Gisselquist Technology, LLC

Digital Resampling and Interpolation

Dan Gisselquist, Ph.D.

License

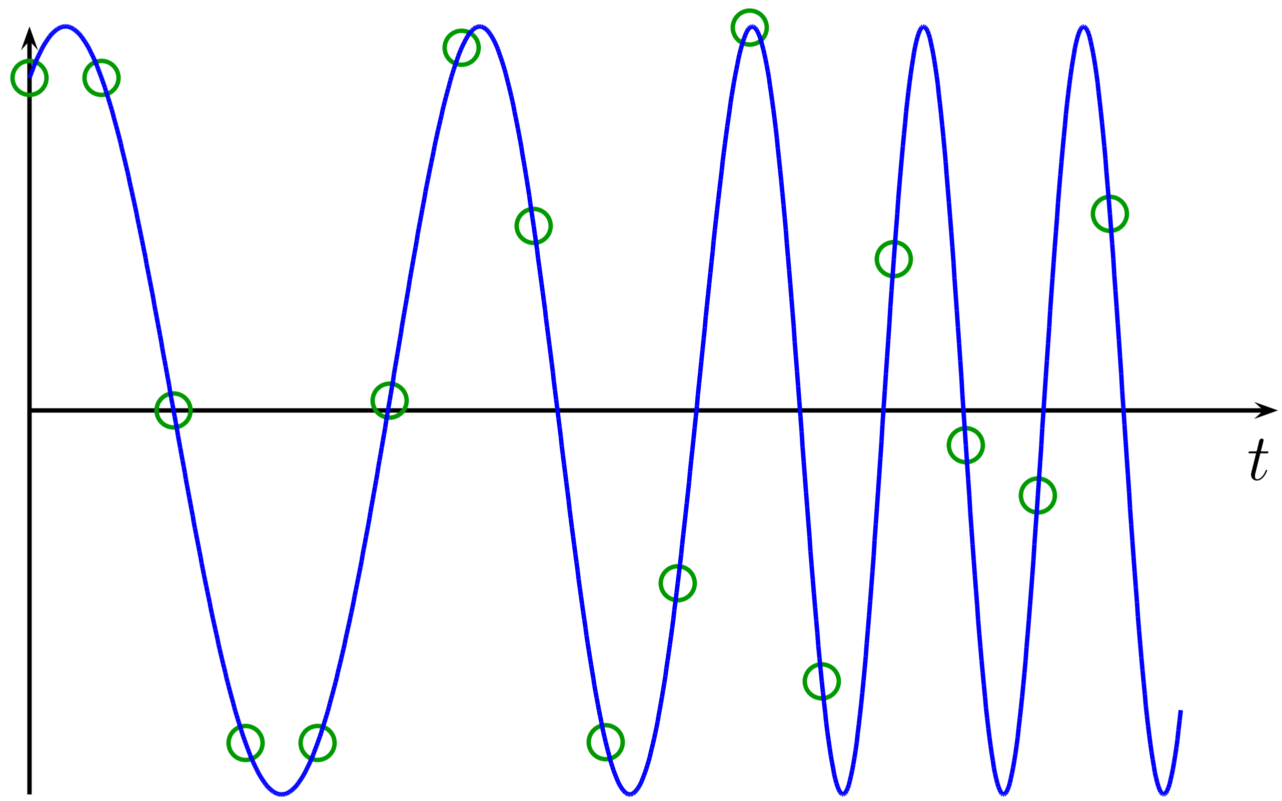
Overview

- Traditional Interpolation
- Rational Resampling
- Mathematical Theory
- Quadratic Interpolation
- Optimal Interpolators
- Sample Applications

Traditional Interpolation

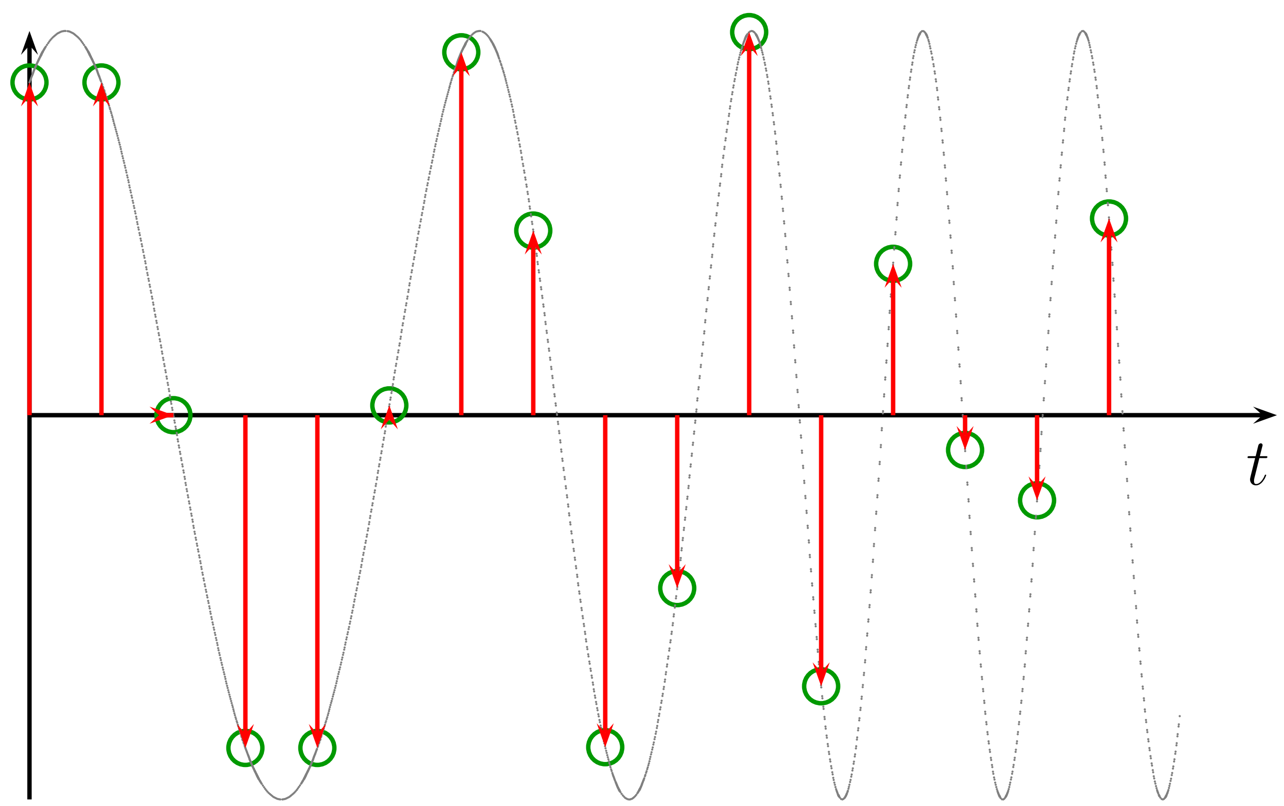
The Obvious

Fact: In digital signal processing,
we use sampled signals.



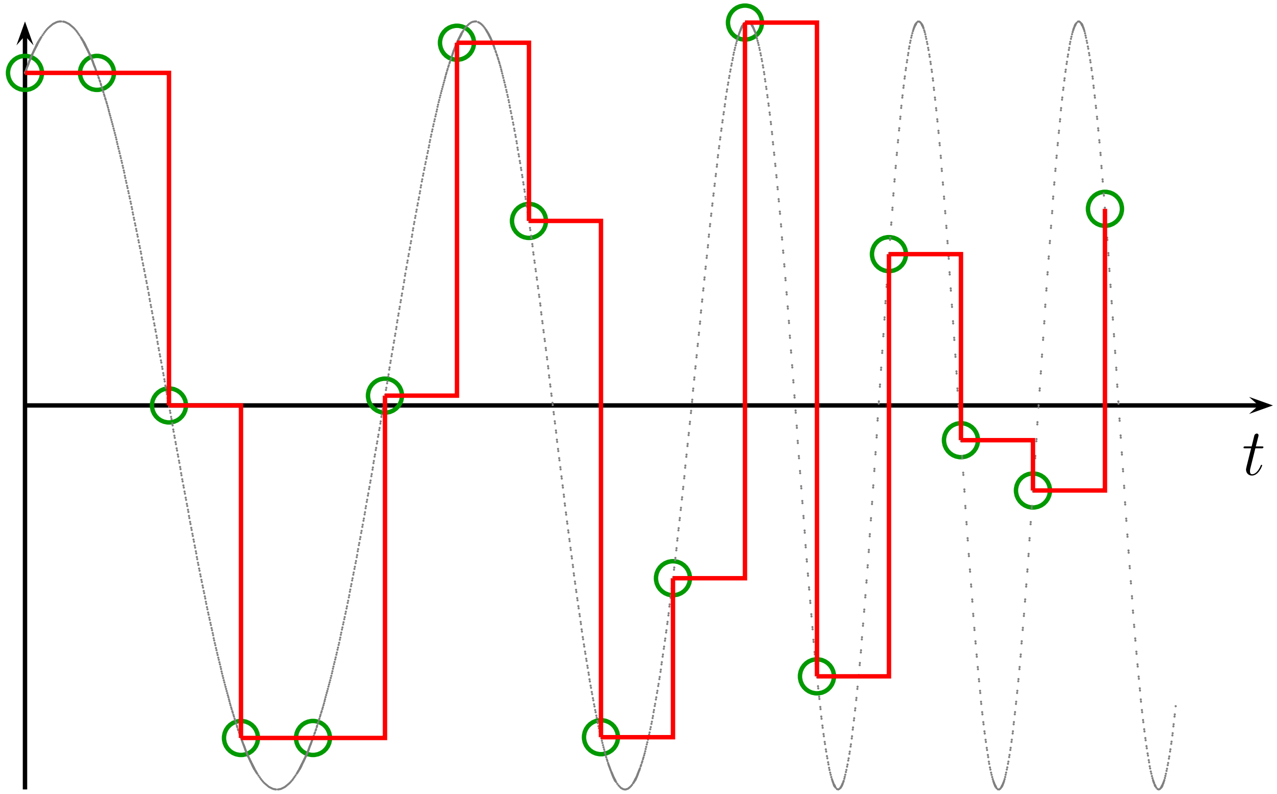
The Problem

Problem: How do you recover the waveform from the samples?



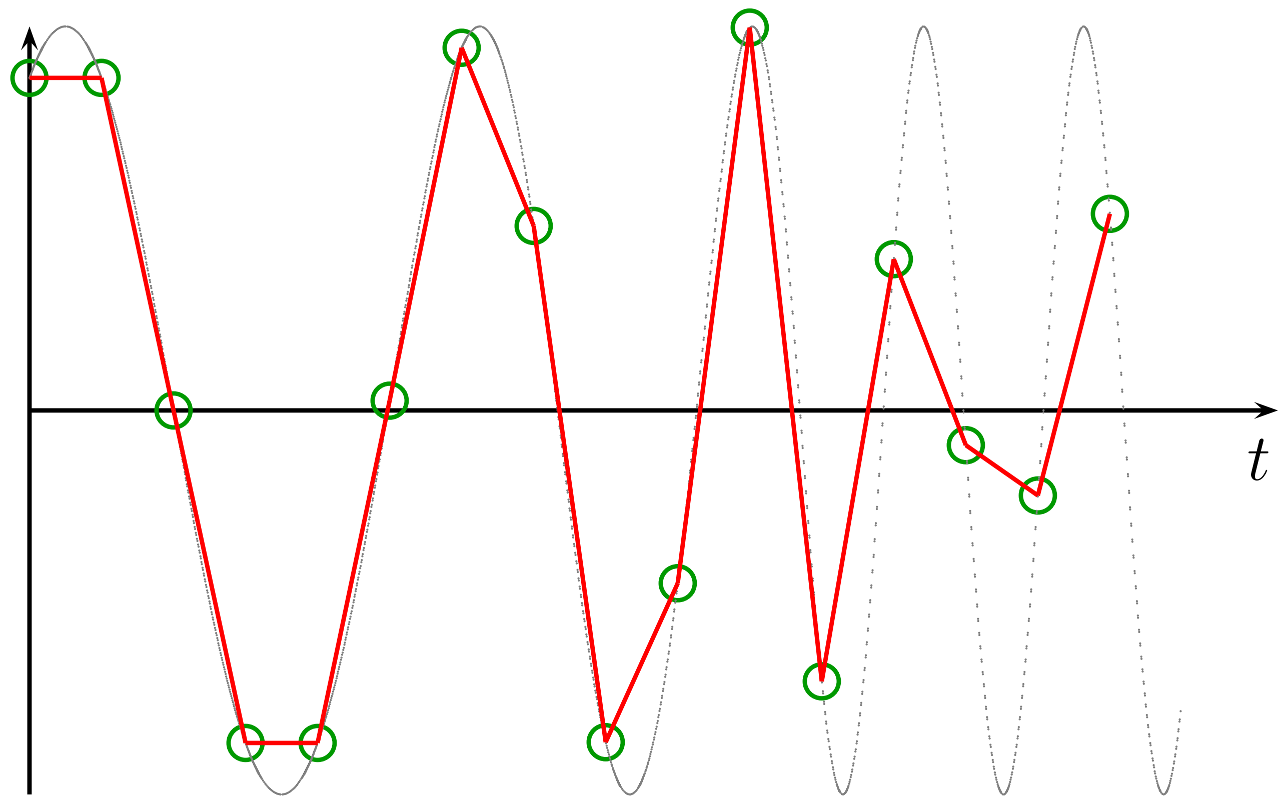
Sample and Hold

Wrong Answer: This doesn't look like my signal!



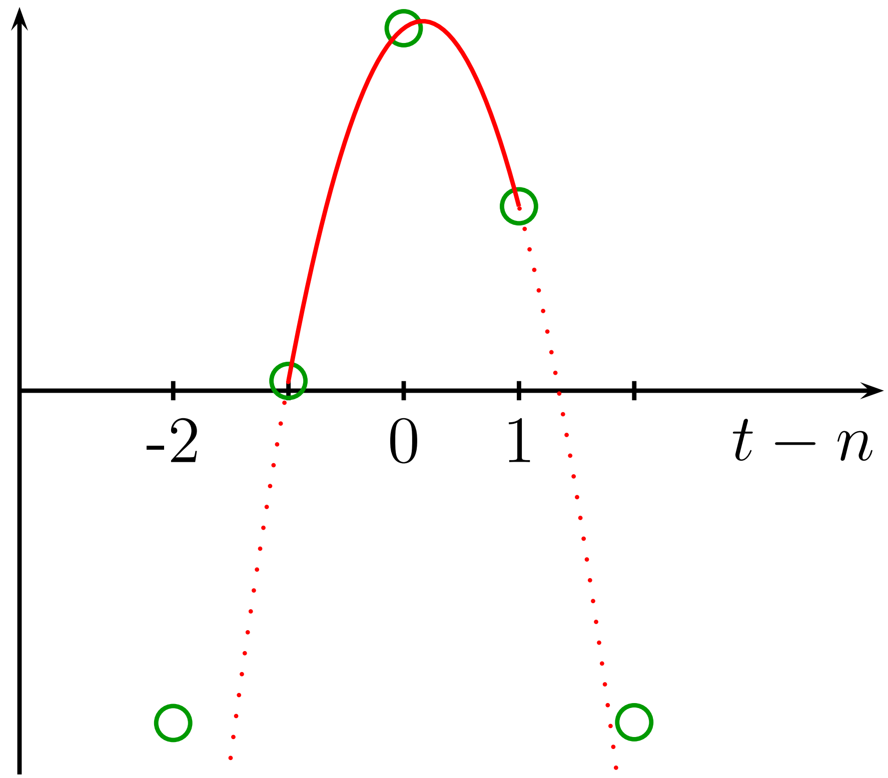
Linear

Wrong Answer: This is better, but my signal was smooth.



