12.5 Multistage Rate Change
Motivation for Multi-Stage Schemes

Consider Decimation:
When $M$ is large (typically > 10 or so) it is usually inefficient to implement decimation in a single step (i.e., in a single stage).

The Culprit: Large $M$ requires the LPF to have a stopband edge of $\theta_s = \pi/M$, which is small for large $M$
  ➞ Need a LPF with a very narrow passband
  ➞ Requires a long FIR filter
  ➞ Inefficient since long filters require a large # of multiplies

Solution: If $M$ can be factored into a product of integers ($M = M_1 M_2 M_3 \ldots M_p$). Then decimation by $M$ can be done by:
Trick to Get Efficiency from Multi-Stage

The design of $H_1(z)$ (& other “front-end” stages) can be relaxed from what you would use for a single-stage design.

Certainly, you need $H_1(z)$ to have $\theta_s = \pi/M_1 > \pi/M$ so no aliasing occurs after $\downarrow M_1$.

But… it is even better than that.

Can let $\theta_s > \pi/M_1$ … which lets some aliasing occur

But… only so much aliasing – such that the aliasing that occurs gets suppressed by the next filter, $H_2(z)$

Higher Stopband Edge ➔ Shorter Filter ➔ More Efficient
Let’s See Why for a 2-Stage Case

Say that the signal $x[n]$ has “spectral content of worth” only up to frequency $\theta = \theta_p < \pi/M$… with $M = M_1M_2$.

**Single-Stage Method**
Suppose we decimate using a single-stage scheme:

Then we need

---

Sharp Transition ➔ Long Filter
Let’s See Why for a 2-Stage Case (cont.)

2-Stage Method

\[
\begin{align*}
&x[n] &\rightarrow H_1(z) &\downarrow M_1 &\rightarrow H_2(z) &\downarrow M_2 &\rightarrow y[n]
\end{align*}
\]

After \( H_1(z) \) but before \( \downarrow M_1 \) we need:

\[
\theta_{s,1} = \frac{(2M_2 - 1)\pi}{M}
\]

Slow Transition \( \rightarrow \) Short Filter

Same Passband as Single-Stage
Let’s See Why for a 2-Stage Case (cont.)

Let’s see the impact of this slower transition on aliasing:

After 1st Filter

\[ H_1(z) \]

\[ \theta_p \quad (2M_2 - 1)\pi / M \]

After 1st Dec

\[ H_1(z) \]

\[ M_1\theta_p \quad \pi \quad 2\pi \]

\[ 2\pi - M_1(2M_2 - 1)\pi / M = \pi / M_2 \]

\[ M_1(2M_2 - 1)\pi / M = 2\pi - \pi / M_2 \]

Stopband Suppresses Aliased Replica

After 2nd Filter

\[ H_2(z) \]

\[ M_1\theta_p \quad \pi \quad \theta \]
Design Requirements
So… say you want to design a 2-stage multirate scheme instead of a 1-stage multirate scheme:

If single stage, say the specs need to be:
- Passband Cutoff = $\theta_p$  Passband Ripple = $\delta_p$
- Stopband Cutoff = $\theta_s$  Stopband Level = $\delta_s$

For a 2-stage scheme, our above results say we need:

- **1\textsuperscript{st} Stage**
  - $\theta_{p,1} = \theta_p$
  - $\theta_{s,1} = (2M_2-1)\pi/M > \theta_s$
  - $\delta_{p,1} = \delta_p/2$
  - $\delta_{s,1} = \delta_s$

- **2\textsuperscript{nd} Stage**
  - $\theta_{p,2} = M_1\theta_p$
  - $\theta_{s,2} = \pi/M_2$
  - $\delta_{p,2} = \delta_p/2$
  - $\delta_{s,2} = \delta_s$

Passband Ripple is split between 2 filters.
Ex. 12.6: How 2-Stage Reduces Computation

Goal: Decimation by $M = 12$

Filter Requirements

$\delta_p = 0.01 \quad \leftarrow$ to give some desired fidelity (application specific)
$\delta_s = 0.001 \quad \leftarrow$ to limit aliasing to desired level (application specific)

$\theta_p = \frac{\pi}{16} \quad \leftarrow$ to pass desired band (application specific)
$\theta_s = \frac{\pi}{12} = \frac{\pi}{M} \quad \leftarrow$ to prevent aliasing for desired decimation rate

Single-Stage Method

Length of filter determines the # of computations

$\Rightarrow$ Use (9.79) to estimate filter order needed:

$$N = \frac{-20 \log_{10} \sqrt{\delta_p \delta_s} - 13}{2.32 \left[ \frac{\theta_s - \theta_p}{\text{transition width}} \right]}$$
Ex. 12.6: Single-Stage Method (cont.)

Using this order estimate for the given filter requirements gives:

\[ N = 244 \quad \Rightarrow \quad \text{Length: } L = N + 1 = 245 \]

Note: #'s given below differ slightly from book because it (wrongly) uses \( N \) instead of \( L \) in its computation estimates

Our chosen complexity measure: \# Multiplies/Input Sample

Each output sample (after decimation): \( L \) Multiplies/Output Sample

There are \( M \) Input Samples/Output Sample (due to decimation)

\[ \Rightarrow \frac{\text{(# Multiplies)}}{\text{(Input Sample)}} = \frac{L}{M} = \frac{245}{12} \approx 20.4 \]

Single-Stage Complexity = 20.4 multiplies/input
Ex. 12.6: Double-Stage Method \((M = M_1M_2: 12 = 3\times4)\)

1st Stage

- \(\theta_{p,1} = \theta_p = \pi/16\) \(\delta_{p,1} = \delta_p/2 = 0.005\)
- \(\theta_{s,1} = (2M_2-1)\pi/M = 7\pi/12\) \(\delta_{s,1} = \delta_s = 0.001\)

Estimated filter order gives: \(N_1 = 11\) \(L_1 = 12\)
So… Mult/Input = \(L_1/M_1 = 12/3 = 4\)

2nd Stage

- \(\theta_{p,2} = M_1\theta_p = 3\pi/16\) \(\delta_{p,2} = \delta_p/2 = 0.005\)
- \(\theta_{s,2} = \pi/M_2 = \pi/4\) \(\delta_{s,2} = \delta_s = 0.001\)

Estimated filter order gives: \(N_2 = 88\) \(L_2 = 89\)
So… Mult/Input = \(L_2/(M_1 M_2) = 89/12 = 7.4\)

Double-Stage Complexity = 4 + 7.4 = 11.4 multiplies/input
2-Stage has \(\approx \frac{1}{2}\) Complexity of 1-Stage
Comments on Multistage Method

Q: What happens in this example when order of stages is switched? i.e., $M_1 = 4$ and $M_2 = 3$ (Left as Exercise!!)

These 2-Stage design ideas can be extended to p-stage designs: $M = M_1 M_2 M_3 \ldots M_p$

The order of these multiple stages matters

See textbook for discussion of multistage interpolation
Application: Multistage Rate Change

Convert Digital Audio Tape (DAT) format to Compact Disk (CD)

DAT uses $F_s = 48$ kHz
CD uses $F_s = 44.1$ kHz

Rate Change Ratio $= \frac{L}{M} = \frac{147}{160}$

Single-Stage Approach:

$F_s = 48$ kHz $F_s = 7.056$ MHz!!!!

Multiple-Stage Approach: $L = 147 = 7 \times 7 \times 3$
$M = 160 = 10 \times 8 \times 2$