Interpolated Delay Lines, Ideal Bandlimited Interpolation, and Fractional Delay Filter Design

Julius Smith and Nelson Lee

RealSimple Project*
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

June 5, 2008

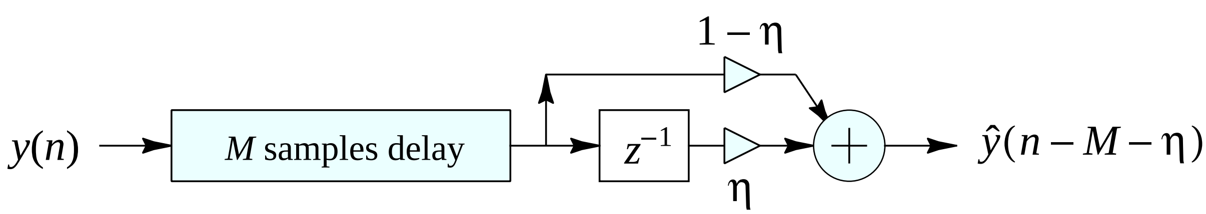
*Work supported by the Wallenberg Global Learning Network

Outline

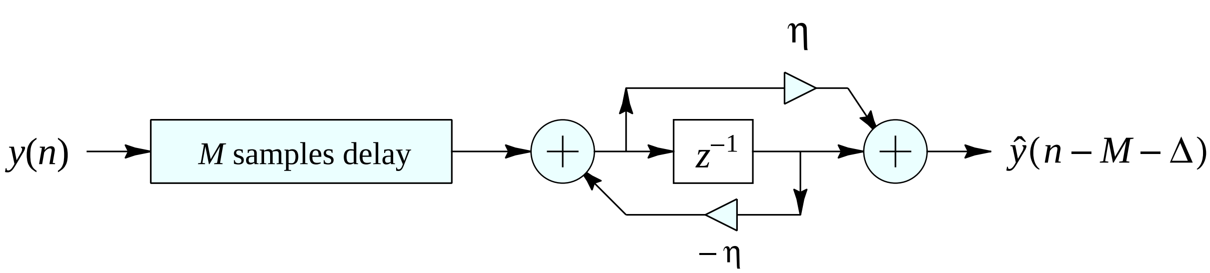
- Low-Order (Fast) Interpolators
  - Linear
  - Allpass
- High-Order Interpolation
  - Ideal Bandlimited Interpolation
  - Windowed-Sinc Interpolation
- High-Order Fractional Delay Filtering
  - Lagrange
  - Farrow Structure
  - Thiran Allpass
- Optimal FIR Filter Design for Interpolation
  - Least Squares
  - Comparison to Lagrange

Simple Interpolators suitable for Real Time Fractional Delay Filtering

Linearly Interpolated Delay Line (1st-Order FIR)



Allpass Interpolated Delay Line (1st-Order)



$$\Delta \approx \frac{1-\eta}{1+\eta}$$

Linear Interpolation

Simplest of all, and the most commonly used:

$$\hat{y}(n-\eta) = (1-\eta)\cdot y(n) + \eta\cdot y(n-1)$$

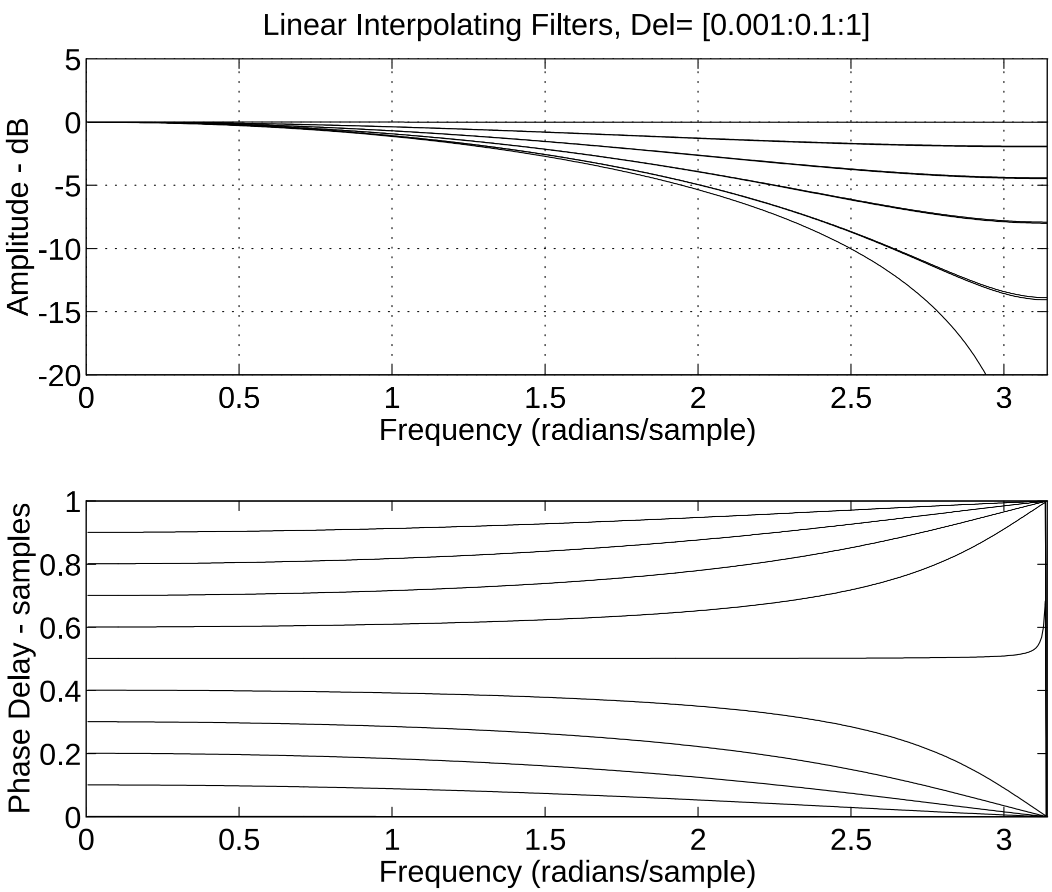
where $\eta =$ desired fractional delay.

One-multiply form:

$$\hat{y}(n-\eta) = y(n) + \eta\cdot[y(n-1) - y(n)]$$

- Works best with *lowpass* signals
  (Natural spectra tend to roll off rapidly)
- Works well with *over-sampling*

Frequency Responses of Linear Interpolation for Delays between 0 and 1



Linear Interpolation as a Convolution

Equivalent to filtering the *continuous-time* impulse train

$$\sum_{n=0}^{N-1} y(nT)\delta(t-nT)$$

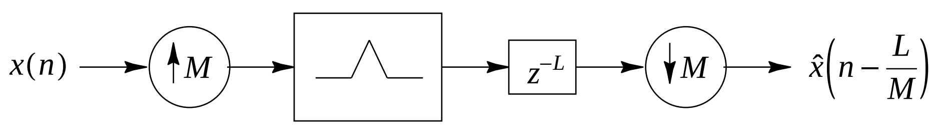
with the *continuous-time* "triangular pulse" FIR filter

$$h_l(t) = \begin{cases} 1-|t/T|\,, & |t| \le T \\ 0, & \text{otherwise} \end{cases}$$

followed by *sampling* at the desired phase

Replacing $h_l(t)$ by $h_s(t) \stackrel{\Delta}{=} \text{sinc}\left(\frac{t}{T}\right)$ converts linear interpolation to *ideal bandlimited interpolation* (to be discussed later)

