Chapter 7: Resampling Filters

7.1 Interpolators 154
   7.1.1 Simple 1-to-M Interpolator 154

7.2 Interpolator Architecture 161
   7.2.1 Polyphase Partition 161

7.3 Band-pass Interpolator 164

7.4 Rational Ratio Resampling 169

7.5 Arbitrary Resampling Ratio 171
   7.5.1 Nearest Neighbor Interpolation 172
   7.5.2 Two Neighbor Interpolation 181

7.6 Farrow Filter 185
   7.6.1 Classical Interpolator 185
   7.6.2 Polynomial Approximation 189
   7.6.3 Farrow Filter 192
Interpolation Concept

- 1:4 Shaping Prior to interpolation processing.

\[ x(n) \rightarrow \exp(j \theta n) \rightarrow \frac{\sin(x)}{x} \rightarrow \text{DAC} \]

**Figure 7.1** Example of Interpolator Following Initial Time Series Generator

**Figure 7.2** Spectrum at Output of Shaping Filter with Embedded 1-to-4 Up Sampler
Interpolation-Filter

- If you are going to interpolate,
  - Interpolate to a sufficient rate to minimize follow-on filter concerns, particularly when a digital to analog converter (DAC) is involved.

- The digital filter can (should) be based on the original signal passband and where the replicas will occur.
  - Example: \( f_{\text{stop}} = \text{original } f_s - f_{\text{pass}} \) ‘may work’
  - Alternately: \( f_{\text{stop}} = \text{original } f_s - (f_{\text{pass}} + f_{\text{stop}})/2 \) ‘may work’
  - Most Conservative: \( f_{\text{stop}} = f_s/2 \)
Interpolator - Desired Filtering

• Post-interpolation filter to eliminate replicated components
• Filter Specification

**Figure 7.3** Periodic Input Spectrum and Filter Response of 1-to-32 Interpolator Filter

**Table 7-1 Parameters Required for Interpolating Filter**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass-band Ripple</td>
<td>0.1 dB</td>
</tr>
<tr>
<td>Stop-band Attenuation</td>
<td>60 dB</td>
</tr>
<tr>
<td>Pass-band Frequencies</td>
<td>0-to-0.75 (0-to-75 kHz)</td>
</tr>
<tr>
<td>Stop-band Frequencies</td>
<td>3.25-to-64 (325 kHz-to-6.4 MHz)</td>
</tr>
<tr>
<td>Sample Frequency</td>
<td>128 (12.8 MHz)</td>
</tr>
</tbody>
</table>
Interpolated Replicas and Filter

Filter Design:
NN=firpmord([0.75 3.25],[1 0],[0.01 0.001],128)

The response to this call is:
NN = 130 use 128 tap as sufficient
Composite and Resulting Responses

- Above: replicated bands and the filter to be applied
- Below: Post-filtering signal spectrum
Zoom to Narrowband Result

- Above: replicated bands and the filter to be applied
- Below: Post-filtering signal spectrum
An Alternate Filter: Don’t Care Bands

- We only care about attenuation were the replicas exist!
- Therefore, why “over-constrain” the FIR filter and firpm?
Composite and Resulting Responses

- Above: replicated bands and the filter to be applied
- Below: Post-filtering signal spectrum
Zoom to Narrowband Result

- Above: replicated bands and the filter to be applied
- Below: Post-filtering signal spectrum
Comparison

- Above: Original Filter
- Below: Don’t Care Band Filter
Comparison - Narrowband

- Above: Original Filter
- Below: Don’t Care Band Filter
Filter Coefficients

Above: Original Filter

Below: Don’t Care Band Filter
Making the filter work

1. Adjust the passband weight for firpm.
   • At equal weights the passband ripple is “too small”
   • Reduce the passband weight to $2/M$ for better results

2. Narrow the zero region around the spectral replicas
   • Use Nyquist passband with 0 roll-off. The combined effect of the roll-off and filter provide sufficient attenuation
   • This also widens the 1st transition from the passband to stopband.
MATLAB Simulation

- See chap7_1.m
  - Same as given in text book and slides
- See chap7_1a.m
  - Tweaked (increased ntaps) to meet stop-band attenuations
- See chap7_1b.m
  - Don’t Care regular and myfrf
- See chap7_1c.m
  - ASK modulated test signal (4 samples per symbol, nyquist filter)
Interpolator Architecture

• Basic Interpolator
• Interpolator with Complex Frequency Mixing
Inverse Architectures

- We can invert the filter decimation process.
  - In general, whatever architecture flows in one direction, an inverse that flows in the exact opposite direction is possible!
Channelizer

- Select one of $M$ spectral bins in an FDM spectrum.

