Ch. 14 Subband Coding

Perfect Reconstruction Filterbanks
Recall the general structure of subband coding:

**To Design the Filters**: Imagine removing the encoders/decoders… Then design so that the output is a “perfect reconstruction” of the input.

\[
\hat{x}[n] = cx[n - n_o]
\]
\[
\hat{X}(z) = cX(z)z^{-n_o}
\]
We’ll limit here to $M = 2$ Channels…

\[ H_1(z) \quad y_{1,n} \quad \downarrow 2 \quad w_{1,n} \quad \uparrow 2 \quad v_{1,n} \quad K_1(z) \quad u_{1,n} \]

\[ H_2(z) \quad y_{2,n} \quad \downarrow 2 \quad w_{2,n} \quad \uparrow 2 \quad v_{2,n} \quad K_2(z) \quad u_{2,n} \]

\[ x_n \]

\[ \hat{x}_n \]

**Figure 14.17** Two-channel subband decimation and interpolation.

The “analysis” side filters are half-band LPF & HPF.

**Q:** To ensure PR how do we choose:

Analysis Filters: $H_1(z), H_2(z)$

Synthesis Filters: $K_1(z), K_2(z)$

**Note:** If $H_1(z), H_2(z), K_1(z)$ & $K_2(z)$ are all *ideal* half-band filters then PR is easily achieved.

But we can’t build ideal filters… So is it even possible to really get PR????
Impact of Non-Ideal Filters

Stop-Band Issues

Analysis Filters: $H_1(z) & H_2(z)$ will leave some content outside their half-band passbands that gets aliased into the passband after decimation.

Synthesis Filters: $K_1(z) & K_2(z)$ will not completely eliminate the images created by upsampling that lie outside their half-band passbands.

Pass-Band Issues

Magnitude: For non-ideal filters the passbands are not perfectly flat and will change the shape of the signal’s DTFT magnitude in the passband.

Phase: Because PR allows a delay and a delay corresponds to a linear phase response (as a function of frequency) it seems natural to focus on linear phase filters – which puts our focus on FIR Filters.

Our Goal: Choose filters such that the aliasing & imaging errors cancel out!!! (Fixes the stop-band issues)

Then… make what is left combine to give the desired composite passband to achieve the PR condition.

Let’s see how to do this mathematically
Math Analysis of PR Requirements

Start at input & work toward the output using z-transform methods:

Top Channel (Bottom Channel Similar):

\[ Y_1(z) = H_i(z)X(z) \quad \text{Filter} \]

\[ W_1(z) = \frac{1}{2}Y_1(z^{1/2}) + \frac{1}{2}Y_1(-z^{1/2}) \quad \text{Down Sampling} \]

\[ = \frac{1}{2} \left[ H_1(z^{1/2})X(z^{1/2}) + H_1(-z^{1/2})X(-z^{1/2}) \right] \]
\[ V_1(z) = W_1(z^2) \]

\[ = \frac{1}{2} \left[ H_1(z)X(z) + H_1(-z)X(-z) \right] \]

\[ U_1(z) = K_1(z)V_1(z) \]

\[ = \frac{1}{2} K_1(z) \left[ H_1(z)X(z) + H_1(-z)X(-z) \right] \]

Now the output of the whole structure is:

\[ \hat{X}(z) = U_1(z) + U_2(z) \]
Substitute results for \( U_i(z) \) & Group \( X(z) \) terms & group \( X(-z) \) terms…

\[
\hat{X}(z) = \frac{1}{2} \left[ H_1(z)K_1(z) + H_2(z)K_2(z) \right] X(z) + \frac{1}{2} \left[ H_1(-z)K_1(z) + H_2(-z)K_2(z) \right] X(-z)
\]

\[
\hat{X}(z) = T(z)X(z) + S(z)X(-z)
\]

Aliasing Term…
Don’t Want It!

\[
X(-z) = X(e^{j\pi z}) \quad \rightarrow \quad X_{DTFT}(\Omega + \pi)
\]

\[
e^{j\pi z} \bigg|_{z=e^{j\Omega}} = e^{j(\Omega + \pi)}
\]
We want to eliminate this “Aliasing Term”:  
\[ S(z) = 0 \]
\[ H_1(-z)K_1(z) + H_2(-z)K_2(z) = 0 \]