Upsample, Shift, Downsample View



First-Order Allpass Interpolation

$$\begin{aligned}\hat{x}(n-\Delta) \stackrel{\Delta}{=} y(n) &= \eta \cdot x(n) + x(n-1) - \eta \cdot y(n-1) \\ &= \eta \cdot [x(n) - y(n-1)] + x(n-1)\end{aligned}$$

$$H(z) = \frac{\eta + z^{-1}}{1 + \eta z^{-1}}$$

- Low frequency delay given by

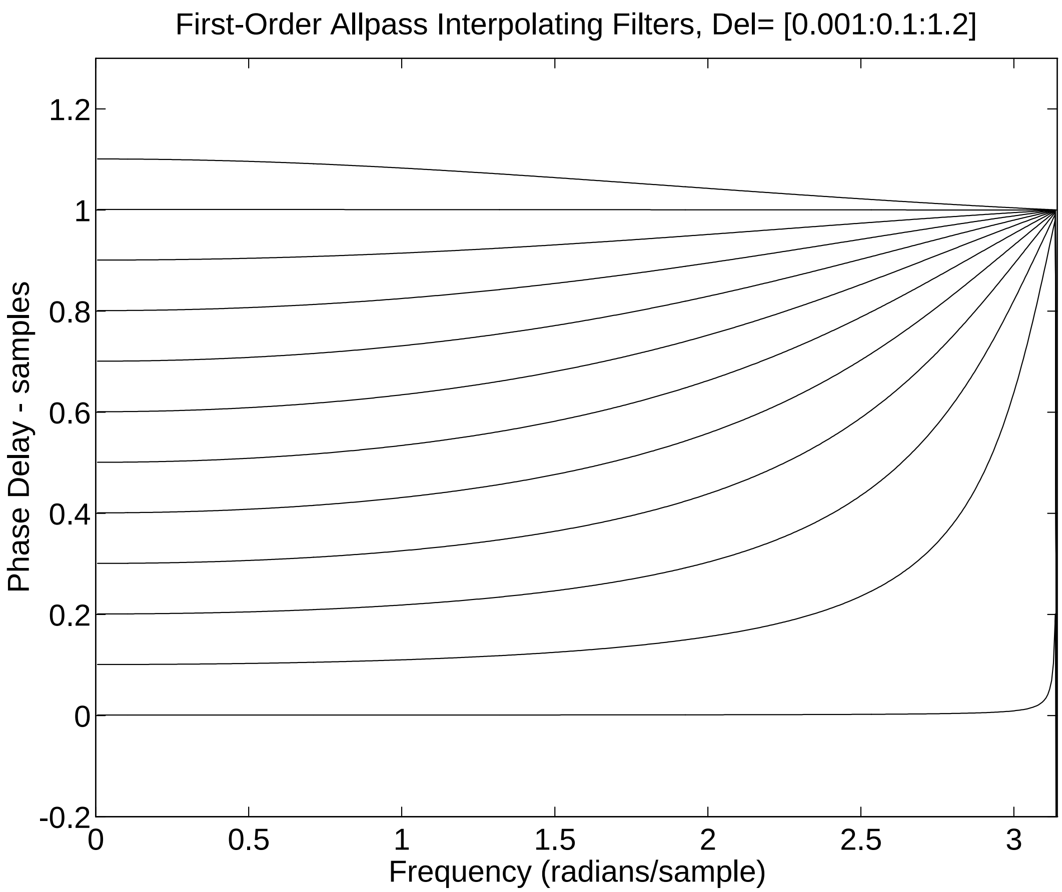
$$\Delta \approx \frac{1-\eta}{1+\eta} \quad \text{(exact at DC)}$$

- Same complexity as linear interpolation
- Good for delay-line interpolation, *not* random access
- Best used with *fixed* fractional delay Δ
- To avoid pole near $z = -1$, use offset delay range, e.g.,

$$\Delta \in [0.1, 1.1] \;\leftrightarrow\; \eta \in [-0.05, 0.82]$$

Intuitively, ramping the coefficients of the allpass gradually "grows" or "hides" one sample of delay. This tells us how to handle resets when crossing sample boundaries.

Phase Delays of First-Order Allpass Interpolators for Various Desired Delays



Ideal Bandlimited Interpolation

Ideal interpolation for digital audio is *bandlimited interpolation*, i.e., samples are *uniquely* interpolated based on the assumption of zero spectral energy for $|f| \geq f_s/2$.

Ideal *bandlimited interpolation* is *sinc interpolation*:

$$y(t) = (y * h_s)(t) = \sum_{n=0}^{N-1} y(nT) h_s(t - nT)$$

where

$$\begin{aligned} h_s(t) &\stackrel{\Delta}{=} \text{sinc}(f_s t) \\ \text{sinc}(x) &\stackrel{\Delta}{=} \frac{\sin(\pi x)}{\pi x} \end{aligned}$$

(Proof: sampling theorem)

Applications of Bandlimited Interpolation

Bandlimited Interpolation is used in (e.g.)

- Sampling-rate conversion
- Wavetable/sampling synthesis
- Virtual analog synthesis
- Oversampling D/A converters
- Fractional delay filtering

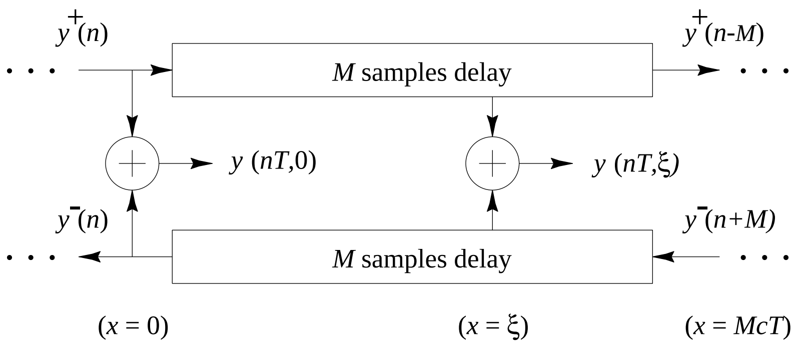
Fractional delay filtering is a *special case* of bandlimited interpolation:

- Fractional delay filters only need *sequential access* $\Rightarrow$ *IIR filters* can be used
- General bandlimited interpolation requires *random access* $\Rightarrow$ *FIR filters* normally used

Fractional Delay Filters are used for (among other things)

- Time-varying delay lines (flanging, chorus, leslie)
- Resonator tuning in digital waveguide models
- Exact tonehole placement in woodwind models
- Beam steering of microphone / speaker arrays

Example Application of Fractional Delay Filtering and Bandlimited Interpolation

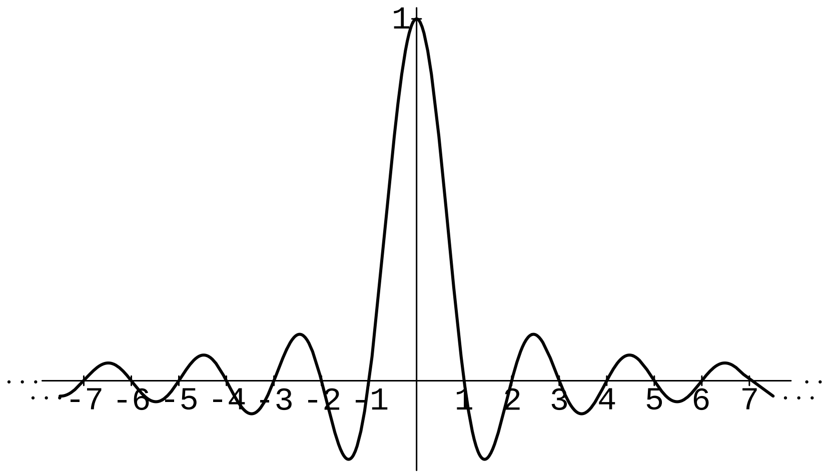


Digital Waveguide String Model

- "Pick-up" needs Bandlimited Interpolation
- "Tuning" needs Fractional Delay Filtering

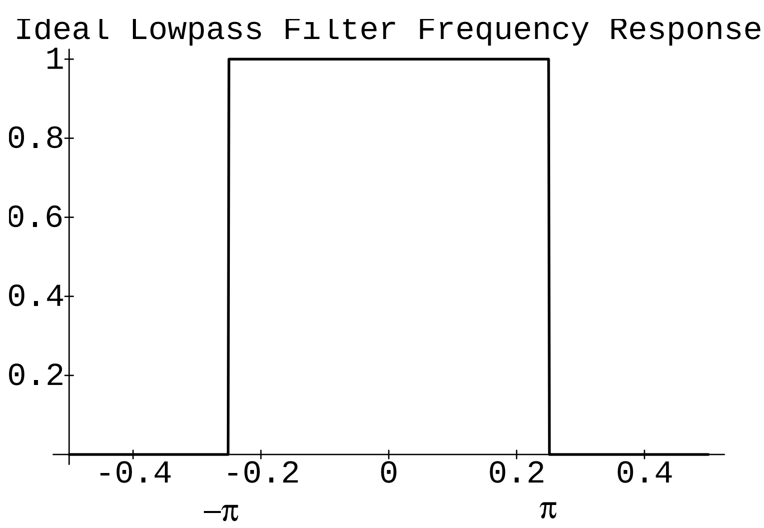
The Sinc Function ("Cardinal Sine")

$$\text{sinc}(t) \triangleq \frac{\sin(\pi t)}{\pi t}$$



Sinc Function

The sinc function is the impulse response of the ideal lowpass filter which cuts off at half the sampling rate