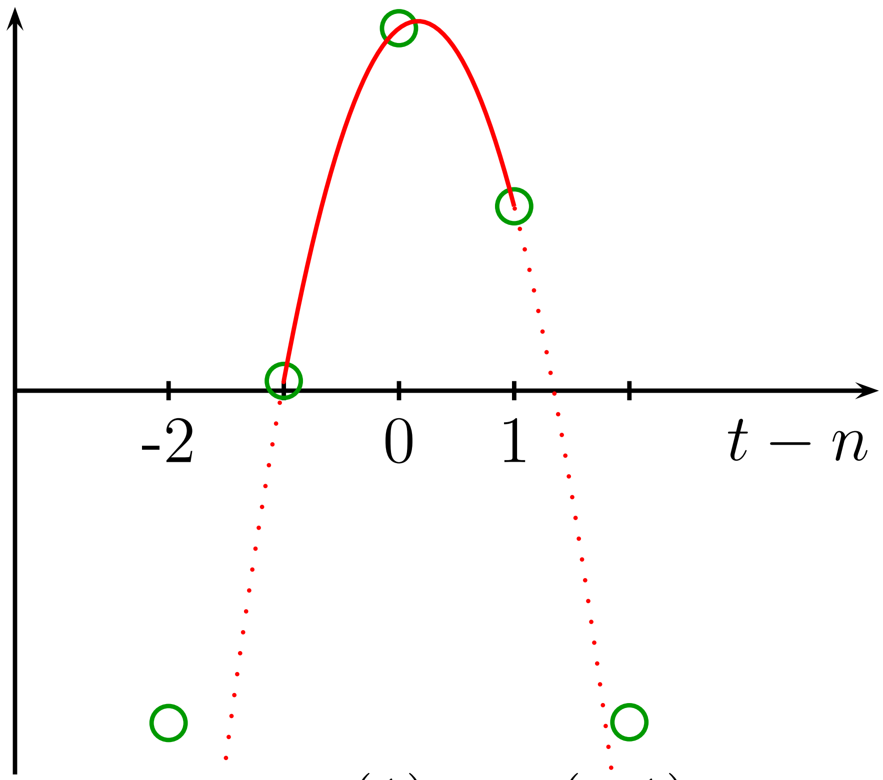
$$y(t) = (x[n+1] - x[n])(t-n) + x[n]$$

Building a Quadratic



We could fit a quadratic to each set of three points, and then move from one quadratic fit to another as we move through the signal.

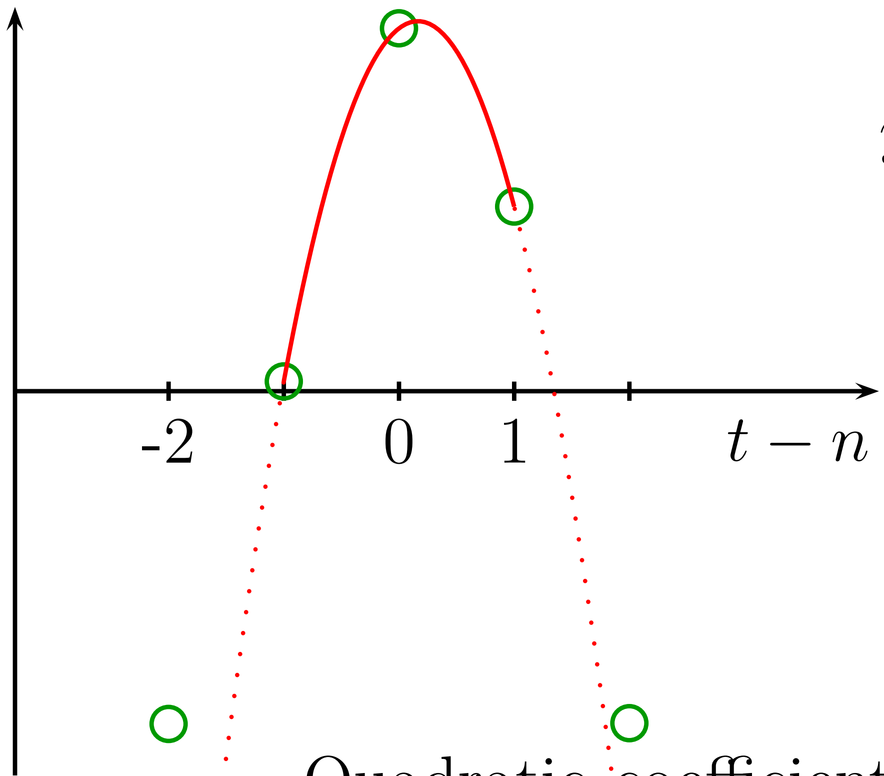
Quadratic: Equations



$$
\begin{aligned}
y(t) &= at^2 + bt + c \\
y(0) &= x[0] \Rightarrow c = x[0] \\
y(1) &= a + b + c = x[1] \\
y(-1) &= a - b + c = x[-1]
\end{aligned}
$$

$$
\begin{aligned}
y(1) - y(-1) &= x[1] - x[-1] = 2b \\
&\Rightarrow \quad b = \frac{1}{2}\left(x[1] - x[-1]\right) \\
y(1) + y(-1) &= x[1] + x[-1] = 2a + 2x[0] \\
&\Rightarrow \quad a = -x[0] + \frac{1}{2}\left(x[1] + x[-1]\right)
\end{aligned}
$$

Quadratic: Solution



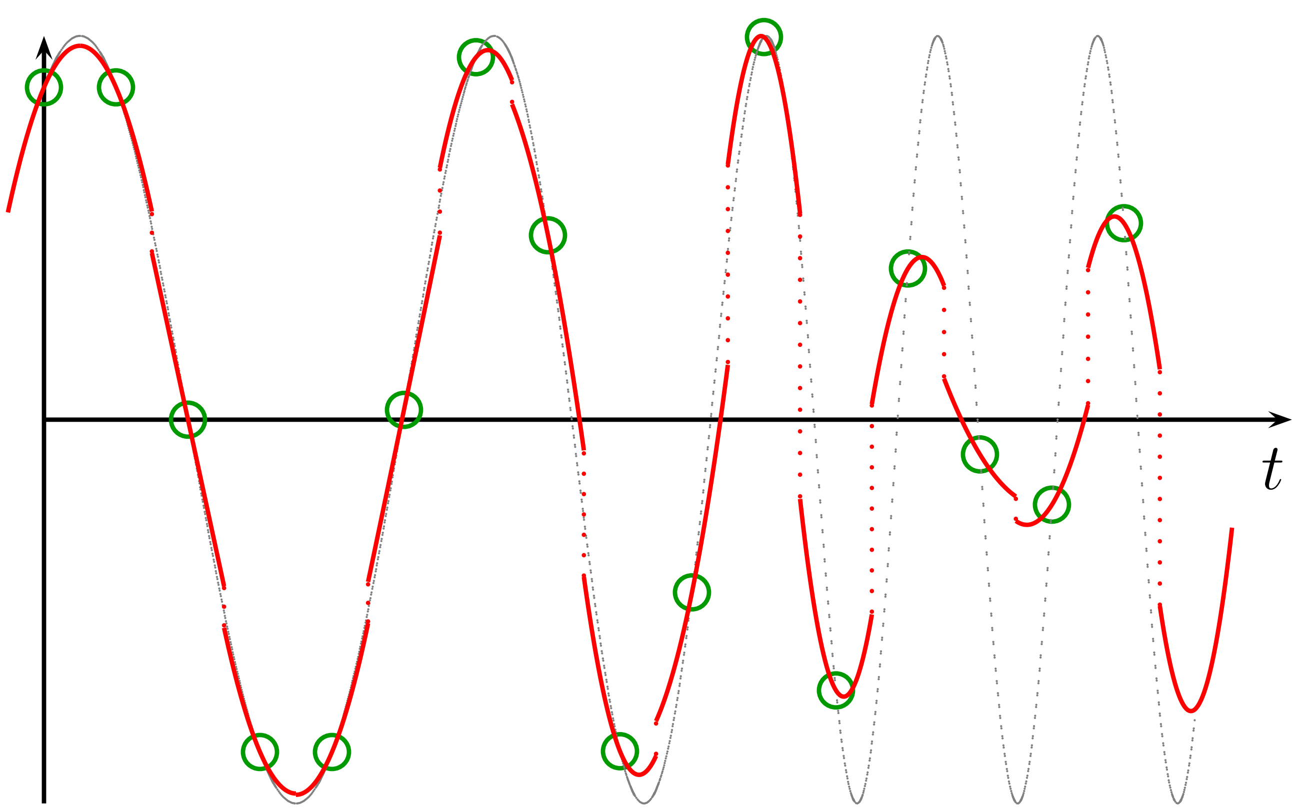
$$
\begin{aligned}
y(t) &= at^2 + bt + c \\
a &= -x[0] + \frac{1}{2}(x[1] + x[-1]) \\
b &= \frac{1}{2}(x[1] - x[-1]) \\
c &= x[n]
\end{aligned}
$$

Quadratic coefficients are given by linear combinations of input parameters. *Filter?*

$$
\begin{aligned}
a[n] &= x[n] \otimes \{\ \tfrac{1}{2} \quad -1 \quad \tfrac{1}{2}\ \} \\
b[n] &= x[n] \otimes \{\ \tfrac{1}{2} \quad 0 \quad -\tfrac{1}{2}\ \} \\
c[n] &= x[n] \otimes \{\ 0 \quad 1 \quad 0\ \}
\end{aligned}
$$

Quadratic

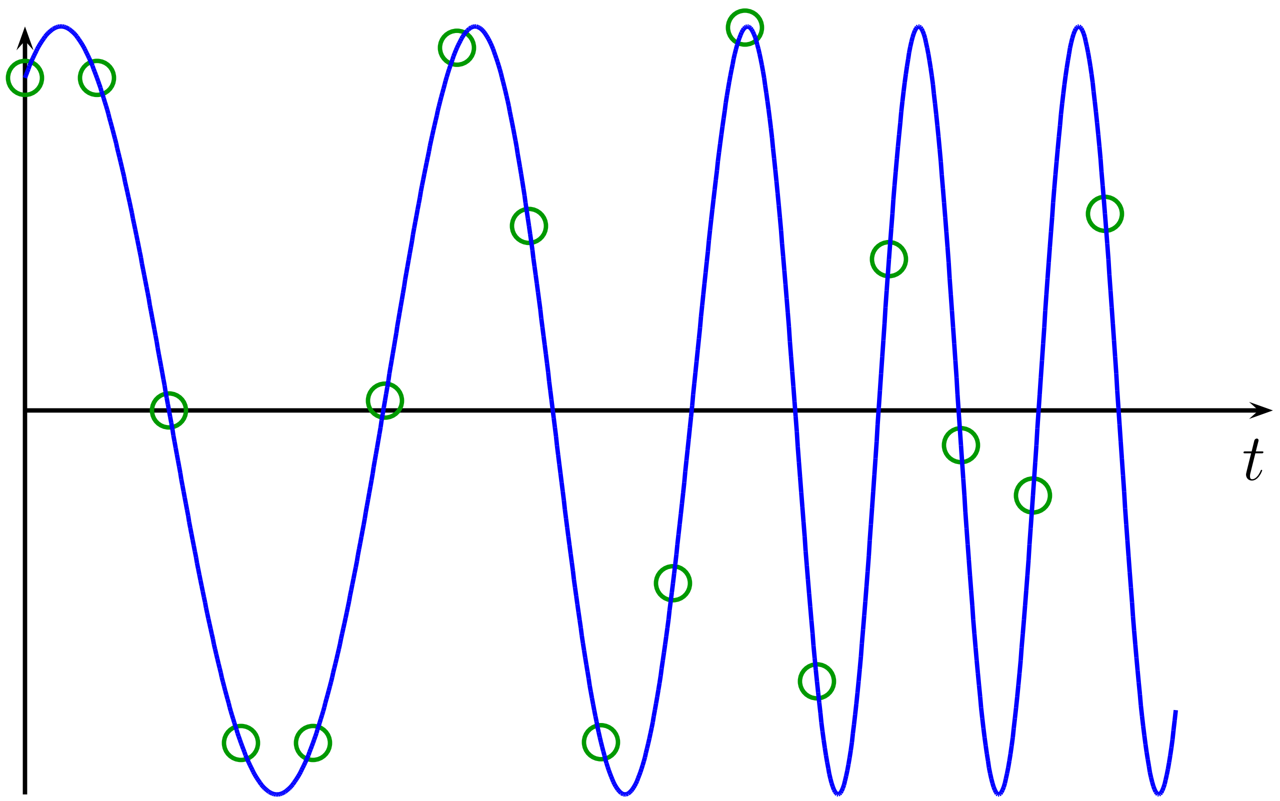
Wrong Answer: This is closer, but my signal was continuous.



$$y(t) = a[n](t-n)^2 + b[n](t-n) + x[n]$$

Desired

Now what? This is what I want. How do I get back there?



Rational Resampling

Rational Resampling

Traditionally, resampling is done by upsampling by some integer amount, L, and then downsampling by some other integer amount, M. This creates a rational, $\frac{L}{M}$, resampler. In this section we'll discuss:

- How to do it
- A frequency perspective and aliasing
- Implementation
- The resulting waveform

Integer Upsampling

- Step one: Insert $L-1$ zeros between every sample

$$x_u[n] = \begin{cases} x\left[\frac{n}{L}\right] & n \text{ divisible by } L \\ 0 & \text{Otherwise} \end{cases}$$

- Step two: Filter the result

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x_u[k]\, h[n-k] = \sum_{k=-\infty}^{\infty} x_u[kL]\, h[n-kL] \\ &= \sum_{k=-\infty}^{\infty} x[k]\, h[n-kL] \end{aligned}$$

Result: Fundamentally, you still have a sampled signal.

Upsampling Results

What did we just do? In frequency,

$$
\begin{aligned}
Y\left(e^{j2\pi f}\right) &= \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x[k]\, h[n-kL]\right) e^{-j2\pi f n} \\
&= \sum_{k=-\infty}^{\infty} x[k]\left(\sum_{n=-\infty}^{\infty} h[n-kL]\, e^{-j2\pi f(n-kL)}\right) e^{-j2\pi f kL} \\
&= \left(\sum_{k=-\infty}^{\infty} x[k]\, e^{-j2\pi f kL}\right) H\left(e^{j2\pi f}\right) \\
&= X\left(e^{j2\pi f L}\right) H\left(e^{j2\pi f}\right),
\end{aligned}
$$

we just expanded things.

Interpolation

Definition: A discrete filter, $h[n]$, is called an *interpolating filter* if the resulting $y[nL]$ passes through a scaled version of the original points, $\frac{1}{L}x[n]$.

These filters are also called Nyquist filters or M–band filters.

Interpolation

Theorem: If $h[n]$ is a discrete interpolating filter, then $h[n]$ must be constrained by,

$$h[nL] = \begin{cases} \frac{1}{L} & n = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Proof: Working from the definition of an interpolating filter, we have,

$$\begin{aligned} y[nL] &= \frac{1}{L}x[n], \text{ and by the definition of } y[n] \\ &= \sum_{k=-\infty}^{\infty} x[k]\, h[nL - kL] = \sum_{k=-\infty}^{\infty} x[k]\, h[(n-k)L] \end{aligned}$$

This can only be true if $h[(n-k)L]$ is zero for all $n \neq k$, and exactly $\frac{1}{L}$ for $n = k$. $\mathcal{Q}.\mathcal{E}.\mathcal{D}$

Summed Frequency

Theorem: If $h[n]$ is a discrete interpolation filter, supporting an upsampling by L operation, then $\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) = 1.$

This property is important to filter design, so we present it here. The proof is rather short.

Proof: Start with the summation, and apply the definition of $H\left(e^{j2\pi f}\right)$,

$$\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) = \sum_{k=0}^{L-1} \sum_{n=-\infty}^{\infty} h[n]\, e^{-j2\pi\left(f+\frac{k}{L}\right)n}.$$

Now, separate out the summation by k,

$$= \sum_{n} h[n]\, e^{-j2\pi f n} \left[\sum_{k=0}^{L-1} e^{-j2\pi\frac{k}{L}n}\right]$$

Summed Frequency

Theorem: If $h[n]$ is a discrete interpolation filter, supporting an upsampling by L operation, then $\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) = 1$.