**Figure 6.2** Standard Single Channel Down Converter
Figure 6.3 Spectra Up/Down Conversion

Spectra Observed At Various Points in Processing Chain of Standard Down Converter

Upconverter operates from bottom to top!
Z-Transform Interpolation Filter Math

\[ H(z) = \sum_{n=0}^{N-1} h(n) \cdot z^{-n} \]

\[ H(z) = \sum_{r=0}^{M-1} \sum_{m=0}^{N/M-1} h(r + m \cdot M) \cdot z^{-(r + m \cdot M)} \]

\[ H(z) = \sum_{m=0}^{N/M-1} z^{-m \cdot M} \cdot \sum_{r=0}^{M-1} h(r + m \cdot M) \cdot z^{-r} \]

\[ H(z) = \sum_{m=0}^{N/M-1} z^{-m \cdot M} \cdot H_r(z^M) \]

- A similar structure in \( z^M \) as seen before
- Therefore we can use a polyphase structure to perform filtering as follows …
Interpolation Polyphase Structure

\[ H(z) = \sum_{m=0}^{N/M-1} z^{-mM} \cdot H_r(z^M) \]

**Figure 7.10** Initial Structure of 1-to-M Polyphase Interpolator

- The noble identity can be applied
Post Polyphase Interpolation

Notice that the delay elements identify a commutator.

**Figure 7.11** Transition Structure of 1-to-M Polyphase Interpolator
Polyphase Interpolator

Figure 7.12 Standard Structure of 1-to-M Polyphase Interpolator

• The “inverse structure” from the filter-decimator
Bandpass Interpolators

- Incorporate output mixing ($x_m$ to the $m^{th}$ Nyquist Band)
- Use Bandpass filters instead of Lowpass filters

\[
y(n) = \exp\left(i \cdot 2\pi \cdot \frac{n \cdot m}{L}\right) \cdot \sum_{k=0}^{\frac{L-1}{2}} h(k) \cdot x_m\left(\frac{n-k}{L}\right)
\]

Let \( k = rL + \rho \) \quad \( n = sL + t \)

\[
y(s \cdot L + t) = \exp\left(i \cdot 2\pi \cdot \frac{(s \cdot L + t) \cdot m}{L}\right) \cdot \sum_{\rho=0}^{L-1} \sum_{r=0}^{\lambda-1} h(r \cdot L + \rho) \cdot x_m\left(\frac{s \cdot L + t - r \cdot L - \rho}{L}\right)
\]
Bandpass Interpolation (2)

\[ y(s \cdot L + t) = \exp\left(i \cdot 2\pi \cdot \frac{t \cdot m}{L}\right) \cdot \sum_{\rho=0}^{L-1} \sum_{r=0}^{\lambda-1} h(r \cdot L + \rho) \cdot x_m\left(s-r + \frac{t-\rho}{L}\right) \]

\[ y(s \cdot L + t) = \exp\left(i \cdot 2\pi \cdot \frac{t \cdot m}{L}\right) \cdot \sum_{r=0}^{\lambda-1} h(r \cdot L + t) \cdot x_m(s-r) \]

\[ y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} \left[ \exp\left(i \cdot 2\pi \cdot \frac{t \cdot m}{L}\right) \cdot h_t(r) \right] \cdot x_m(s-r) \]

- This form uses a complex filter on the replicated spectra

\[ y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} h_t(r) \cdot \left[ \exp\left(i \cdot 2\pi \cdot \frac{t \cdot m}{L}\right) \cdot x_m(s-r) \right] \]

- This form uses a complex pre-multiplication for each polyphase path
Complex Shift

- Filter Equivalence

$y(n) = h_0(m)x(m) + h_1(m)x(m - 1) + h_{M-1}(m)x(m - M) + W_{M}^{nk}$

$y(n) = h_0(m)x(m) + h_1(m)x(m - 1) + h_2(m)x(m - 2) + W_{M}^{nk}$
Complex Shift (2)

