Queen Mary
University of London

# EBU6018
# Advanced Transform Methods

Week 4.4 – Perfect Reconstruction and Daubechies

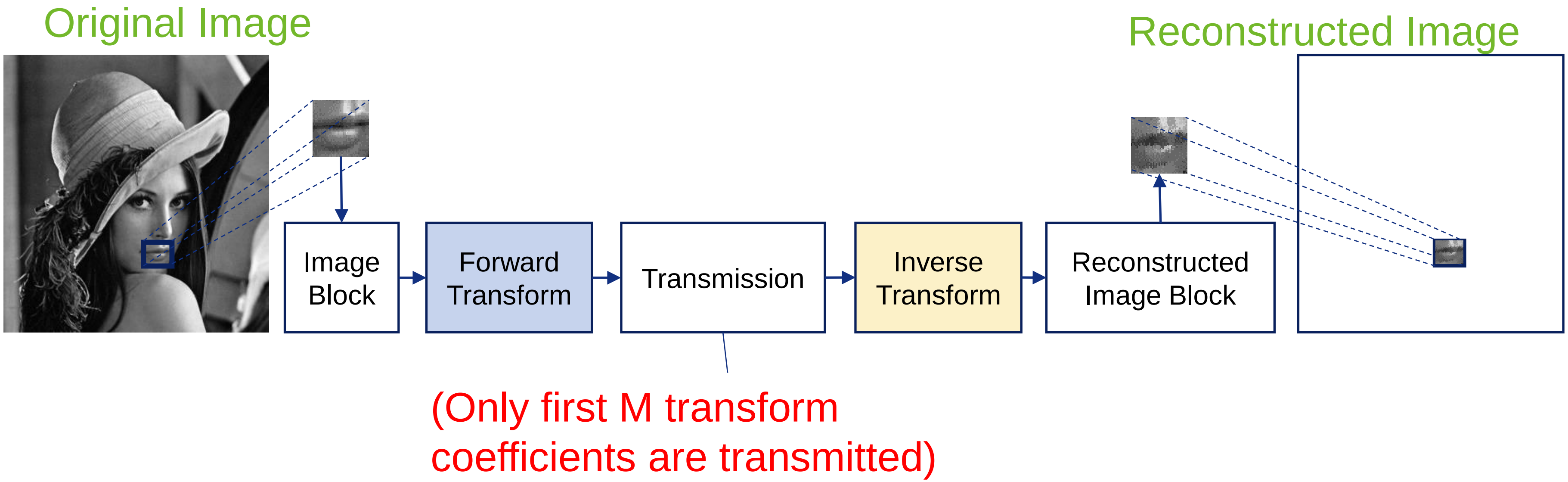
Dr Yixuan Zou

# Lecture Outline

- Perfect Reconstruction
  - Filter Banks
  - Z-transform
- Daubechies Wavelet Family
  - Orthogonal Filter Banks
  - Daubechies Wavelet Transform Matrix
  - Daubechies vs. Haar

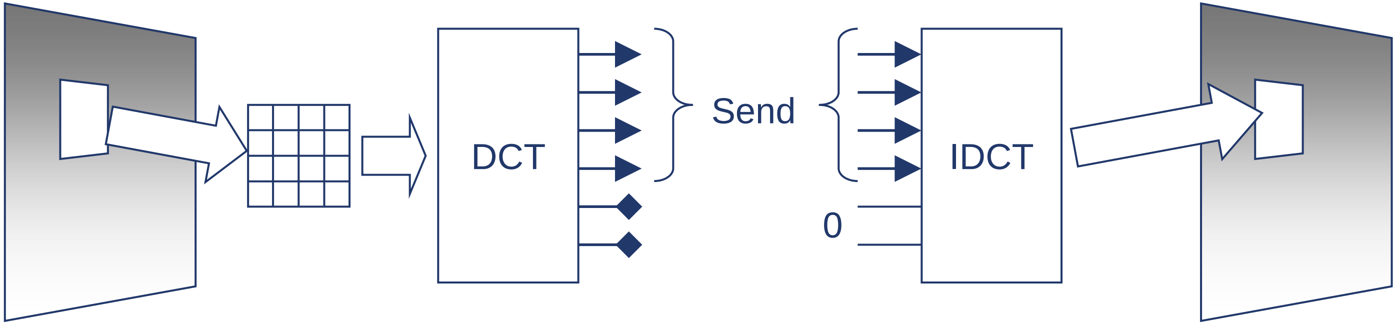
# Linear Transform Coding

- General procedures of linear transform coding:



# Linear Transform Coding (LTC)

- Discrete Cosine Transform (DCT) is a type of linear transform coding
- Advantages of DCT
  - most energy concentrated in a few coefficients, so
  - can discard some coeffs, while keeping most of signal