Proof: (Continued ...) Evaluate the summation over k,

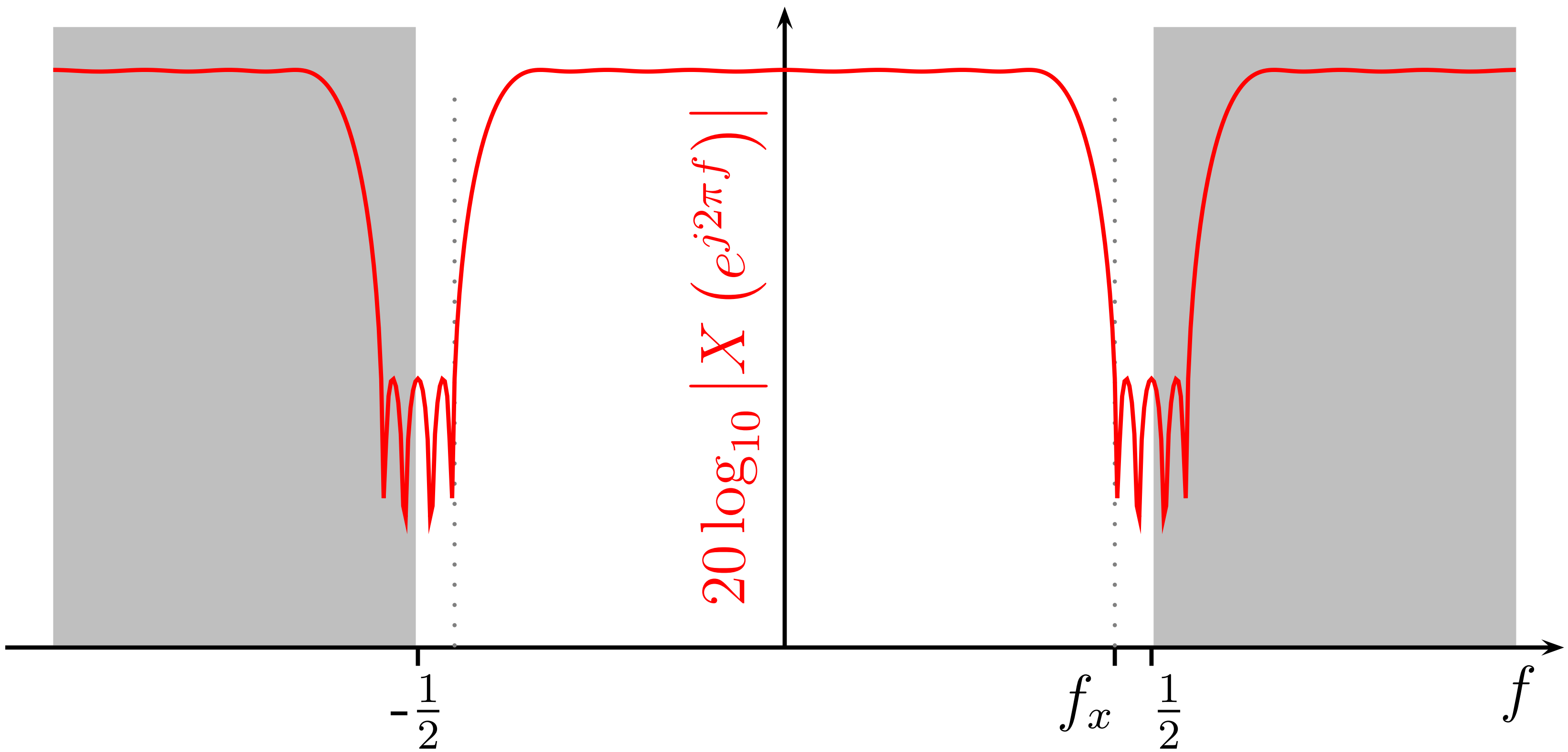
$$
\begin{aligned}
&\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \\
&= \sum_{n} h[n]\, e^{-j2\pi f n} \times \begin{cases} L & n \text{ is divisible by } L \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}
$$

and apply the fact that h is an interpolator,

$$
= L \sum_{n} h[nL]\, e^{-j2\pi f n L} = L h[0] = 1
$$

$\mathcal{Q}.\mathcal{E}.\mathcal{D}$

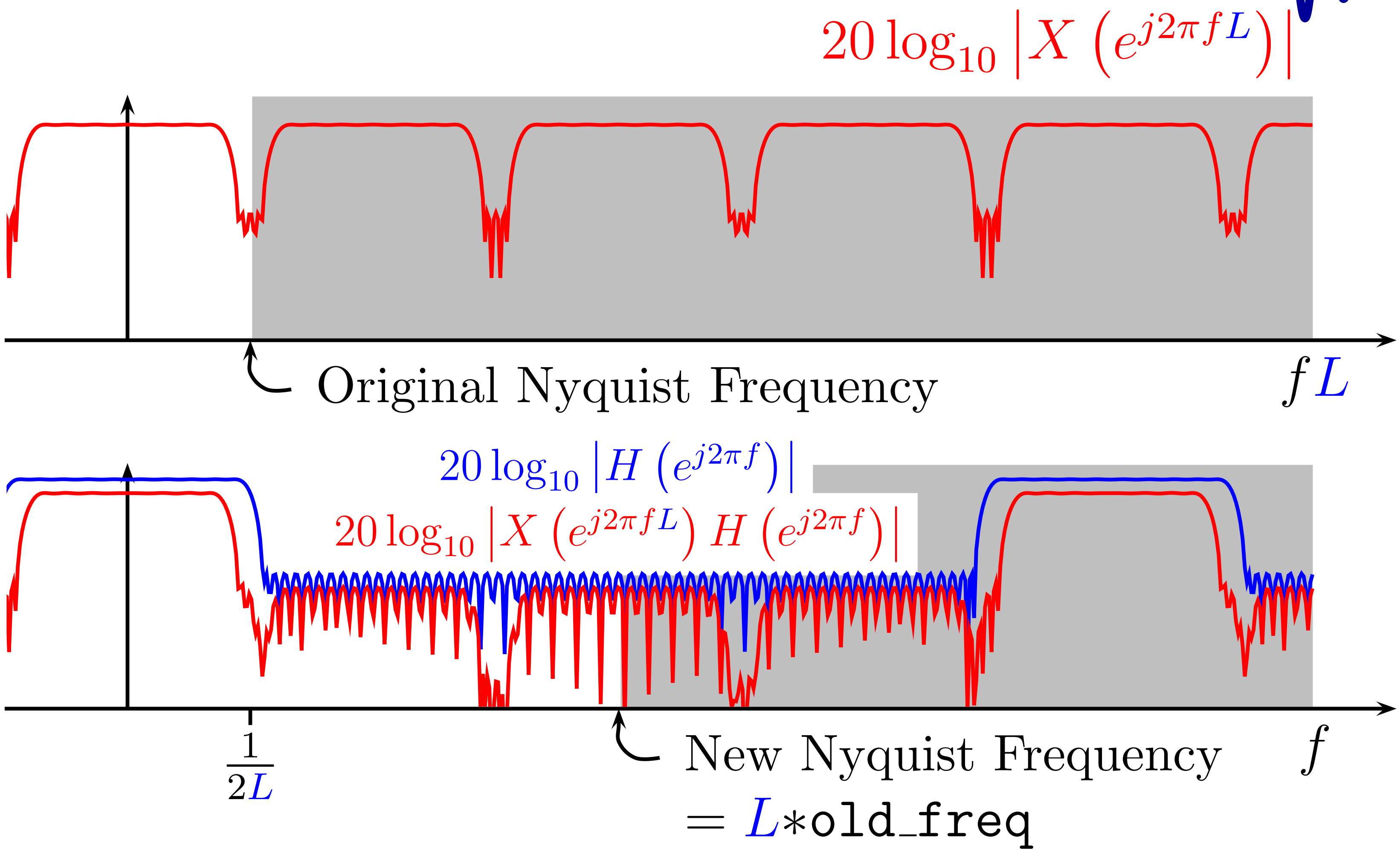
Example Signal



Two frequencies of interest:

1. The Nyquist frequency, $f = \frac{1}{2}$, beyond which the spectrum repeats, and
2. The highest frequency containing valid data, f_x.

Aliasing



Result: Nyquist cutoff moves over.

Limitation: Can only upsample by integer amounts.

Filter Design

A good interpolating filter must:

1. Pass the signal unaltered whever $|fL| < f_x$,

$$X\left(e^{j2\pi fL}\right) H\left(e^{j2\pi f}\right) = X\left(e^{j2\pi fL}\right)$$

2. Interpolate between sample points. (i.e. be an interpolator.)

The first requirement is equivalent to saying that

$$H\left(e^{j2\pi f}\right) = 1 \text{ whenever } |fL| < f_x,$$

or equivalently whenever $|f| < f_p = \frac{1}{L} f_x$. This defines the passband of $H\left(e^{j2\pi f}\right)$.

Filter Design

If $h[n]$ is an interpolating filter, then

$$\begin{aligned} 1 &= \overbrace{H\left(e^{j2\pi f}\right)}^{\text{This is 1 whenever } |f| < f_p} + \sum_{k=1}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \\ 0 &= \sum_{k=1}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right). \\ &\quad \text{This will be true if} \\ 0 &= H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \forall k \in [1, L-1] \end{aligned}$$

This equation describes a region, for $|f| < f_p$, where H must be zero. Adding $\frac{k}{L}$ to this inequality gives,

$$\frac{k}{L} - f_p < \quad \frac{k}{L} + f \quad < \frac{k}{L} + f_p.$$

Filter Design

From the last slide, if h is an interpolating filter then $H\left(e^{j2\pi f}\right)$ must be zero any time

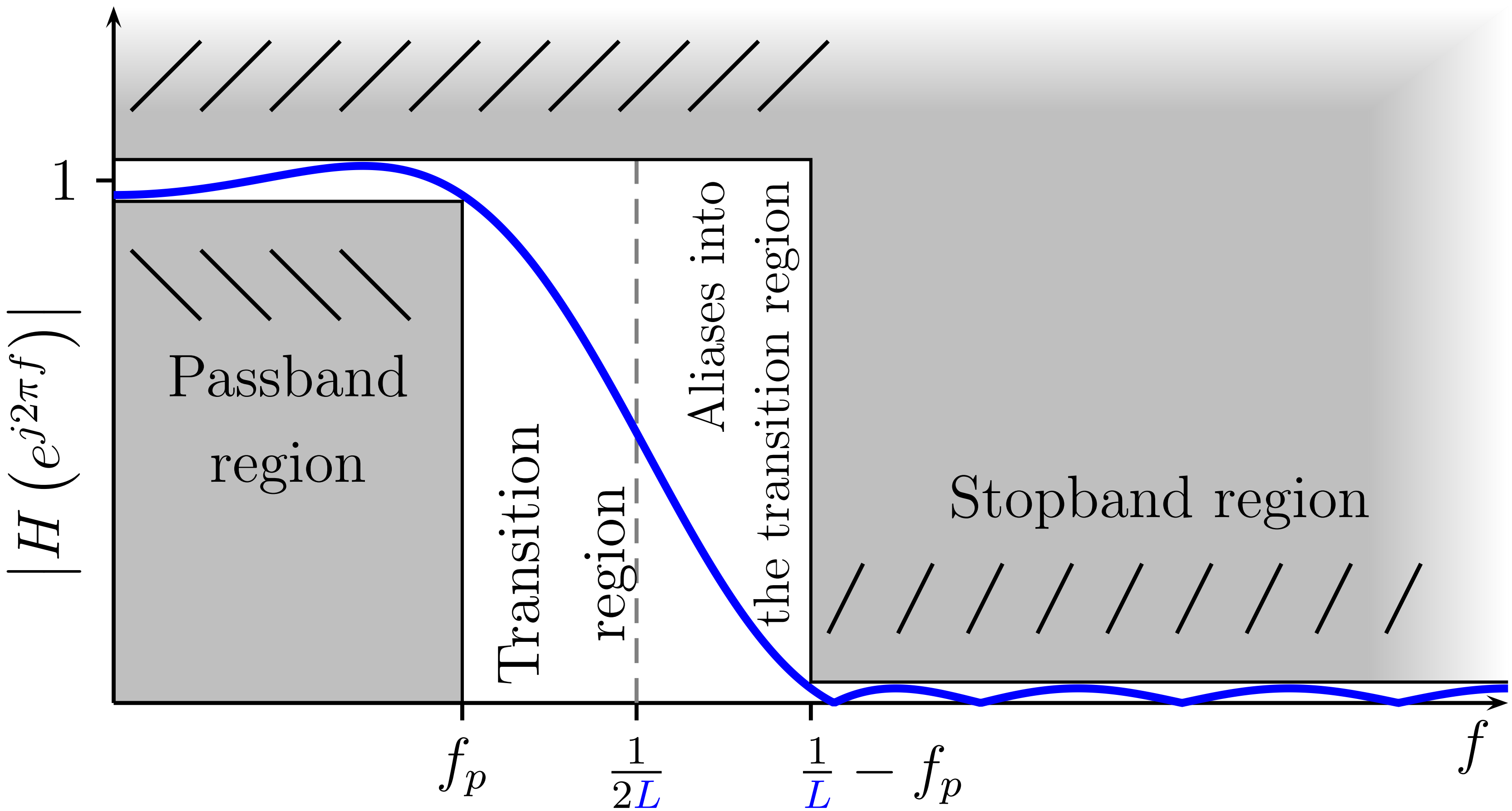
$$\frac{k}{L} - f_p < \quad f \quad < \frac{k}{L} + f_p, \; \forall k \in [1, L-1]\,.$$

This describes a set of regions, $2f_p$ in width, surrounding frequencies $\frac{k}{L}$ in frequency. The least of these regions starts at $\frac{1}{L} - f_p$. This gives us our criteria for designing h:

$$H\left(e^{j2\pi f}\right) \quad = \begin{cases} 1 \quad \text{for} \quad |f| \quad < f_p \quad = \quad \dfrac{1}{L} f_x, \\ 0 \quad \text{for} \quad |f| \quad > f_s \quad = \quad \dfrac{1}{L} - f_p \\ \qquad \text{and don't care otherwise.} \end{cases}$$

Perhaps a picture would help ...

Filter Design



Although the transition band could be made sharper for the same f_p, the resulting filter would no longer be an interpolator.

Design Methods

I want to design an interpolating filter, with a passband of f_p and a stopband cutoff of δ or better. How shall I do it?

Four basic methods of designing Nyquist filters:

1. Ad–Hoc

2. Eigenfilter

3. Remez–Exchange Algorithms

4. Cosine–Series Approximations

Ad–Hoc Method

To design a Nyquist lowpass filter:

1. Design a lowpass filter, with a passband from 0 to f_p, and a stopband from $\frac{1}{L} - f_p$ through $\frac{1}{2}$.

   The Parks–McClellan filter design algorithm works well here.

2. After designing the filter, adjust the taps so that $h[nL]$ is zero for $n \neq 0$.

3. All done!

This method claims no optimal property, yet it is often used for its simplicity.

Eigenfilter Design

1. First, we'll insist that our filter be symmetric, $h[n] = h[-n]$.

2. The Fourier transform of $h[n]$ is given by,

$$H\left(e^{j2\pi f}\right) = h[0] + 2\sum_{k=1}^{\infty} h[n]\cos(2\pi f n).$$

We'll write this as a linear combination of the terms in $h[n]$,

$$H\left(e^{j2\pi f}\right) = \Phi(f)^{T}\boldsymbol{h}$$

Eigenfilter Design

3. Define a stopband error,

$$\begin{aligned}\xi_S^2 &\triangleq \int_{\frac{1}{L}-f_p}^{\frac{1}{2}} \left|H\left(e^{j2\pi f}\right)\right|^2 df \\ &= \boldsymbol{h}^T \left[\int \boldsymbol{\Phi}(f)\,\boldsymbol{\Phi}(f)^T df\right] \boldsymbol{h}\end{aligned}$$

4. and a passband error,

$$\begin{aligned}\xi_P^2 &\triangleq \int_{0}^{f_p} \left|H\left(e^{j0}\right) - H\left(e^{j2\pi f}\right)\right|^2 df \\ &= \boldsymbol{h}^T \left[\int \left[\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}(f)\right] \left[\boldsymbol{\Phi}(0) - \boldsymbol{\Phi}(f)^T\right] df\right] \boldsymbol{h}\end{aligned}$$

Although these integrals look hideous, they are not.

