/sqrt(2);
h1 = 1/sqrt(2);
h=[h0 h1] ;
stages=length(nus);
for stage=1:stages
   nu=nus(stage);
   nuc=conj(nu);
   N=length(h)+2;
   h=[0 0 conj(h) 0 0];
   for i=2:2:N+1
      newh(i)= h(i)-nu*h(i-1)+nu*h(i+1)+ ...
                 nu*nuc*h(i+2);
      i1=i+1;
      newh(i1)=h(i1)-nuc*h(i1-1)+ ...
                nuc*h(i1+1)+ nu*nuc*h(i1-2);
   end
   h=newh(2:N+1)/(1+nu*nuc);
end
```

Figure 3. MATLAB function for generating orthonormal wavelets using parameterization of Section 2.2.

wavelets that exhibit linear phase. Linear phase is highly valuable in many applications because the position of details is preserved in the filtered signal.

An efficient implementation in MATLAB is given in Figure 3.

3 Daubechies Wavelets

3.1 Generating Daubechies Maxflat Wavelets

The Daubechies maxflat wavelet of length 2*N* is the unique length 2*N* wavelet whose frequency response has both minimum phase and also exhibits *N* zero derivatives at $\omega = 0$ [2]. These wavelets are widely used because of their regularity. The wavelet is obtained by a spectral factorization of the following halfband frequency response [8]:

$$P(\omega) = 2\left(\frac{1+\cos\omega}{2}\right)^{N} \sum_{k=0}^{N-1} \binom{N+k-1}{k} \left(\frac{1-\cos\omega}{2}\right)^{k}$$

Introduce $y = \dfrac{1-\cos\omega}{2}$; then y, ω, and the z-transform variable are related by $\cos\omega = \dfrac{z+z^{-1}}{2} = 1-2y$. Expressing $P(\omega)$ in terms of y yields $\widetilde{P}(y) = 2(1-y)^{N} B_{N}(y)$, where $B_{N}(y) = \sum_{k=0}^{N-1} \binom{N+k-1}{k} y^{k}$ is the binomial series truncated after *N* terms.

The spectral factorization of $P(\omega)$ and the subsequent determination of the scaling filter coefficients are performed as follows:

1. Find the *N*-1 zeros $\{y_i\}$ of the polynomial $B_N(y)$.
2. Transform these to the z domain using $z_i = 1-2y_i \pm \sqrt{(1-2y_i)^2 - 1}$. Because of the square root, this yields a total of 2*N*-2 values.
3. Of the 2*N*-2 values of z_i, retain only those *N*-1 that have $|z_i| < 1$. This choice results in a minimum phase filter.
4. Include a further *N* zeros at $z = -1$. The coefficients of the resultant polynomial are the taps of the scaling filter for the Daubechies maxflat filter of length 2*N*.

This procedure is implemented in the MATLAB program of Figure 4, which generates these wavelets.

```
function dau=makedau(N)
% Function to generate coefficients of
% Daubechies maxflat orthonormal wavelet
% filter of length N.
%
if mod(N,2) ~=0
  error('N must be even');
end
p=N/2;
b(p)=1;
for i=p-1:-1:1
   b(i)= b(i+1)*(2*p-i-1)/(4*(p-i));
end
r=roots(b)/4;
z1= (1-2*r) + sqrt((1-2*r).^2 -1);
z2= (1-2*r) - sqrt((1-2*r).^2 -1);
z=[z1;z2];
zchoose = z( abs(z)<1 );  % for minimum phase
q=poly(zchoose);
q=real(q);
%add p zeros at -1
flat=1;
for i=1:p
   flat=conv(flat,[1 1]);
end
dau= conv(q,flat);
dau= sqrt(2)*dau/sum(dau);
```

Figure 4. MATLAB function for generating Daubechies maxflat wavelets.

