EEE6440

Advanced Signal Processing (ASP)

- Transforms, Filter banks and Wavelets
  - Convolution as a matrix multiplication.
  - What are the uses of transforms?
  - Transform example.
  - Orthogonal Transforms. (Perfect reconstruction and Parseval's Theorem)
  - The Discrete Cosine Transform.
  - N-point transforms on signals.

- Filter Banks and wavelets
  - Orthogonal filter banks
  - Perfect reconstruction condition
  - Filter bank design
  - Dyadic decomposition
  - What is a wavelet?
  - Wavelet implementation
  - Wavelet decomposition schemes

Dr Charith Abhayaratne

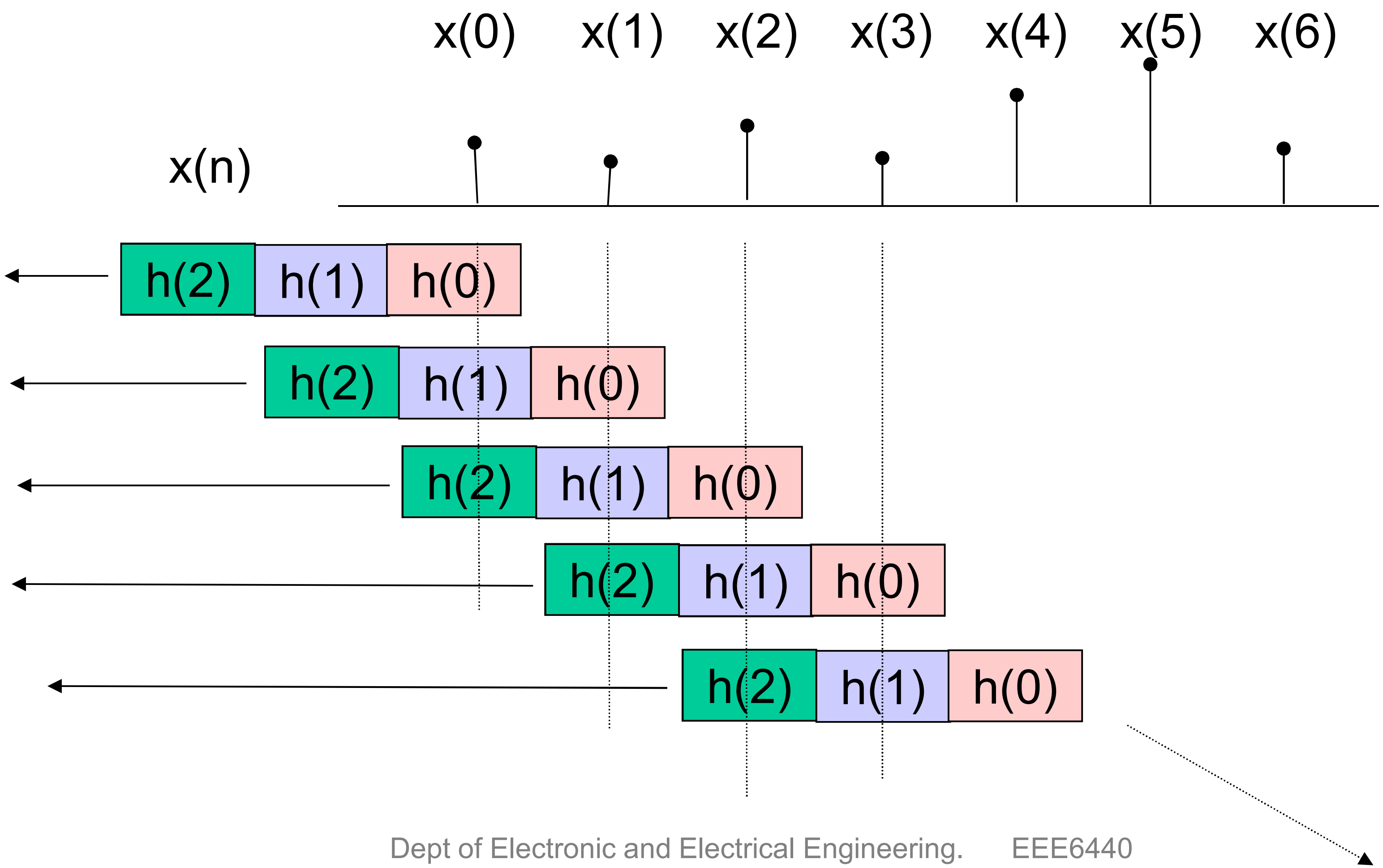
Email: c.abhayaratne@sheffield.ac.uk
Office: F176 Tel: 25893

Convolution as a matrix multiplication

$$y = x * h$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

h(k)= [h(0) h(1) h(2)]



The University Of Sheffield.

Convolution as a matrix multiplication

What are the rest of the elements in the matrix?

$$
\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ . \end{bmatrix} = \begin{bmatrix} h(0) & & & & \\ h(1) & h(0) & & & \\ h(2) & h(1) & h(0) & & \\ & h(2) & h(1) & h(0) & \\ & & . & . & . \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ . \end{bmatrix}
$$

output

Convolution matrix
(a block diagonal matrix)
Let's call it H (The transform matrix)

input

Invertibility property: If $H^{-1}H = I$ (The Identity matrix)

For most filters: Either H is not invertible or there exists an H^{-1}, but is not stable. Therefore, filters are usually lossy transforms.

Transforms

A transform of a signal is a new representation of that signal.

Consider the following system of equations:

$$y(0) = x(0) + x(1)$$
$$y(1) = x(0) - x(1)$$
$$y(2) = x(2) + x(3)$$
$$y(3) = x(2) - x(3)$$

This transforms x into y using a 4-point transform.

In other words the signal is represented in y-domain using the linear combinations of signal components in the x domain.

What is the inverse transform for the above transform?

Transforms

$y(0) = x(0) + x(1)$

$y(1) = x(0) - x(1)$

$y(2) = x(2) + x(3)$

$y(3) = x(2) - x(3)$

1. Write down this transform in matrix representation:
2. Repeat the same for the inverse transformation
3. Check the Invertibility condition

Transforms

$y(0)= x(0) + x(1)$
$y(1)= x(0) - x(1)$
$y(2)= x(2) + x(3)$
$y(3)= x(2) - x(3)$

1. Why do we use transforms?
   (We will discuss the solution throughout this topic
So, write them down here when you have learned them)

   1.
   2.
   3.

An example: Consider $x(0)= 12$ $x(1)=10$ $x(2)= -9$ $x(3)= -10$ #Plot X

Compute: $y(0)$ $y(1)$ $y(2)$ $y(3)$ #Plot Y

What can you learn about this data from the y-domain representation?
How do you interpret the transform domain values.

Now set $y(1)=y(3)=0$ and compute the new x values. #Plot new X

What have you learned about transform domain processing?

Transforms

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}}_{H} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \qquad \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{2} \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}}_{H^{-1}} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix}$$

Is H the transpose of H^{-1} ?

We can split the factor ½ into (1/sqrt(2) x 1sqrt(2)) and use as the normalistaion constant for both H and H^{-1}.

Now, the inverse is the transpose of the original matrix.
This is true only when the transform is an **orthogonal transform**:

Compute the sum of squares of the output (y) and show that $||x||^2 = ||y||^2$.
In this case we call the transform is **unitary**.

Orthogonal Transforms

- Discrete transforms "map" data from one domain into another.

$$x \longrightarrow \boxed{\text{Transform}} \longrightarrow c$$

- x is input data on time or space domain.
- c is the transform coefficient domain (For the Fourier transform it is frequency domain).

- 1D transforms have the form:

(also called an N-point transform)

$$c_n = \sum_{i=0}^{N-1} f_{ni} x_i \qquad for\ n = 0, \ldots, N-1.$$

N coefficients

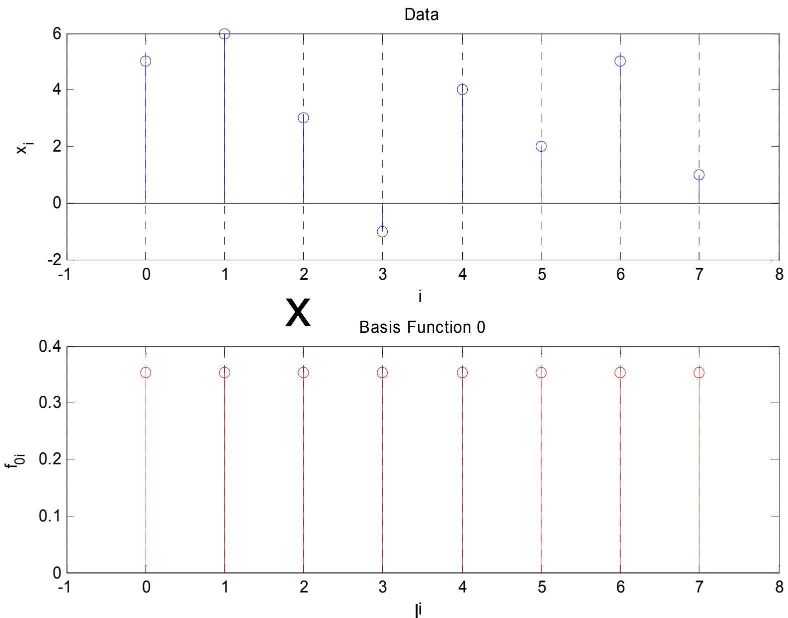
N basis functions

N data values

- The corresponding matrix notation: C=FX ,
- Rows of F represent corresponding **basis functions** of the transform.

