Ch. 13 Subband Coding

Introduction

Idea: split signal \( x[n] \) into \( M \) signals \( X_1[n] \ldots X_M[n] \) such that each signal can be more easily/effectively coded.

Goal: signals \( X_1[n] \ldots X_M[n] \) should be s.t.

- each \( X_i[n] \) is uncorrelated \( \Rightarrow \) can use SQ

- some \( X_i[n] \) have smaller dynamic range \( \Rightarrow \) can use fewer bits for a given desired distortion

- should be a clear way to exploit psychological effects

Consider: DTFT of \( x[n] \)

\[ X(e^{j\omega}) \]

\[ H_b(e^{j\omega}) \]

Ideal BPF
Motivational Form (Not Practical)

Problems w/ This
1. Can't build Ideal BPF's
   ⇒ Reconstruction can't be simply adding $X_i[n_i]$
2. Increases # of samples/sec by $Mx$ after subband filtering

Each $X_i[n_i]$ is called a subband signal.
We'll fix these problems by first fixing #2 then #1

Take a look at $x_i[n]$

\[ x_i(n) \]

\[ \frac{\pi}{M} \quad \pi \]

This signal is oversampled by factor $M$

To sample it slower just throw away samples

\[ x_i[0] x_i[1] x_i[2] x_i[3] x_i[4] \]

\[ M = 3 \]

\[ \langle \text{Decimation} \rangle \]

\[ \langle \text{Expansion} \rangle \]

\[ M = 3 \]

\[ \hat{x}_i[0] \quad 0 \quad 0 \quad \hat{x}_i[3] \quad 0 \quad 0 \quad \ldots \]
Need a filter to smooth out the jumps due to inserted zeros:

\[ T \xrightarrow{} X_i[n] \xrightarrow{} \text{Interpolation} \]

Result: Can do same thing for each \( X_i[n] \)

**Figure 13.7** Block diagram of the subband coding system.

**Figure 13.8** Nonoverlapping and overlapping filter banks.
Subband Coding Method

Filter Design Goal: If we remove then encode/decode, then output = input

Analysis Filters must also provide frequency decomposition into subbands - to give "easy" to code signals

Synthesis Filters are chosen to give the desired "perfect Reconstruction"

Encoding/Decoding Goals:

1. Choose methods matched to channels

2. Allocate bit budget across channels

Decimation reduces each channel's sample rate to keep total filter bank's output sample rate equal to input sample rate

Interpolation returns each channel to original rate before reconstruction.
To understand how filter banks work we need to understand:

- effect of decimation
- effect of expansion
- How to choose Analysis & synthesis filters to achieve perfect reconstruction (PR)

13.5.1 Decimation (Down Sampling)

Consider $X[n] \rightarrow H(\omega) \rightarrow Y[n]$ (sampled @ $F_s$)

First let $H(\omega)$ be an ideal LPF w/ cutoff freq. $\omega_c = \pi/2 \Rightarrow "Half\ Band\\ Filter"

\[ X(\omega) \]

\[ H(\omega) \]

\[ Y(\omega) \]
Now imagine what continuous-time signal would give $y(t)$ if it were sampled at $F_s$.

Since the largest frequency component in $y(t)$ is $\frac{F_s}{2} = \frac{F_s}{4}$, the Nyquist rate for this signal is $F_s(Nyq) = 2f_{\text{max}} = \frac{F_s}{2}$

$\Rightarrow y[n]$ is oversampled by 2 times.

So if we sampled at $F_s(Nyq)$ we'd have
In general, if the filter passes only in range $-\pi/2 < \omega_0 < \pi/2$, we can downsample by $M$.

Now that was all "intuition" and for an **ideal** LPF.

Math Analysis of Decimation by $M=2$.

**Time Domain Result**: $W[n] = y[2n]$

```
    """"""""""""
```
Freq. Domain Result

Use $Z$-Transform, then convert to DFT.

Need a trick: $y'[n] = \frac{1}{2} [1 + (-1)^n] y[n]$

$y[n] \rightarrow y_0, y_1, y_2, y_3, y_4, y_5, \ldots$

$y'[n] \rightarrow y'_0, 0, y'_2, 0, y'_4, 0, \ldots$

Then still have: $W[n] = y'[2n]$

$\Rightarrow W(z) = \sum_{n=-\infty}^{\infty} w[n] z^{-n} = \sum_{n=-\infty}^{\infty} y'[2n] z^{-n}$

$\left(\text{let } m = 2n\right) = \sum_{m=-\infty}^{\infty} y'[m] z^{-\frac{m}{2}}$

Here is the point of the trick!

The sum now includes the odd samples of $y'[m]...$ but these are all zero! 