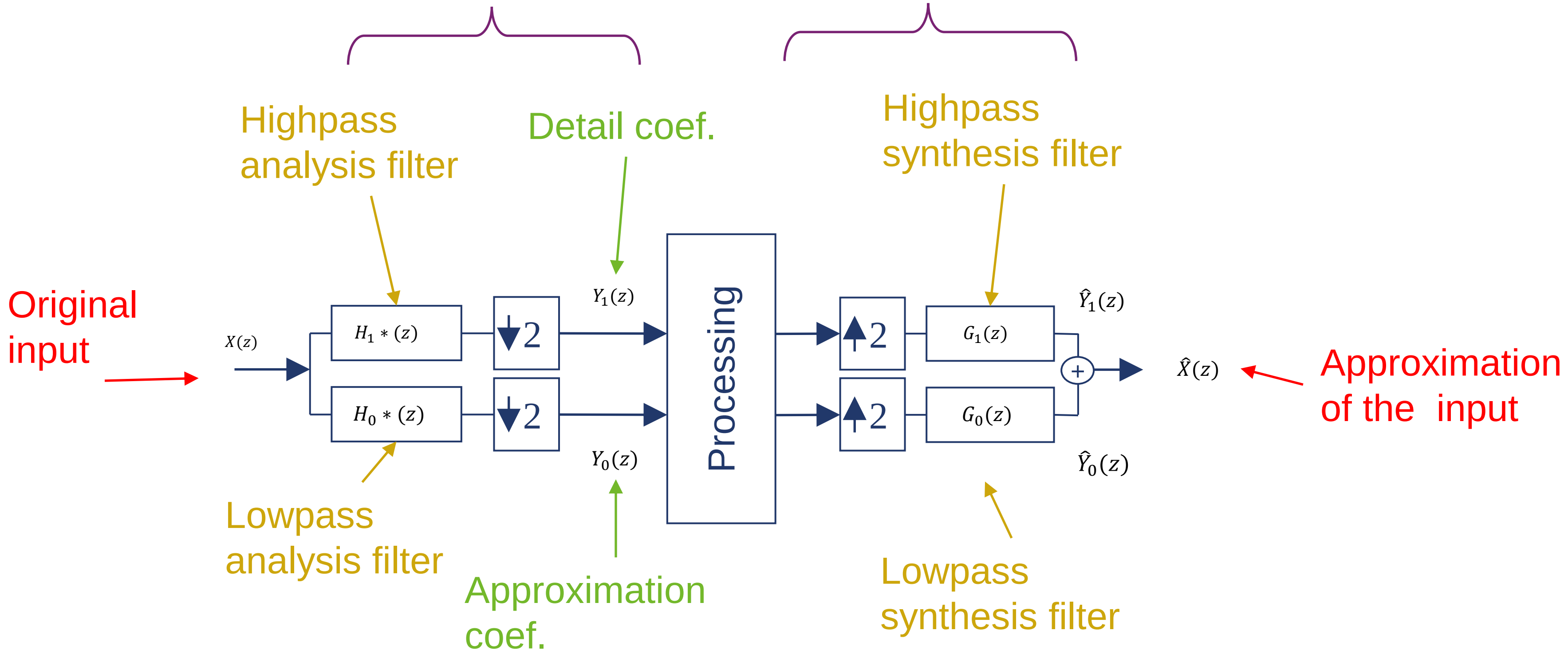

- Fourier transform, wavelet transform can also achieve this, depending on the signal.
- This type of application is known as Linear Transform Coding

# Perfect Reconstruction in LTC


- The transmission error is calculated as
  - $J = E(|\mathbf{x} - \hat{\mathbf{x}}|^2)$
  - Mean squared error (MSE)
  - $E(v)$ is the expected value (mean) of v
- We want to choose A (forward transform matrix) and B (inverse transform matrix) to minimize J

- Perfect reconstruction: If we keep all the coefficients and let $\mathbf{B} = \mathbf{A}^{-1}$ , we have
  - $\hat{\mathbf{y}} = \mathbf{y}$, hence $\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{y}} = \mathbf{B}\mathbf{y} = \mathbf{B}A\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x}$