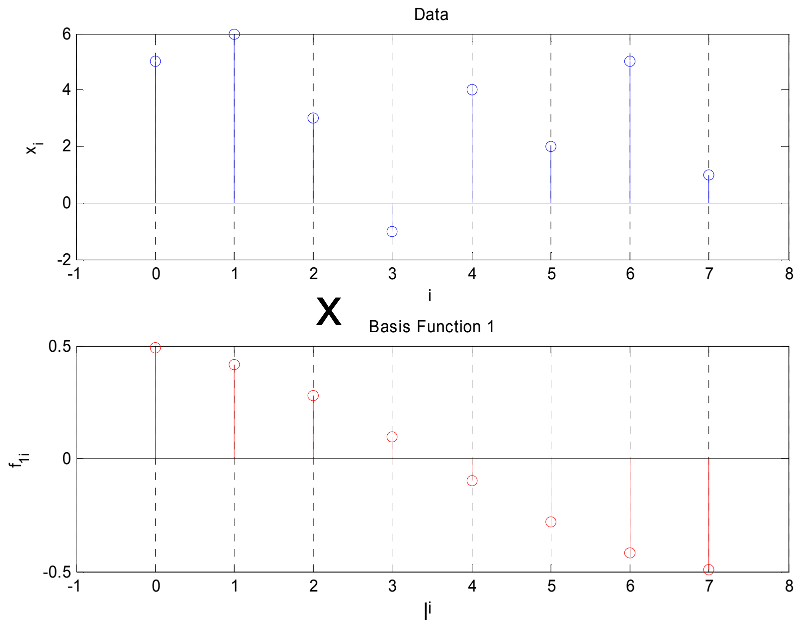
Orthogonal Transforms

x



f_0

x



f_1

c_0= sum of all products

$$=\sum f_{0i} x_i$$

c_1= sum of all products

$$=\sum f_{1i} x_i$$

How do you find the n^{th} coefficient c_n ?

Orthogonal Transforms

- Inverse transform reconstructs data.

$$x_j = \sum_{n=0}^{N-1} g_{jn} c_n \qquad for\ \ j = 0, \ldots, N-1.$$

- We need perfect reconstruction.
- Let's expand the inverse transform:

$$x_j = \sum_{n=0}^{N-1} g_{jn} \sum_{i=0}^{N-1} f_{ni} x_i$$

$$= \sum_{i=0}^{N-1} x_i \sum_{n=0}^{N-1} g_{jn} f_{ni}$$

- We will get perfect reconstruction if

$$\sum_{n=0}^{N-1} g_{jn} f_{ni} = 1 \quad \text{when} \ \ i = j$$

$$= 0 \quad \text{when} \ \ i \neq j$$

- i.e., the Identity matrix.
- For Orthogonal Transforms --------------------------- $g_{jn} = f_{jn}$ (transpose)
- The orthogonality condition:

$$\boxed{\sum_{n=0}^{N-1} f_{jn} f_{ni} = \delta_{ji}}$$

Orthogonal Transforms

- Consider the total power of the data:

$$P = \sum_j (x_j)^2 = \sum_j \left(\sum_n f_{jn} c_n \right)^2$$

- When you multiply this out, you get the sum of all possible pair products.

$$
\begin{aligned}
P &= \sum_j \sum_m \sum_n f_{jn} c_n f_{jm} c_m \\
&= \sum_n \sum_m c_n c_m \sum_j f_{nj} f_{jm} \\
&= \sum_n \sum_m c_n c_m \delta_{nm} \\
&= \sum_n c_n^{\ 2}
\end{aligned}
$$

Homework:
Prove the same using matrix representation.

- Parseval's Theorem: $\boxed{\sum_i x_i^{\ 2} = \sum_i c_i^{\ 2}, \quad \text{provided} \quad \sum_{j=0}^{N-1} f_{nj} f_{mj} = \delta_{nm},}$ i.e., the orthogonality condition.

The University Of Sheffield.

The Discrete Cosine Transform (DCT)

- Uses Cosines as basis functions:
- The N-point DCT

$$c_n = \sqrt{\frac{e_n}{N}} \sum_{i=0}^{N-1} \left[\cos\left(\frac{(2i+1)n\pi}{2N} \right) \right] x_i$$

$$e_n = \begin{cases} 1 & \text{when } n = 0 \\ 2 & else \end{cases}$$

Please bring your results to the next lecture

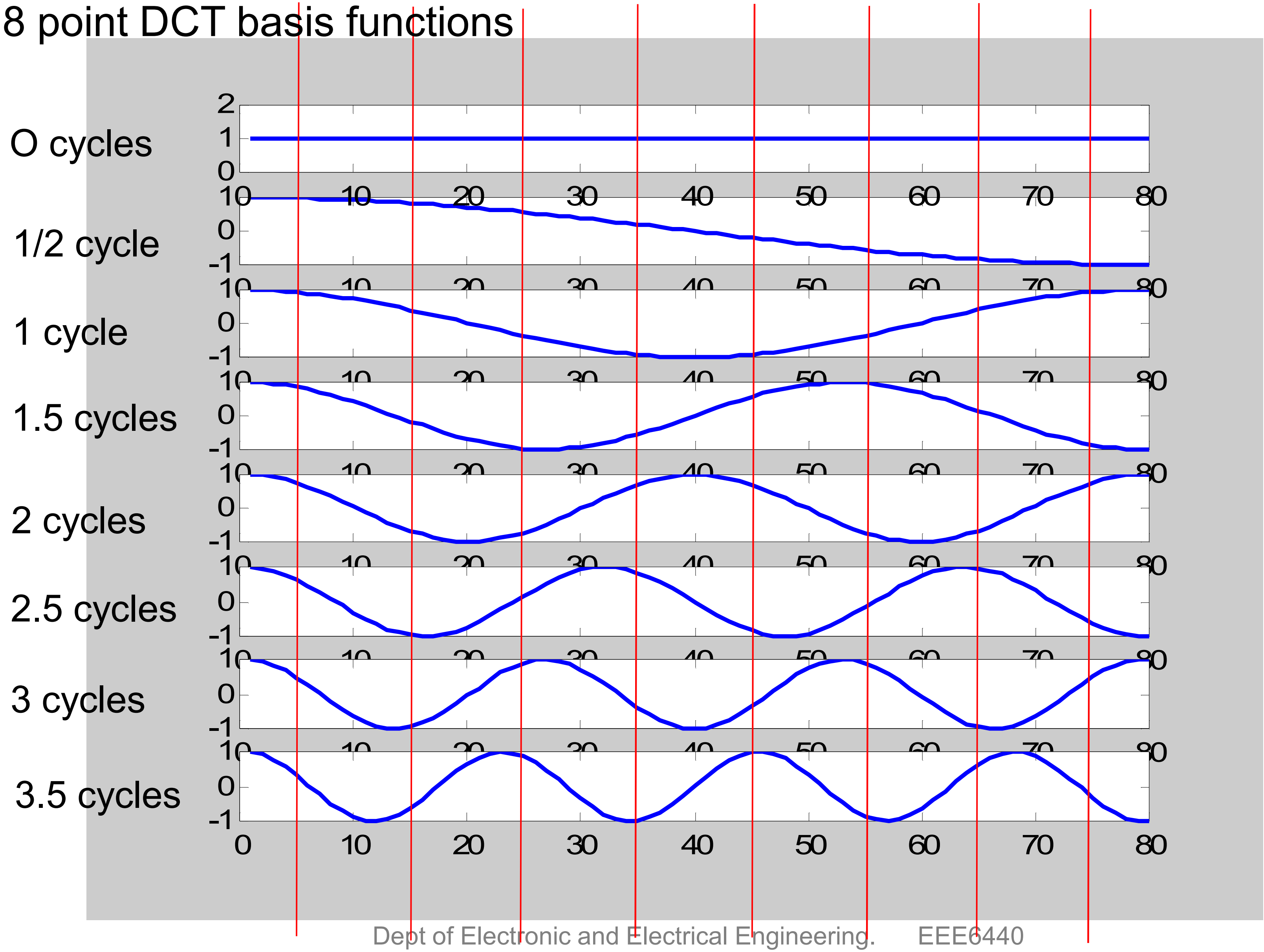
<u>Homework:</u>

Using MATLAB

1. Find out the N-point DCT transforms matrix for N=2, 4 & 8.
   Hint: >lookfor dct %to find out command for computing dct in Matlab
   We Know Y=HX for transforms in matrix notation
   What is Y when X=I, where I is the Identity matrix of NxN elements.
2. Plot them using the “stem” command.
   Hint: >help stem
3. Verify that these DCTs are orthogonal
4. Compute the Inverse of H for all DCTs and derive an expression.