Eigenfilter Design

5. Define a total error,

$$\xi^2 \;=\; \xi_S^2 + \xi_P^2 = \boldsymbol{h}^T \boldsymbol{\Xi} \boldsymbol{h}$$

6. Calculate $\mathbf{A} = \boldsymbol{\Xi}^{-1}$,

7. Find the largest eigenvector of $\mathbf{A}$. This vector corresponds to the filter of interest.

   This eigenvector is easily found. Let $\boldsymbol{x}_0$ be a random vector in the domain of $\mathbf{A}$. Then, repeat $\boldsymbol{x}_{n+1} = \mathbf{A}\boldsymbol{x}_n$ until it converges. $\boldsymbol{x}_\infty$ will then correspond to largest eigenvector of interest.

8. Scale as necessary.

Homework

Given $L = 10$, and $f_p = 0.04$, answer the questions below:

1. What is $\mathbf{\Phi}(f)$?

2. What is $\mathbf{\Xi}$?

Answers

Given $L = 10$, and $f_p = 0.04$, answer the questions below:

1. What is $\mathbf{\Phi}(f)$?

$$\mathbf{\Phi}(f) \quad = \quad \begin{bmatrix} 1 \\ 2\cos(2\pi f) \\ 2\cos(2\pi 2 f) \\ \vdots \\ 2\cos(2\pi f(m-1)) \\ \vdots \end{bmatrix}$$

Answers

2. What is $\boldsymbol{\Xi}$? For $n, m > 1$, $\boldsymbol{\Xi}_s$ is given by,

$$
\begin{aligned}
\left(\boldsymbol{\Xi}_s\right)_{nm} &= \int_{f_s}^{\frac{1}{2}} \phi_m(f)\, \phi_n(f)\, df \\
&= 4 \int_{f_s}^{\frac{1}{2}} \cos\left(2\pi f\left(m-1\right)\right) \cos\left(2\pi f\left(n-1\right)\right) df \\
&= 2 \int_{f_s}^{\frac{1}{2}} \cos\left(2\pi f\left(m+n-2\right)\right) + \cos\left(2\pi f\left(m-n\right)\right) df \\
&= \left.\frac{\sin\left(2\pi f\left(m+n-2\right)\right)}{\pi\left(m+n-2\right)}\right|_{f_s}^{\frac{1}{2}} + \left.\frac{\sin\left(2\pi f\left(m-n\right)\right)}{\pi\left(m+n-2\right)}\right|_{f_s}^{\frac{1}{2}} \\
&= -\frac{\sin\left(2\pi f_s\left(m+n-2\right)\right)}{\pi\left(m+n-2\right)} - \frac{\sin\left(2\pi f_s\left(m-n\right)\right)}{\pi\left(m+n-2\right)}
\end{aligned}
$$

Answers

2. What is $\boldsymbol{\Xi}$? For $n = m > 1$, $\boldsymbol{\Xi}_s$ is given by,

$$
\begin{aligned}
(\boldsymbol{\Xi}_s)_{nn} &= \int_{f_s}^{\frac{1}{2}} \phi_n(f)\,\phi_n(f)\,df \\
&= 4\int_{f_s}^{\frac{1}{2}} \cos(2\pi f(n-1))\cos(2\pi f(n-1))\,df \\
&= 2\int_{f_s}^{\frac{1}{2}} 1+\cos(2\pi f(2n-2))\,df \\
&= 1-2f_s-\frac{\sin(2\pi f_s(2n-2))}{\pi(2n-2)}
\end{aligned}
$$

Answers

2. What is $\boldsymbol{\Xi}$? For $n > 1$, $\boldsymbol{\Xi}_s$ is given by,

$$\begin{aligned}
(\boldsymbol{\Xi}_s)_{1n} = (\boldsymbol{\Xi}_s)_{n1} &= \int_{f_s}^{\frac{1}{2}} \phi_1(f)\,\phi_n(f)\,df \\
&= 2\int_{f_s}^{\frac{1}{2}} \cos(2\pi f(n-1))\,df \\
&= -\frac{\sin(2\pi f_s(n-1))}{2\pi(n-1)} \\
(\boldsymbol{\Xi}_s)_{11} &= \int_{f_s}^{\frac{1}{2}} df = \frac{1}{2} - f_s
\end{aligned}$$

The rest of the terms in $\boldsymbol{\Xi} = \boldsymbol{\Xi}_s + \boldsymbol{\Xi}_p$ are left as (tedious) exercises for the student.

Remez Exchange

First, note that if $h[n]$ is an interpolating filter, then $\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) = 1$. That means that the passband is entirely determined by the stopband:

$$H\left(e^{j2\pi f}\right) = 1 - \sum_{k=1}^{L-1} H\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right).$$

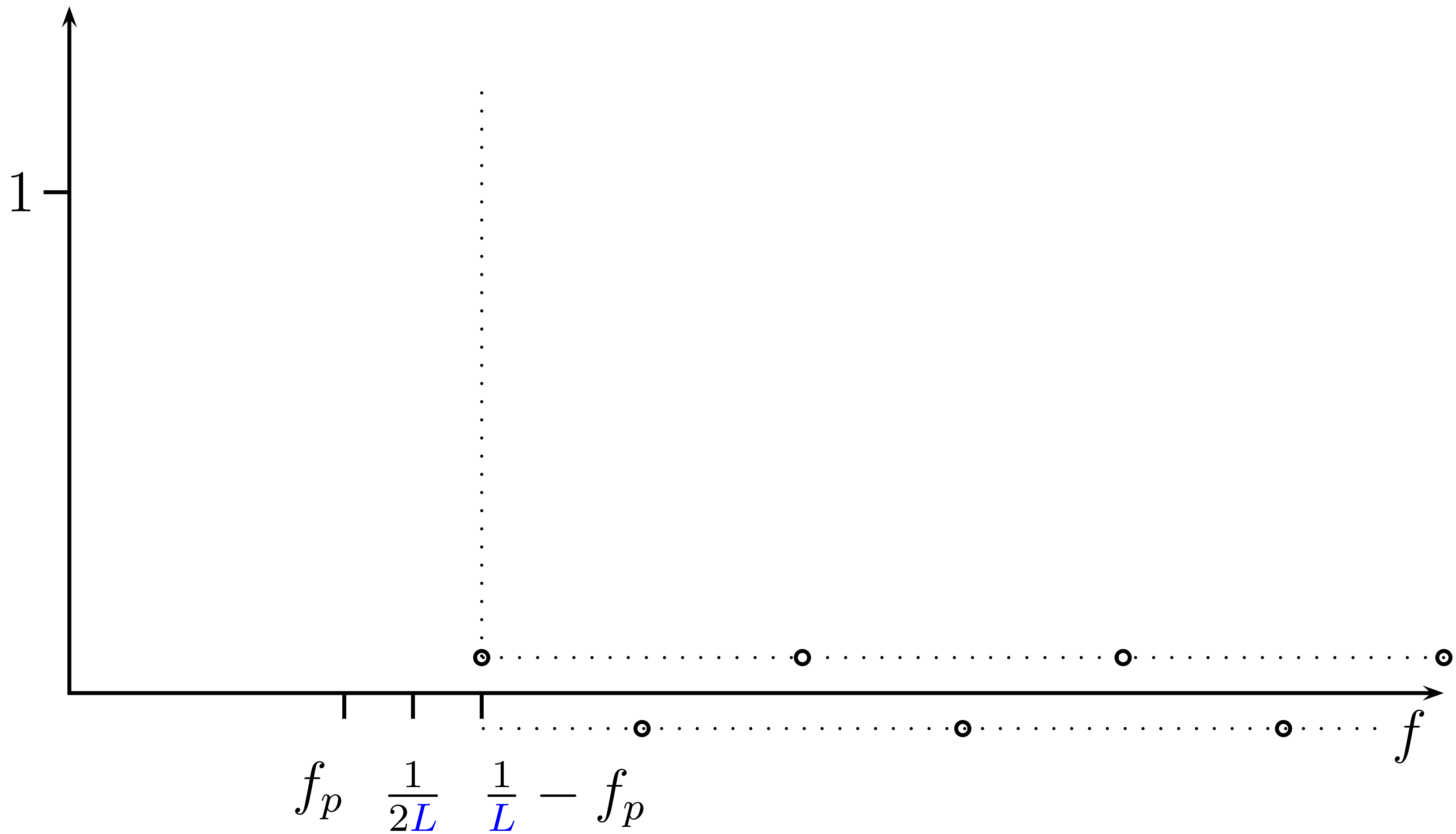
Further, if each term in the stopband summation is absolutely less than δ, the passband will be within $\pm(L-1)\delta$ of one.

We'll use the Remez–Exchange algorithm to achieve the equiripple passband response we are interested in.

Remez Exchange

Steps:

1. Pick $\frac{N}{2}+1$ evenly spaced frequencies over the range from $f = f_s = \left(\frac{1}{L} - f_p\right)$ to $f = \frac{1}{2}$.



Remez Exchange

Steps: (Continued ...)

2. Form a linear system from the interpolation equations,

$$h[nL] = \begin{cases} \frac{1}{L} & n = 0 \\ 0 & n \neq 0 \end{cases}$$

together with the stopband frequency equations,

$$H\left(e^{j2\pi f_k}\right) + (-1)^k \delta = \mathbf{\Phi}^T(f_k)\,\boldsymbol{h} + (-1)^k \delta = 0.$$

Remez Exchange

Steps: (Continued ...)

2. (Continued ...) This linear system will look something like,

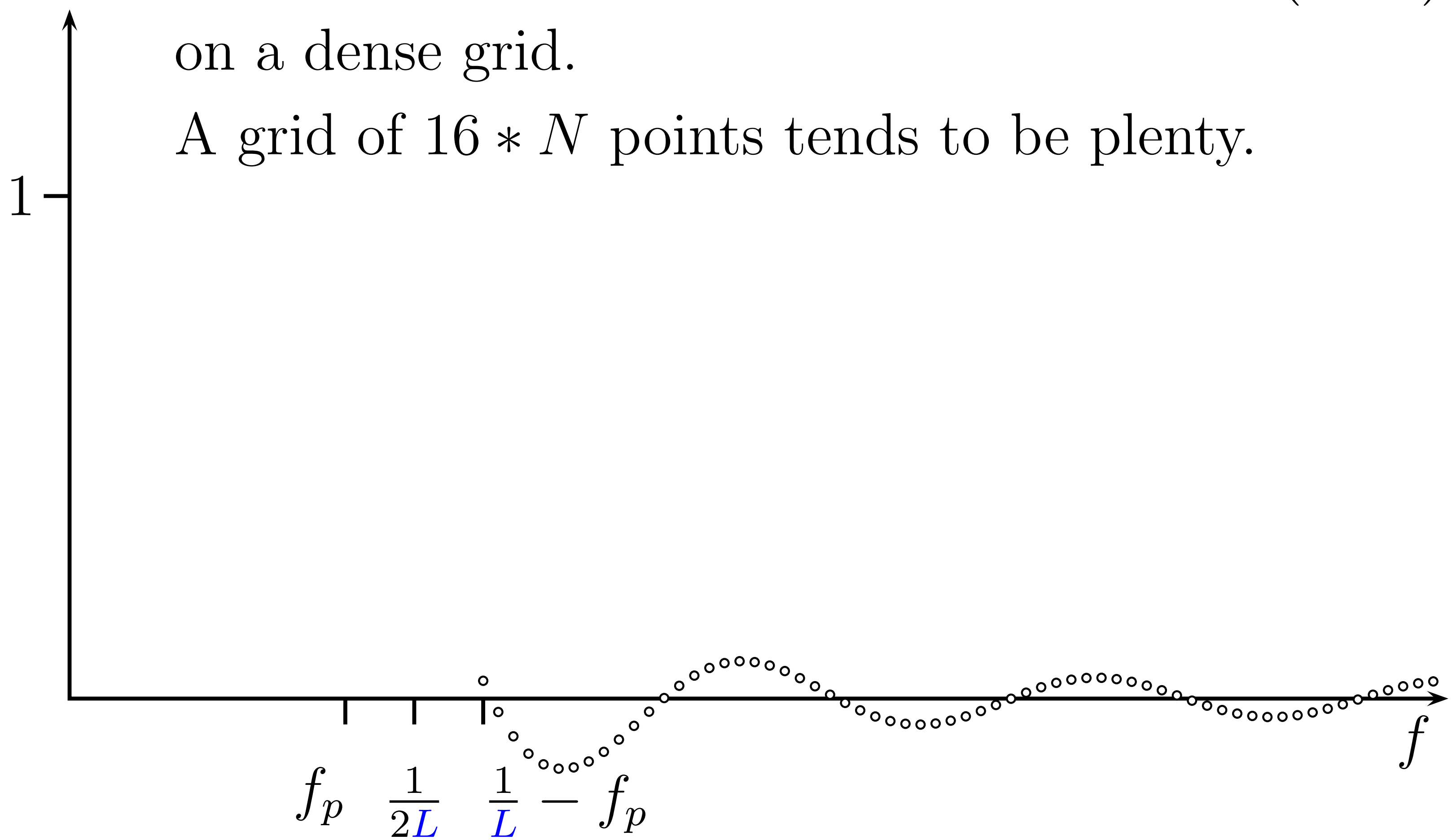
$$\begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & & & & \cdots & 0 \\ & & \boldsymbol{\Phi}(f_0)^T & & & 1 \\ & & \boldsymbol{\Phi}(f_1)^T & & & -1 \\ & & \boldsymbol{\Phi}(f_2)^T & & & 1 \\ & & \boldsymbol{\Phi}(f_3)^T & & & -1 \\ & & \vdots & & & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{h} \\ \delta \end{bmatrix} = \begin{bmatrix} \frac{1}{L} \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

Remez Exchange

Steps: (Continued . . .)

3. Solve this linear system for $\boldsymbol{h}$ and δ.
4. Evaluate the filter's frequency response, $H\left(e^{j2\pi f}\right)$ on a dense grid.

   A grid of $16 * N$ points tends to be plenty.

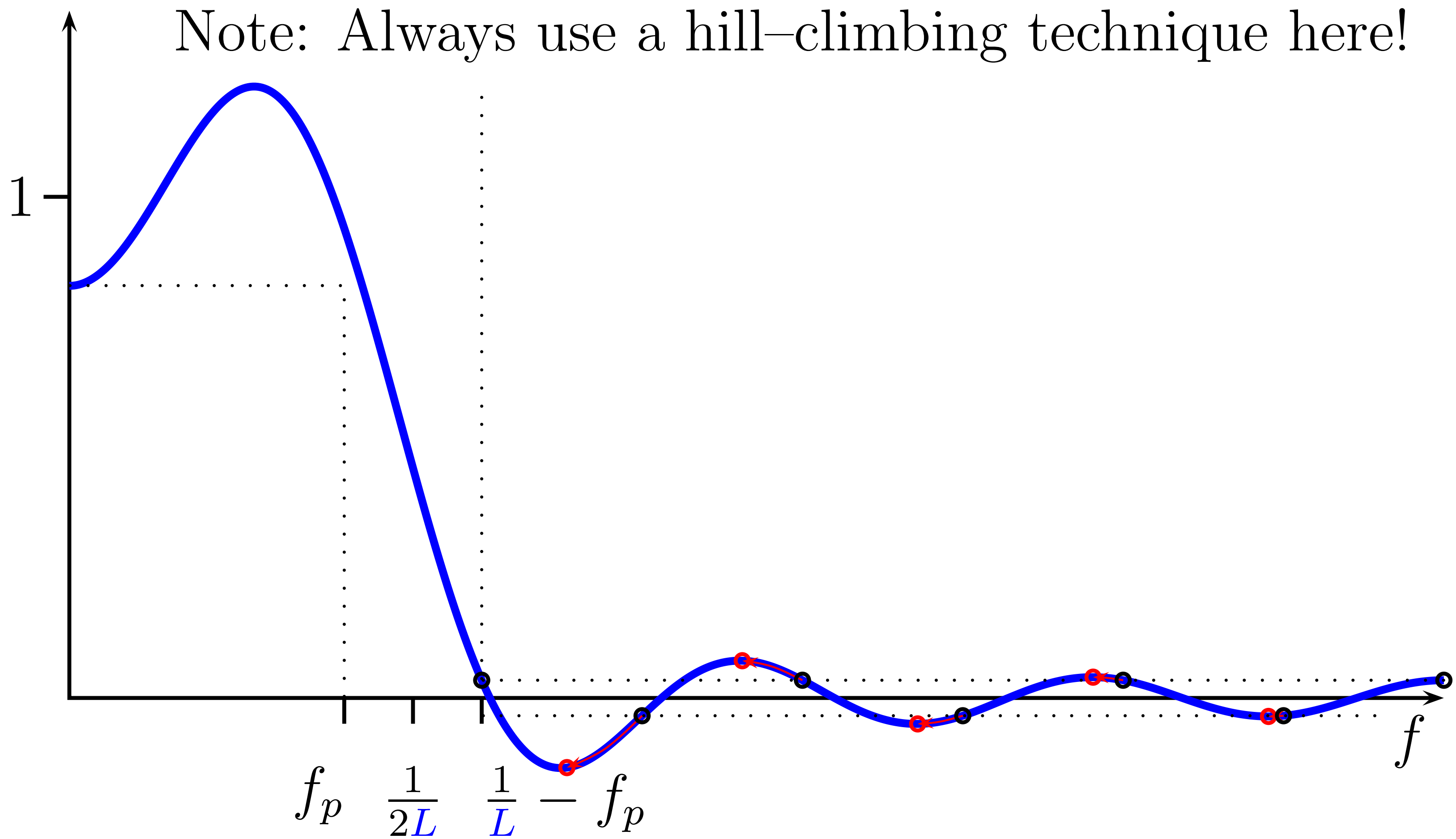


Remez Exchange

Steps: (Continued ...)