- A version with real filtering of complex coefficients
- The input is output in the Nyquist zone defined by \( k \).
- Question: is there a structure where we could fill all the bands simultaneously?

\[
y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} h_t(r) \cdot \exp \left( i \cdot 2\pi \cdot \frac{t \cdot m}{L} \right) \cdot x_m(s-r)
\]
X data provided for every band

- One Bandpass band

\[
y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} h_t(r) \cdot \left[ \exp \left( i \cdot 2\pi \cdot \frac{t \cdot m}{L} \right) \cdot x_m(s - r) \right]
\]

- Sum of all bandpass bands

\[
y(s \cdot L + t) = \sum_{m=0}^{L-1} \sum_{r=0}^{\lambda-1} h_t(r) \cdot \left[ \exp \left( i \cdot 2\pi \cdot \frac{t \cdot m}{L} \right) \cdot x_m(s - r) \right]
\]

\[
y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} h_t(r) \sum_{m=0}^{L-1} \left[ \exp \left( i \cdot 2\pi \cdot \frac{t \cdot m}{L} \right) \cdot x_m(s - r) \right]
\]

- Filtering the (r-s) “vector” outputs of an M-point IFFT
A Trans-multiplexer

- This structure generates a frequency-division-multiplexed output from the $M$ discrete narrowband inputs!
  - Extensively used in early telephony for analog FDM

$$y(s \cdot L + t) = \sum_{r=0}^{\lambda-1} h_r(r) \cdot \sum_{m=0}^{L-1} \left[ \exp \left( i \cdot 2\pi \cdot \frac{t \cdot m}{L} \right) \cdot x_m(s - r) \right]$$
MATLAB Simulation

- See Chap7_interpfilterv.m
  - FFT based interpolation
- See Chap7_interpfilterv2.m
  - IFFT based interpolation

FDM Generation and Processing

Forming Wideband and Reforming Narrowband

Filters narrower than the Nyquist regions are used for generating FDM waveforms. A guard band between adjacent frequencies is typically used.

Synthesis

Analysis

OFDM Symbol Basis
Quadrature Mirror Filter Processing

Observing Narrowband and Reforming Wideband

Significant filter restriction are required if the output is required to approximate the input!

Quadrature Mirror Filter Definition and Requirements

Analysis

Synthesis
QMF Equations

• Analysis

\[ y_k(m) = \sum_{\rho=0}^{M-1} \exp\left( j2\pi \cdot \frac{\rho \cdot k}{M} \right) \cdot \left[ \sum_{r=0}^{\lambda-1} h_{\rho}(r) \cdot x_{\rho}(m-r) \right] \]

• Synthesis (substitute \(-t\) and \(L=M\))

\[ \hat{x}(s \cdot M - t) = \sum_{r=0}^{\lambda-1} h_{t}(r) \cdot \sum_{m=0}^{M-1} \exp\left( -i \cdot 2\pi \cdot \frac{t \cdot m}{M} \right) \cdot y_{m}(s-r) \]

• Inverse mathematical processes
  – IFFT with “inverse” FFT
  – Broadband to narrowband and back to broadband
  – Filters are critical for overall performance.
QMF Applications

- Frequency domain filtering or equalization
- Time-Spectral Analysis with reconstruction

- Arbitrarily take signals apart and then reconstruct them
  - Partial-Band Synthesis to one or more arbitrary bandwidths (universal base station receiver)
  - Partial-Band Analysis with frequency domain summation and full-band synthesis (universal base station transmitter)
  - Applications: cellular telephone base stations, satellite relay stations, etc.
Quadrature Mirror Filters

- Architecturally, the structure is mirrored across the central axis in terms of the analysis and synthesis processing.
- Perfect Reconstruction of the input signal is desired, but …
  - Quasi-Perfect is usually accomplished
  - Look for PR-QMF or Quasi-PR QMF
- The QMF operation is typically described in terms of two filters.
2-Bank QMF