# Filterbank

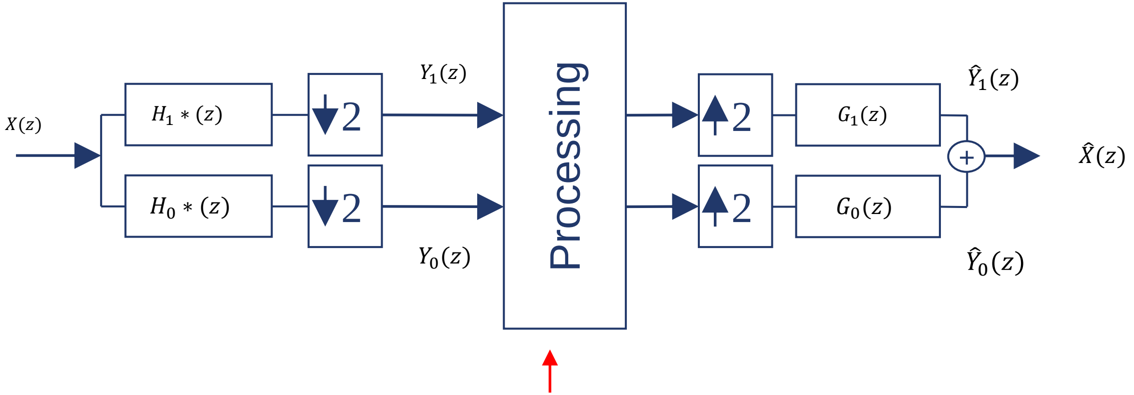


Multiple levels
Multiple levels
Highpass
analysis filter
Detail coef.
Highpass
synthesis filter
Original
input
$X(z)$
$H_1 * (z)$
$H_0 * (z)$
↓2
↓2
$Y_1(z)$
$Y_0(z)$
Processing
↑2
↑2
$G_1(z)$
$G_0(z)$
$\hat{Y}_1(z)$
$\hat{Y}_0(z)$
$\hat{X}(z)$
Approximation
of the input
Lowpass
analysis filter
Approximation
coef.
Lowpass
synthesis filter

# Perfect Reconstruction in Filterbank

We know that it would be good if we can invert a transform to reconstruct the original sequence from the output of the transform.

If we can do so, then we have Perfect Reconstruction:

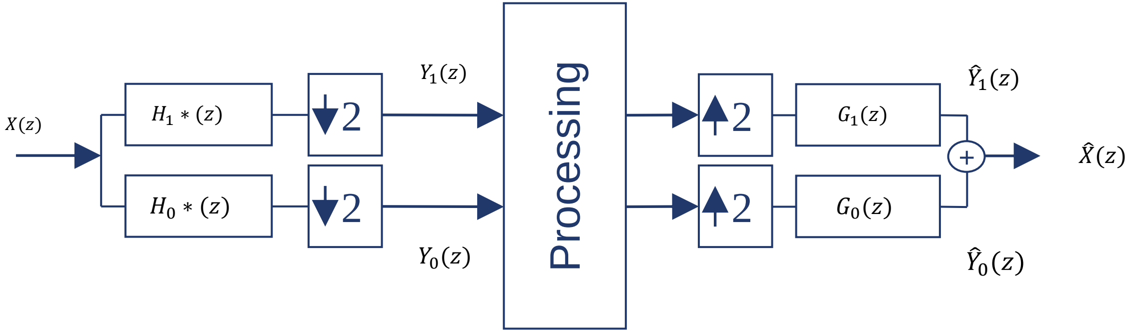



It is only possible to have Perfect Reconstruction if we do not process the output of the transform before we invert it.

# Perfect Reconstruction in Filterbank

We know that it would be good if we can invert a transform to reconstruct the original sequence from the output of the transform.

If we can do so, then we have Perfect Reconstruction:




- We want to derive possible relationships between H and G
- Notice that the signals and the filters are functions of z, i.e., operator of Z-transform

It is only possible to have Perfect Reconstruction if we do not process the output of the transform before we invert it.

# Z-Transform – Operator z

- What is z?
  - The Z-transform of a function $x(n)$ is given by

$$\boxed{Z[x(n)] = x(z) = \sum_{n} x(n) z^{-n}}$$

  - $n$ is an integer
  - $z = re^{j\omega} = r(\cos(\omega) + isin(\omega))$ is a complex variable, $r > 0$
    - $r$ is the magnitude of $z$, i.e., $r = |z|$
    - $\omega$ is the phase/angle of $z$

# Z-Transform of Sequences and Filters

- For example:
  - the sequence s[n] = [3, 6, 2, 8, 5…….] can be presented as

$$s(z) = 3 + 6z^{-1} + 2z^{-2} + 8z^{-3} + \ldots\ldots..$$

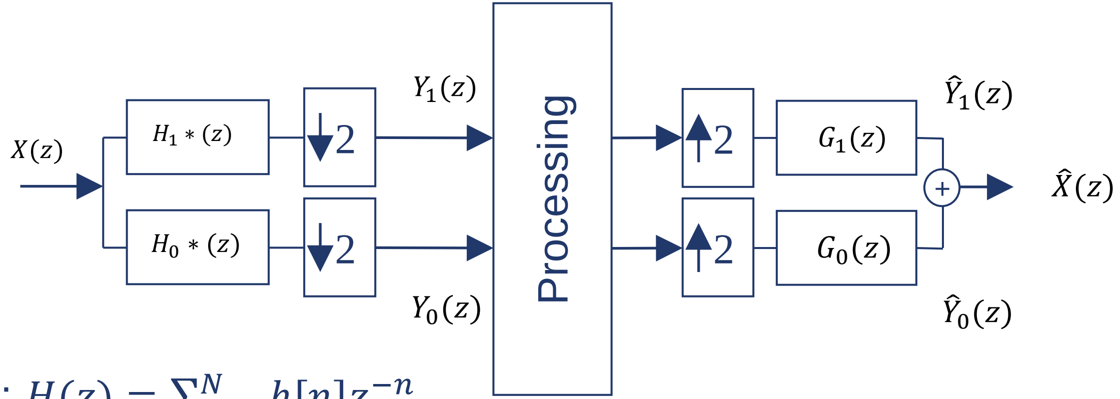
  - the Haar low-pass filter $h_0 = \frac{1}{\sqrt{2}}[1 \quad 1]$ (normalised) could be written