5. Pick new frequencies, f_k, where the previous filter had a maxima.

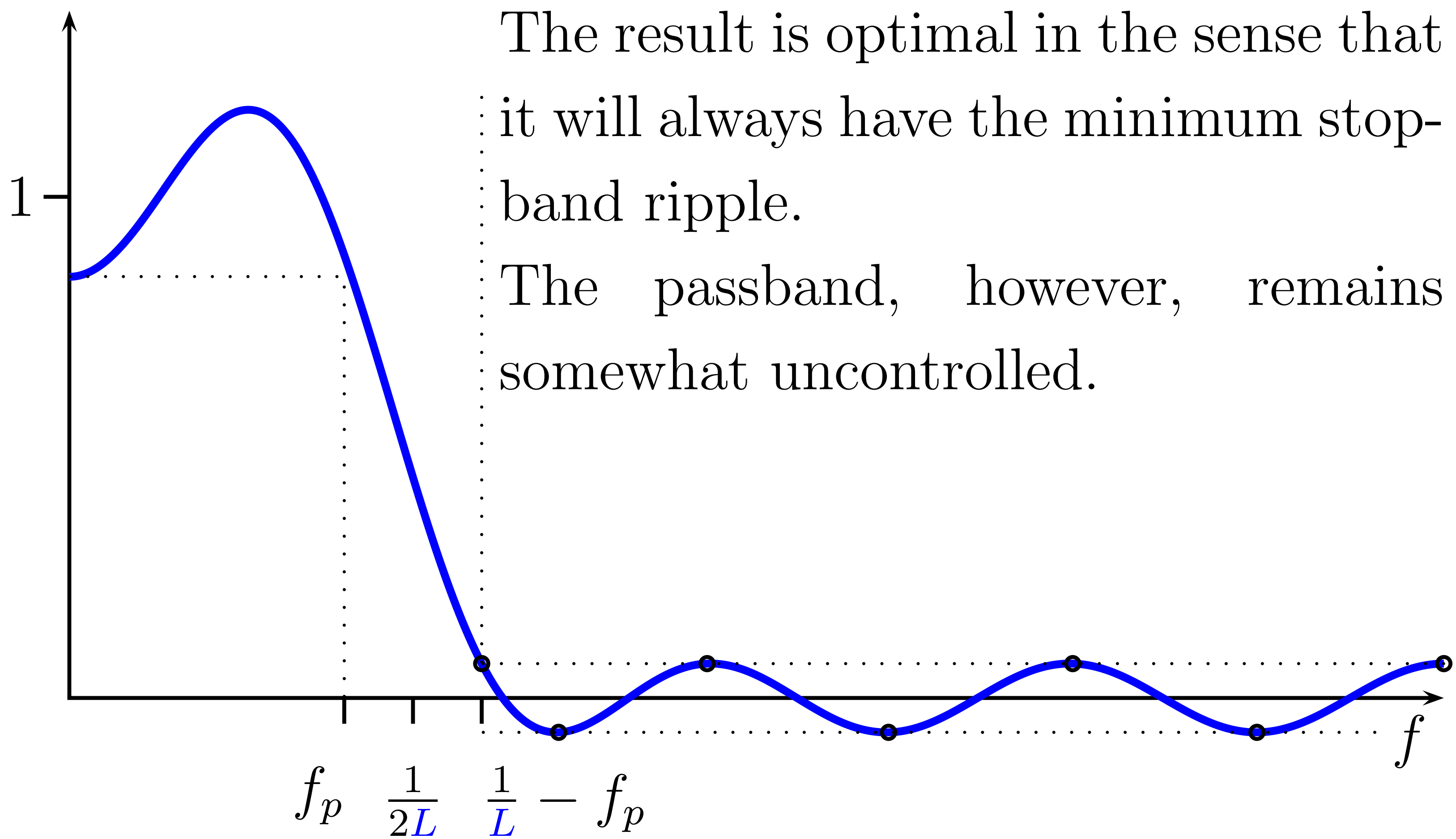
   Note: Always use a hill–climbing technique here!



Remez Exchange

Steps: (Continued . . .)

7. Repeat until the frequencies stop moving.



The result is optimal in the sense that it will always have the minimum stop-band ripple.

The passband, however, remains somewhat uncontrolled.

Cosine–Series

Problem: Designing long filters via the Remez–Exchange algorithm as just presented, requires inverting large matrices. Such matrix inversions are subject to a compounding numerical error, often rendering solution difficult.

Solution: Describe the filter by a small number of coefficients, $\{a_k\}_{k=0}^{N_k-1}$ only. Solving for the optimal values of these smaller number of coefficients keeps the matrix size small.

Note: Cosine–Series filters are in no way *optimal*. However, they are often *good enough*.

Cosine–Series

Suppose we restrict our filter to be of the form,

$$
\begin{aligned}
h[n] &= \begin{cases} \dfrac{1}{NL} \displaystyle\sum_{k=0}^{N_k-1} a_k \cos\left(2\pi \dfrac{k}{NL} n\right) & |n| < \frac{NL}{2} \\ 0 & \text{otherwise.} \end{cases} \\
&= \boldsymbol{\phi}^T [n]\, \boldsymbol{a}
\end{aligned}
$$

for $N_k = \left\lfloor \left(\frac{1}{L} - f_p\right) NL \right\rfloor$. As a result, we have only N_k unknown values of a_k to solve. For these N_k unknowns, we have $\left\lceil \frac{N}{2} \right\rceil + 1$ equations to ensure $h[n]$ is an interpolator, and perhaps one to ensure it ends at zero.

The optimal vector of coefficients, $(\boldsymbol{a})_k = a_k$, may be found by using the Remez–Exchange algorithm as well.

Cosine–Series

1. Pick a set of frequencies, $f_m = f_s + \frac{0.5+m}{N}$, for $m \in \left[0, N_k - \left\lceil \frac{N}{2} \right\rceil\right] - 1$.
2. Form a linear system of constraints on $\boldsymbol{a}$.
   - One equation forces $h\left[n\right] = 0$ at $n = \frac{NL}{2}$.
   - $\left\lfloor \frac{N}{2} \right\rfloor + 1$ equations force $h\left[n\right]$ to be an interpolator.
   - That leaves $N_k - \left\lceil \frac{N}{2} \right\rceil$ equations to solve for $\boldsymbol{a}$ and δ while ensuring that $H\left(e^{j2\pi f_m}\right) = \left(-1\right)^m \delta$.

Sound familiar?

Cosine–Series

2. (Continued . . .) This set of equations will look something like:

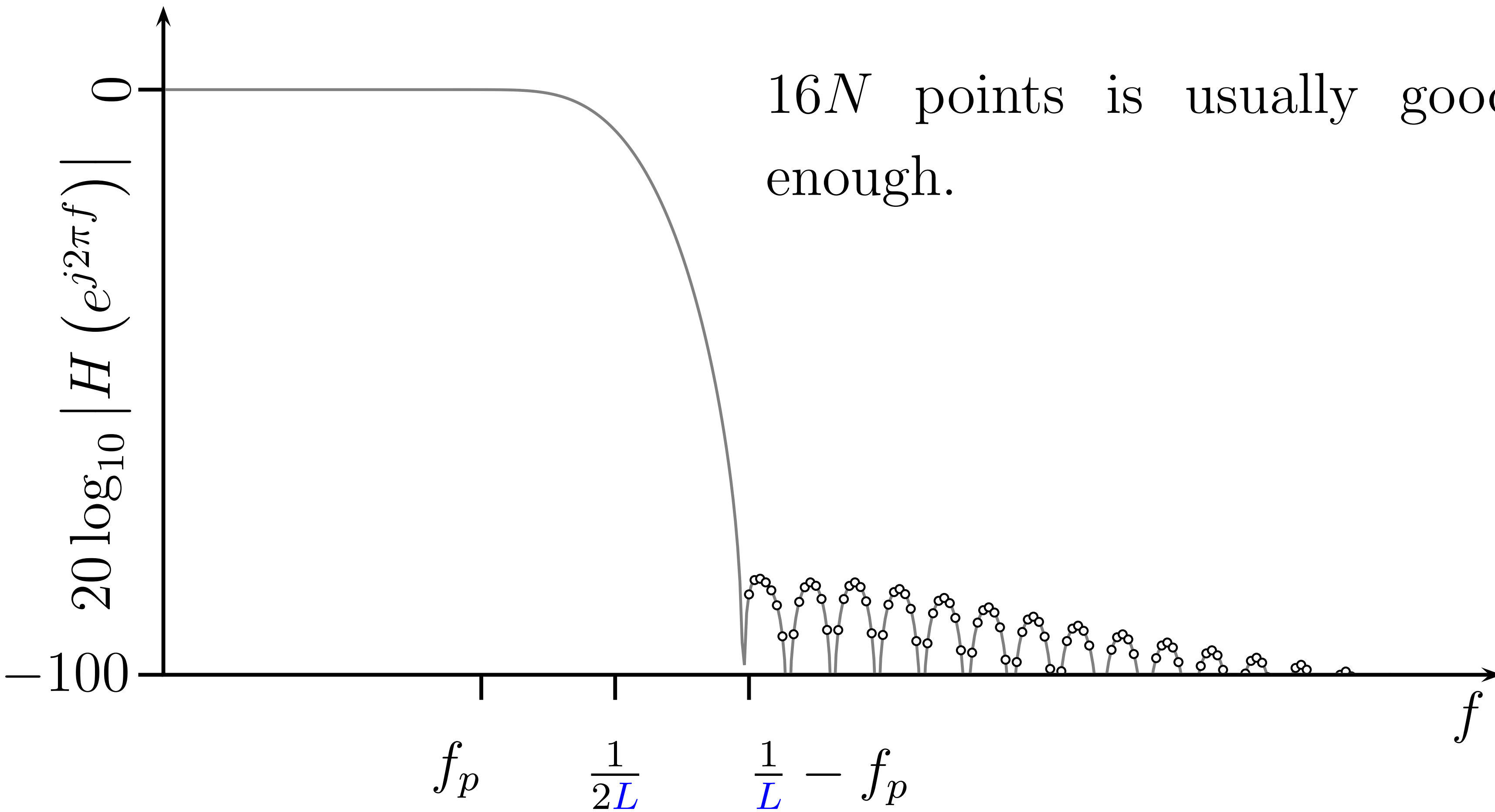
$$\begin{bmatrix} \boldsymbol{\phi}\left[0\right]^{\dagger} & 0 \\ \boldsymbol{\phi}\left[L\right]^{\dagger} & 0 \\ \vdots & \vdots \\ \boldsymbol{\phi}\left[\frac{N}{2}L\right]^{\dagger} & 0 \\ \boldsymbol{\Phi}\left(f_s + \frac{0.5}{N}\right)^{\dagger} & -1 \\ \boldsymbol{\Phi}\left(f_s + \frac{1.5}{N}\right)^{\dagger} & 1 \\ \boldsymbol{\Phi}\left(f_s + \frac{2.5}{N}\right)^{\dagger} & -1 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \delta \end{bmatrix} = \begin{bmatrix} \frac{1}{L} \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

3. Solve for $\boldsymbol{a}$ and δ.

Cosine–Series

Steps: (Continued . . .)

4. Evaluate the filter's frequency response on a dense grid.

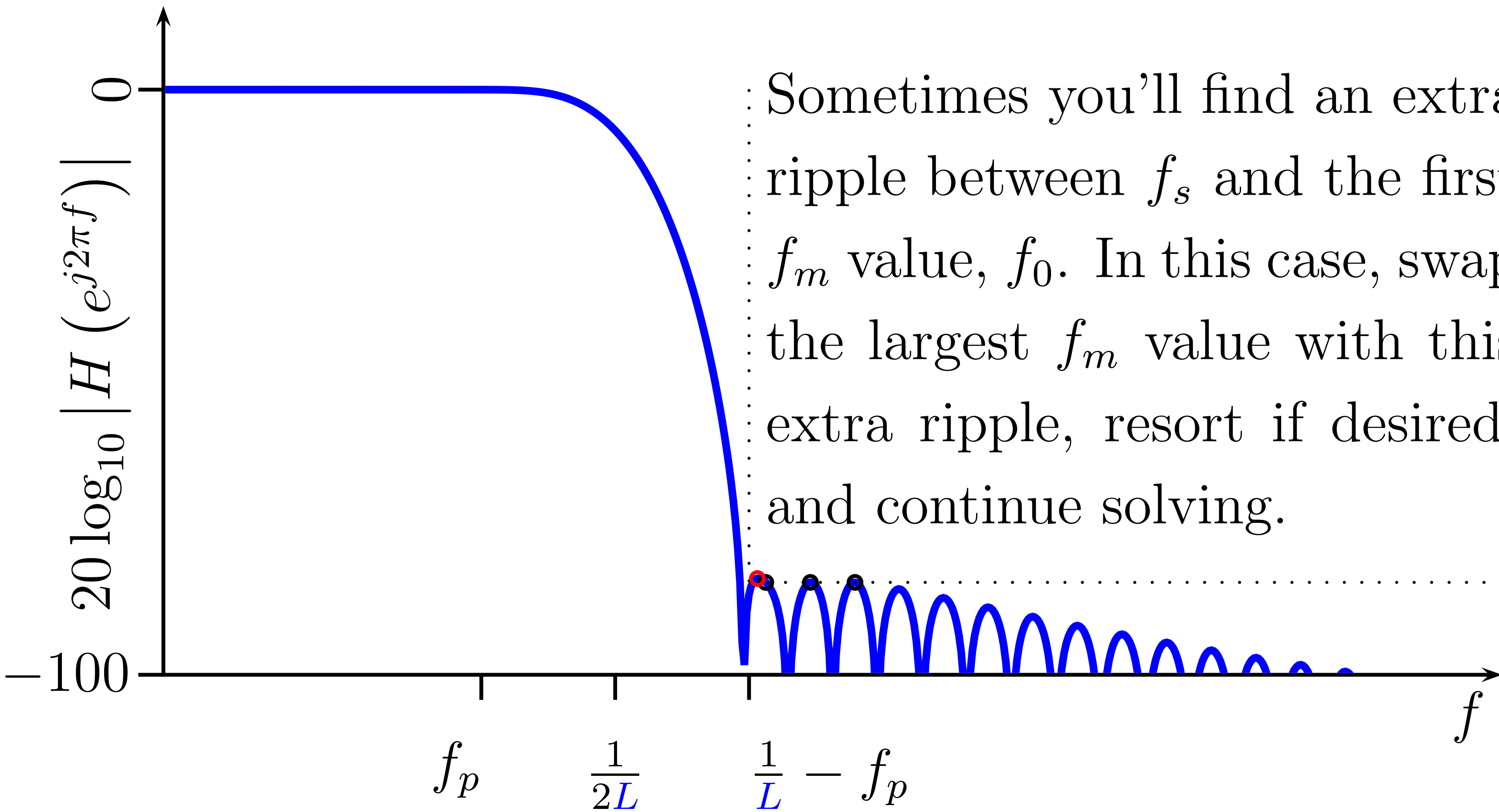


$16N$ points is usually good enough.