- Perfect Reconstruction Conditions

\[
\hat{X}(z) = \frac{1}{2} \left[ H_0(z) \cdot H_0(z) - H_1(z) \cdot H_1(z) \right] \cdot X(z) \\
+ \frac{1}{2} \left[ H_0(-z) \cdot H_0(z) - H_1(-z) \cdot H_1(z) \right] \cdot X(-z)
\]

\[
H_1(z) = H_0(-z)
\]

\[
\hat{X}(z) = \frac{1}{2} \left[ H_0(z) \cdot H_0(z) - H_0(-z) \cdot H_0(-z) \right] \cdot X(z) \\
+ \frac{1}{2} \left[ H_0(-z) \cdot H_0(z) - H_0(-z) \cdot H_0(-z) \right] \cdot X(-z)
\]

\[
\hat{X}(z) = \frac{1}{2} \left[ H_0(z) \cdot H_0(z) - H_0(-z) \cdot H_0(-z) \right] \cdot X(z)
\]
2-Bank QMF

- Perfect Reconstruction Conditions

\[ \hat{X}(z) = \frac{1}{2} \cdot [H_0(z) \cdot H_0(z) - H_0(-z) \cdot H_0(-z)] \cdot X(z) \]

\[ H_1(z) = H_0(-z) \]

\[ \|T(w)\| = \|H_0(w)\| + \|H_0(w + \pi)\| = 1 \]

\[ \left\| H_0\left(\frac{\pi}{2}\right) \right\| = \frac{1}{2} \]

- Additional Conditions (optimization)

\[ \|H_0(w)\| = 1, \quad \text{for} \quad 0 < w < w_{\text{Passband}} < \frac{\pi}{2} \quad \|H_0(w)\| = 0, \quad \text{for} \quad \pi - \varepsilon < w < \pi \]
Project Assignment Discussion

1. Find a paper describing the implementation of a filter that can be used in a Quadrature Mirror Filter
   • Look in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing
   • Dr. Bazuin must approve of all papers/filters selected.

2. Implement the filter using MATLAB
   • Provide a function that generates the filter coefficients based on a desired set of input criteria (i.e. filtercoef=Name(inputparameters))
   • Verify that you have generated the correct filter coefficients by comparing them to values or curves provided in the paper.

3. Use the QMF analysis-synthesis MATLAB script that will be developed or provided in class to characterize the results of your filter.
   • Post-analysis filter ripple, bandwidth, stopbands, etc.
   • Post-synthesis input to output error, error frequency response, etc.
Rational Rate Resampling

- What happens if we want an output rate that is $P/Q$ where $P$ and $Q$ are integers.
  - Filter-decimate and then interpolate-filter, or
  - Interpolate-filter and then filter decimate

Which one can use fewer filters?

Interpolate-decimate
Rational Rate Filter

- The filter is defined based on the larger of P or Q
  - If P is greater, the baseband Nyquist Zone must be isolated from the others prior to decimation
  - If Q is greater, the filter that eliminates aliasing when decimation is performed is narrower than the baseband Nyquist Zone.
Rational Rate Filter Implementation

- Implementation
  - Build a polyphase filter structure for interpolation, but then only take every Qth output from the commutator as shown in Fig. 7.20

![Figure 7.20 Detail Presentation of 5/3 Interpolator](image)
Arbitrary Resampling Ratio

• In the previous example, the interpolators output commutator was sampled at fixed intervals according to the integer denominator of P/Q.

• What would happen if Q were non-integer and we selected the commutator output time sample closest to the one desired? Or interpolate between two samples?
Arbitrary Ratio Resampling

- Interpolation

- Nearest Neighbor
Performance Example (same ratio)

• 1 to 5 Interpolation: 5/1

• 32/6.4 Interpolate-Decimate

Stopband sidelobes increased
Performance Example (2)

• 32/6.4 Interpolate-Decimate

![Spectral Response of Up 32 down 6.4 Interpolated Filter](image1)

Figure 7.29 Time and Frequency Response of 32/6.4 Nearest Neighbor Interpolator

• 32/6.37 Interpolate-Decimate

![Spectral Response of Up 32 down 6.37 Interpolated Filter](image2)

Figure 7.31 Time and Frequency Response of 32/6.37 Nearest Neighbor Interpolator

Stopband sidelobes worse
Matlab Example

- Chap7_2.m
  - Rational Rate Example with integers: 4/17 rate.

- Chap7_3.m
  - Rational Rate Example with non-integers: 32/6.37 rate.
  - Nearest neighbor output sample