The Discrete Cosine Transform (DCT)

8 point DCT basis functions



1. The Coefficients are real.

2. Has half as well as full period cosines.

3. Symmetry can be either odd or even.

4. Can compute using the FFT

The Discrete Cosine Transform (DCT)

- Consider the input data:

  X= [5 6 3 4 3 4 2 3]
- H= 8-point DCT transform
- Y=HX gives

  Y= [10.6066 2.4635 0.6533 0.6539 0 -1.0878 -0.2706 -1.8222]

- Is X or Y more correlated?

- Y_{new}=[11 2 1 1 0 -1 0 -2] by rounding.

- X_{new}=[5 6 3 4 3 5 2 3] by inverse DCT.

- Sum of x^2=124 in Y, 112.5 out of 124 is coming from a single coefficient.

Transforms

- Why do we need transforms?

1. To analyse data or signals. (Different features can be identified in different representations)
2. To decorrelate data.
3. To compact power of data into a fewer coefficients.
4. To use above 2 & 3 to compress data.

- Other transforms

1 Discrete Sine Transform (DST) – Sine waves as bases

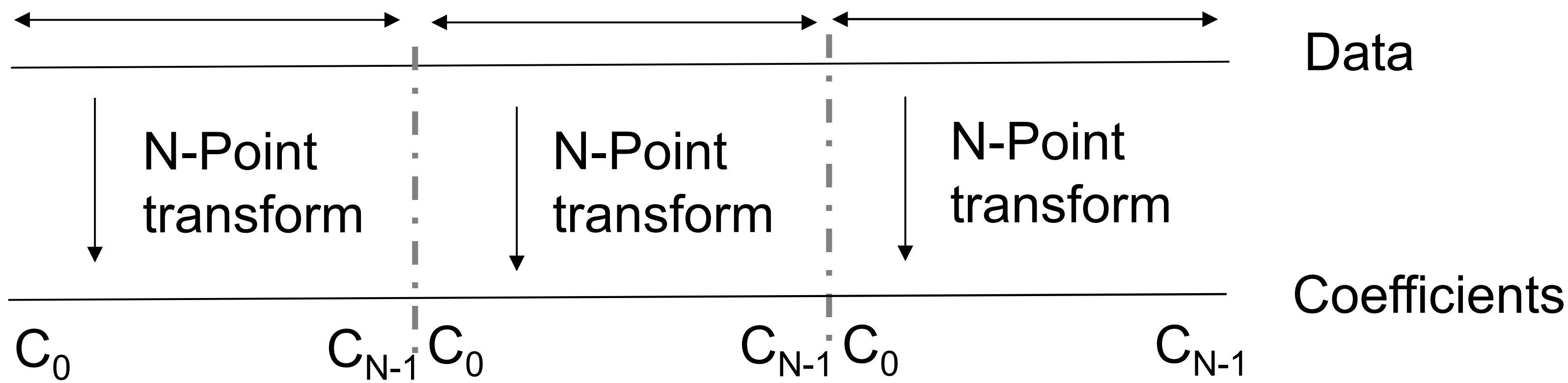
2 Walsh Hadamard Transform – Square waves as bases

3 Wavelet Transforms - Short localised waves as bases

Transforms

- An N-point transform
  - Contains N basis functions
  - When applied on N data points, results in N coefficients.

- If the length of data (L) is larger than N,
  - First the data is partitioned into segments with N data points
  - and then each segment is transformed using the N-point transform.



Filter Banks

- There are many applications where it is desirable to separate a signal into a set of subband signals, each occupying a portion (subband) of the original frequency range.

- Each subband is processed independently.

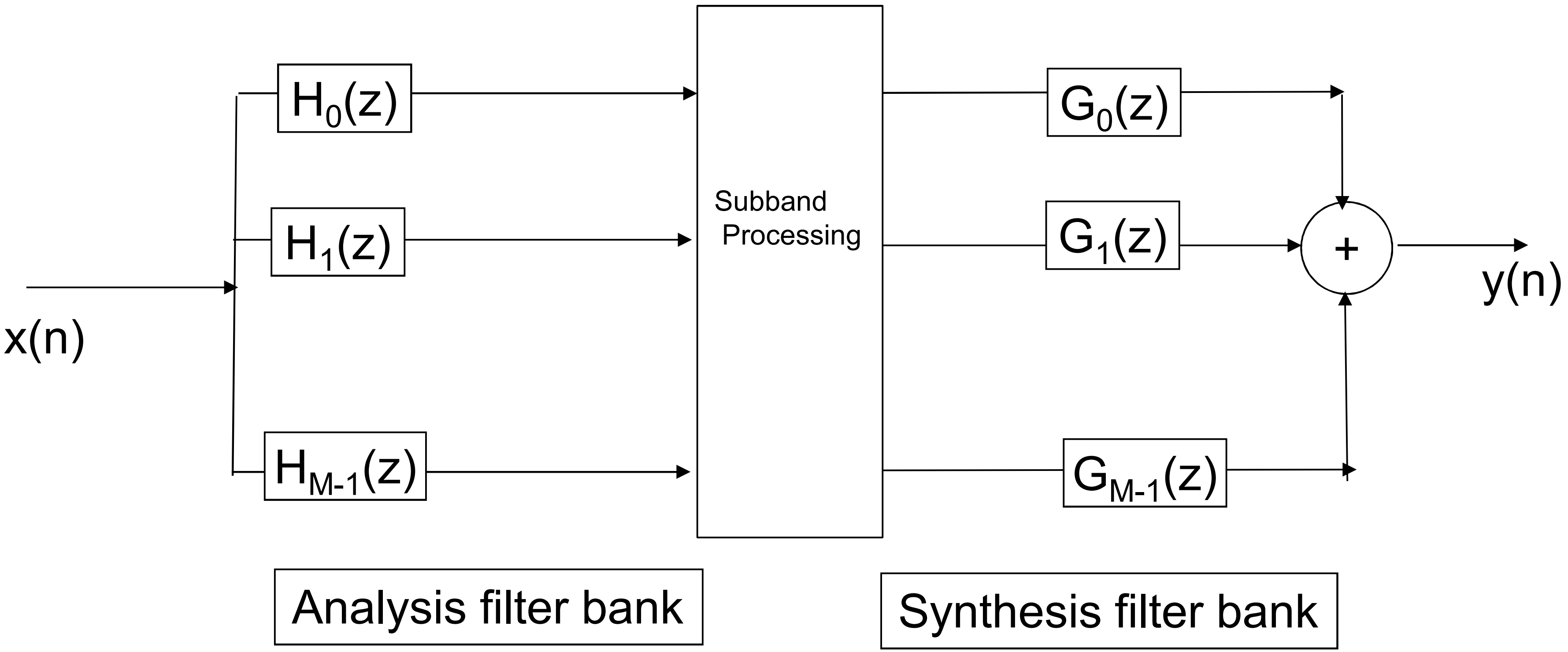
- In some applications, it may be necessary to recombine the subband signals into a single composite signal occupying the whole Nyquist range.

- Example applications:
  - Subband coding of speech signal and images
  - Spectrum analysis and signal synthesis
  - Frequency division multiplexing

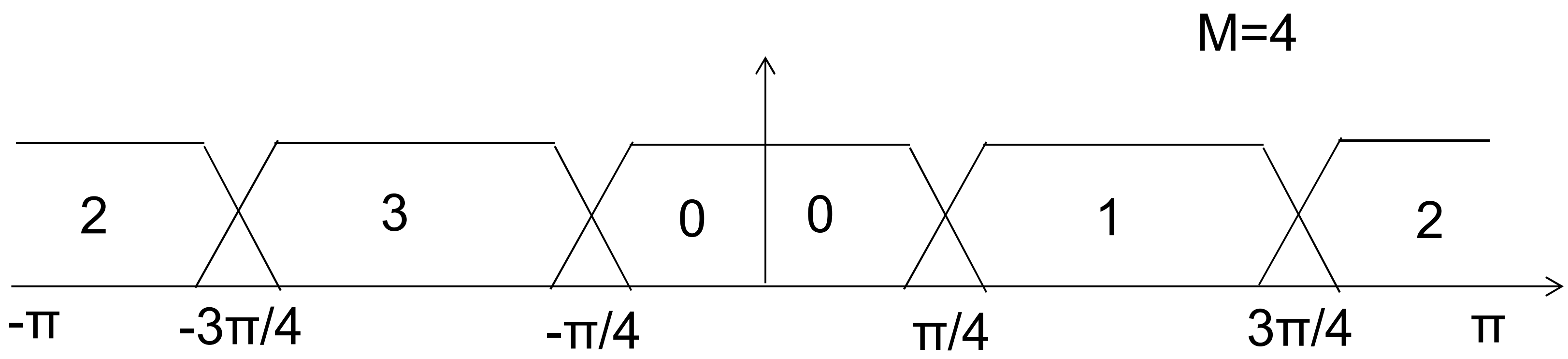
Analysis and synthesis filter banks

- *An analysis filter bank* is a set of M parallel bandpass filters which separate an input signal into M subbands signals.