Can cancel aliasing term by choosing:

\[ K_1(z) = H_2(-z) \quad \& \quad K_2(z) = -H_1(-z) \]
\[ K_1(\Omega) = H_2(\Omega + \pi) \quad \& \quad K_2(\Omega) = -H_1(\Omega + \pi) \]

“ACC”

Aliasing Cancellation Condition

 Doesn’t constrain the filters… but constrains the relationship between the K’s and H’s

**Note:** Since…  
\[ H_1(z) \text{ is lowpass} \quad \& \quad H_2(z) \text{ is highpass} \ldots \]

we have:
\[ K_1(z) \text{ is lowpass} \quad \& \quad K_2(z) \text{ is highpass.} \]

To see this:  
\[ K_1(z) = H_2(-z) \quad \rightarrow \quad K_1(z) = H_2(e^{j\pi}z) \quad \rightarrow \quad K_1(\Omega) = H_2(\Omega + \pi) \]

Similarly:  
\[ K_2(z) = -H_1(-z) \quad \rightarrow \quad K_2(z) = -H_1(e^{j\pi}z) \quad \rightarrow \quad K_2(\Omega) = -H_2(\Omega + \pi) \]
Once the \( K \)s are chosen this way we get
\[
\hat{X}(z) = T(z)X(z)
\]
\[
= \frac{1}{2} \left[ H_1(z)H_2(-z) - H_1(-z)H_2(z) \right] X(z)
\]
Want this = \( Cz^{-n_o} \) for PR

So the condition the \( H \)s must meet for PR is:

\[
\left[ H_1(z)H_2(-z) - H_1(-z)H_2(z) \right] = Cz^{-n_o}
\]  \( \text{(★)} \)

Note: the \( K \)s are chosen to cancel aliasing
the \( H \)s are chosen to give PR

**Comment:** For compression we not only want to cancel aliasing but we often need to minimize it in *each* channel… which requires all filters to have sharp transition bands and low stop bands

*Why do we need this?* Because in compression we often throw away some subbands (those having small energy)… and that upsets the balance used to cancel aliasing!
Focus of Design Process

So… our design process now focuses on designing the analysis filters $H_1(z) \& H_2(z)$ so that they meet ($\star$) for PR

**Note**: The aliasing cancelation puts no constraint on the design of the filters… it only says: “if the analysis filters are this… then the synthesis filters must be that”.

There are several design methods to get analysis filters $H_1(z) \& H_2(z)$ that give PR… various researchers have proposed these over the years.

We’ll look at two:

- Quadrature Mirror Filters (QMF)
- Power Symmetric Filters
  – also called Conjugate Mirror Filters (CMF)
Quadrature Mirror Filters (QMF)
These were proposed in 1977 by Esteban & Galand
Their definition of QMF leads to:

- Useful filters for filterbanks
- But… not able to give PR (except in a trivial case)

**QMF Definition**: A pair of analysis filters are QMFs if

\[ H_2(z) = H_1(-z) \]

\[ H_2(\theta) = H_1(\theta \pm \pi) \]

**QMF Condition**

**Note**: Once \( H_1(z) \) is designed then the QMF condition nails down \( H_2(z) \)… …and remember that \( K_1(z) \) & \( K_2(z) \) are also nailed down by the ACC

So… enforcing QMF & ACC reduces the design problem to only designing \( H_1(z) \)
QMF “Facts”

1. If $H_1(z)$ is linear phase, so is $H_2(z)$

2. QMFs can only achieve PR if the $h_1[n]$ and $h_2[n]$ each have only 2 non-zero “taps”
   - E.g., $h_1[n] = [1 \ 1]$ or $h_1[n] = [1 \ 0 \ 1]$ or $h_1[n] = [1 \ 0 \ 0 \ 1]$ etc.
   - Note: 2-tap Filters Stink! (See poor $\frac{1}{2}$-band characteristics shown in Fig. 14.18)

3. If $H_1(z)$ has linear phase then $T(z)$… the analysis/synthesis total transfer function… also has linear phase.
   - Note that this is necessary for PR, where we need
     
     $T(z) = Cz^{-n_0} \Rightarrow T(\Omega) = Ce^{-j\Omega n_0}$

So… for QMF (w/ # taps > 2) we can’t get the amplitude part of PR:

$T(\Omega) = \underbrace{C(\Omega)}_{\text{real-valued \& } \geq 0} e^{-j\Omega n_0}$
QMF Design Process (One Way to Do It)

1. Once we get a design for the Analysis Filters… the $H_i(\Omega)$…

Choose the Synthesis Filters … the $K_i(\Omega)$… to cancel aliasing

\[ K_1(z) = H_2(-z) \quad \& \quad K_2(z) = -H_1(-z) \]

“ACC”

\[ T(z) = \frac{1}{2} \left[ H_1(z)H_2(-z) - H_1(-z)H_2(z) \right] \]

2. Eliminate Phase Distortion by constraining the $H_i(\Omega)$ to be linear phase FIR filters… which ensures that you get:

\[ T(\Omega) = C(\Omega)e^{-j\Omega n_o} \]

3. Enforce the QMF relationship…

\[ H_2(z) = H_1(-z) \]

“QMF”

\[ T(z) = \frac{1}{2} \left( [H_1(z)]^2 - [H_1(-z)]^2 \right) \]

$T(z)$ now depends only on $H_1(z)$!!
4. Design $H_1(\Omega)$ to be a good LPF and to minimize the amplitude distortion of the end-to-end frequency response $T(\Omega)$. This can be done numerically by minimizing

$$J_{\alpha}(h_1) = \alpha \int_{\Omega_s} ^{\pi} |H_1(\Omega)|^2 d\Omega + (1-\alpha) \int_{0} ^{\pi} \left[1 - |T(\Omega)|^2\right] d\Omega$$

See book “Numerical Recipes in C” for details on numerical minimization methods.

$\alpha$ controls relative priority of the two goals… $0 \leq \alpha \leq 1$

4. Once you have a $H_1(\Omega)$ that minimize $J_{\alpha}$ for your chosen $\alpha$, then use it to generate all the other filters…

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

$\{\text{"QMF"}\}$

$$K_1(z) = H_2(-z) = H_1(z)$$

$\{\text{"ACC"}\}$

$$K_2(z) = -H_1(-z)$$
Power Symmetric FIR Filters (or Conjugate Mirror Filters)

This **Does Allow** PR!!!

We’ll get rid of aliasing the same way as before:

\[
K_1(z) = H_2(-z) \quad \& \quad K_2(z) = -H_1(-z)
\]

“ACC”

Error in Book in (14.75)

This gives \( S(z) = 0 \) (as before) and gives (as before)

\[
T(z) = \frac{1}{2} \left[ H_1(z)H_2(-z) - H_1(-z)H_2(z) \right]
\]

Error in Book

Now… here is the new condition to use instead of QMF:

\[
H_2(z) = (-z)^{-N} H_1(-z^{-1})
\]

“CMF Z-D”

where \( N \) = “Order” of the FIR filter \( H_1(z) \)

**Recall**: FIR has \( h_1[n] = 0 \) for \( n \neq 0, 1, 2, ..., L-1 \)

Length = \( L \)  \quad Order = Length – 1 = \( L-1 \)
Can show that “CMF” is equivalent to

\[ h_2[n] = (-1)^n h_1(N - n) \]  

“CMF T-D”

For Order-N FIR with Odd \( N \)...

\[
\begin{align*}
\begin{array}{cccc}
  h_2[n] : & h_1[N] & -h_1[N - 1] & \cdots & h_1[1] & -h_1[0]
\end{array}
\end{align*}
\]

Using “CMF Z-D” in \( T(z) \) gives:

\[
T(z) = \frac{1}{2} z^{-N} \left[ H_1(z) H_1(z^{-1}) + H_1(-z) H_1(-z^{-1}) \right] \]

\[ \overset{\Delta}{=} R(z) \]

\[ = \frac{1}{2} z^{-N} \left[ R(z) + R(-z) \right] \]

Want = constant for PR
Recall: 1. If $H_1(z) \leftrightarrow h_1[n]$ then $H_1(z^{-1}) \leftrightarrow h_1[-n]$