Cosine–Series

Steps: (Continued . . .)

5. Pick a new set of frequencies f_m, by hill climbing from the previous set.

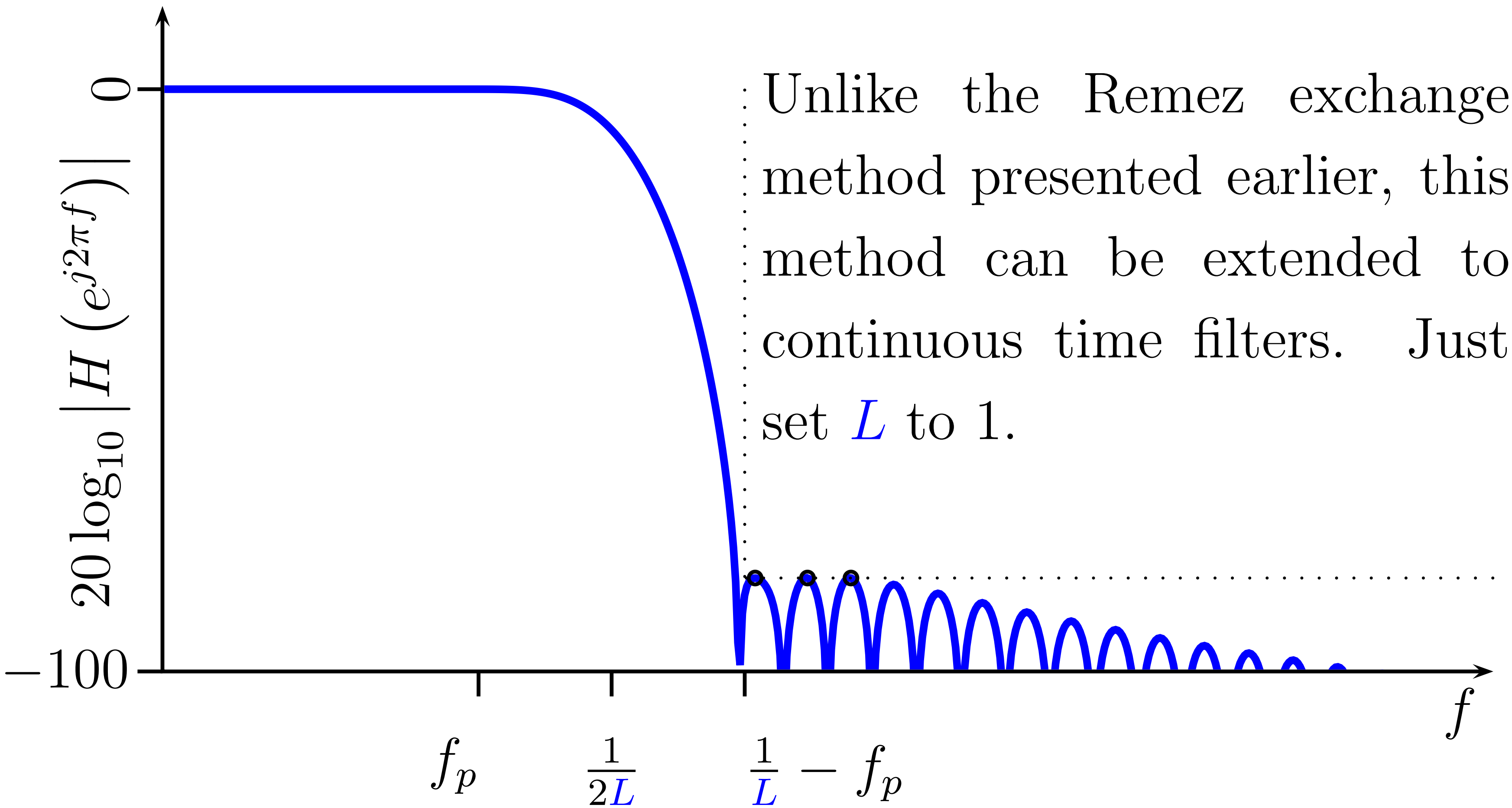


Sometimes you'll find an extra ripple between f_s and the first f_m value, f_0. In this case, swap the largest f_m value with this extra ripple, resort if desired, and continue solving.

Cosine–Series

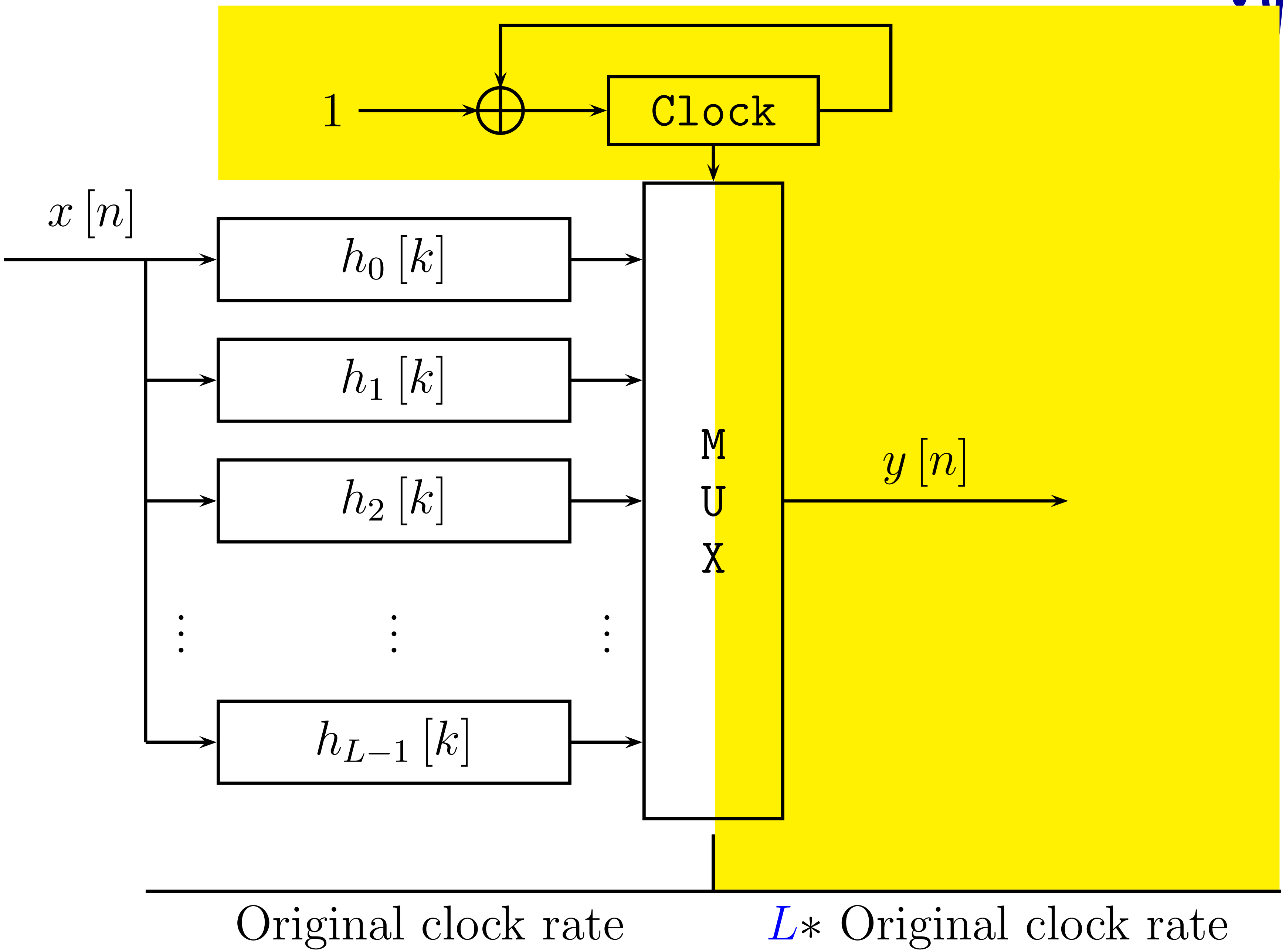
Steps: (Continued . . .)

6. Repeat until the frequency set stops changing.



Unlike the Remez exchange method presented earlier, this method can be extended to continuous time filters. Just set L to 1.

Implementation

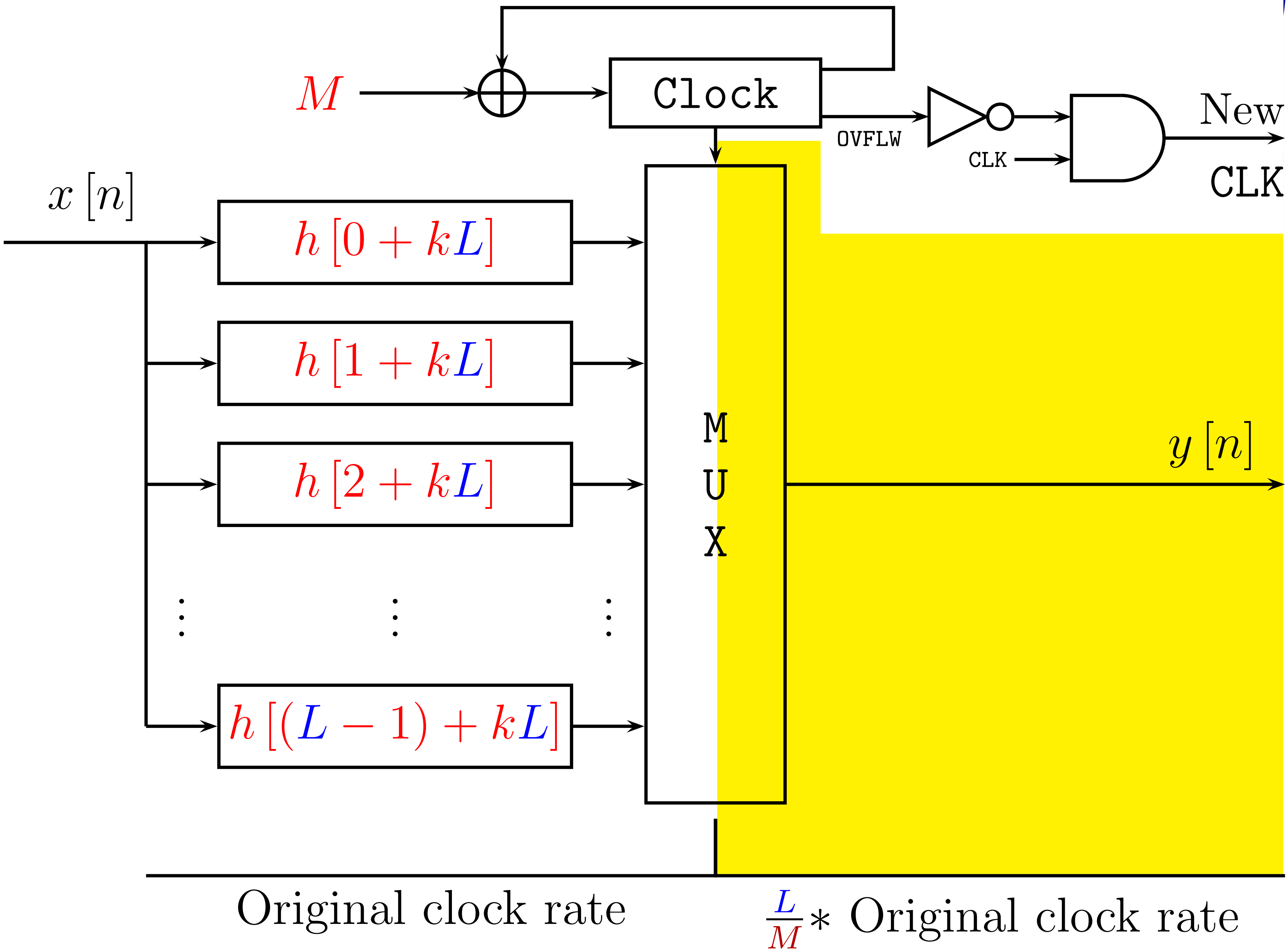


Filters

What filters?

$$
\begin{aligned}
y[nL+m] &= \sum_k h[k]\, x_u[nL+m-k] \\
&\quad \text{And if } k = aL+b \\
&= \sum_a \sum_b h[aL+b]\, x_u[nL+m-(aL+b)] \\
&\quad \text{If } m = b,\ x \text{ is non–zero, otherwise it is zero.} \\
&= \sum_k h[kL+m]\, x_u[(n-k)\,L] \\
&= \sum_k h[kL+m]\, x[n-k] \\
h_m[k] &= h[kL+m]\, x[n-k]
\end{aligned}
$$

Resampler



M
Clock
OVFLW
CLK
New
CLK
$x[n]$
$h[0+kL]$
$h[1+kL]$
$h[2+kL]$
$h[(L-1)+kL]$
M
U
X
$y[n]$
Original clock rate
$\frac{L}{M}*$ Original clock rate

Integer Downsampling

Did you notice the downsampler? It's as simple as taking only every M^{th} output,

$$y_r[n] = y[nM],$$

yet it has some nasty consequences in frequency. To see this, let's look at the inverse Fourier transform of $Y\left(e^{j2\pi f}\right)$,

$$y_r[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} Y\left(e^{j2\pi f}\right) e^{j2\pi f n M} df.$$

Splitting this band into M equal subbands,

$$= \sum_{k=0}^{M-1} \int_{-\frac{1}{2M}}^{\frac{1}{2M}} Y\left(e^{j2\pi\left(f+\frac{k}{M}\right)}\right) e^{j2\pi\left(f+\frac{k}{M}\right)nM} df.$$

From here on out, it's just simplification.

Integer Downsampling

We'll start by a u substitution. That is, let,

$$\begin{aligned} u &= fM & f &= \frac{u}{M} \\ du &= M\,df & df &= \frac{1}{M}du, \text{ then} \end{aligned}$$

$$y_r[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left[\frac{1}{M} \sum_{k=0}^{M-1} Y\left(e^{j2\pi\left(\frac{u+k}{M}\right)}\right)\right] e^{j2\pi un} du$$

This, however, is just the expression for the inverse z—transform of a sequence. Indeed, it is the inverse z—transform of $Y_r\left(e^{j2\pi f}\right)$ and thus,

$$Y_r\left(e^{j2\pi f}\right) = \frac{1}{M}\sum_{k=0}^{M-1} Y\left(e^{j2\pi(f+k)\frac{1}{M}}\right)$$

Resampling

But where did $Y\left(e^{j2\pi f}\right)$ come from?

$$
\begin{aligned}
Y\left(e^{j2\pi f}\right) &= X\left(e^{j2\pi f L}\right) H\left(e^{j2\pi f}\right), \text{ and thus} \\
Y_r\left(e^{j2\pi f}\right) &= \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{j2\pi (f+k)\frac{L}{M}}\right) H\left(e^{j2\pi \frac{f+k}{M}}\right), \text{ or} \\
&= \overbrace{X\left(e^{j2\pi f \frac{L}{M}}\right)}^{\text{My desired output}} \frac{1}{M} H\left(e^{j2\pi f \frac{1}{M}}\right) \\
&\quad \underbrace{+ \frac{1}{M}\sum_{k=1}^{M-1} X\left(e^{j2\pi (f+k)\frac{L}{M}}\right) H\left(e^{j2\pi \frac{f+k}{M}}\right).}_{\text{Alias terms}}
\end{aligned}
$$

How can I make it so that $Y_r\left(e^{j2\pi f}\right) \approx X\left(e^{j2\pi f \frac{L}{M}}\right)$?