- *A synthesis filter bank* is a set of M parallel bandpass filters whose outputs are combined together to form a single composite signal.

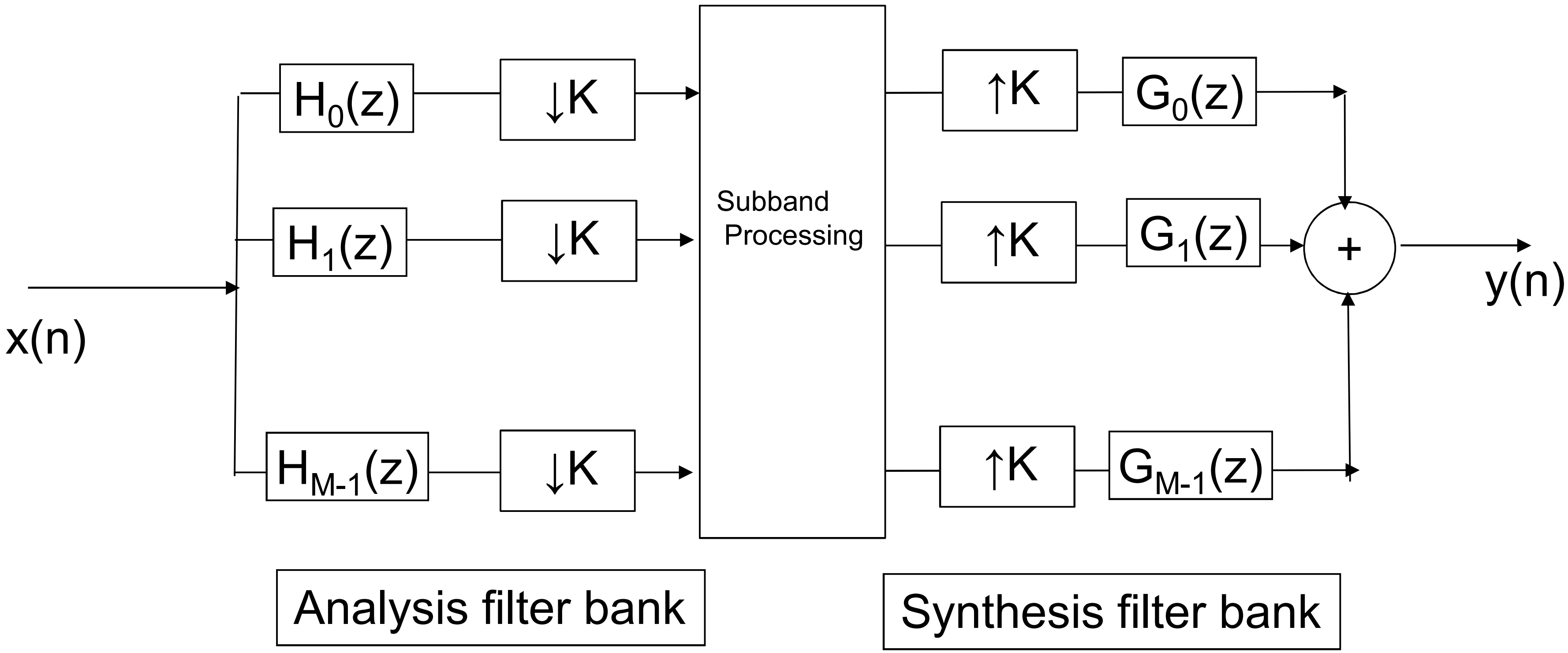


- The following points are worth noting:
  - 1. The zeroth subband usually occupies the range [$-\pi/M$, π/M], and the filters $h_0(n)$ and $g_0(n)$ have real coefficients.
  - 2. All other frequency subbands are nonsymmetric; hence the corresponding filters have complex coefficients.
  - 3. For M even, the subband M/2 is split evenly between positive and negative frequencies.



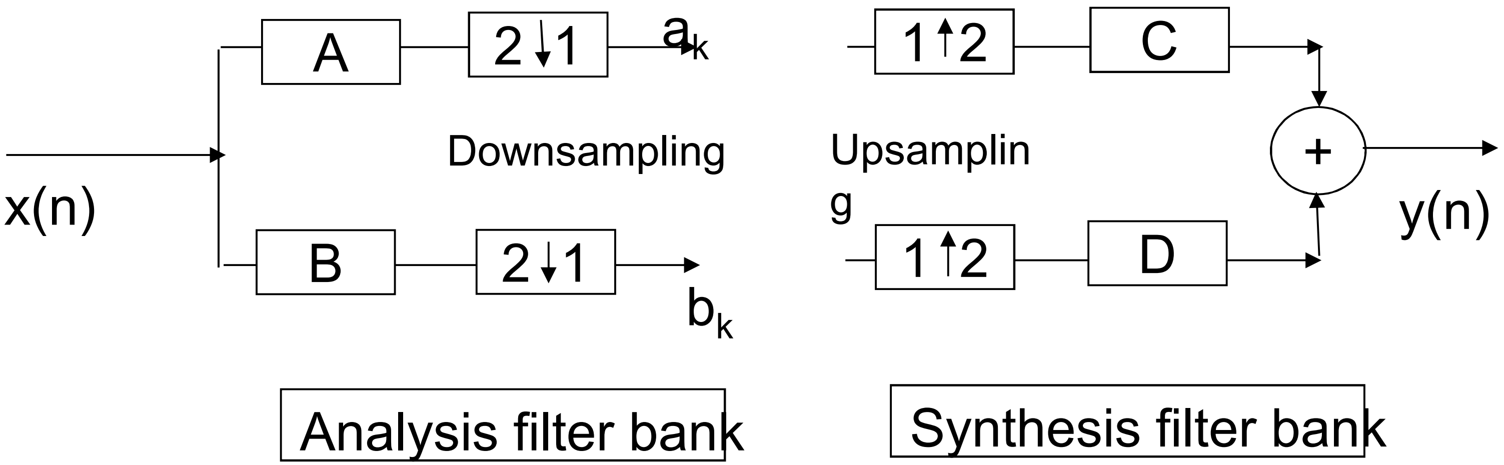
Decimated filter banks

- For an uniform filter bank, each subband has a width of $2\pi/M$.
- Thus, each subband signal can be decimated by a factor $K \le M$.
- If $K=M$, we have a maximally decimated filter bank.



Two Channel Filter Banks

- The simplest filter bank containing two channels is shown below.



- A = $H_0(z)$; B = $H_1(z)$; C = $G_0(z)$; D = $G_1(z)$; What filter types are they?
- Filters are generally not perfect reconstructing (not invertible or not lossless).
- However, using filter banks (i.e., a bank of filters), results in low complexity transforms giving perfect reconstruction.
- The forward transform is obtained by the “analysis filter bank”
- The inverse transform is realised by the “synthesis filter bank”
- Derive the conditions for perfect reconstruction (x(n) = y(n))

Two Channel Filter Banks

- Consider the z-transform representation
  - Input signal $X(z)$
  - Outout signal $Y(z)$
  - The filters $A(z)$, $B(z)$, $C(z)$ and $D(z)$
  - The Downsampling operator $\boxed{2 \downarrow 1}$ (2:1 downsampling) on an input signal $F(z)$ is $F(z^2)$.
  - The Interpolation operator $\boxed{1 \uparrow 2}$ (1:2 upsampling) of an input signal $F(z)$ is $\frac{1}{2}[F(z^{1/2})+F(-z^{1/2})]$

- Now for the filter bank: For the upper banch:
  - After the low pass filter A: $A(z)X(z)$
  - After downsampling: $A(z^2)X(z^2)$
  - After Upsampling: $\frac{1}{2}[A(z)X(z) + A(-z)X(-z)]$
  - After the filter C: $\frac{1}{2}C(z)[A(z)X(z) + A(-z)X(-z)]$ -----(1)

- Similarly for the lower Branch
  - We can write: $\frac{1}{2}D(z)[B(z)X(z) + B(-z)X(-z)]$ ------(2)

- Now by (1)+(2) we can get $Y(z)$

Two Channel Filter Banks

- The output of the filter bank

  $Y(z) = ½\, C(z)[A(z)X(z) + A(-z)X(-z)] + ½\, D(z)[B(z)X(z) + B(-z)X(-z)]$

  $½\ [A(z)C(z)+B(z)D(z)]\ X(z)\ + ½\ [A(-z)C(z)+B(-z)D(z)]\ X(-z)$ ----------(3)

- For the Perfect Reconstruction (PR)

  $A(z)C(z) + B(z)D(z) = 2z^{-l}$ For no distortion (i.e., the Coefficient of X(z)=1)

  $A(-z)C(z) + B(-z)D(z) = 0$ For no aliasing (i.e, the Coefficient of X(-z)=0)

- To satisfy the PR conditions, choose C(z)=B(-z), D(z)=-A(-z)
- and choose B(z) as the corresponding high pass filter of the low pass filter A(z)

A= a, b, c, d

Alternating flip

B= d, -c, b, -a

Alternating signs

C= d, c, b, a

D= -a, b, -c, d

- That means if we know A(z), we can find the other 4 filters.