2. $F(z)G(z) \leftrightarrow f[n]^*g[n]$

So… since $R(z) = H_1(z)H_1(z^{-1})$  

$R(z) \leftrightarrow \rho[n] = h_1[n]^*h_1[-n]$

“Time Auto Correlation” of $h_1[n]$

\[
\rho[n] = 0 \quad \text{for} \quad |n| > N \\
\rho[-n] = \rho[n] \quad \text{(even symmetry)}
\]

So… (recalling that $N$ is odd)…

\[
R(z) = \rho[N]z^N + \rho[N-1]z^{N-1} + \cdots + \rho[1]z^1 + \rho[0] + \rho[1]z^{-1} + \cdots + \rho[N]z^{-N} \\
R(-z) = -\rho[N]z^N + \rho[N-1]z^{N-1} - \cdots - \rho[1]z^1 + \rho[0] - \rho[1]z^{-1} + \cdots - \rho[N]z^{-N}
\]

Odd-Indexed Terms Cancel when $R(z)$ & $R(-z)$ are added
\[ R(z) + R(-z) = \text{Only even-order terms} \]

\[ = \rho[0] + \sum_{n=2,4,\ldots,N-1} \rho[n](z^n + z^{-n}) \]

Want = constant for PR

\[ \rho[n] = \begin{cases} 
C, & n = 0 \\
0, & n \text{ even, } n \neq 0 \\
don't \text{ care, } & n \text{ odd} 
\end{cases} \]

Requirement for PR: \( \rho[2n] = C \delta[n] \)

\[ \rho[2n] = \sum_{k=0}^{N} h_1[k]h_1[k+2n] = C\delta[n] \]

Time-Domain Requirement for PR
To convert this into Freq. Domain: \[ \mathcal{F} \{ \rho[2n] \} = \mathcal{F} \{ C\delta[n] \} \] (A)

DTFT of decimated sequence

From F-D result for decimation: \[ \mathcal{F} \{ \rho[2n] \} = \frac{1}{2} \left[ R \left( \frac{\Omega}{2} \right) + R \left( \frac{\Omega - 2\pi}{2} \right) \right] \]

\[ \Omega \triangleq \frac{\Omega}{2} \]

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

Now since \( R(z) = H_1(z) H_1(z^{-1}) \) The DTFT form is \( R(\Omega) = H_1(\Omega)H_1(-\Omega) \)

\[ = H_1(\Omega)H_1^*(\Omega) \] (C)

From (A) – (C) we get: \[ \left| H_1(\Omega) \right|^2 + \left| H_1(\Omega - \pi) \right|^2 = C \] Freq-Domain Requirement for PR
To see this condition:

\[
|H_1(\Omega - \pi)|^2 + |H_1(\Omega - \pi)|^2
\]

Filters satisfying this are called…

“Power Symmetric Filters” or “Conjugate Mirror Filters”

For Design Details… See Books on Filter Banks
Perfect Reconstruction for M Channels

There are two ways to get PR for $M > 2$ Channels:

1. Extend all previous results to general $M > 2$ case
   - Same basic ideas but much more complicated
   - See books on Filter Banks

2. Cascade 2-Channel Stages…
Analysis Side of 3-Stage, 8-Channel PR Filter Bank
Design a 2-Channel PR Filterbank… Get M Channel PR:

Gives PR… so can “remove”

\[ \hat{x}[n] \]
After “removal” of Center:

Gives PR…

So… a cascade followed by the reverse cascade “collapses” to give M-Channel PR
What do the various channels in a cascaded analysis filter bank look like?

Can be shown that each channel has transfer function that looks like this (for the 3-stage case):

\[ H_{\text{channel}}(z) = H_i(z)H_j(z^2)H_k(z^4) \]

\[ \text{1 or 2 (1st Stage)} \quad \text{1 or 2 (2nd Stage)} \quad \text{1 or 2 (3rd Stage)} \]

The cascade method is useful but has a limitation:

If \( H_i(z) \) has order \( N \), then \( H_i(z^2) \) has order \( 2N \), and \( H_i(z^4) \) has order \( 4N \)...

… and then the cascade of them has order \( N + 2N + 4N \)

**BUT**…. You only have \( N \) degrees of freedom in “choosing all those” \( N + 2N + 4N \) coefficients!!!
Bit Allocation

Same ideas as for Bit Allocation for TC….

Each subband has its own quantizer and you want to allocate bits to the quantizers