$$h_0(z) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}z^{-1}$$

- We use z-transform to represent the position of values in a sequence

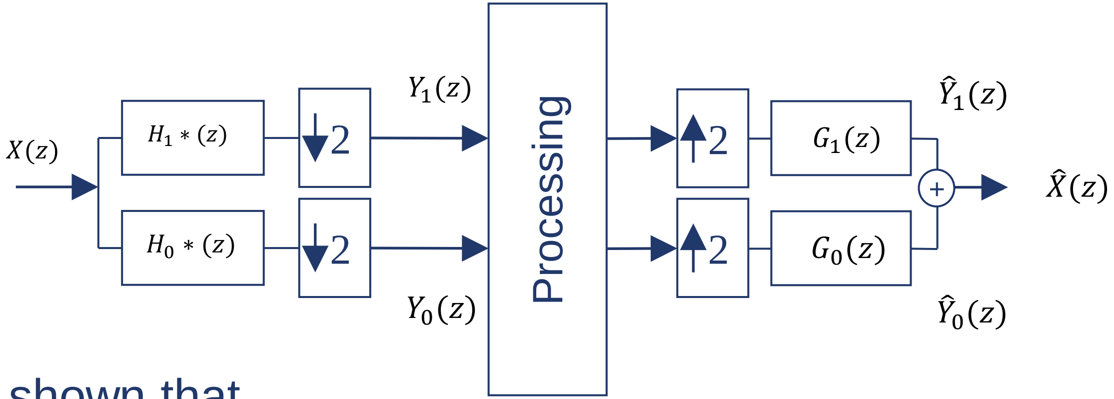
# Z-Transform and Perfect Reconstruction

- Analysis filters $H_0, H_1$ and synthesis filters $G_0, G_1$ may differ.




- Recall z-transform: $H(z) = \sum_{n=0}^{N} h[n]z^{-n}$
- Assuming no processing, we typically want Perfect Reconstruction (PR) - i.e. that $\hat{X}(z)$ is equal to $X(z)$
- This can be achieved with a +ve delay only, i.e. $\hat{X}(z) = z^{-l}X(z)$ for some $l \geq 0$.

# Z-Transform and Perfect Reconstruction




- It can be shown that

$$\hat{X}(z) = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)]X(z) + \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)]X(-z)$$

- For PR, want 2nd ("alias") term to be zero:

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0 \quad \text{biorthogonal filter bank}$$

# Z-Transform and Perfect Reconstruction

- It can be shown that

$$\hat{X}(z) = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)]X(z) + \frac{1}{2}[G_0(z)H_0(-z) + G_1(z)H_1(-z)]X(-z)$$

- For PR, want 2nd (“alias”) term to be zero:
  - **Condition 1**: $G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0$ biorthogonal filter bank
- A possible solution is

$$G_0(z) = H_1(-z) \quad \text{and} \quad G_1(z) = -H_0(-z)$$

This satisfies the alias cancellation condition

- For PR, also want 1st term to be a pure delay, e.g.
  - **Condition 2**: $2z^{-l} = G_0(z)H_0(z) + G_1(z)H_1(z) = G_0(z)H_0(z) - H_0(-z)G_0(-z)$
- i.e.

$$P_0(z) - P_0(-z) = 2z^{-l} \quad \text{where} \quad P_0(z) = H_0(z)G_0(z)$$

# Z-Transform and Perfect Reconstruction

- $P_0$(z) = $H_0$(z) $G_0$(z) is the product of two Low-Pass filters
  - There are many possible types of filter, each requiring specific design criteria, e.g, linear phase, maximally flat, etc
  - There are many ways of factoring $P_0$(z) into $H_0$(z) and $G_0$(z)

- One of the common ways is defined by:

$$P_0(z) = (1 + z^{-1})^{2k} Q(z)$$

  - where k is some constant and Q(z) can be chosen to give PR
  - $Q(z) = -1 + 4z^{-1} - z^{-2}$ gives filters that are “maxflat”, that is, the passband is maximally flat, e.g., Butterworth filters

# Z-Transform and Perfect Reconstruction

- $H_0$(z) and $G_0$(z) are low-pass filters

- The number of zeroes that they have and the positions of those zeroes affect the performance of the filters
- These are FIR filters, and the position of the zeroes affects many of the filter characteristics, including phase response and orthogonality.
- For minimum phase lag, all zeroes must lie inside the unit circle, if all zeroes lie outside the unit circle then the filter is maximum phase.
- For example, if the factors in $P_0(z) = H_0(z)G_0(z) = (1+z^{-1})^{2k}Q(z)$ are

$$H_0(z) = (1+z^{-1})^2(c - z^{-1}) \text{ and } G_0(z) = (1+z^{-1})^2\left(\frac{1}{c} - z^{-1}\right)$$

then they are orthogonal, although their phase is not linear.