Two Channel Filter Banks

- Filter Bank design Criteria:
- Let's say the low pass filter A has the coefficients: $\{h_0, h_1, h_2, \ldots\}$

- (1) Orthogonality condition for the filter bank:

$$\sum_i h_i h_{i+2k} = \delta_{0k}$$

  – We only require to retain the orthogonality only for double shifts of the filter (why?)

- (2) Regularity condition for the filter bank:
  – B is a high pass filter. So its coefficients add up to zero. This requirement and the Perfect Reconstruction condition mean,

$$\sum_i h_i = \sqrt{2}$$

- We can use these two conditions to design filter banks:
  – Exercise: Design length N=2, N=3 and N=4 two-channel filter banks

Two Channel Filter Banks

- Length N=2 filter bank

$$A = \{h_0, h_1\}$$

(1)

$$\sum_i h_i h_{i+2k} = \delta_{0k}$$

$$-> k = 0: \quad h_0^{\,2} + h_1^{\,2} = 1$$

(2)

$$\sum_i h_i = \sqrt{2}: \quad h_0 + h_1 = \sqrt{2}$$

$$h_0 = h_1 = \frac{1}{\sqrt{2}}$$

Length N=3 filter bank

$$A = \{h_0, h_1, h_2\}$$

(1)

$$\sum_i h_i h_{i+2k} = \delta_{0k}$$

$$-> k = 0: \quad h_0^{\,2} + h_1^{\,2} + h_2^{\,2} = 1$$

$$-> k = 1: \quad h_0 h_2 = 0$$

(2)

$$\sum_i h_i = \sqrt{2}: \quad h_0 + h_1 + h_2 = \sqrt{2}$$

$$h_2 = 0; \quad h_0 = h_1 = \frac{1}{\sqrt{2}};$$

- Not possible to design odd length filter banks.

- Homework: For N=2, filter bank, verify the perfect reconstruction for the input data sequence X={ 0 1 2 3 4 0}

Two Channel Filter Banks

- Length N=4 filter bank

$$A = \{h_0, h_1, h_2, h_3\}$$

$$(1) \qquad \sum_i h_i h_{i+2k} = \delta_{0k}$$

$$-> k = 0: \quad h_0^{\ 2} + h_1^{\ 2} + h_2^{\ 2} + h_3^{\ 2} = 1$$

$$-> k = 1: \quad h_0 h_2 + h_1 h_3 = 0$$

$$(2) \qquad \sum_i h_i = \sqrt{2}: \quad h_0 + h_1 + h_2 + h_3 = \sqrt{2}$$

- 3 equations & 4 unknowns. So there is one free choice. We can use this to optimise the filter bank performance. (Daubechies - 4 filter bank is one example).
- Can we make the filter symmetric?
  - i.e., $h_0 = h_3$ and $h_1 = h_2$ in N=4

$$-> k = 1: \quad h_0 h_2 + h_1 h_3 = 0$$

$$h_0 h_1 + h_1 h_0 = 0$$

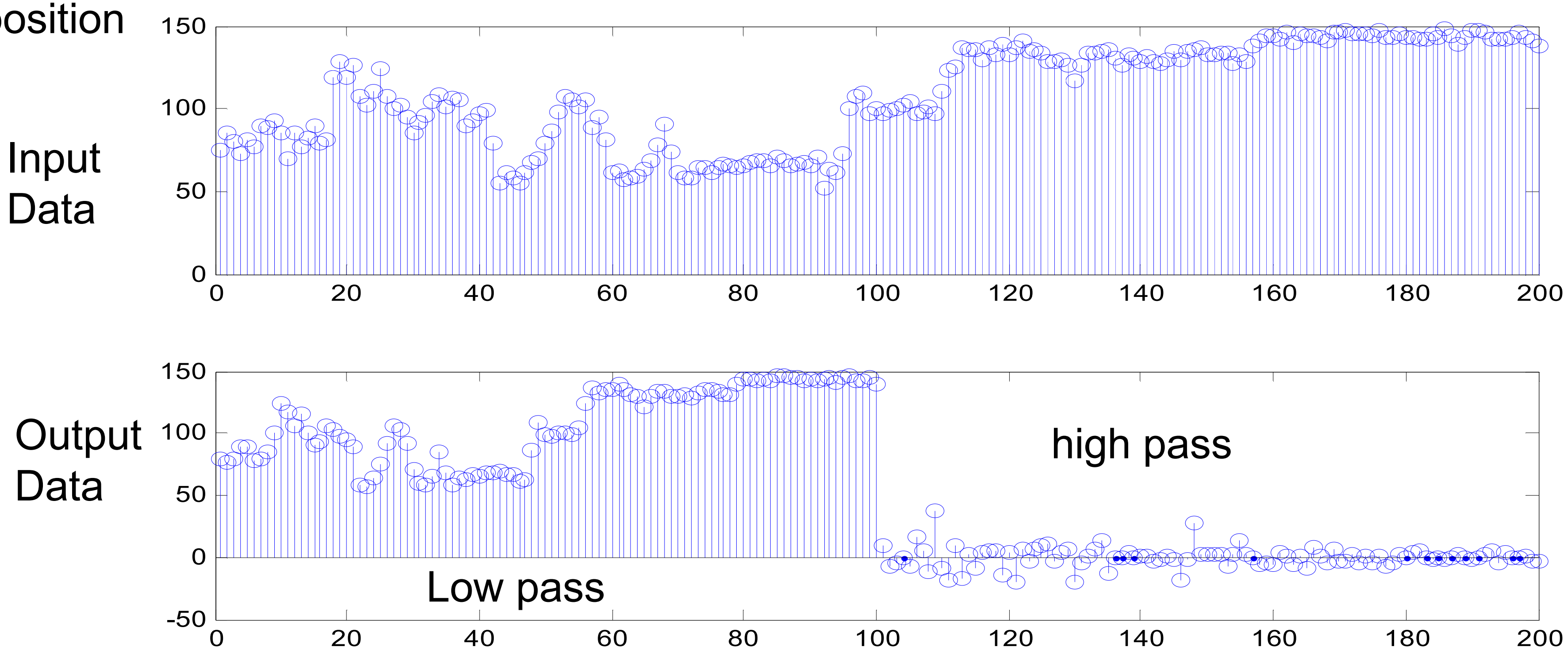
$$2 h_1 h_0 = 0$$

- Filters in orthogonal filter banks can't be symmetric. Therefore, they have phase distortion (no linear phase response).

Two Channel Filter Banks

- A 2- channel filter bank decomposes data into two sub bands low pass an high pass.
- Non-expanding --- i.e., The length of output data = The length of input data
- Remember for filters, The length of output data= The length of input data + The length of filter -1.

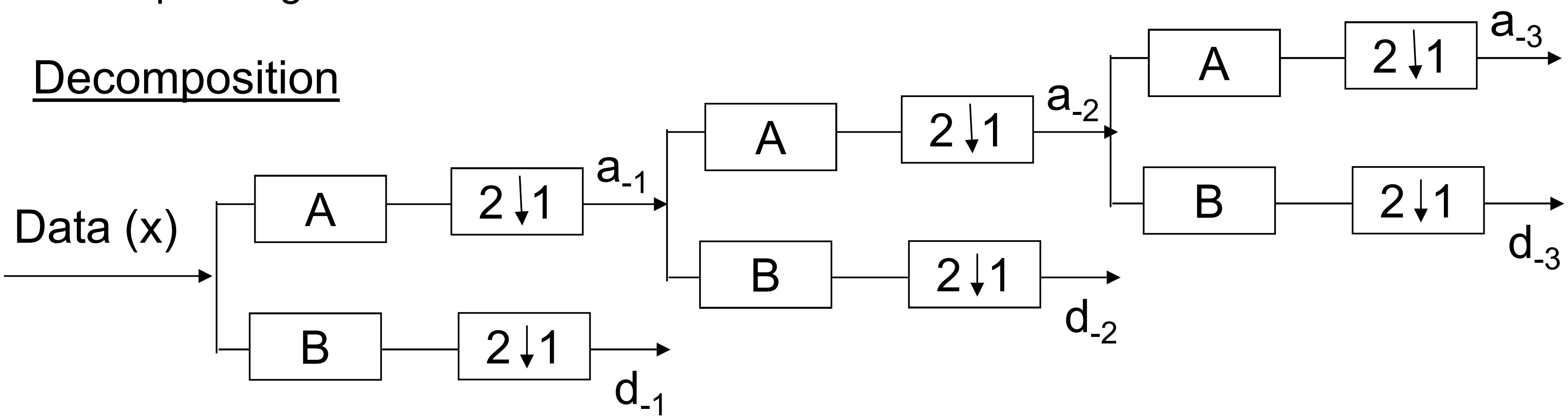
- The low pass signal looks the same as the original (only smoothed)
- The filter bank can be applied repeatedly on the low pass signal. This is called Dyadic decomposition