Ideal D/A Conversion

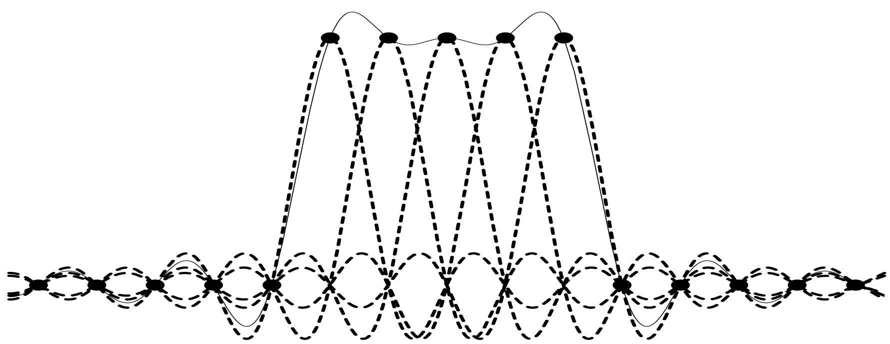
Each sample in the time domain scales and locates one *sinc function* in the unique, continuous, bandlimited interpolation of the sampled signal.

Convolving a sampled signal $y(n)$ with sinc$(n-\eta)$ "evaluates" the signal at an arbitrary continuous time $\eta \in \mathbf{R}$:

$$\begin{aligned} y(\eta) &= \sum_{n=0}^{N-1} y(n)\mathsf{sinc}(\eta - n) \\ &= \textsc{Sample}\{y * \textsc{Shift}_n(\delta)\} \end{aligned}$$

Ideal D/A Example

Reconstruction of a bandlimited rectangular pulse $x(t)$ from its samples $x = [\ldots, 0, 1, 1, 1, 1, 1, 0, \ldots]$:



Bandlimited Rectangular Pulse Reconstruction

Catch

- Sinc function is infinitely long and noncausal
- Must be available in *continuous* form

Optimal Least Squares Bandlimited Interpolation Formulated as a Fractional Delay Filter

Note that interpolation is a special case of *linear filtering*. (Proof: Convolution representation above.)

Consider a filter which delays its input by Δ samples:

- Ideal impulse response = *bandlimited delayed impulse* = *delayed sinc*

$$h_\Delta(t) = \mathsf{sinc}(t-\Delta) \stackrel{\Delta}{=} \frac{\sin[\pi(t-\Delta)]}{\pi(t-\Delta)}$$

- Ideal frequency response = *"brick wall" lowpass* response, cutting off at $f_s/2$ and having *linear phase* $e^{-j\omega\Delta T}$

$$\begin{aligned} H_\Delta(e^{j\omega}) &\stackrel{\Delta}{=} \mathrm{DTFT}(h_\Delta) = \begin{cases} e^{-j\omega\Delta}, & |\omega| < \pi f_s \\ 0, & |\omega| \geq \pi f_s \end{cases} \\ \rightarrow \quad H_\Delta(e^{j\omega T}) &= e^{-j\omega\Delta T}, \quad -\pi \leq \omega T < \pi \\ &\leftrightarrow \mathsf{sinc}(n-\Delta), \quad n = 0, \pm 1, \pm 2, \ldots \end{aligned}$$

The sinc function is an infinite-impulse-response (IIR) digital filter with no recursive form $\Rightarrow$ *non-realizable*

To obtain a *finite* impulse response (FIR) interpolating filter, let's formulate a *least-squares filter-design problem:*

Desired Interpolator Frequency Response

$$H_\Delta\left(e^{j\omega T}\right) = e^{-j\omega\Delta T}, \quad \Delta = \text{Desired delay in samples}$$

FIR Filter Frequency Response

$$\hat{H}_\Delta\left(e^{j\omega T}\right) = \sum_{n=0}^{L-1} \hat{h}_\Delta(n) e^{-j\omega n T}$$

Error to Minimize

$$E\left(e^{j\omega T}\right) = H_\Delta\left(e^{j\omega T}\right) - \hat{H}_\Delta\left(e^{j\omega T}\right)$$

L^2 Error Norm

$$\begin{aligned} J(\underline{h}) \stackrel{\Delta}{=} \| E \|_2^2 &= \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \left| E\left(e^{j\omega T}\right)\right|^2 d\omega \\ &= \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \left| H_\Delta\left(e^{j\omega T}\right) - \hat{H}_\Delta\left(e^{j\omega T}\right)\right|^2 d\omega \end{aligned}$$

By Parseval's Theorem

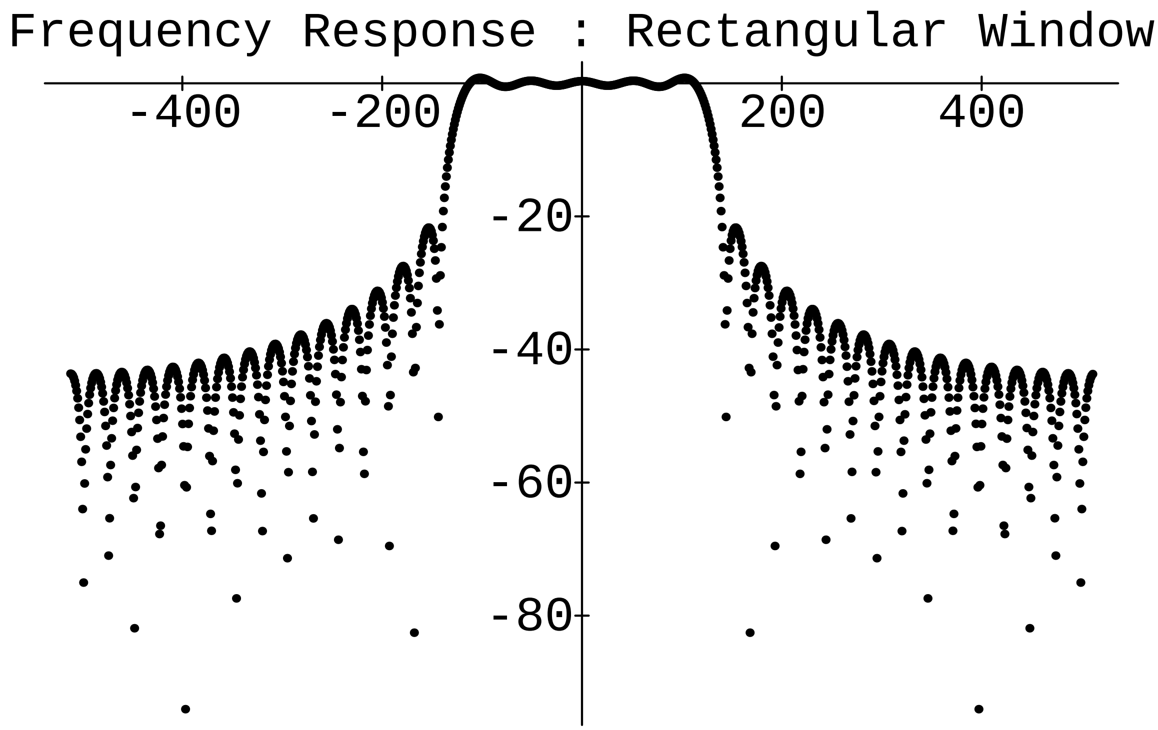
$$J(\underline{h}) = \sum_{n=0}^{\infty} \left| h_\Delta(n) - \hat{h}_\Delta(n)\right|^2$$

Optimal Least-Squares FIR Interpolator

$$\hat{h}_\Delta(n) = \begin{cases} \text{sinc}(n-\Delta), & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

Truncated-Sinc Interpolation

Truncate sinc(t) at 5th zero-crossing to left and right of time 0 to get



Truncated-Sinc Transform

- Vertical axis in dB, horizontal axis in spectral samples
- Optimal in least-squares sense
- Poor stop-band rejection (≈ 20 dB)
- "Gibbs Phenomenon" gives too much "ripple"
- Ripple can be reduced by *tapering* the sinc function to zero instead of simply truncating it.

