MULTIRATE DIGITAL SIGNAL PROCESSING

Signal processing can be enhanced by changing sampling rate:

• Up-sampling before D/A conversion in order to relax requirements of analog antialiasing filter. Cf. audio CD, where the sampling frequency 44.1 kHz is increased fourfold to 176.4 kHz.

• Need to connect systems in digital (audio) signal processing which operate at different sampling rates.

• Decomposition of a signal into $M$ components corresponding to various frequency bands. If original signal is sampled at the sampling frequency $f_s$ (with frequency band of width
\(fs/2\), the components then contain a frequency band of width \(\frac{1}{2}fs/M\), and can be represented using the sampling rate \(fs/M\). This allows for:

- parallel signal processing with processors operating at lower sampling rates.
- data compression in subband coding, by representing different frequency band components with different word lengths (e.g., in speech processing).
- implementation of high-performance filtering operations with a very narrow transition band. Decomposing the signal into passband, stopband and transition band components, each component can be processed at a lower rate, and the transition band will be less narrow. This results in significant simpler filter complexity.
Basic multirate signal processing operations

Sampling rate conversion

Decimation

\[ y(m) = x(mM) \quad \text{(every } M\text{th element of } x) \]

Matlab routines:

\[ y = \text{downsample}(x, M) \]

\[ y = \text{decimate}(x, M) \text{ applies antialiasing lowpass filtering followed by downsampling} \]
Spectrum of decimated signal:

For example, for $M = 2$,
- spectrum of $\{x(n)\}$ consists of frequencies in $[0, \pi]$, 
- spectrum of $\{y(m)\}$ consists of frequencies in $[0, \pi/2]$, given by

$$Y(\omega) = \frac{1}{2} (X(\omega/2) + X(\pi - \omega/2)^*)$$

The second term is due to frequency folding, factor $1/2$ is due to sampling rate conversion

This implies that if $x$ is bandlimited; $X(\omega) = 0, \pi/2 \leq \omega \leq \pi$, then

$$Y(\omega) = \frac{1}{2} X(\omega/2)$$
Expansion

\[ y(m) = \begin{cases} 
  x(m/L), & \text{for } m = 0, L, 2L, \ldots \\
  0, & \text{otherwise} 
\end{cases} \]

Expansion followed by interpolation is equivalent to sampling rate increase by factor \( L \).

Matlab routine:

\[ y = \text{upsample}(x, L) \]
Spectrum of expanded signal:

For example, for \( M = 2 \),

- spectrum of \( \{x(n)\} \) consists of frequencies in \([0, \pi]\)
- spectrum of \( \{y(m)\} \) consists of frequencies in \([0, \pi]\), given by

\[
Y(\omega) = \begin{cases} 
X(2\omega), & \text{for } 0 \leq \omega \leq \pi/2 \\
X(2(\pi - \omega))^* \ [= Y(\pi - \omega)^*], & \text{for } \pi/2 \leq \omega \leq \pi 
\end{cases}
\]

It is seen that \( Y(\omega) \) is uniquely defined by its value in frequency band \([0, \pi/2]\)

Correct interpolation is achieved by a low-pass filter which eliminates the frequencies in the band \( \pi/2 \leq \omega \leq \pi \).
Sampling rate conversion by non-integer factors

Sampling rate conversion with factor $F = \frac{L}{M}$ can be performed by applying:

- expansion by factor $L$
- low-pass filtering (corresponding to interpolation)
- decimation by factor $M$

Matlab routine:

$y=\text{resample}(x,L,M)$
Analysis and synthesis filter banks

Simple Analysis filter bank:

- $H_1(z)$: low-pass filter to extract subband $[0, \pi/2]$
- $H_2(z)$: high-pass filter to extract subband $[\pi/2, \pi]$
- subbands components can be processed using half the sampling rate
Analysis filter bank with three subbands:

- $x_{D1}$: subband component $[0, \pi/2]$
- $x_{D2}$: subband component $[\pi/2, \pi]$
- $x_{D21}$: subband component $[\pi/2, 3\pi/4]$
- $x_{D22}$: subband component $[3\pi/4, \pi]$
Simple multirate signal processing system