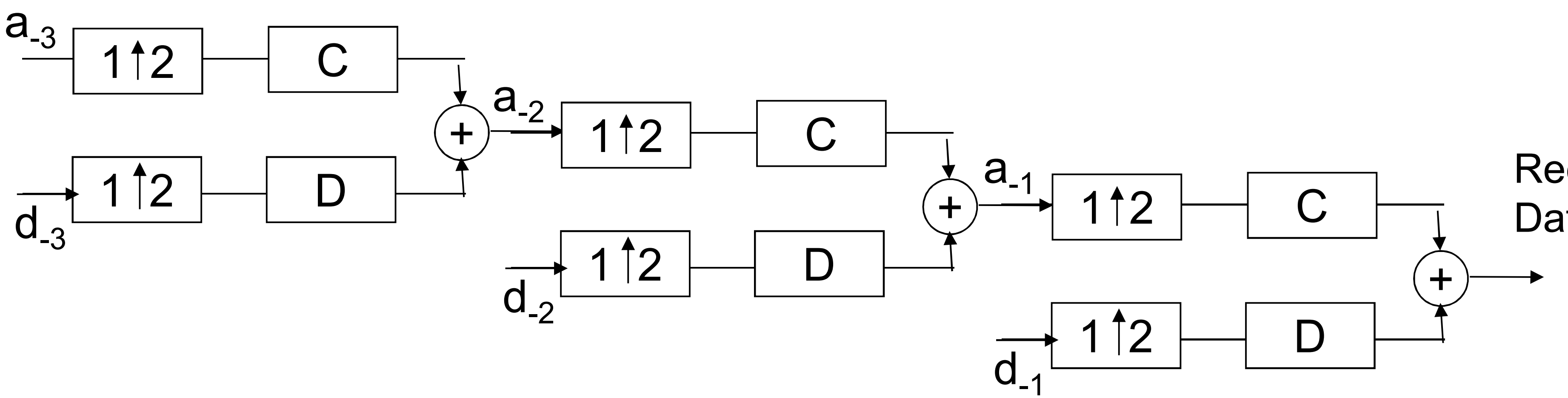
Two Channel Filter Banks

- Dyadic Decomposition (Draw a diagram for a 3-level dyadic decomposition) and its corresponding reconstruction.

<u>Decomposition</u>

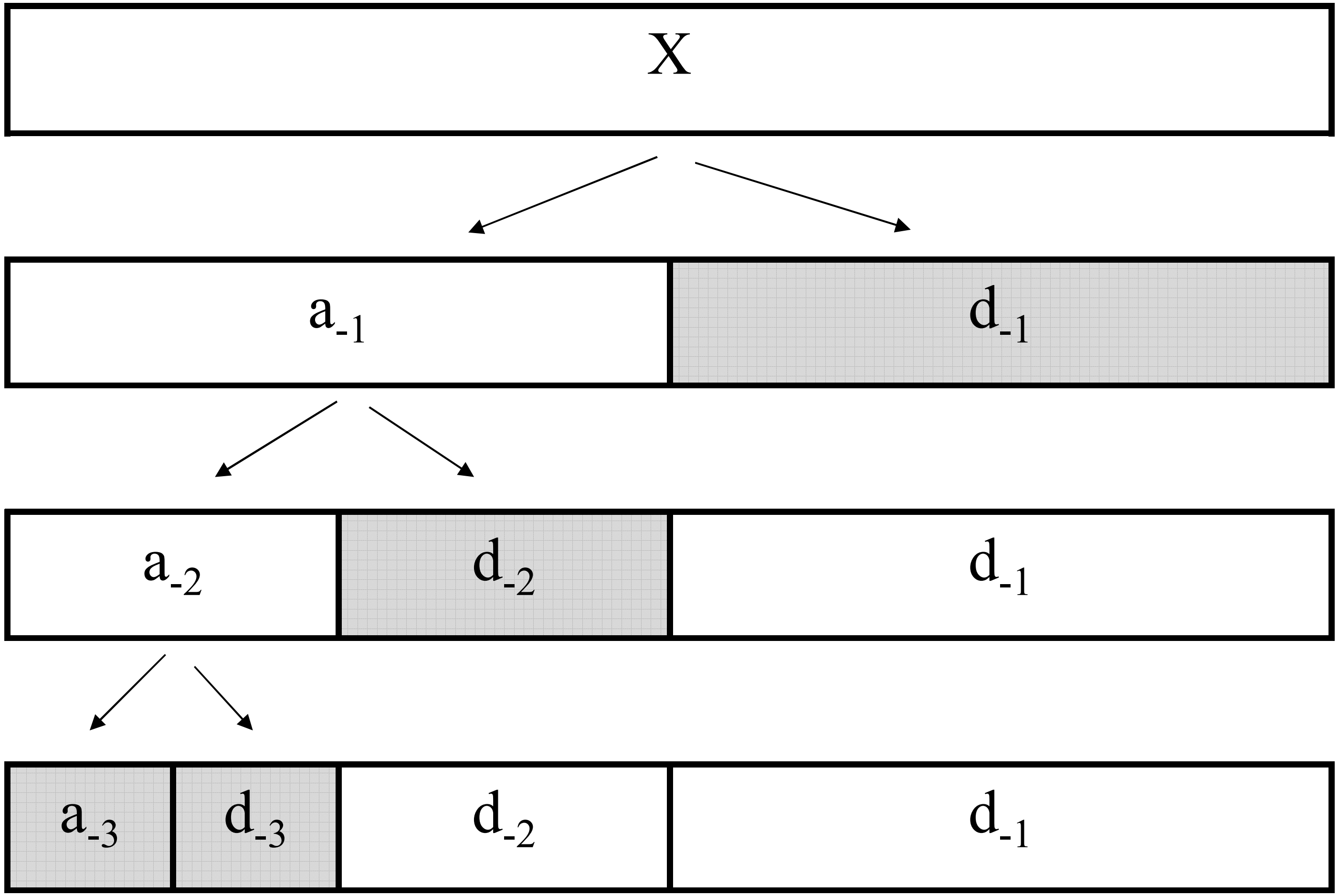


<u>Reconstruction</u>



Two Channel Filter Banks

- Dyadic Decomposition



a_n = Low pass filtered data
d_n = High pass filtered data

Wavelet Transforms

- A wavelet is a short localised wave used as a basis function in a wavelet transform.
- What are the basis functions used in the Fourier transform?
- In a wavelet transform the main wavelet, usually called mother wavelet (w(n)), is defined first.
- Then in the transformation, the mother wavelet is scaled (by a factor s) or translated by k points to obtain the other wavelets $w_{(s,k)}(n)$ as basis functions:

$$w_{(s,k)}(n) = 2^{s/2} w(2^s n - k)$$

- Wavelet transforms can be implemented using two different methods:
  - Filter banks (In the current topic)
  - Lifting (in EEE6081)

- It can be shown that a wavelet is the impulse response of the high pass filter of the inverse filter bank iterated to infinity. ☹

[Beyond the scope of this module ☺]

Wavelet Transforms

- Wavelet basis functions: $$w_{(s,k)}(n) = 2^{s/2} w(2^s n - k)$$

- The translation by a factor k
  - corresponds to the location of the wavelet
  - the high pass filter operation in the filter bank (Convolution) corresponds to this.

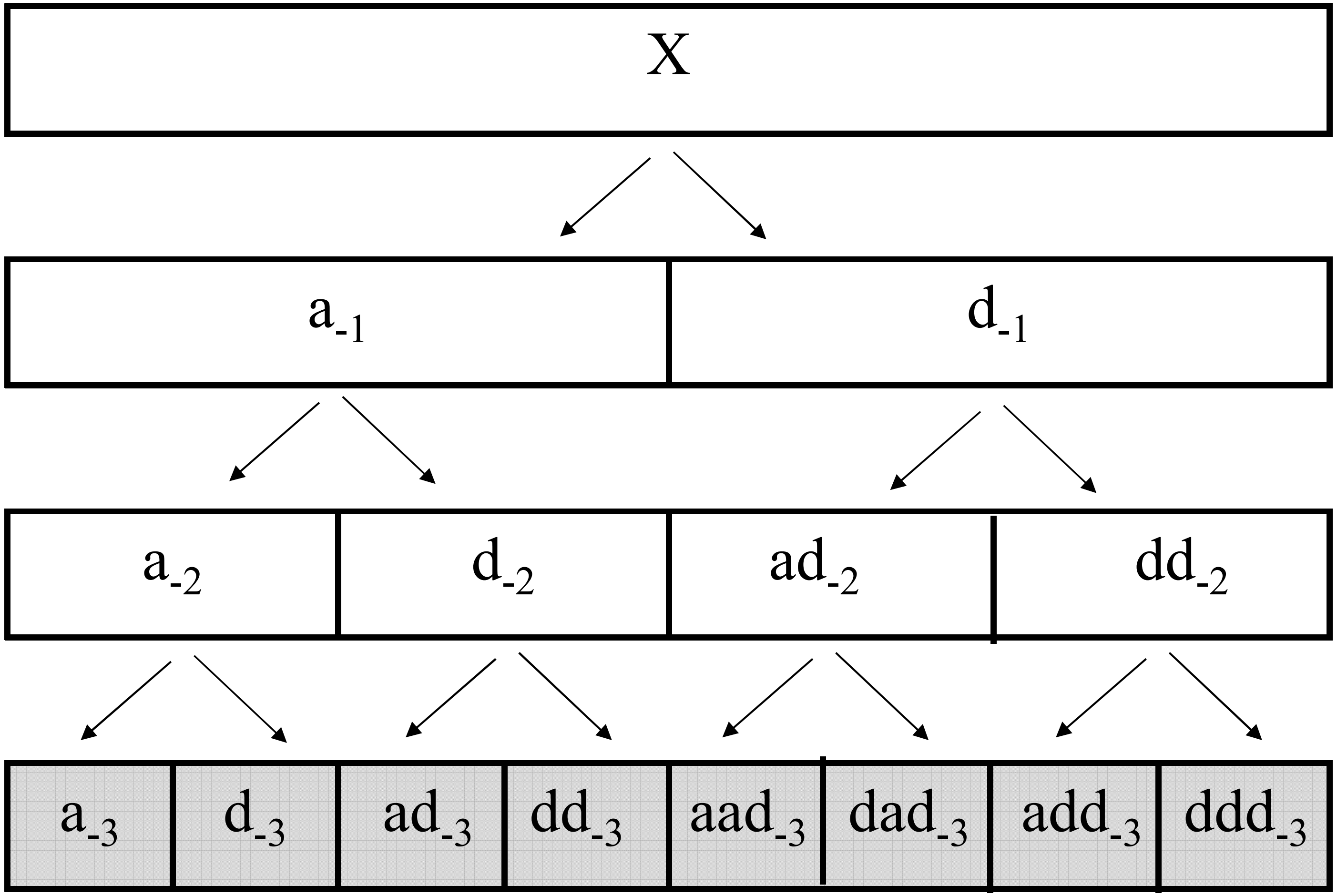
- The scaling of the mother wavelet by a factor s is represented in the filter bank
  - when the high pass filter is applied on the output of one level of decomposition.
  - And corresponds to wavelet operation on the down-sampled low passed signal.