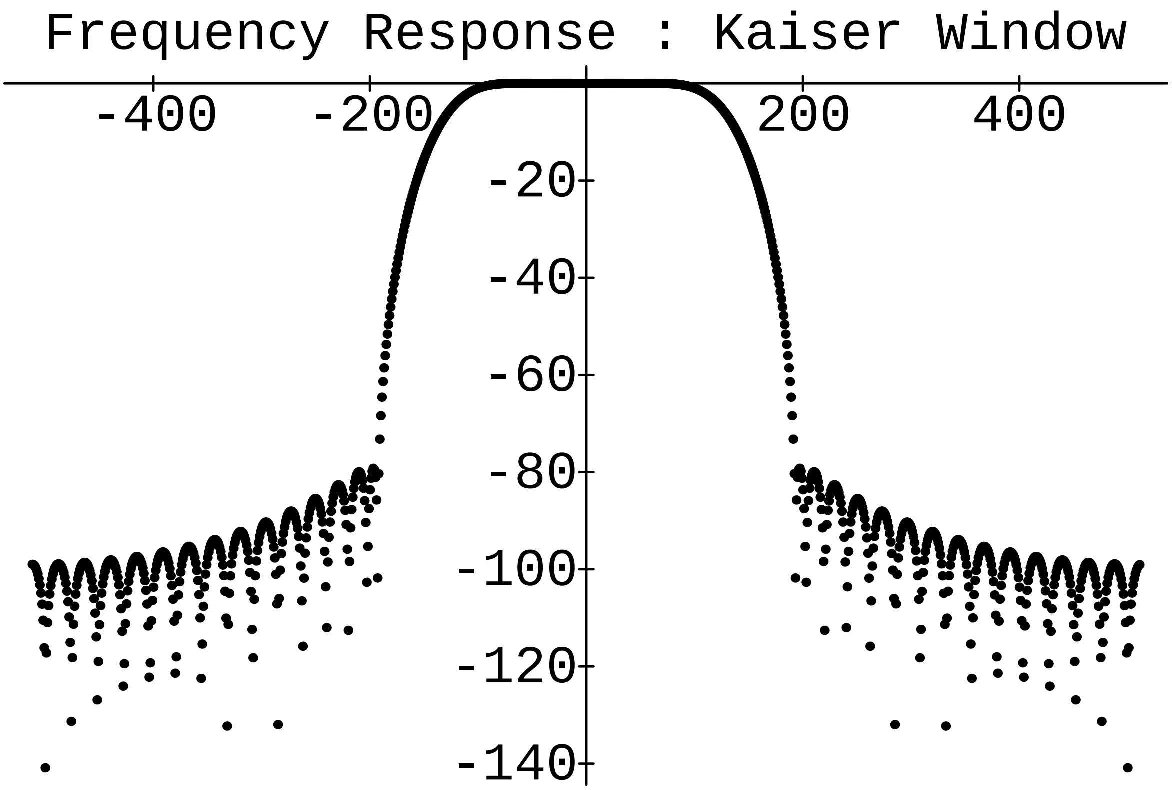
Windowed Sinc Interpolation

- Sinc function can be *windowed* more generally to yield

$$\hat{h}_\Delta(n) = \begin{cases} w(n-\Delta)\mathrm{sinc}[\alpha(n-\Delta)], & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

- Example of *window method* for FIR lowpass filter design applied to sinc functions (ideal lowpass filters) sampled at various phases (corresponding to desired delay between samples)
- For best results, $\Delta \approx L/2$
- $w(n)$ is any real symmetric window (e.g., Hamming, Blackman, Kaiser).
- Non-rectangular windows *taper* truncation which *reduces* Gibbs phenomenon, as in FFT analysis

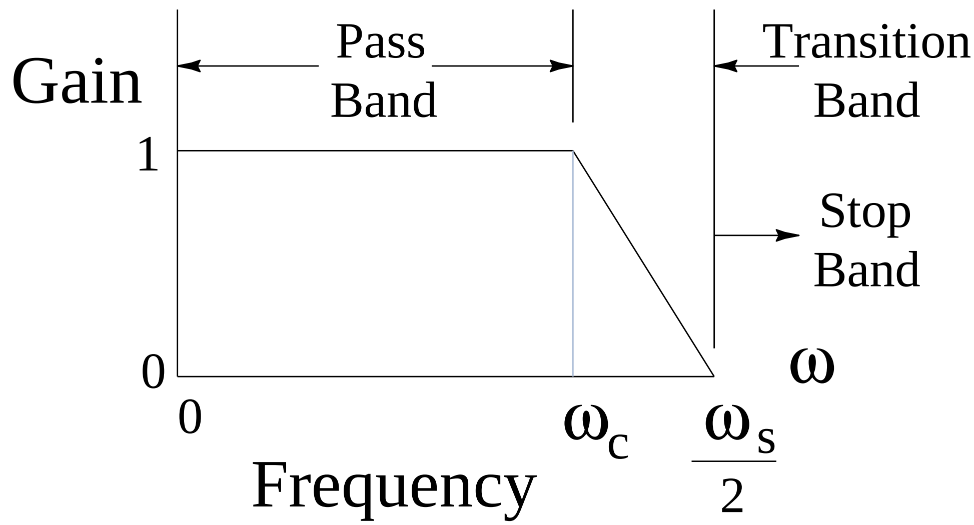
Spectrum of Kaiser-windowed Sinc



Kaiser-Windowed Sinc Transform

- Stopband now starts out close to -80 dB
- Kaiser window has a single parameter which trades off stop-band attenuation versus transition-bandwidth from pass-band to stop-band

Lowpass Filter Design



Lowpass Filter Design Parameters

- In the *transition band*, frequency response "rolls off" from 1 at $\omega_c = \omega_s/(2\alpha)$ to zero (or ≈ 0.5) at $\omega_s/2$.
- Interpolation can remain "perfect" in pass-band

Online references (FIR interpolator design)

- Music 421 Lecture 2 on Windows[1]
- Music 421 Lecture 3 on FIR Digital Filter Design[2]
- Optimal FIR Interpolator Design[3]

[1] http://ccrma.stanford.edu/~jos/Windows/
[2] http://ccrma.stanford.edu/~jos/WinFlt/
[3] http://ccrma.stanford.edu/ jos/resample/optfir.pdf

Oversampling Reduces Filter Length

- Example 1:
  - $f_s = 44.1$ kHz (CD quality)
  - Audio upper limit $= 20$ kHz
  - *Transition band* $= 2.05$ kHz
  - *FIR filter length* $\stackrel{\Delta}{=} L_1$
- Example 2:
  - $f_s = 48$ kHz (e.g., DAT)
  - Audio upper limit $= 20$ kHz
  - *Transition band* $= 4$ kHz
  - *FIR filter length* $\approx L_1/2$
- Required FIR filter length varies inversely with transition bandwidth
  $\Rightarrow$ Required filter length in example 1 is almost *double* ($\approx 4/2.1$) the required filter length for example 2
- Increasing the sampling rate by less than ten percent reduces the filter expense by almost fifty percent

The Digital Audio Resampling Home Page

- C++ software for windowed-sinc interpolation
- C++ software for FIR filter design by window method
- Fixed-point data and filter coefficients
- Can be adapted to time-varying resampling
- Open source, free
- First written in 1983 in SAIL
- URL: http://ccrma.stanford.edu/~jos/resample/
- *Most needed upgrade*:
  - Design and install a set of *optimal* FIR interpolating filters.[4]

[4]http://ccrma.stanford.edu/ jos/resample/optfir.pdf

Lagrange Interpolation

- Lagrange interpolation is just *polynomial interpolation*
- Nth-order polynomial interpolates $N+1$ points
- First-order case $=$ *linear interpolation*

Problem Formulation

Given a set of $N+1$ known samples $f(x_k)$, $k = 0, 1, 2, \ldots, N$, find the *unique* order N *polynomial* $y(x)$ which *interpolates* the samples