Analysis filter bank

Synthesis filter bank

- subband components $x_{D1}, x_{D2}$ are processed independently at lower sampling frequency

- processed components are upsampled to original sampling frequency

- synthesis filters $G_1(z), G_2(z)$ remove alias components

- filtered components are combined back to give the processed signal $y$
Subband decomposition

Decomposition into low-frequency component and high-frequency component

- $H_1(z)$: low-pass filter to extract subband $[0, \pi/2]$
- $H_2(z)$: high-pass filter to extract subband $[\pi/2, \pi]$
Brickwall filters:

- $H_1$ ideal low-pass filter
- $H_2$ ideal high-pass filter
**Real filters:** frequency bands cannot be exactly separated

- $H_1$ real low-pass filter
- $H_2$ real high-pass filter

**Frequency response**
Solution:

Construct $H_1, H_2, G_1, G_2$ so that perfect reconstruction is achieved (up to a constant and time delay):

$$y(n) = K x(n - P)$$
Here:
\[ \{x_{D1}(n)\} = \{x_1(0), x_1(2), x_1(4), \ldots\} \]
and
\[ \{v_1(n)\} = \{x_{D1}(0), 0, x_{D1}(1), 0, x_{D1}(2), 0, \ldots\} \]
\[ = \{x_1(0), 0, x_1(2), 0, x_1(4), 0, \ldots\} \]

But as
\[ \hat{x}_1(\pm z) = x_1(0) \pm x_1(1)z^{-1} + x_1(2)z^{-2} \pm x_1(3)z^{-3} + \cdots \]
we have
\[ \hat{v}_1(z) = v_1(0) + v_1(1)z^{-1} + v_1(2)z^{-2} + v_1(3)z^{-3} + \cdots \]
\[ = x_1(0) + x_1(2)z^{-2} + x_1(3)z^{-4} + \cdots \]
\[ = \frac{1}{2}(\hat{x}_1(z) + \hat{x}_1(-z)) \]

Similarly: \[ \hat{v}_2(z) = \frac{1}{2}(\hat{x}_2(z) + \hat{x}_2(-z)) \]
We have the relations

\[ \hat{v}_1(z) = \frac{1}{2} \left( H_1(z) \hat{x}(z) + H_1(-z) \hat{x}(-z) \right) \]

\[ \hat{v}_2(z) = \frac{1}{2} \left( H_2(z) \hat{x}(z) + H_2(-z) \hat{x}(-z) \right) \]

and

\[ \hat{y}(z) = \hat{x}_{E1}(z) + \hat{x}_{E2}(z) \]

\[ = G_1(z) \hat{v}_1(z) + G_2(z) \hat{v}_2(z) \]

\[ \Longrightarrow \]

\[ \hat{y}(z) = \frac{1}{2} \left( G_1(z) H_1(z) + G_2(z) H_2(z) \right) \hat{x}(z) \]

\[ + \frac{1}{2} \left( G_1(z) H_1(-z) + G_2(z) H_2(-z) \right) \hat{x}(-z) \]
Here $\hat{x}(-z)$ is an alias component (Fourier transform $= \hat{x}(-e^{j\omega}) = \hat{x}(e^{j(\omega-\pi)})$)

Alias avoided if:

$$G_1(z)H_1(-z) + G_2(z)H_2(-z) = 0$$

$$\Rightarrow$$

$$G_1(z) = 2H_2(-z)$$
$$G_2(z) = -2H_1(-z)$$

Scaling factor 2 introduced to compensate factor $1/2$ in expression for $\hat{y}(z)$
Typically low- and high-pass filters $H_1, H_2$ are selected so that

$$H_2(z) = H_1(-z)$$

$$\implies \quad H_2(e^{j\omega}) = H_1(-e^{j\omega}) = H_1(e^{j(\pi-\omega)})^*$$

$$\implies \quad \text{Frequency responses are mirror images about the quadrature frequency } \pi/2 = 2\pi/4 \quad (\text{quadrature mirror filters})$$