# Daubechies Wavelet Family

- Recall, wavelet design method is:

  1) Design product filter $P_0(z)$ to satisfy $P_0(z) - P_0(-z) = 2z^{-l}$

  2) Factorize $P_0(z)$ into $H_0(z)$ and $G_0(z)$

- Example: The $k^{th}$ order Daubechies wavelets (db$k$),

$$H_0(z) = (1 + z^{-1})^k \prod_{i=1}^{k-1} (z_i - z^{-1})$$

$$G_0(z) = (1 + z^{-1})^k \prod_{i=1}^{k-1} (\frac{1}{z_i} - z^{-1})$$

where $z_i$ and $1/z_i$ are roots of a polynomial of degree $2k - 2$

Queen Mary University of London

# Daubechies Wavelet Family

- Using $k = 1$ for db$k$ wavelet we get

$$H_0(z) = G_0(z) = (1 + z^{-1}) \text{ or } H_0(\omega) = (1 + e^{-j\omega n})$$ (Not normalised)

- i.e. the Haar wavelet (apart from a scaling factor).

- Using $k = 2$ for db$k$ wavelet we get

$c = 2\text{-}\sqrt{3}$ , $\frac{1}{c} = 2\text{+}\sqrt{3}$

$$H_0(z) = (1 + z^{-1})^2(c - z^{-1}) \text{ and } G_0(z) = (1 + z^{-1})^2(\frac{1}{c} - z^{-1})$$

  - where $c = 2 - \sqrt{3}$ and $1/c$ are the roots of the polynomial

$$Q(z) = -1 + 4z^{-1} - z^{-2}$$

- Daubechies wavelets are actually orthogonal.
  (E.g. check the power complementary condition)

Queen Mary University of London

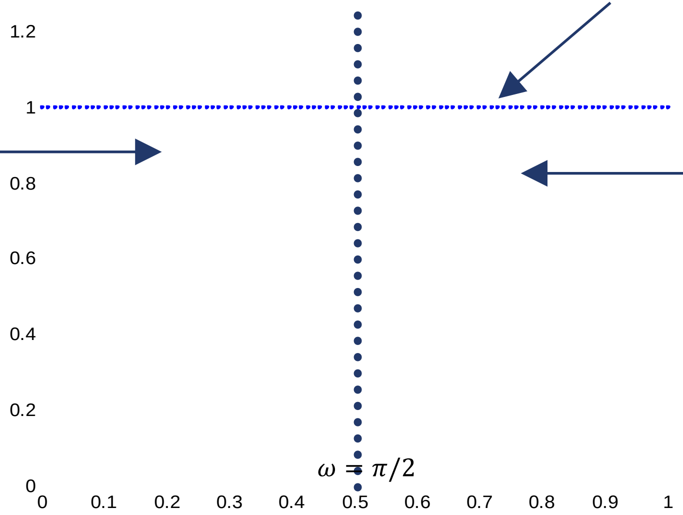
# Power Complementary Condition

- Definition: Transform formed by set of filters $H_0(\omega)$ and $H_1(\omega)$ is energy conserving.

$$|H_0(\omega)|^2 + |H_1(\omega)|^2 = 1$$

Energy in the low-pass filter

$|H_0(\omega)|^2$




$|H_1(\omega)|^2$

Energy in the high-pass filter

This is true if the filters are orthogonal, which we have chosen.

(Biorthogonal filter banks are more difficult to design.)

# Orthogonal Filter Banks

- Orthogonal filter banks are orthogonal in the sense that

$$\sum_n h_i[n-2k]h_i[n] = \delta(k) \ \text{ and } \ \sum_n h_i[n-2k]h_l[n] = 0 \text{ for } i \neq l$$

which can be achieved by e.g. (given without proof)

For Haar $h_0$ = [$h_0$(0), $h_0$(1)] = [0.707, 0.707]

$$H_1(z) = (-z)^{-N} H_0(-z^{-1})$$

So $h_1$ = [0.707, -0.707]

$\text{i.e. that high}-\text{pass analysis filter } h_1 \text{ is alternating flip of } h_0\text{:}$

$$(h_1[0],\ h_1[1],\ h_1[2],\ \dots, h_1[N]) = (h_0[N],\ -h_0[N-1],\ h_0[N-2],\ \dots)$$

- Recall, $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$, we get e.g

$$G_0(z) = z^{-N} H_0(z^{-1})$$

- So, e.g. the resynthesis filter $\gamma_0[n] \Leftrightarrow G_0(z)$ is flip of $h_0[n]$:

$$(\gamma_0[0],\ \gamma_0[1],\ \gamma_0[2],\ \dots, \gamma_0[N]) = (h_0[N],\ h_0[N-1],\ \dots,\ h_0[0])$$

# Orthogonal Filter Banks – Summary

For our special case of Orthogonal Filter Banks:

- Choose $H_1$(z) = -$z^{-N}H_0$(-$z^{-1}$)     ie, delayed
- $G_0$(z) = $H_1$(-z) = $z^{-N}H_0(z^{-1})$     ie, delayed
- $G_1$(z) = -$H_0$(-z) = $z^{-N}H_1(z^{-1})$     ie, delayed

So, if we know $H_0$(z) then the others can be derived.