Solution (Waring, Lagrange):

$$y(x) = \sum_{k=0}^{N} l_k(x) f(x_k)$$

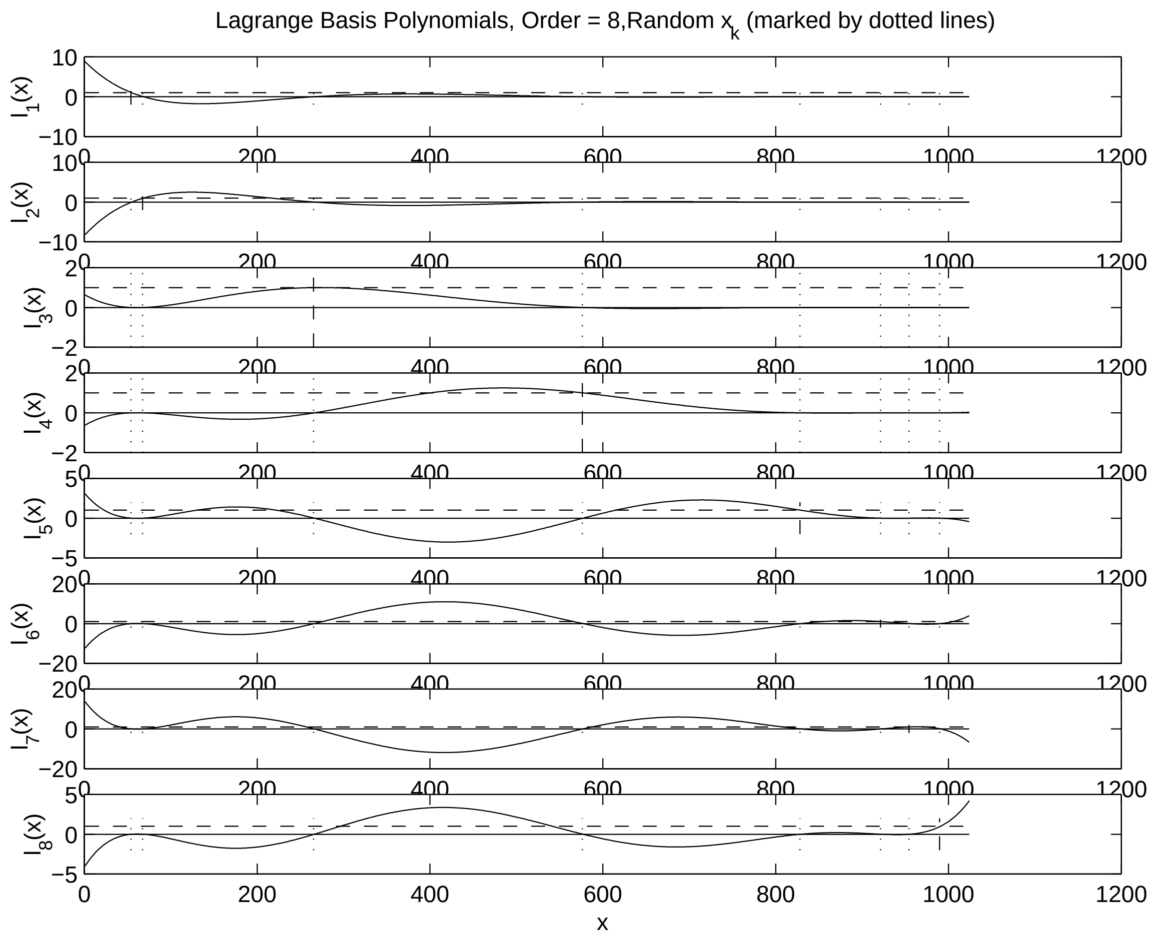
where

$$l_k(x) \triangleq \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_N)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_N)}$$

- Numerator gives a *zero* at all samples but the kth
- Denominator simply *normalizes* $l_k(x)$ to 1 at $x = x_k$
- As a result,
$$l_k(x_j) = \delta_{kj} \triangleq \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$
- Generalized bandlimited impulse = generalized sinc function:
  Each $l_k(x)$ goes through 1 at $x = x_k$ and zero at all other sample points
  I.e., $l_k(x)$ is analogous to sinc$(x - x_k)$

- Lagrange interpolaton is *equivalent* to *windowed sinc* interpolation using a *binomial window*
- Can be viewed as a *linear, spatially varying filter* (in analogy with linear, time-varying filters)

Example Lagrange Basis Functions



Lagrange Interpolation Optimality

In the uniformly sampled case, Lagrange interpolation can be viewed as ordinary FIR filtering:

- Lagrange interpolation filters *maximally flat* in the frequency domain about dc:

  $$\left.\frac{d^m E(e^{j\omega})}{d\omega^m}\right|_{\omega=0} = 0, \quad m = 0, 1, 2, \ldots, N,$$

  where

  $$E(e^{j\omega}) \stackrel{\Delta}{=} e^{-j\omega\Delta} - \sum_{n=0}^{N} h(n) e^{-j\omega n}$$

  and Δ is the desired delay in samples.
- Same optimality criterion as *Butterworth filters* in classical analog filter design
- Can also be viewed as "Pade approximation" to a constant frequency response in the frequency domain

Proof of Maximum Flatness at DC

The maximumally flat fractional-delay FIR filter is obtained by equating to zero all $N+1$ leading terms in the Taylor (Maclaurin) expansion of the frequency-response error at dc:

$$\begin{aligned} 0 &= \left.\frac{d^k}{d\omega^k} E(e^{j\omega})\right|_{\omega=0} \\ &= \left.\frac{d^k}{d\omega^k}\left[e^{-j\omega\Delta} - \sum_{n=0}^{N} h(n)e^{-j\omega n}\right]\right|_{\omega=0} \\ &= (-j\Delta)^k - \sum_{n=0}^{N}(-jn)^k h(n) \\ \Longrightarrow & \boxed{\sum_{n=0}^{N} n^k h(n) = \Delta^k,\ k=0,1,\ldots,N} \end{aligned}$$

This is a linear system of equations of the form $V\underline{h} = \underline{\Delta}$, where V is a Vandermonde matrix. The solution can be written as a ratio of Vandermonde determinants using Cramer's rule. As shown by Cauchy (1812), the determinant of a Vandermonde matrix $[p_i^{j-1}]$, $i,j=1,\ldots,N$ can be expressed in closed form as

$$\begin{aligned} \left|\left[p_i^{j-1}\right]\right| &= \prod_{j>i}(p_j - p_i) \\ &= (p_2-p_1)(p_3-p_1)\cdots(p_N-p_1)\cdots \\ &\quad (p_3-p_2)(p_4-p_2)\cdots(p_N-p_2)\cdots \\ &\quad (p_{N-1}-p_{N-2})(p_N-p_{N-2})\cdot \\ &\quad (p_N-p_{N-1}) \end{aligned}$$