It follows that

$$G_1(z) = 2H_2(-z) = 2H_1(z)$$
$$G_2(z) = -2H_1(-z) = -2H_2(z)$$
Introducing $G_1, G_2$ and $H_2(z) = H_1(-z)$ into

\[ \hat{y}(z) = \frac{1}{2} \left( G_1(z)H_1(z) + G_2(z)H_2(z) \right) \hat{x}(z) + \frac{1}{2} \left( G_1(z)H_1(-z) + G_2(z)H_2(-z) \right) \hat{x}(-z) \]

gives

\[ \hat{y}(z) = \left( H_1(z)^2 - H_1(-z)^2 \right) \hat{x}(z) \]

\[ \implies \text{Condition for perfect signal recovery} \]

\[ H_1(z)^2 - H_1(-z)^2 = K z^{-P} \]
Haar filters

Only quadrature mirror FIR filters which satisfy perfect signal recovery:

\[ H_1(z) = \frac{1}{2} (1 + z^{-1}), \quad G_1(z) = 1 + z^{-1} \]

\[ H_2(z) = \frac{1}{2} (1 - z^{-1}), \quad G_2(z) = -1 + z^{-1} \]

giving \( \hat{y}(z) = z^{-1} \hat{x}(z) \), or \( y(n) = x(n - 1) \)

Alternatively, when noncausal filters can be used,

\[ H_1(z) = \frac{1}{2} (z + 1), \quad G_1(z) = 1 + z^{-1} \]

\[ H_2(z) = \frac{1}{2} (z - 1), \quad G_2(z) = -1 + z^{-1} \]

giving \( \hat{y}(z) = \hat{x}(z) \), i.e., \( y(n) = x(n) \)
Subband coding and multiresolution analysis

Pyramid algorithm:

- $x$: frequency band $[0, \pi]$, length $N$
- $x_H$: frequency band $[\pi/2, \pi]$, length $N/2$
- $x_{LH}$: frequency band $[\pi/4, \pi/2]$, length $N/4$
- $x_{LLL}$: frequency band $[\pi/8, \pi/4]$, length $N/8$
- $x_{LLL}$: frequency band $[0, \pi/8]$, length $N/8$
Example \((N = 8)\):

\[
\begin{array}{c|c|c|c|c}
\text{Time} & \text{Frequency} & \pi & \pi/2 & \pi/4 \\
\hline
N & x_{LLL} & x_{LLH} & x_{LH} & x_H \\
\end{array}
\]

\textit{Time-frequency resolution}

Low-frequency components \(x_{LLL}, x_{LLH}\) have high frequency resolution and low time resolution.

High-frequency component \(x_H\) has low frequency resolution and high time resolution.
The pyramid algorithm generates a transform of input signal
\( \{x(n)\} = \{x(0), x(1), \ldots, x(N - 1)\} \) to the signal transform
(when using three stages as above)

\[
x_{\text{Haar}} = [x_{\text{LLL}}, x_{\text{LLH}}, x_{\text{LH}}, x_{\text{H}}]
\]

where

\[
x_{\text{LLL}} = \{x_{\text{LLL}}(0), \ldots, x_{\text{LLL}}(N/8 - 1)\}
\]
\[
x_{\text{LLH}} = \{x_{\text{LLH}}(0), \ldots, x_{\text{LLL}}(N/8 - 1)\}
\]
\[
x_{\text{LH}} = \{x_{\text{LH}}(0), \ldots, x_{\text{LH}}(N/4 - 1)\}
\]
\[
x_{\text{H}} = \{x_{\text{H}}(0), \ldots, x_{\text{H}}(N/2 - 1)\}
\]
Note: sequences $x$ and $x_{\text{Haar}}$ have both length $N$

Inverse transform:

Use filters $G_1, G_2$ to reconstruct

- $x_{LL}$ from $x_{LLH}$ and $x_{LLL}$

- $x_L$ from $x_{LH}$ and $x_{LL}$

- $x$ from $x_H$ and $x_L$
Signal compression using multiresolution

Signal can be compressed by discarding small elements of $x_{\text{Haar}}$:

$$\tilde{x}_{\text{Haar}}(m) = \begin{cases} x_{\text{Haar}}(m), & \text{if } |x_{\text{Haar}}(m)| > d \\ 0, & \text{if } |x_{\text{Haar}}(m)| \leq d \end{cases}$$

where $d > 0$ is a specified threshold.