Filter Design

Neglecting aliasing for a moment, in order to pass my signal undistorted, two conditions must hold:

1. The highest frequency in $X\left(e^{j2\pi f}\right)$ must be less than the Nyquist frequency,

$$\frac{M}{L} f_x \quad < \quad \frac{1}{2}$$

2. $H\left(e^{j2\pi f}\right)$ must continue to pass $X\left(e^{j2\pi f L}\right)$ undistorted. This leads to nearly the same criteria as before,

$$H\left(e^{j2\pi f}\right) \quad = \quad M, \text{ whenever } |f| < f_p = \frac{1}{L} f_x$$

That leaves the stopband and the alias terms.

Filter Design

In order to eliminate aliasing, I desire,

$$H\left(e^{j2\pi\frac{f+k}{M}}\right) = 0$$

forall $k \in [1, M-1]$ and $\left|f\frac{L}{M}\right| < f_x$. Breaking out the range of frequency values, we have,

$$\begin{aligned} -f_x < & \quad \frac{L}{M}f \quad < f_x \\ \frac{k}{M} - \frac{1}{L}f_x < & \quad \frac{k+f}{M} \quad < \frac{k}{M} + \frac{1}{L}f_x \end{aligned}$$

This inequality describes multiple bands where $H\left(e^{j2\pi f}\right)$ must be zero in order to avoid alasing. I can either control $H\left(e^{j2\pi f}\right)$ over each stopband individually, or aggregate these regions into one stopband, $|f| > f_s = \frac{1}{M} - f_p$.

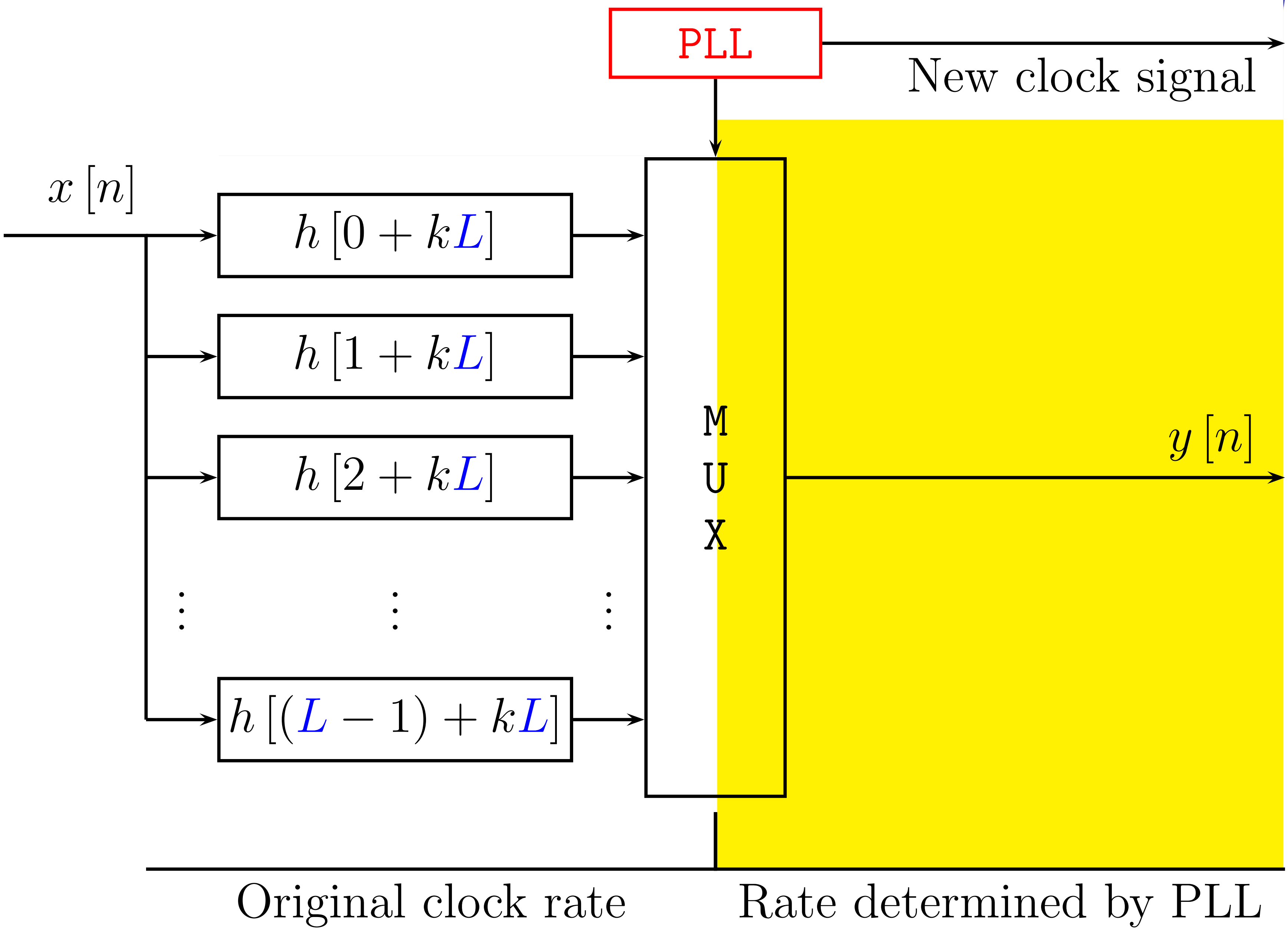
Filter Design

Our final resampling filter design requirements are:

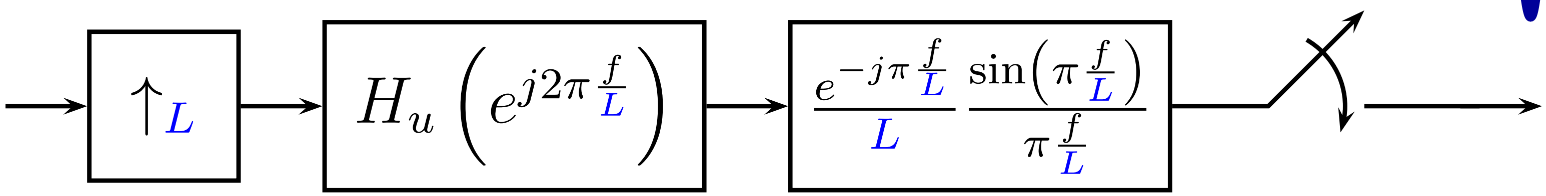
$$H\left(e^{j2\pi f}\right) \approx \begin{cases} M & |f| < f_p = \dfrac{1}{L} f_x < \dfrac{1}{2M} \\ 0 & |f| > f_s = \dfrac{1}{M} - f_p \\ \text{and don't care otherwise.} \end{cases}$$

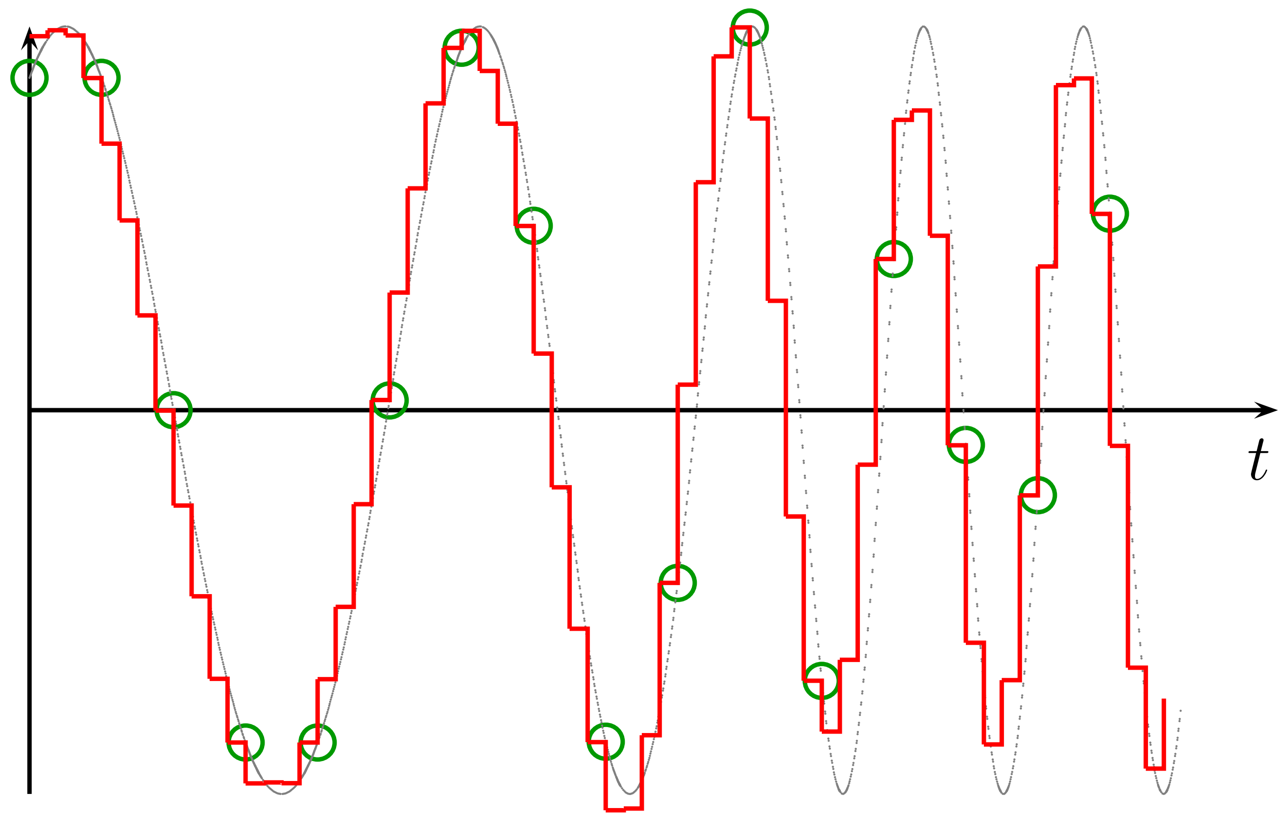
This filter will only be an interpolator if $\dfrac{1}{L} - f_p < \dfrac{1}{M} - f_p$, or equivalently if $M < L$. That is to say, an interpolating filter is only possible when the sample rate is increasing overall.

Tracking Resampler



Equivalent Circuit





Conclusion: We gotta be able to do better than this!

Questions

1. You have a 5 MHz wide complex signal, sampled at 14 MHz. What filter cutoff values, f_p, would you use to resample this signal at 12 MHz?
2. Given a 44.1 ksps real signal, what cutoff value would you use to design a resampler to give you 8 ksps samples? (Hint: Toll quality audio ends at 3.3 kHz.)
3. Following your 44.1 kHz to 8 kHz down sampling operation, you find an annoying tone in your signal at 1.4 kHz. You don't recall this tone being in the original signal. What happened?
4. Suppose the tone occurred at 3.7 kHz. Should you be concerned?

Questions

5. Consider a digital signal sampled at 4.1 samples per symbol. You would like to resample this signal at one sample per symbol. What cutoff would you use?

Project

- Build a resampler to convert a 44.1 kHz audio signal to 8 kHz. Use 3.3 kHz as a cutoff frequency.

  You could use a 441:80 resampler ...

- Suppose instead you have a digital signal sampled at 2π samples per symbol. Build a resampler to bring this signal down to one sample per symbol.

Mathematical Theory for Continuous Resamplers

Mathematical Theory

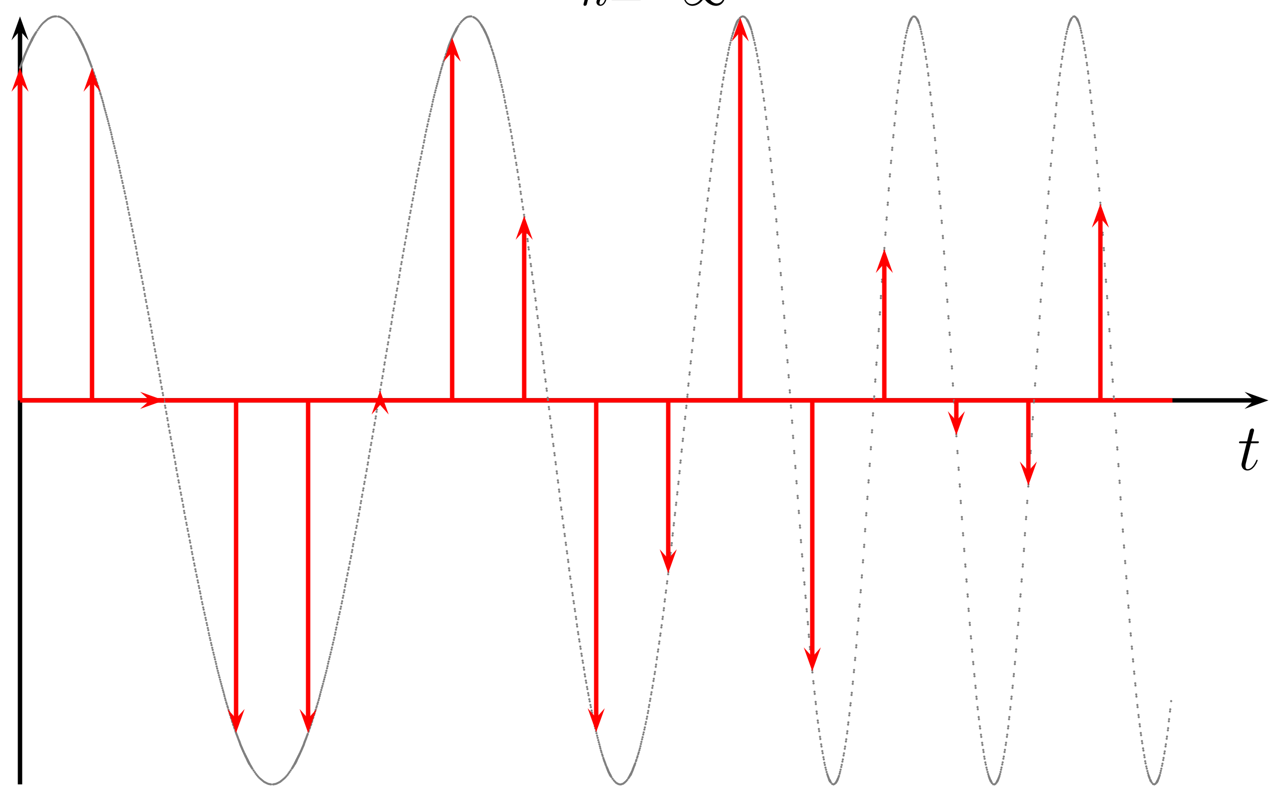
We'll solve this problem by forming a continuous signal to resample from on at an arbitrary sampling rate. To see how this is done, we'll discuss,

- Basic theory, or the mathematics behind how it is done
- Some mathematical properties—Fourier transforms, continuity, etc.
- Aliasing
- Filter design criteria

Basic Theory

- Step one: Replace the incoming sampled sequence by a sequence of impulses,

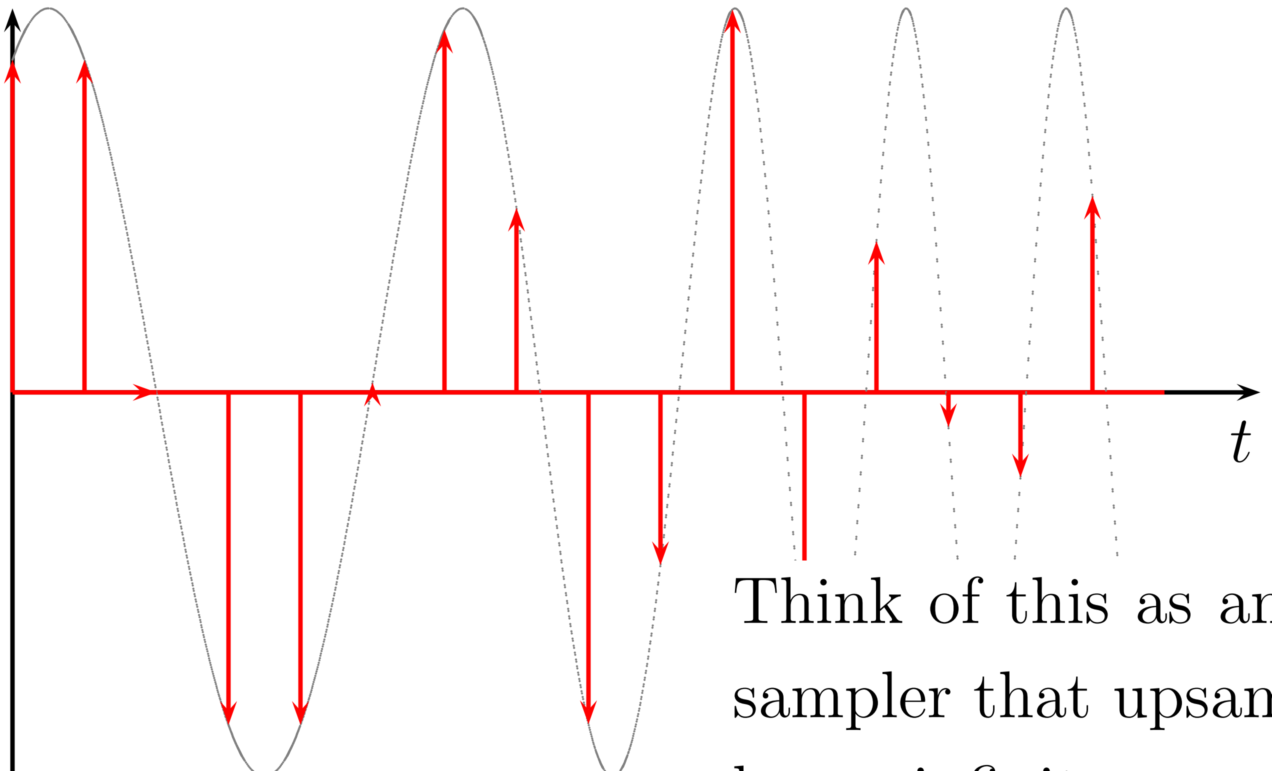
$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\,\delta(t-n)$$



Basic Theory

- Step one: Replace the incoming sampled sequence by a sequence of impulses,

$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\,\delta(t-n)$$



Think of this as an upsampler that upsamples by an infinite amount.