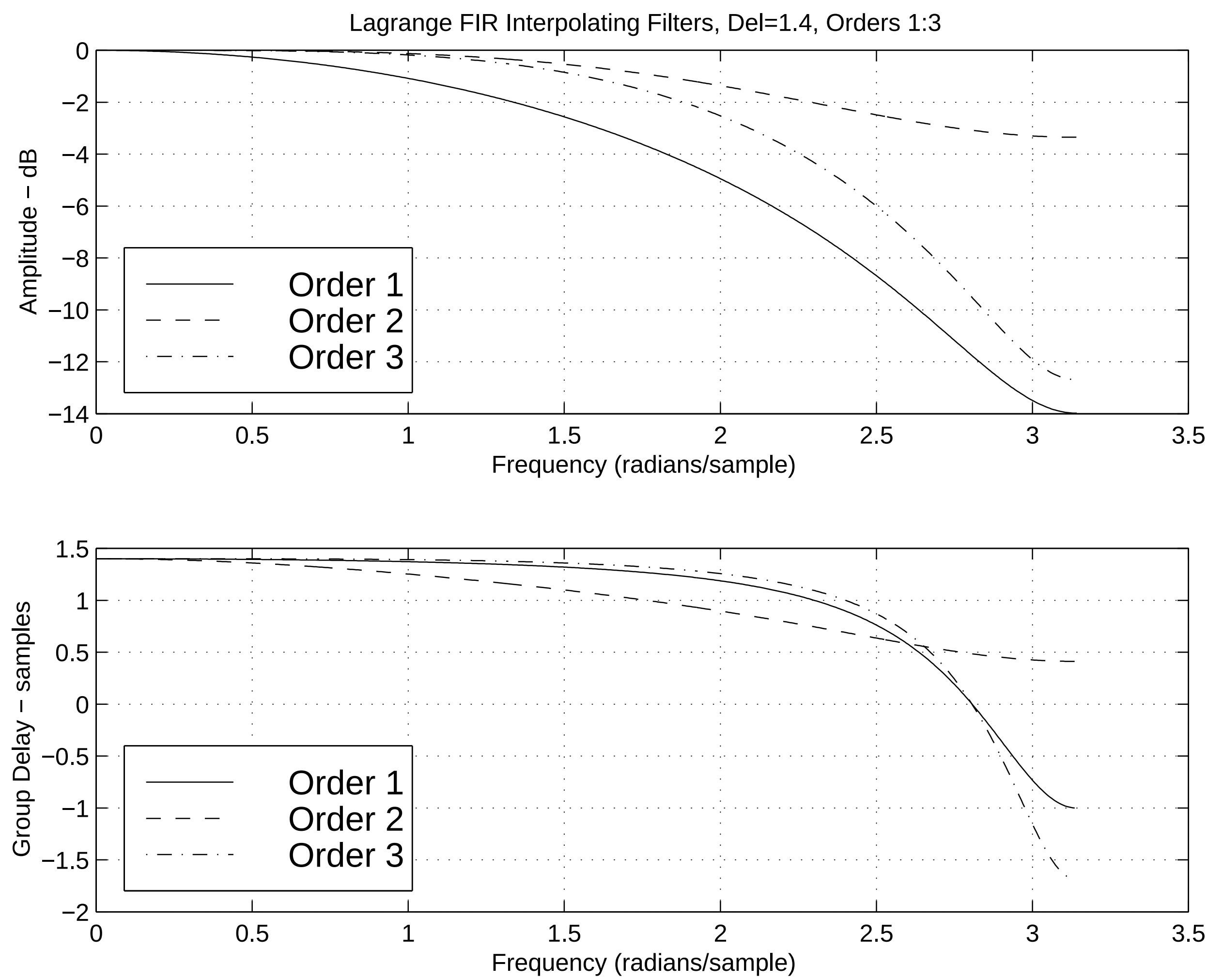
Making this substitution in the solution obtained by Cremer's rule yields that the impulse response of the order N maximally flat fractional-delay FIR filter may be written in closed form as

$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{D-k}{n-k}$$

which coincides with the formula for Lagrange interpolation when the abscissae are equally spaced on the integers from 0 to $N-1$. (Online Reference:[5] Vesa Välimäki's thesis, Chapter 3, Part 2, pp. 82–84)

[5]http://www.acoustics.hut.fi/~vpv/publications/vesan_vaitos/ch3_pt2_lagrange.pdf

Lagrange Interpolator Frequency Responses: Orders 1, 2, and 3



$$\Delta = 1.4$$

Explicit Formula for Lagrange Interpolation Coefficients

$$h_\Delta(n) = \prod_{\substack{k=0 \\ k\neq n}} \frac{\Delta - k}{n - k}, \quad n = 0, 1, 2, \ldots, N$$

Lagrange Interpolation Coefficients
Orders 1, 2, and 3

$h_\Delta Order$	$h_\Delta(0)$	$h_\Delta(1)$	$h_\Delta(2)$	$h_\Delta(3)$
$N = 1$	$1 - \Delta$	Δ		
$N = 2$	$\frac{(\Delta-1)(\Delta-2)}{2}$	$-\Delta(\Delta - 2)$	$\frac{\Delta(\Delta-1)}{2}$	
$N = 3$	$-\frac{(\Delta-1)(\Delta-2)(\Delta-3)}{6}$	$\frac{\Delta(\Delta-2)(\Delta-3)}{2}$	$-\frac{\Delta(\Delta-1)(\Delta-3)}{2}$	$\frac{\Delta(\Delta-1)(\Delta-2)}{6}$

– For $N = 1$, Lagrange interpolation reduces to *linear interpolation* $h = [1 - \Delta, \Delta]$, as before

– For order N, desired delay should be in a one-sample range centered about $\Delta = N/2$

Matlab Code For Lagrange Fractional Delay

```
function h = lagrange(N, delay)
%LAGRANGE  h=lagrange(N,delay) returns order N FIR
%          filter h which implements given delay
%          (in samples).  For best results,
%          delay should be near N/2 +/- 1.
n = 0:N;
h = ones(1,N+1);
for k = 0:N
    index = find(n ~= k);
    h(index) = h(index) *  (delay-k)./ (n(index)-k);
end
```

Relation of Lagrange Interpolation to Windowed Sinc Interpolation

- For an *infinite* number of *equally spaced* samples, with spacing $x_{k+1} - x_k = \Delta$, the Lagrange-interpolation basis polynomials converge to shifts of the *sinc function*, i.e.,

$$l_k(x) = \text{sinc}\left(\frac{x - k\Delta}{\Delta}\right), \quad k = \ldots, -2, -1, 0, 1, 2, \ldots$$

  Proof: As order $\to \infty$, the binomial window $\to$ Gaussian window $\to$ constant (unity).

  Alternate Proof: Every analytic function is determined by its zeros and its value at one nonzero point. Since $\sin(\pi x)$ is zero on all the integers except 0, and since $\text{sinc}(0) = 1$, it therefore coincides with the Lagrangian basis polynomial for $N = \infty$ and $k = 0$.

Variable FIR Interpolating Filter

Basic idea: Each FIR filter coefficient h_n becomes a *polynomial* in the delay parameter Δ:

$$\begin{aligned}
h_\Delta(n) &\isdef \sum_{m=0}^{P} c_n(m)\Delta^m, \quad n = 0,1,2,\ldots,N \\
\Leftrightarrow \; H_\Delta(z) &\isdef \sum_{n=0}^{N} h_\Delta(n) z^{-n} \\
&= \sum_{n=0}^{N} \left[\sum_{m=0}^{P} c_n(m)\Delta^m\right] z^{-n} \\
&= \sum_{m=0}^{P} \left[\sum_{n=0}^{N} c_n(m) z^{-n}\right] \Delta^m \\
&\isdef \sum_{m=0}^{P} C_m(z)\Delta^m
\end{aligned}$$

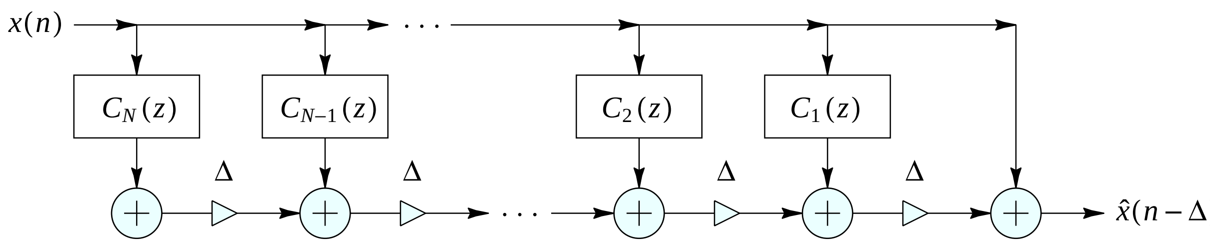
- More generally: $H_\Delta(x) = \sum_m \alpha(\Delta) C_m(z)$
  where $\alpha(\Delta)$ is provided by a *table lookup*
- Basic idea applies to any *one-parameter* filter variation
- Also applies to *time-varying* filters ($\Delta \leftarrow t$)

Farrow Structure for Variable Delay FIR Filters

When the polynomial in Δ is evaluated using *Horner's rule*,

$$\hat{X}_{n-\Delta}(z) = X + \Delta\left[C_1 X + \Delta\left[C_2 X + +\Delta\left[C_3 X + \cdots\right]\right]\right],$$

the filter structure becomes



As delay Δ varies, "basis filters" $C_k(z)$ remain *fixed*
$\Rightarrow$ *very convenient for changing* Δ over time