- Resolution in both time and frequency domain
- Fourier transform gives only frequency domain resolution
EXAMPLE

\[ \{x(n)\} = \{37, 35, 28, 28, 58, 18, 21, 15\} \]

\[ \{x_{\text{Haar}}(m)\} = \{30, -2, -4, -10, -1, 0, -20, -3\} \]

(a): \( d = 2 \), (b): \( d = 3 \), (c): \( d = 4 \)
Wavelets

*Haar wavelets*

Recall that the Fourier transform expresses a signal \( \{ x(n) \} \) in terms of frequency components \( e^{j2\pi kn/N} \).

In a similar way, the Haar transform \( x_{\text{Haar}} \) can be expressed as an expansion of the signal \( x \) in terms of a function set.

For the cases with three stages (resolution levels), we can show that

\[
\begin{align*}
x(n) &= \sum_{m=0}^{N/2^3-1} x_{LLL}(m) \phi_H(2^{-3}n - m) - \sum_{m=0}^{N/2^3-1} x_{LLH}(m) \psi_H(2^{-3}n - m) \\
&\quad - \sum_{m=0}^{N/2^2-1} x_{LH}(m) \psi_H(2^{-2}n - m) - \sum_{m=0}^{N/2-1} x_H(m) \psi_H(2^{-1}n - m)
\end{align*}
\]
Here:

\[ \psi_H(t) = \begin{cases} 
1, & 0 \leq t < 1/2 \\
-1, & 1/2 \leq t < 1 \\
0, & \text{otherwise}
\end{cases} \]

and

\[ \phi_H(t) = \begin{cases} 
1, & 0 \leq t < 1 \\
0, & \text{otherwise}
\end{cases} \]

- \( \psi_H(2^{-i}n) \) is defined for \( 0 \leq n < 2^i \) (dilated, or stretched, version of \( \psi_H(n) \))

- \( \psi_H(2^{-i}n - m) \) is defined for \( m2^i \leq n < (m + 1)2^i \) (dilated and translated version of \( \psi_H(n) \))
Examples

(a): $\psi_H(t)$; (b): $\psi_H(2^{-2}t)$; (c): $\psi_H(2^{-2}t - 1)$
Expansion of $x$ in terms of $\phi_H(2^{-i}n - m)$ and $\psi_H(2^{-i}n - m)$ is a wavelet expansion.

$\phi_H(t)$ and $\psi_H(t)$: Haar wavelets
Generalization

Discrete wavelet expansion of \{x(n)\} of length \(N\) and \(J\) resolution levels has the form

\[
x(n) = \sum_{m=0}^{N/2^J-1} X_{DWT}(0, m)\phi(2^{-J}n-m) + \sum_{i=1}^{J} \sum_{m=0}^{N/2^i-1} X_{DWT}(i, m)\psi(2^{-i}n-m)
\]

- \(\psi(2^{-i}n)\) is defined for \(0 \leq n < 2^i\) (dilated, or stretched, version of \(\psi(n)\))

- \(\psi(2^{-i}n - m)\) is defined for \(m2^i \leq n < (m + 1)2^i\) (dilated and translated version of \(\psi(n)\))

\(\phi(t)\): father wavelet

\(\psi(t)\): mother wavelet

\(\psi(2^i n - m)\): daughter wavelets
Daubechies wavelets

Main restriction of Haar wavelets: Haar filter does not provide good separation of frequency bands.

For more powerful wavelet transforms, we should require that:

- $H_1(z), H_2(z)$ are FIR filters, which give good frequency separation

- $H_1(z), H_2(z)$ give a filter bank which corresponds to a wavelet expansion for some wavelet functions $\phi(t)$ and $\psi(t)$.

- Filter bank should have the perfect reconstruction property
In 1988 Ingrid Daubechies found a class of filters satisfying the conditions:

Instead of using quadrature mirror filters, $H_1, H_2$ should be related according to

$$H_2(z) = -z^M H_1(-z^{-1})$$

where $M =$ order of $H_1(z)$

Elimination of alias components gives condition for synthesis filters $G_1, G_2$:

$$G_1(z) = z^{-M} H_2(-z)$$
$$G_2(z) = -z^{-M} H_1(-z)$$
Daubechies filters are defined only for odd values of $M$.