That is, the synthesis filters are time-reversed versions of the analysis filters with a delay.

# Daubechies Wavelet Family – Example k=2

- The values of the scaling coefficients are:

$$h_0[0] = \frac{1+\sqrt{3}}{4\sqrt{2}}$$

$$h_0[1] = \frac{3+\sqrt{3}}{4\sqrt{2}}$$

$$h_0[2] = \frac{3-\sqrt{3}}{4\sqrt{2}}$$

$$h_0[3] = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

The dbk-2 wavelet is also called the D4 wavelet because it has 4 coefficients

# Daubechies Wavelet Family – Example k=2

- Daubechies wavelet: analysis filters

$$h_0[n] = (h_0[0], h_0[1], h_0[2], h_0[3]) = \left(\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}}\right)$$ = (0.483, 0.837, 0.224, -0.129)

$$h_1[n] = (h_1[0], h_1[1], h_1[2], h_1[3]) = (h_0[3], -h_0[2], h_0[1], -h_0[0])$$

- Synthesis filters:

$$\gamma_0[n] = (h_0[3],\ h_0[2],\ h_0[1],\ h_0[0])$$

$$\gamma_1[n] = (h_1[3],\ h_1[2],\ h_1[1],\ h_1[0])$$ = (- $h_0$[0], $h_0$[1], - $h_0$[2], $h_0$[3])

Question: What are the elements of the high-pass analysis filter and of the synthesis filters?

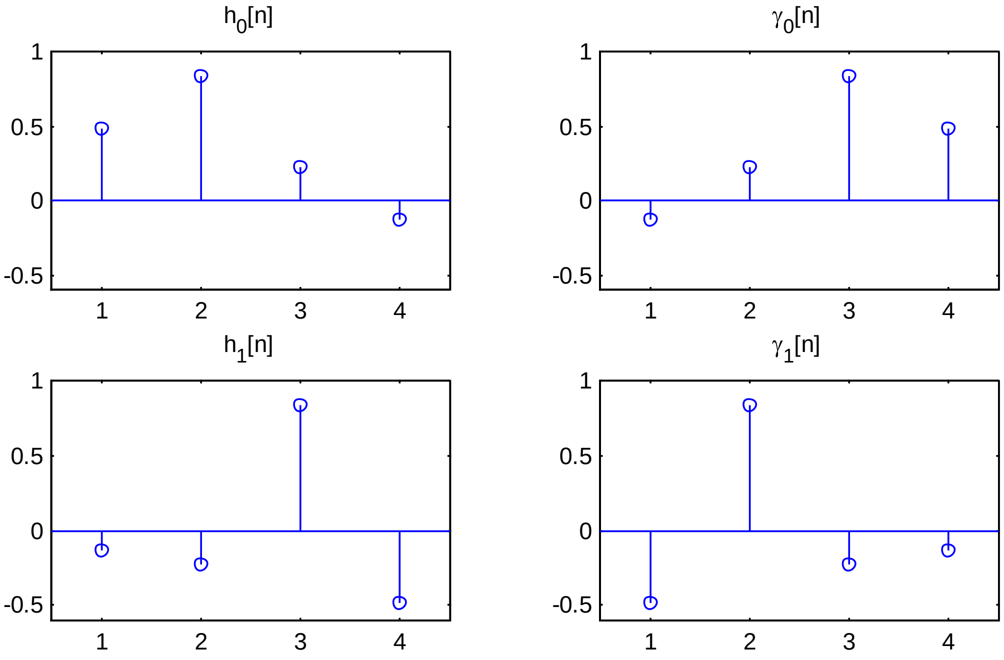
# Daubechies Wavelet Family – Example k=2

- Answer:
  - $h_o[n]=[h_0[0], h_0[1], h_0[2], h_0[3]] =[0.483, 0.837, 0.224, -0.129]$
  - ✓ $h_1[n]=[\ h_0[3],-\ h_0[2], h_0[1], -h_0[0]] = [-0.129, -0.224, 0.837, -0.483]$
  - ✓ $\gamma_0[n]=[\ h_0[3], h_0[2], h_0[1], h_0[0]] = [-0.129, 0.224, 0.837, 0.483]$
  - ✓ $\gamma_1[n]=[\ -h_0[0], h_0[1], -h_0[2], h_0[3]] = [-0.483, 0.837, -0.224, -0.129]$