Farrow Structure Design Procedure

Solve the N_Δ equations

$$z^{-\Delta_i} = \sum_{k=0}^{N} C_k(z)\Delta_i^k, \quad i = 1, 2, \ldots, N_\Delta$$

for the $N+1$ FIR transfer functions $C_k(z)$, each order N_C in general

References: Laakso et al., Farrow

Thiran Allpass Interpolators

Given a desired delay $\Delta = N + \delta$ samples, an order N allpass filter

$$H(z) = \frac{z^{-N} A\left(z^{-1}\right)}{A(z)} = \frac{a_N + a_{N-1} z^{-1} + \cdots + a_1 z^{-(N-1)} + z^{-N}}{1 + a_1 z^{-1} + \cdots + a_{N-1} z^{-(N-1)} + a_N z^{-N}}$$

can be designed having *maximally flat group delay* equal to Δ at DC using the formula

$$a_k = (-1)^k \left(\begin{array}{c} N \\ k \end{array} \right) \prod_{n=0}^{N} \frac{\Delta - N + n}{\Delta - N + k + n}, \; k = 0, 1, 2, \ldots, N$$

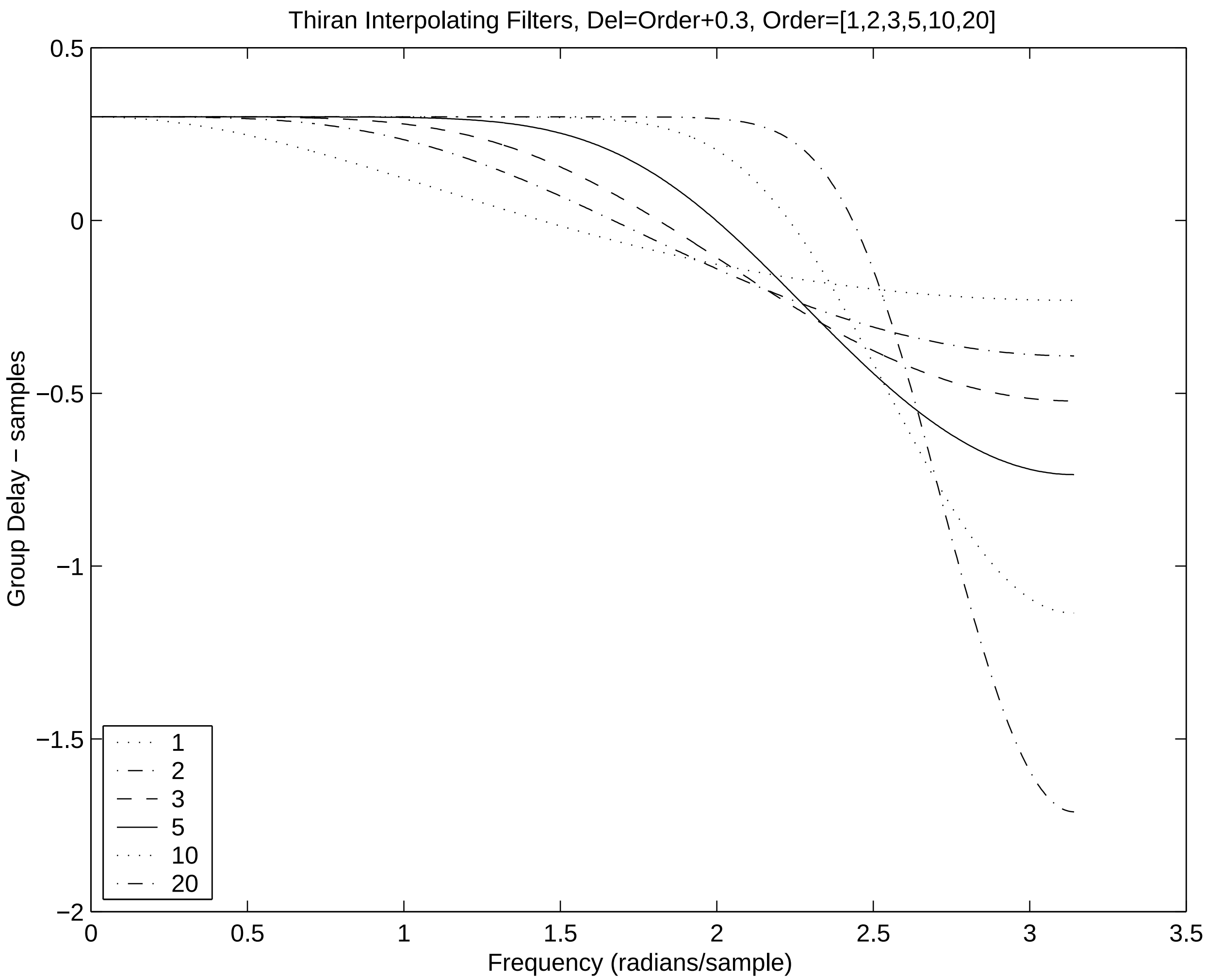
where

$$\left(\begin{array}{c} N \\ k \end{array} \right) = \frac{N!}{k!(N-k)!}$$

denotes the kth *binomial coefficient*

- $a_0 = 1$ without further scaling
- For sufficiently large Δ, stability is guaranteed
  rule of thumb: $\Delta \approx$ order
- Mean group delay is always N samples
  (for any stable Nth-order allpass filter):
  $$\frac{1}{2\pi} \int_0^{2\pi} D(\omega) d\omega \stackrel{\Delta}{=} -\frac{1}{2\pi} \int_0^{2\pi} \Theta'(\omega) d\omega = -\frac{1}{2\pi} \left[\Theta(2\pi) - \Theta(0)\right] = N$$
- Only known closed-form case for allpass interpolators of arbitrary order
- Effective for delay-line interpolation needed for *tuning* since pitch perception is most acute at low frequencies.

Frequency Responses of Thiran Allpass Interpolators for Fractional Delay



Large Delay Changes

When implementing large delay-length changes (by many samples), a useful implementation is to *cross-fade* from the initial delay line configuration to the new configuration.

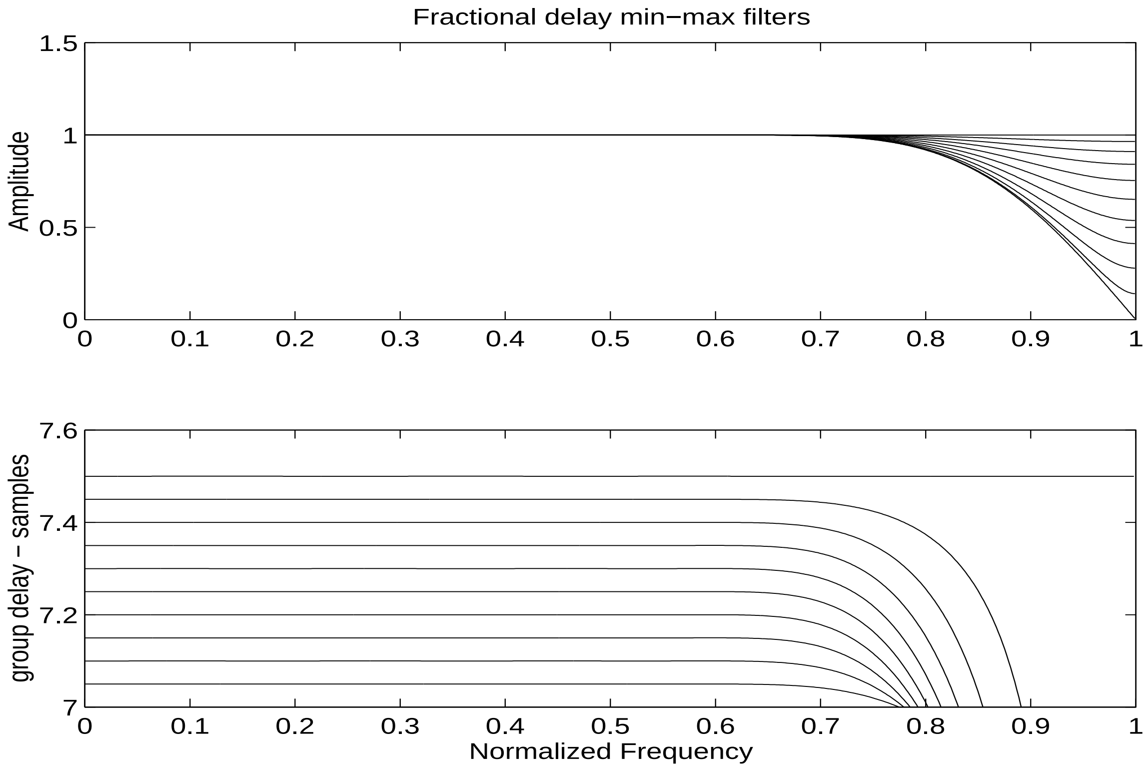
- Computation doubled during cross-fade
- Cross-fade should be long enough to sound smooth
- Not a true "morph" from one delay length to another, since we do not pass through the intermediate delay lengths.
- A single delay line can be *shared* such that the cross-fade occurs from one *read-pointer* (plus associated filtering) to another.