3.2 Generating Daubechies Symmetric Complex Wavelets

A Daubechies symmetric complex wavelet of length 2*N* is a length-2*N* wavelet whose frequency response exhibits *N* zero derivatives at $\omega = 0$ and also has linear phase [3]. Linear phase is a highly desirable property that cannot be achieved in orthonormal wavelets unless the coefficients are allowed to be complex. These wavelets are obtained by the spectral factorization of the same halfband filter as in Section 3.1, i.e. :

$$P(\omega) = 2\left(\frac{1+\cos\omega}{2}\right)^{N} \sum_{k=0}^{N-1} \binom{N+k-1}{k} \left(\frac{1-\cos\omega}{2}\right)^{k}$$

The coefficients are calculated according to the same four steps shown in Section 3.1, except that now the roots of the factorization to be retained in step 3 are chosen in order to achieve symmetry rather than minimum phase. Symmetry is achieved by ensuring that roots are retained in pairs (z_i, z_i^{-1}). (Since more than one solution achieving symmetry can be obtained, we choose also to ensure that retained zeros within the unit circle are evenly distributed above and below the real axis.) Therefore, step 3 of the process becomes:

3a. Of the 2*N*-2 values of z_i, retain only those $\dfrac{2N-2}{4}$ that have $|z_i| > 1$ and positive real part. What this does is to reduce each quadruplet

(z_i, z_i^{-1}, $\bar{z}_i$, $\bar{z}_i^{-1}$) of roots to a single root outside the unit circle and having positive imaginary part.

3b. Sort these $\frac{N-1}{2}$ roots into increasing order of real part.

3c. Distribute these $\frac{N-1}{2}$ roots evenly above and below the real axis, by taking the conjugate of every second root.

3d. Include the inverse of each root. This results in a total of N-1 roots.

Note that for step 3a to succeed, it is necessary that $\frac{2N-2}{4}$ be an integer, i.e. that N be odd. Consequently, symmetric complex Daubechies wavelets exist only for odd N, i.e. filters of length $2N$ = 2, 6, 10, 14, etc.

A MATLAB function to generate these wavelets is presented in Figure 5.

```
function dau=makecxdau(N)
% Function to generate coefficients of
% Daubechies maxflat symmetric complex
% orthonormal wavelet filters.
%
if mod(N,2) ~=0
  error('N must be even');
end
p=N/2;
b(p)=1;
for i=p-1:-1:1
   b(i)= b(i+1)*(2*p-i-1)/(4*(p-i));
end
r=roots(b)/4;
z1= (1-2*r) + sqrt((1-2*r).^2 -1);
z2= (1-2*r) - sqrt((1-2*r).^2 -1);
z=[z1;z2];
% Keep one of every quadruplet z, 1/z, zbar,
% 1/zbar:
zchoose = z( abs(z)>1 & imag(z) >0);
% sort in order of increasing real part:
[junk,index]= sort(real(zchoose));
zchoose = zchoose(index);
% Distribute evenly above / below real axis:
zchoose(1:2:length(zchoose))=...
   conj(zchoose(1:2:length(zchoose)));
zchoose = [zchoose; 1./zchoose];
if(length(zchoose) ~= length(z)/2)
   disp('No symmetric solutions for this N:')
   error('N must be of form 2*(odd number)')
end
q=poly(zchoose);
% add p zeros at -1
flat=1;
for i=1:p
   flat=conv(flat,[1 1]);
end
dau= conv(q,flat);
dau= sqrt(2)*dau/sum(dau);
```

Figure 5. MATLAB function for generating complex symmetric Daubechies maxflat wavelets.

4 Conclusion

We have presented MATLAB implementations of two parameterizations of the space of orthonormal wavelets. In addition to their obvious use in generating wavelets with real coefficients, we have indicated how these may be used to generate symmetric wavelets with complex coefficients. This software would be useful in an optimization task in which the "best" wavelet for a given application is sought.

We also presented MATLAB functions that produce the commonly encountered Daubechies max-flat wavelets, as well as the symmetric complex Daubechies wavelets.

References:

[1] A. Cohen, I. Daubechies and J.C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, vol. 45, 1992, pp 485-560.

[2] I. Daubechies, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, vol. 41, 1988, pp 909-966.

[3] J.M. Lina and M. Mayrand, Complex Daubechies Wavelets, University of Montreal Laboratory of Nuclear Physics Report *UdeM-PHYSNUM-ANS-15*, December 1993.

[4] J.M. Lina and M. Mayrand, Parameterizations for Daubechies Wavelets, *Physical Review E*, Vol. 48, No. 6, 1993, pp. R4160-R4163.

[5] D. Pollen, SU1(2,F(z,1/z)) for F a subfield of C, *Journal of the American Mathematical Society,* Vol.3, No.3, 1990, pp. 611-624.

[6] B.G. Sherlock and D.M. Monro, On the Space of Orthonormal Wavelets, *IEEE Transactions on Signal Processing*, Vol. 46, No. 6, 1998, pp. 1716-1720.

[7] B.G. Sherlock and D.M. Monro, Optimized Wavelets for Fingerprint Compression, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-96)*, Atlanta, Georgia, May 1996, Vol. III, pp. 1447-1450

[8] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[9] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1992.