# Daubechies Wavelet Family – Example k=2

Low-pass analysis
Scaling Function

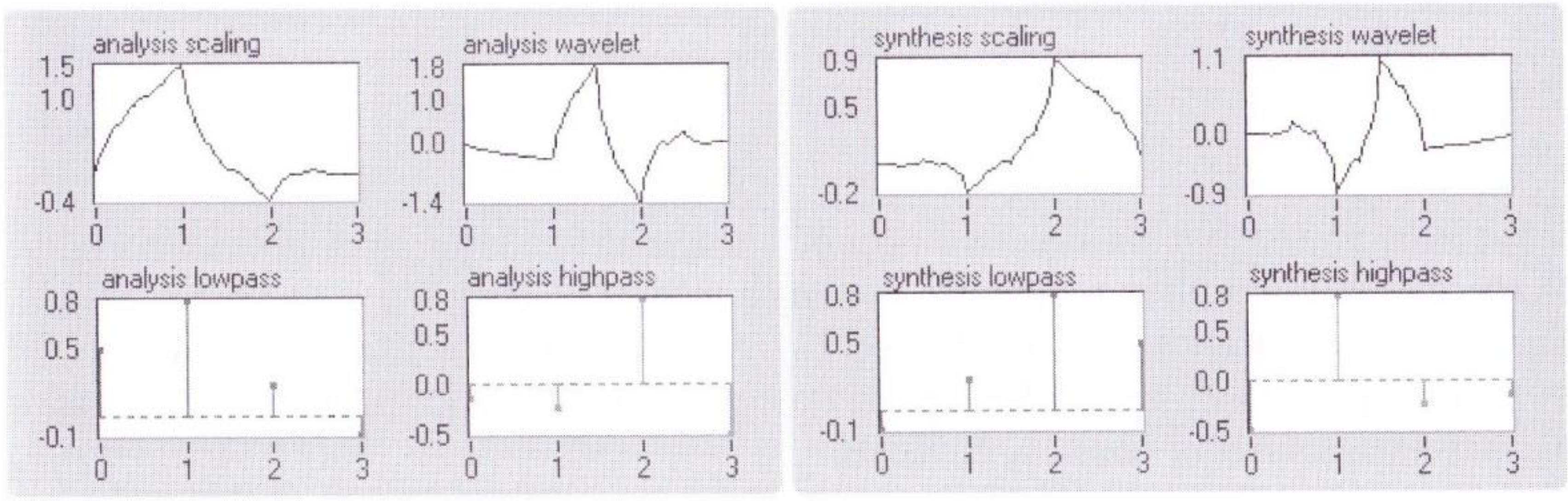
High-pass analysis
Wavelet function




Low-pass synthesis

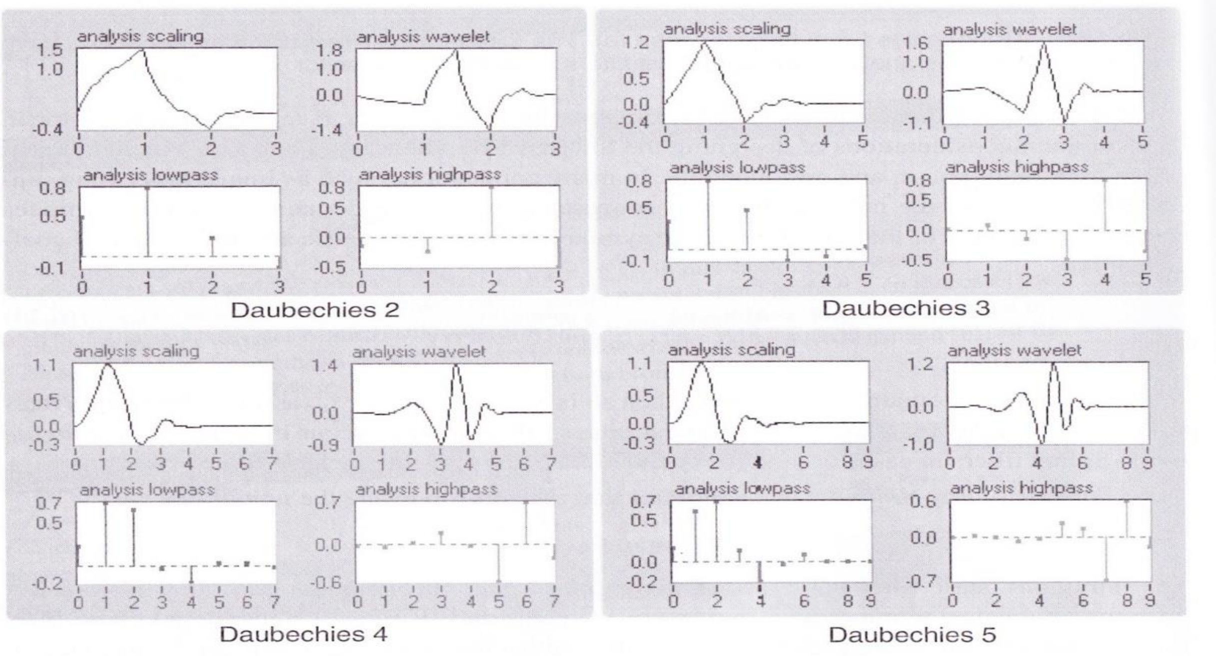
High-pass synthesis

# Daubechies Wavelet Family – Example k=2




$H(z)$ $G(z)$

# Daubechies Wavelet Family – Example k=2, 3, 4, 5




As the order increases,
the wavelets become more and more smooth
with more oscillations.