- Why do we need a low pass filter in the filter bank?

- Different forms of wavelet decomposition schemes
  - Dyadic wavelet transform (Using the dyadic filter bank decomposition)
  - Wavelet Packet transform (either as a full tree or an optimum tree decomposition)

Wavelet Transforms

- Full tree wavelet packet transform:
  - Both the low pass and high pass sub bands are decomposed further following a complete binary tree:

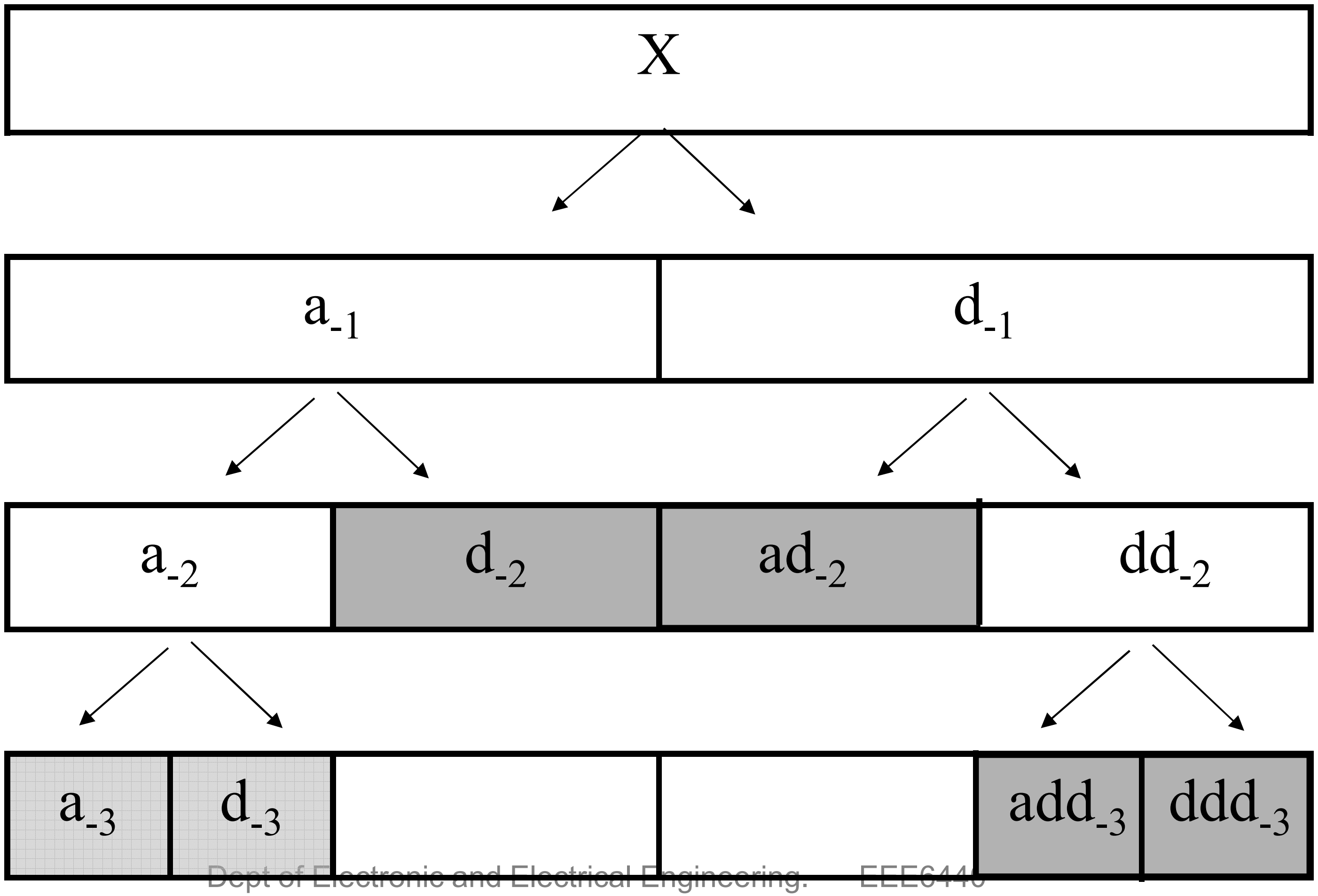


Wavelet Transforms

- Full tree wavelet packet transform:
  - Draw the corresponding filter bank based realisation for the forward transform (decomposition) and the inverse transform (reconstruction)

Wavelet Transforms

- Wavelet packet transform :
  - Can have various decompositions
  - An example:



Wavelet Transforms

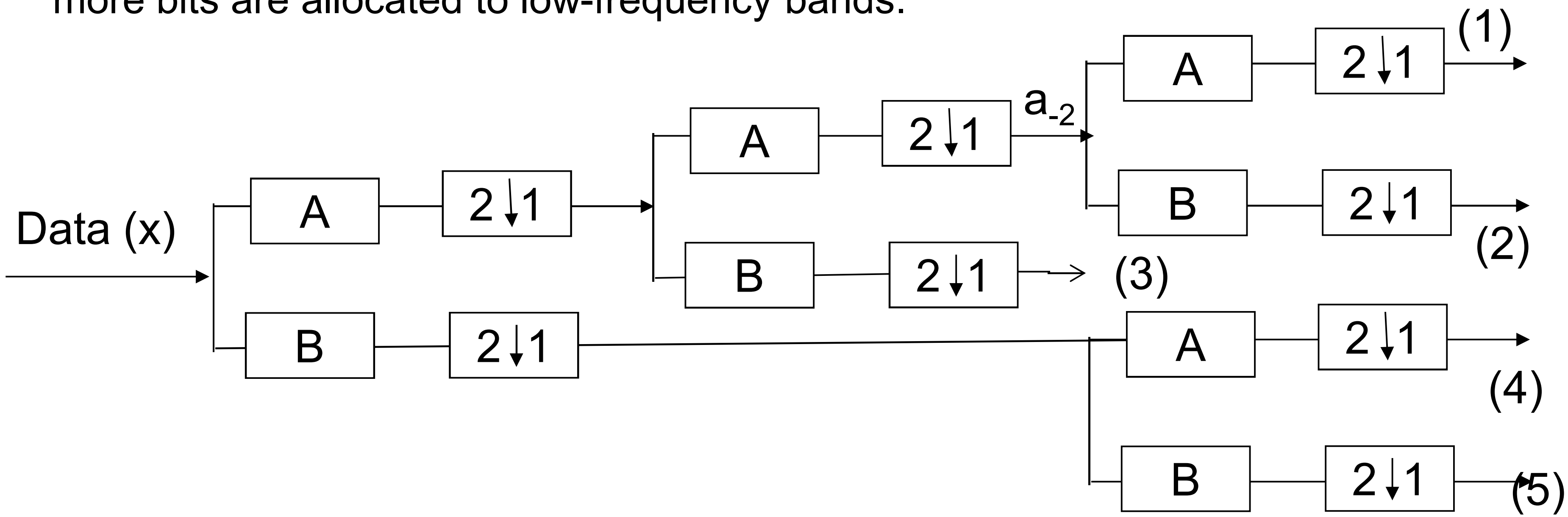
- Wavelet packet transform (With the optimum tree):
  - Draw the corresponding filter bank based realisation for the forward transform (decomposition) and the inverse transform (reconstruction)

Wavelet Transforms

- Frequencies shown in each sub band
  - For the Fourier transform, we know a signal in time domain representation is transformed and shown in frequency domain.
    - e.g. The axes are “time” and “frequency” in the 2 domains.
- But for the Wavelet transform,
  - It shows a joint time-frequency (or space-frequency) representation.
    - The original signal (a_0) represents the full resolution signal with all normalised frequencies 0 – π. High spatial resolution and low frequency resolution.
    - The First level decomposition:
      - ½ spatial resolution
      - (a_{-1}) represents frequencies 0 – π/2.
      - (d_{-1}) represents frequencies π/2 – π.
      - i.e, low spatial resolution, but high frequency resolution.
    - Similarly for level 2 of the decomposition.
      - ¼ spatial resolution
      - (a_{-2}) represents frequencies 0 – π/4.
      - (d_{-2}) represents frequencies π/4 – π/2.
    - These define the bandwidths of the filters A and B.