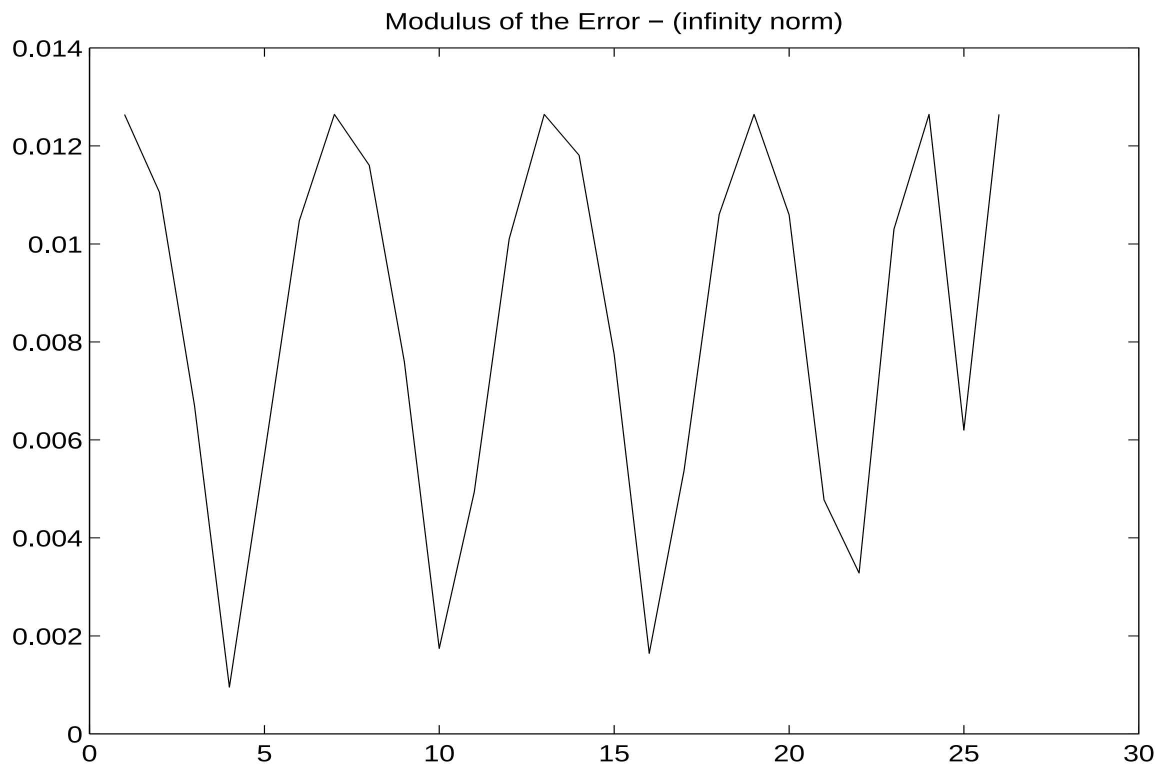
L-Infinity (Chebyshev) Fractional Delay Filters

- Use Linear Programming (LP) for real-valued L_∞-norm minimization
- Remez exchange algorithm (`remez`, `cremez`)
- In the complex case, we have a problem known as a *Quadratically Constrained Quadratic Program*
- Approximated by sets of linear consraints (e.g., a *polygon* can be used to approximate a *circle*)
- Can solve with code developed by Prof. Boyd's group
- See Mohonk-97 paper[6] for details.

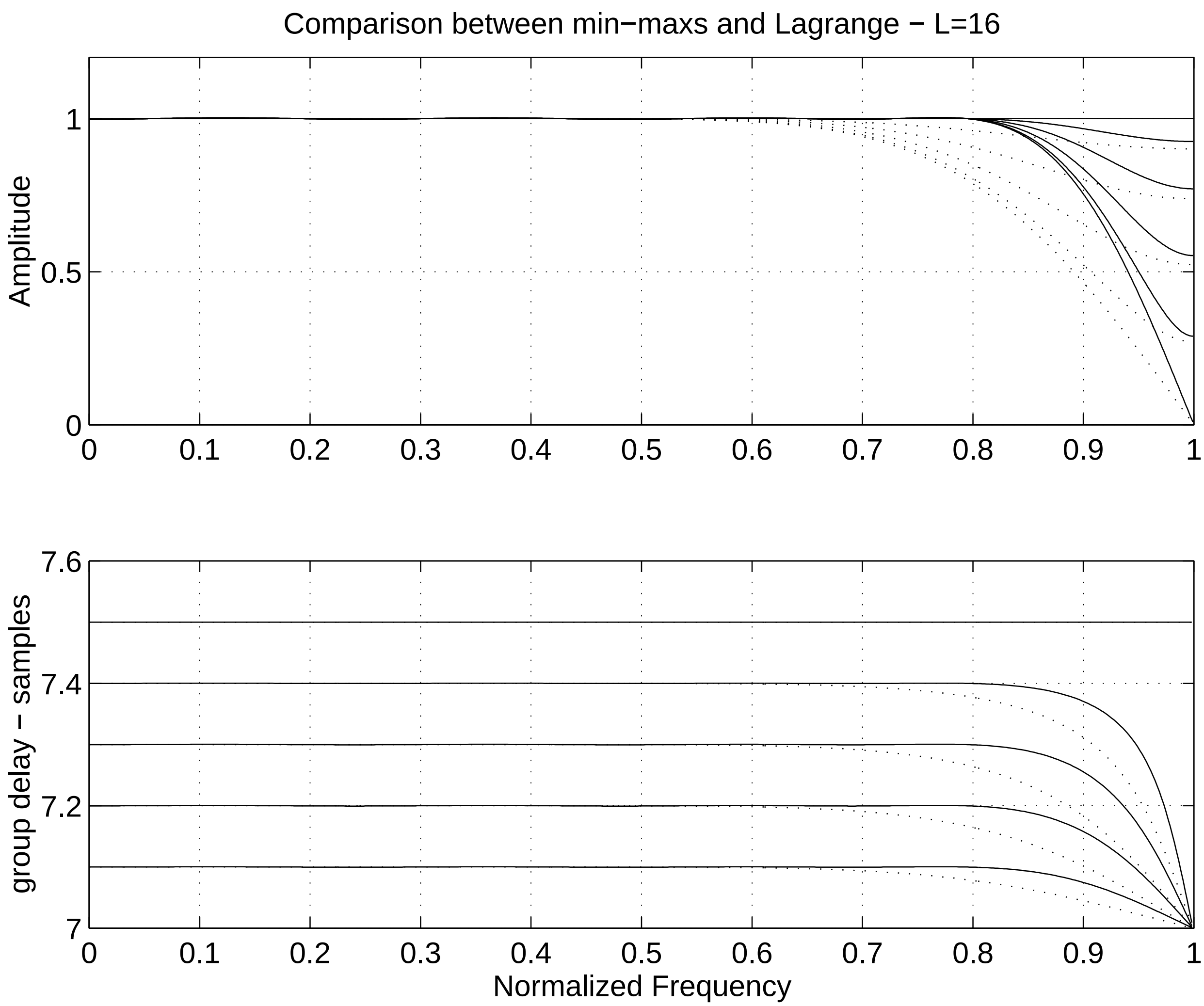
[6] http://ccrma.stanford.edu/ jos/resample/optfir.pdf

Chebyshev FD-FIR Design Example




Modulus of the Error − (infinity norm)
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
5
10
15
20
25
30

Comparison of Lagrange and Optimal Chebyshev Fractional-Delay Filter Frequency Responses



Interpolation Summary

	Order			
	1	N	Large N	∞
FIR	Linear	Lagrange	Windowed Sinc	Sinc
IIR	Allpass_1	Thiran		Sinc