They do not have a linear phase response.

# Daubechies Wavelet  Transform

Calculating the dbk-2 transform is performed by taking the inner product (dot product) of the filter coefficients and 4 input data values of s[n].

If the output of the low pass filter is a $c$ coefficient and the output of the high pass filter is a $d$ coefficient:

$$c_i = h_0[0]s_{2i} + h_0[1]s_{2i+1} + h_0[2]s_{2i+2} + h_0[3]s_{2i+3}, \qquad i = 0,1,\ldots$$

$$d_i = h_1[0]s_{2i} + h_1[1]s_{2i+1} + h_1[2]s_{2i+2} + h_1[3]s_{2i+3}, \qquad i = 0,1,\ldots$$

# Daubechies Wavelet Transform Matrix – Forward

$$\begin{bmatrix} h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 & 0 & 0 & 0 & & \\ h_1[0] & h_1[1] & h_1[2] & h_1[3] & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 & 0 & & \\ 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] & 0 & 0 & & \\ 0 & 0 & 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] & & \\ 0 & 0 & 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] & & \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] \\ 0 & 0 & 0 & 0 & 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} c_0 \\ d_0 \\ c_1 \\ d_1 \\ c_2 \\ d_2 \\ c_3 \\ d_3 \end{bmatrix}$$

The problem of not being able to calculate the final values could be solved by assuming that the input sequence is periodic, that is, assuming that $s_0$ and $s_1$will be the next two input values, that is, putting $s_8 = s_0$, and $s_9 = s_1$

# Daubechies Wavelet Transform Matrix – Inverse

$$
\begin{bmatrix}
h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 & 0 & 0 & 0 & 0 & 0 \\
h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 & 0 & 0 & 0 \\
 & & h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & h_0[2] & h_1[2] & h_0[0] & h_1[0] & 0 & 0 \\
 & & 0 & 0 & h_0[3] & h_1[3] & h_0[1] & h_1[1] & 0 & 0 \\
 & & 0 & 0 & 0 & 0 & h_0[2] & h_1[2] & h_0[0] & h_1[0] \\
 & & 0 & 0 & 0 & 0 & h_0[3] & h_1[3] & h_0[1] & h_1[1]
\end{bmatrix}
\begin{bmatrix} c_1 \\ d_1 \\ c_2 \\ d_2 \\ c_3 \\ d_3 \\ c_4 \\ d_4 \end{bmatrix}
=
\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}
$$

# Daubechies vs. Haar – Summary

- The Daubechies family is an orthogonal set of wavelet functions.
  - When k=1, we have the Haar Function
  - When k=2, we have the dbk-2 Function
- For the Haar Forward Transform, there is no overlap between successive pairs of scaling and wavelet functions
- With the dbk-2 Forward Transform there is an overlap
- The Haar high-pass filter produces a result depending on the difference between an even element and an odd element. But will not produce a result depending on the difference between an odd element and its even successor. This change will, however, be picked up in later steps.
- For the dbk-2 high-pass filter, there is an overlap so change between any two elements will be detected.

# Daubechies vs. Haar – Summary

- Since the dbk-2 Wavelet Transform detects any change in the input data at every transform step, then it is more accurate in detecting changes in the input data
- However, the dbk-2 has a higher computation cost than the Haar Transform
- The trade-off depends on the application……accuracy versus speed.
- Daubechies transforms are used for detecting discontinuities and also for "self-similarity", that is, feature extraction. They are not so useful for image processing because the phase is not linear.

# Daubechies Wavelet – Exercise

The forward transform matrix for the dbk-2 transform is:

$$\begin{bmatrix} h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 & 0 & 0 & 0 \\ h_1[0] & h_1[1] & h_1[2] & h_1[3] & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] & 0 & 0 \\ 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] \\ 0 & 0 & 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0[0] & h_0[1] & h_0[2] & h_0[3] \\ 0 & 0 & 0 & 0 & 0 & 0 & h_1[0] & h_1[1] & h_1[2] & h_1[3] \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} c_1 \\ d_1 \\ c_2 \\ d_2 \\ c_3 \\ d_3 \\ c_4 \\ d_4 \end{bmatrix}$$

Calculate the first 2 c coefficients and the first 2 d coefficients if the input sequence is:

S[n] = [3, 7, 1, 4, 6, 9, 2, 5]

# Daubechies Wavelet – Exercise Solution

- $c_1$ = 3x0.483 + 7x0.837 + 1x0.224 + 4x(-0.129) = 7.016
- $d_1$= 3x(-0.129) + 7x(-0.224) + 1x0.837 + 4x(-0.483) = -3.05
- $c_2$ = 1x0.483 + 4x0.837 + 6x0.224 + 9x(-0.129) = 4.014
- $d_2$ = 1x(-0.129) + 4x(-0.224) + 6x0.837 + 9x(-0.483) = -0.35

Queen Mary
University of London