Subband Coding of Speech

- Speech is typically sampled at 8kHz or higher
- The bit rate of a speech signal at 8 KHz and quantized to 8 bits is 64000 bits/sec.
- Speech signals are encoded for efficient transmission and storage.
- The compression ration is defined as the ratio of the bit rate before compression to the bit rate after compression
- In subband coding, the speech signal is divided into several subbands and each subband is encoded differently.
- Most of the energy in the speech signal is contained in the low frequencies; thus, more bits are allocated to low-frequency bands.



Subband Coding of Speech

- Draw the resulting in subband partitioning indicating the bandwidth of the subbands.

- Compute the final bitrate and the compression ratio for the following two strategies of bit allocation

-

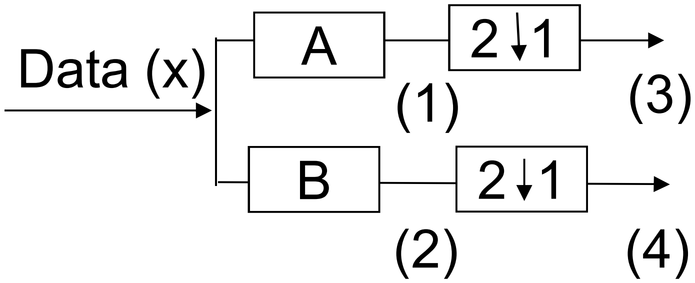
Subband number	**1**	**2**	**3**	**4**	**5**
Strategy 1	5 bits /sample	5 bits /sample	4 bits /sample	3 bits /sample	3 bits /sample
Strategy 2	5 bits /sample	4 bits /sample	3 bits /sample	2 bits /sample	1 bits /sample

Matrix-based implementation

- [Y] = [T][X]
- [X] is the input data vector (Nx1)
- [Y] is the output data vector (Nx1)
- [T] is the transform matrix (NxN)

- Consider the low pass filter $A = \{p, q\}$
- And the high pass filter $B = \{r, s\}$

- Write T for level 1 transform. Assume N=4



At (1)

$$\begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \end{bmatrix} = \begin{bmatrix} q & p & 0 & 0 \\ 0 & q & p & 0 \\ 0 & 0 & q & p \\ 0 & 0 & 0 & q \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

At (2)

$$\begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ h(3) \end{bmatrix} = \begin{bmatrix} s & r & 0 & 0 \\ 0 & s & r & 0 \\ 0 & 0 & s & r \\ 0 & 0 & 0 & s \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

At (3)

$$\begin{bmatrix} a(0) \\ a(2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \end{bmatrix}$$

At (4)

$$\begin{bmatrix} d(0) \\ d(2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} d(0) \\ d(1) \\ d(2) \\ d(3) \end{bmatrix}$$

From (1) and (3), At (3)

$$\begin{bmatrix} a(0) \\ a(2) \end{bmatrix} = \begin{bmatrix} q & p & 0 & 0 \\ 0 & 0 & q & p \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

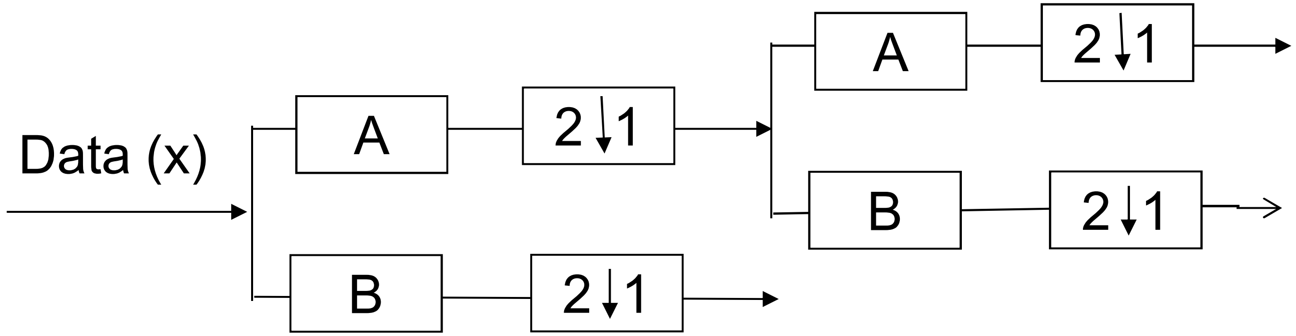
From (2) and (4), At (4)

$$\begin{bmatrix} d(0) \\ d(2) \end{bmatrix} = \begin{bmatrix} s & r & 0 & 0 \\ 0 & 0 & s & r \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

Combining above 2 together

$$\begin{bmatrix} a(0) \\ a(2) \\ d(0) \\ d(2) \end{bmatrix} = \begin{bmatrix} q & p & 0 & 0 \\ 0 & 0 & q & p \\ s & r & 0 & 0 \\ 0 & 0 & s & r \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

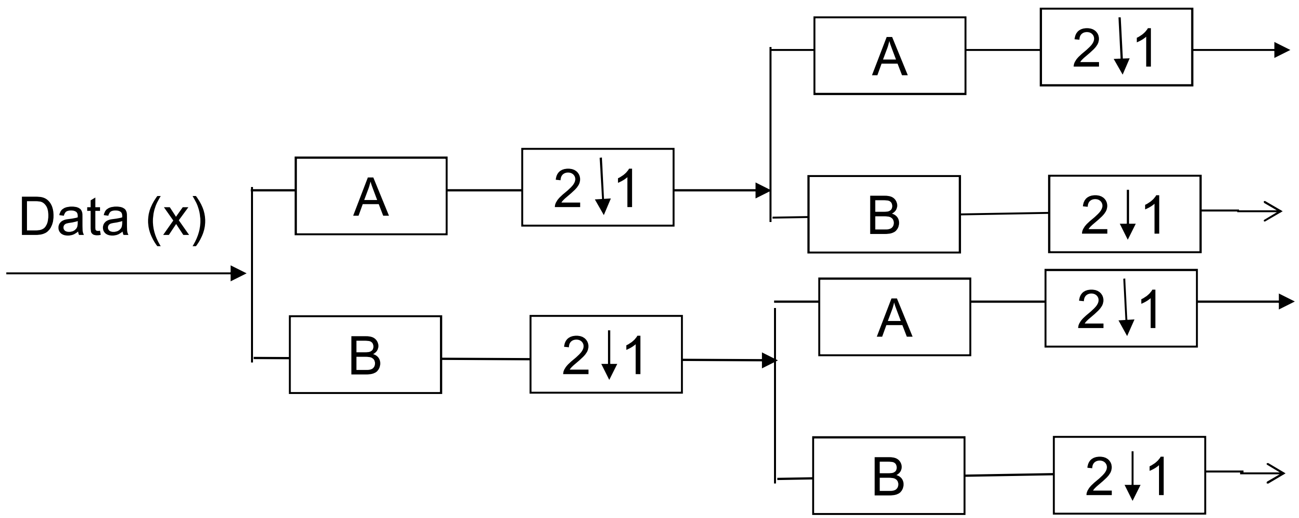
- How do you extend it for level 2 for a dyadic decomposition?



Only a(0) and a(2) are processed. d(0) and d(2) are not processed.

$$\begin{bmatrix} aa(0) \\ ad(0) \\ d(0) \\ d(2) \end{bmatrix} = \begin{bmatrix} q & p & 0 & 0 \\ r & s & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a(0) \\ a(2) \\ d(0) \\ d(2) \end{bmatrix}$$

- How do you extend it for level 2 for a dyadic decomposition?



All a(0), a(2), d(0) and d(2) are processed.

$$\begin{bmatrix} aa(0) \\ ad(0) \\ da(0) \\ dd(0) \end{bmatrix} = \begin{bmatrix} q & p & 0 & 0 \\ r & s & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & r & s \end{bmatrix} \begin{bmatrix} a(0) \\ a(2) \\ d(0) \\ d(2) \end{bmatrix}$$

The University Of Sheffield.

- Find the corresponding matrices for the inverse transform?