$= Y'(z^{1/2})$
So... what is \( Y'(z) \)?

\[
Y'(z) = \sum_{n=-\infty}^{\infty} y'[n] z^{-n}
\]

Use "Trick"

\[
\frac{1}{2} \left[ \left( \sum_{n} y[n] z^{-n} \right) + \left( \sum_{n} y[n] (-z)^{-n} \right) \right] = Y(z) + Y(-z)
\]

\[
\Rightarrow Y'(\frac{z}{2}) = \frac{1}{2} \left[ Y(\frac{z}{2}) + Y(-\frac{z}{2}) \right]
\]

\[
\Rightarrow W(z) = \frac{1}{2} Y(\frac{z}{2}) + \frac{1}{2} Y(-\frac{z}{2})
\]

\[\uparrow \]

Z-T of decimated signal

To get DTFT version: \( Z = e^{j\omega} \)

\[
\Rightarrow z^{\frac{1}{2}} = e^{j\frac{\omega}{2}} \quad \& \quad -z^{\frac{1}{2}} = e^{-j\frac{\omega}{2}}
\]

\[
= e^{j\left(\frac{\omega - 2\pi}{2}\right)}
\]
So...

\[ W(\Omega) = \frac{1}{2} \left[ Y\left(\frac{\Omega}{2}\right) + Y\left(\frac{\Omega - 2\pi}{2}\right) \right] \]

\[ \uparrow \]

(stretches by factor of 2)

Desired point that is within \([\frac{\pi}{2}, \pi]\)

\[ \uparrow \]

(stretches, then shifts by \(\pi\))

Additional Replica due to sampling

< Plots on Page 4 show this >

< Plots on Page 6 show this >

for an Ideal LPF

for a Non Ideal Filter
Decimation w/ Ideal Filter
Down sampling of a weighted filter

$H(e)$

$Y(e)$

$Y(e/2)$

$Y(e - 3\pi/2)$

Aliasing

Even $W(e)$

(etc.)
So in practice the filter leaves some non zero stuff outside $\Omega \in [-\pi/2, \pi/2]$ that will alias back into $[-\pi, \pi]$ after decimation.

Generally, can design filters to give negligible contributions outside $[-\pi/2, \pi/2]$ but ... long FIR filters may be needed to achieve this.

**Note:** Similar analysis can be done for HPF case.

\[
X[n] \xrightarrow{\text{Half Band}} Y[n] \xrightarrow{\text{VM}} W[n]
\]

For $M=2$

\[\text{(See HW Problem)}\]
13.5.2 Interpolation (upsampling)

\[ w[n] \xrightarrow{\uparrow M} V[n] \xrightarrow{H(\Omega)} U[c] \]

Not so easy to see intuitively!

For \( M = 2 \) case: \( V[n] = \begin{cases} \text{if } n \text{ even} \quad w[n/2] \\ 0 \quad \text{if } n \text{ odd} \end{cases} \)

"Zero stuffing"

\[ W[0] \quad W[1] \quad W[2] \quad W[3] \quad \ldots \]

\[ V[0] \quad V[1] \quad 0 \quad V[4] \quad 0 \quad V[6] \quad \ldots \]

What does this look like in freq. domain?

Use ZT then convert to DTFT

\[ V(z) = \sum_{n=-\infty}^{\infty} V[n] z^{-n} = \left( \sum_{m=-\infty}^{\infty} V[2m] z^{-2m} \right) + \left( \sum_{m=-\infty}^{\infty} V[2m+1] z^{-2m} \right) \]

\[ V(z) = \sum_{m=-\infty}^{\infty} W[m] z^{-2m} \]

\[ = (z^2)^{-m} \]

\[ \Rightarrow V(z) = W(z^2) \]
Thus, \( V(\tilde{z}) = W(\tilde{z}^2) \)  

Effect of "zero stuffing" on \( z \)-Transform

Now to see DFT: \( \tilde{z} = e^{j \omega} \Rightarrow \tilde{z}^2 = e^{j 2 \omega} \)

\[ \Rightarrow V(\omega) = W(2 \omega) \]

\[ \text{\uparrow squishes } \, W(\omega) \text{ to get } V(\omega) \]

Anti-imaging Filter
Interpolating Filter

Non-ideal filters leave some imaging error

Time Domain View

"Interpolated" points
Non-ideal Anti-Image filters leave a small amount of the image spectrum → design to minimize image residual

**General (M ≠ 2) Results**

\[ W(\omega) = \frac{1}{M} \sum_{m=0}^{M-1} Y\left(\frac{\omega - 2\pi m}{M}\right) \quad \& \quad Y(\omega) = H(\omega)X(\omega) \]

with \( H(\omega) \) having a passband of width \( \frac{\pi}{M} \)

\[ V(\omega) = W(M\omega) \quad \& \quad U(\omega) = K(\omega)W(M\omega) \]

with \( K(\omega) \) having a passband width \( \frac{\pi}{M} \)