Perfect reconstruction requirement gives condition:

$$\frac{1}{2} \left( G_1(z) H_1(z) + G_2(z) H_2(z) \right) = 1$$

Solution of order $M = 2m - 1, m = 1, 2, \ldots$ is given by

$$H_1(z) = \sqrt{2} \left( \frac{1 + z}{2} \right)^m S(z)$$

where $S(z)$ should satisfy

$$|S(e^{j\omega})|^2 = \sum_{k=0}^{m-1} \binom{m + k - 1}{k} \sin^2 k \left( \frac{\omega}{2} \right)$$
Case $m = 1$:

Daubechies wavelet is equivalent to the Haar wavelet (scaled by factor $\sqrt{2}$)

Case $m = 2$:

$$S(z) = \frac{1 - \sqrt{3}}{2} + \frac{\sqrt{3} + 1}{2}z$$

and

$$H_1(z) = \sqrt{2} \left\{ \frac{1 - \sqrt{3}}{8} + \frac{3 - \sqrt{3}}{8}z + \frac{3 + \sqrt{3}}{8}z^2 + \frac{1 + \sqrt{3}}{8}z^3 \right\}$$

$$= -0.129409522551 + 0.224143868042z$$

$$+ 0.836516303738z^2 + 0.482962913145z^3$$
Efficient algorithms exist for calculating higher-order Daubechies wavelets, and coefficients have been tabulated.

Daubechies wavelets form are an example of orthogonal wavelets, because the associated filters $H_1, H_2, G_1, G_2$ have a certain orthogonality property. Other orthogonal filter families are the *symlets* and the *Coiflets*.

Another important type of wavelets are the biorthogonal wavelets, for which the associated filters satisfy a biorthogonality condition.

An important family of biorthogonal wavelets are the *CDF (Cohen-Daubechies-Feauveau)* wavelets, which are used in the JPEG 2000 standard.
Wavelet transform of 2-dimensional signal \( \{x(n, m)\} \)

- first perform 1-dimensional wavelet transform of each row to give \( X_{\text{DWT, row}}(n, k) \) (=DWT of \( n \)th row of \( x(n, m) \))

- then perform 1-dimensional wavelet transform of each column to give the result \( X_{\text{DWT}}(l, k) \) (=DWT of \( k \)th column of \( X_{\text{DWT, row}}(n, k) \))

*Application*: image compression and denoising
Signal processing with wavelets

Time-frequency resolution makes wavelets powerful for data denoising and compression. A standard approach is to use threshold-based methods:

*Hard thresholds:*

Given a wavelet transform $X_{DWT}(i, m)$, define:

$$
\tilde{X}_{DWT}(i, m) = \begin{cases} 
X_{DWT}(i, m), & \text{if } |X_{DWT}(i, m)| \geq d(i) \\
0, & \text{if } |X_{DWT}(i, m)| < d(i)
\end{cases}
$$

Threshold $d(i) \geq 0$ may be different for different resolution levels $i$. Optimal value of $d(i)$ which depend of signal variance and length have been presented in the literature.
Soft thresholds:

Given a wavelet transform $X_{\text{DWT}}(i, m)$, define:

$$
\tilde{X}_{\text{DWT}}(i, m) = \text{sgn}(X_{\text{DWT}}(i, m))|X_{\text{DWT}}(i, m) - d(i)|
$$

if $|X_{\text{DWT}}(i, m)| \geq d(i)$,

and

$$
\tilde{X}_{\text{DWT}}(i, m) = 0, \text{ if } |X_{\text{DWT}}(i, m)| < d(i)
$$

Threshold $d(i) \geq 0$ may be different for different resolution levels $i$. Optimal value of $d(i)$ which depend of signal variance and length have been presented in the literature.
Hard and soft thresholds can be applied to both data compression and denoising.

JPEG 2000 image compression standard uses biorthogonal CDF wavelets (cf. above) and quantization of the wavelet components to achieve data compression.