Basic Theory

- Step one: $x(t) = \sum_{n=-\infty}^{\infty} x[n]\,\delta(t-n)$
- Step two: Filter the sequence of impulses.

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\,x(t-\tau)\,d\tau.$$

Simplifying with the expression from step one,

$$= \int_{-\infty}^{\infty} h(\tau) \sum_{n=-\infty}^{\infty} x[n]\,\delta(t-\tau-n)\,d\tau.$$

Then, reordering,

$$= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} h(\tau)\,\delta(t-\tau-n)\,d\tau.$$

Basic Theory

- Step two: Filter the sequence of impulses. (Continued . . .)

$$y(t) = \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} h(\tau)\,\delta(t-\tau-n)\,d\tau.$$

Finally, by the sifting property of the δ function.

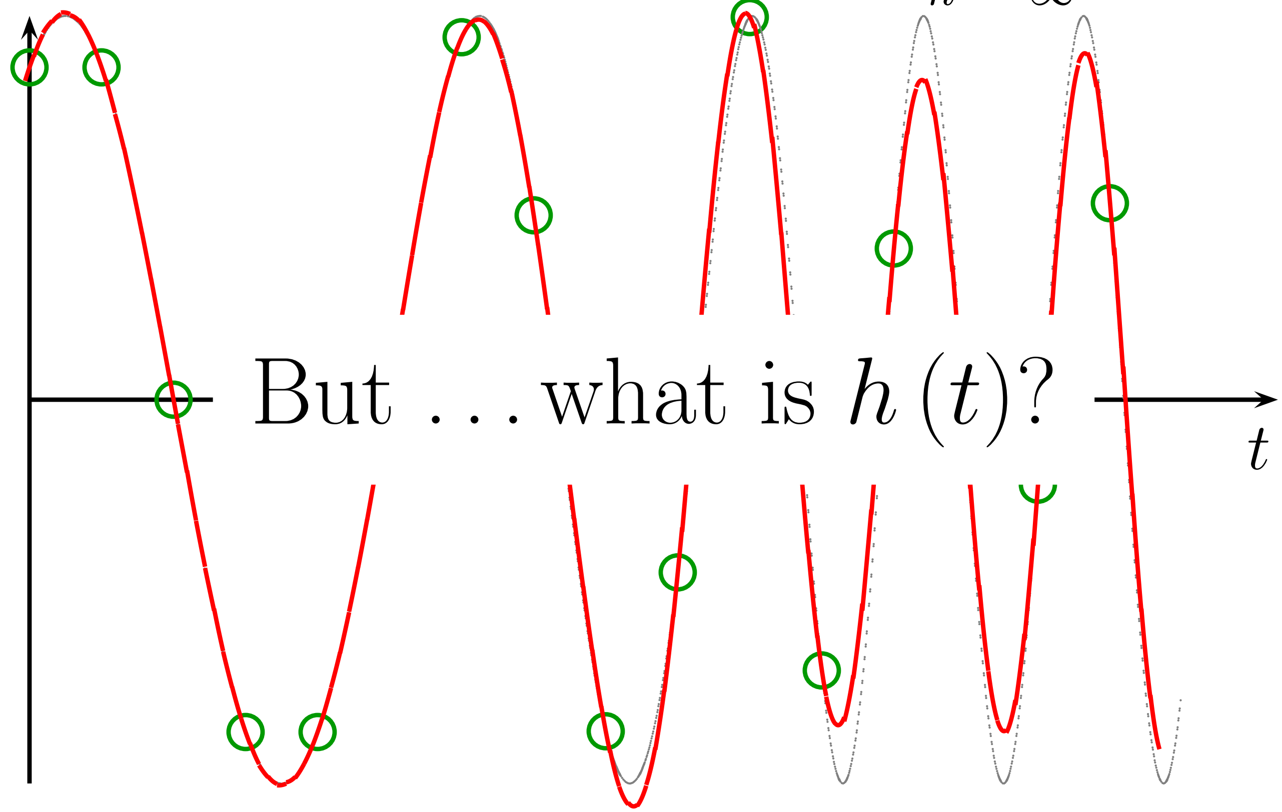
$$y(t) = \sum_{n=-\infty}^{\infty} x[n]\,h(t-n)$$

- Step three: Plug in t and evaluate

Basic Theory

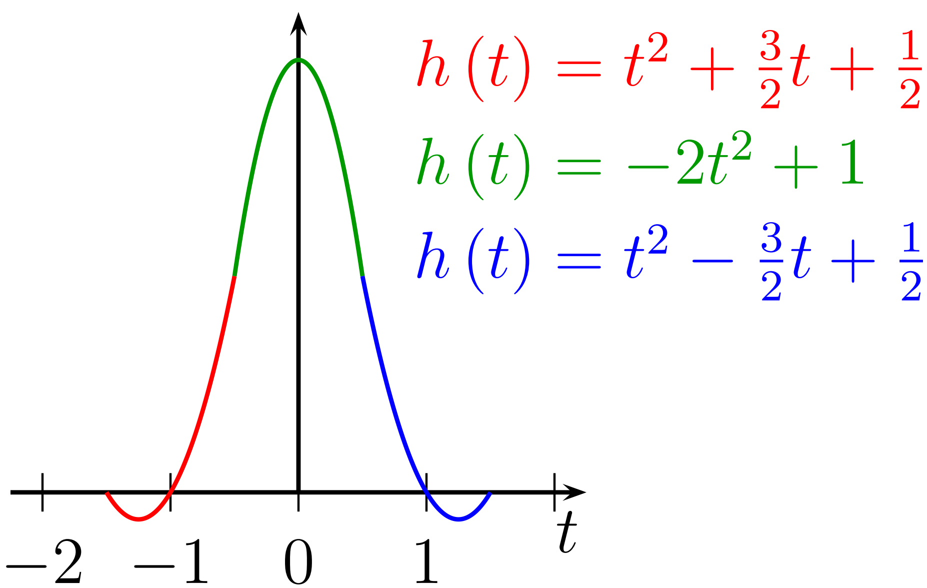
- Step two: Filter the sequence of impulses. (Continued ...)

$$y(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau = \sum_{n=-\infty}^{\infty} x[n]\, h(t-n)$$



Polynomial Filters

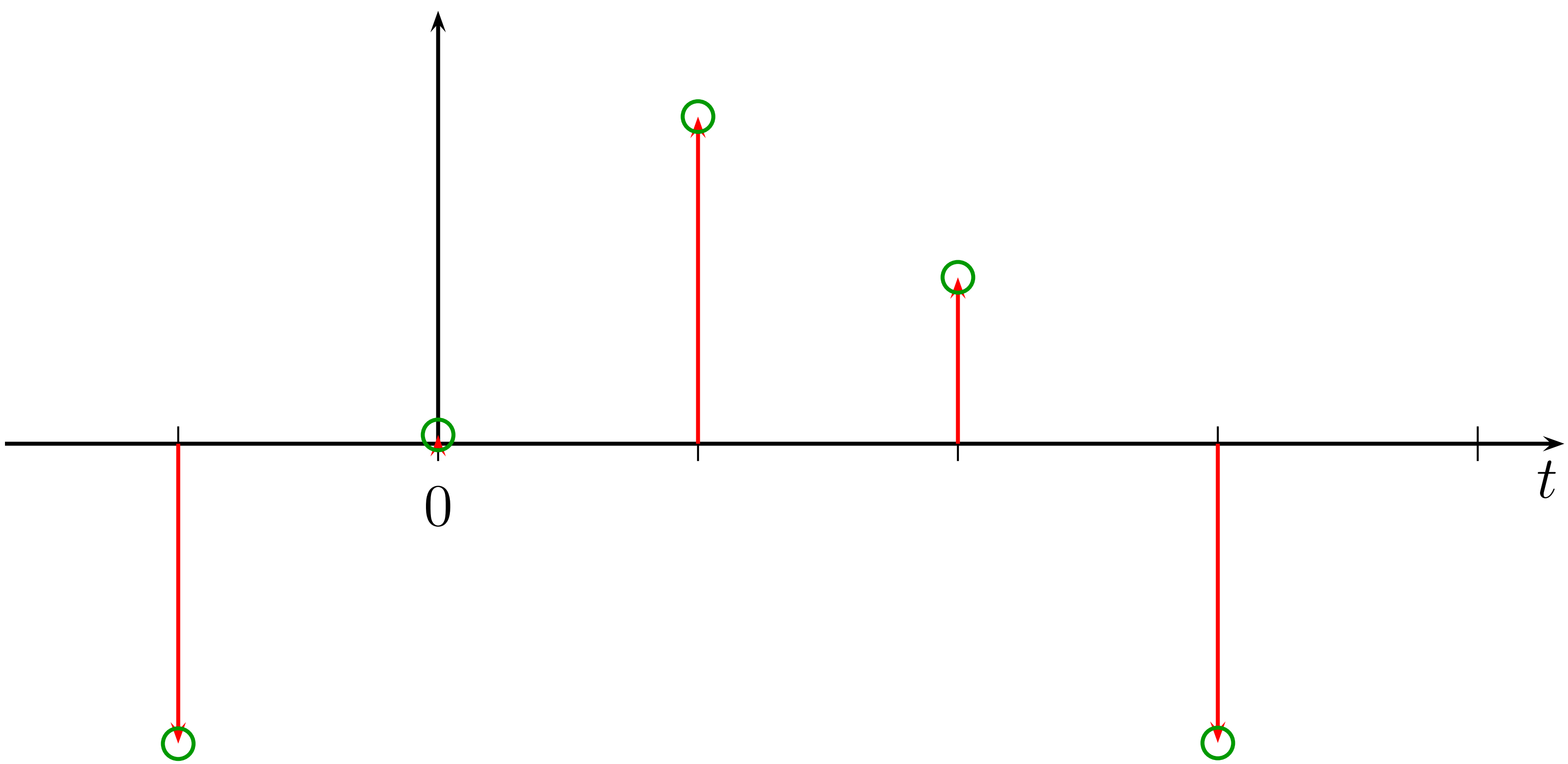
Suppose $h(t)$ is a piecewise polynomial.



Filter Application

How do I apply this filter?

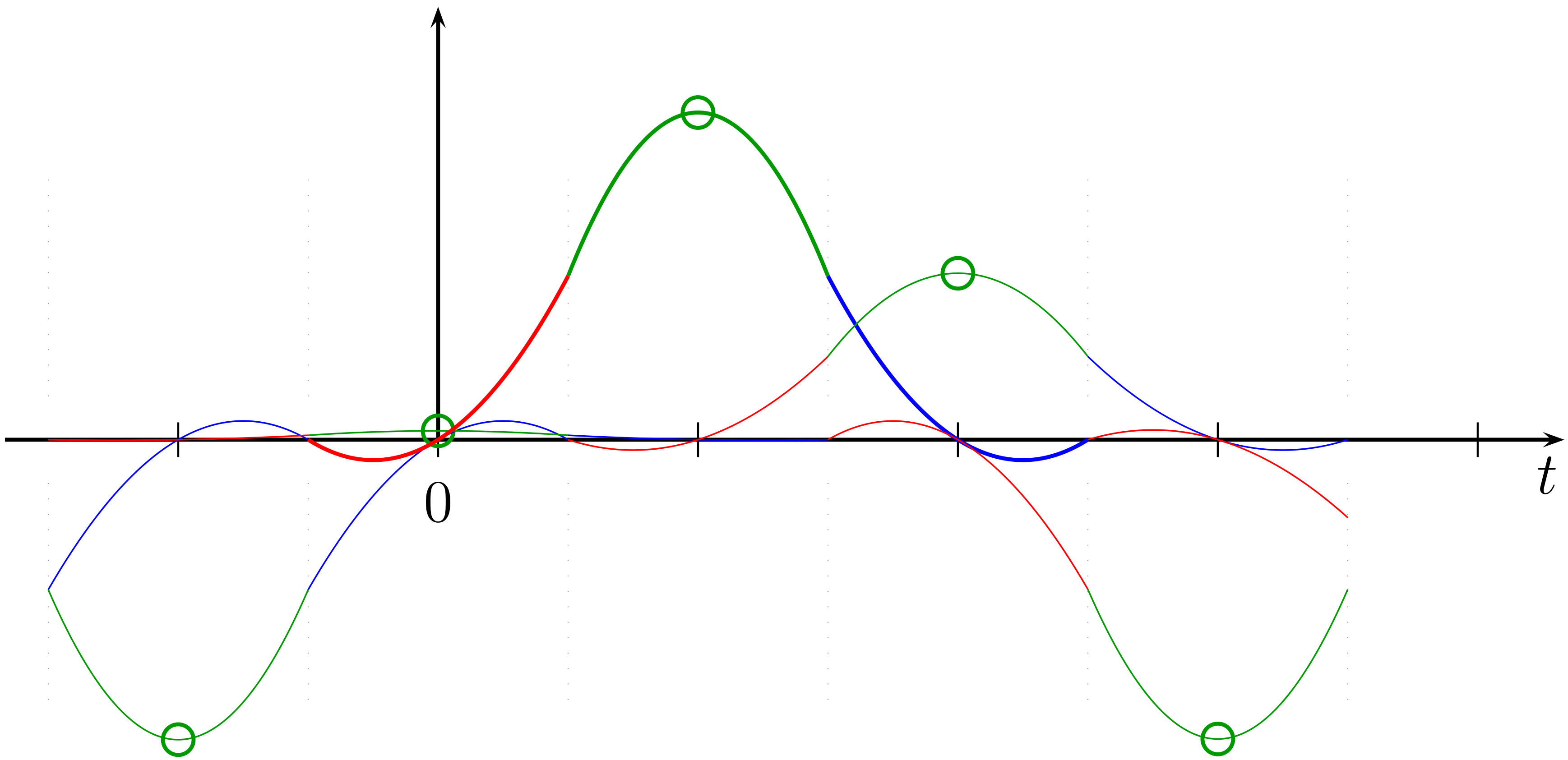
Here are my points.



Filter Application

How do I apply this filter? *Superposition!*

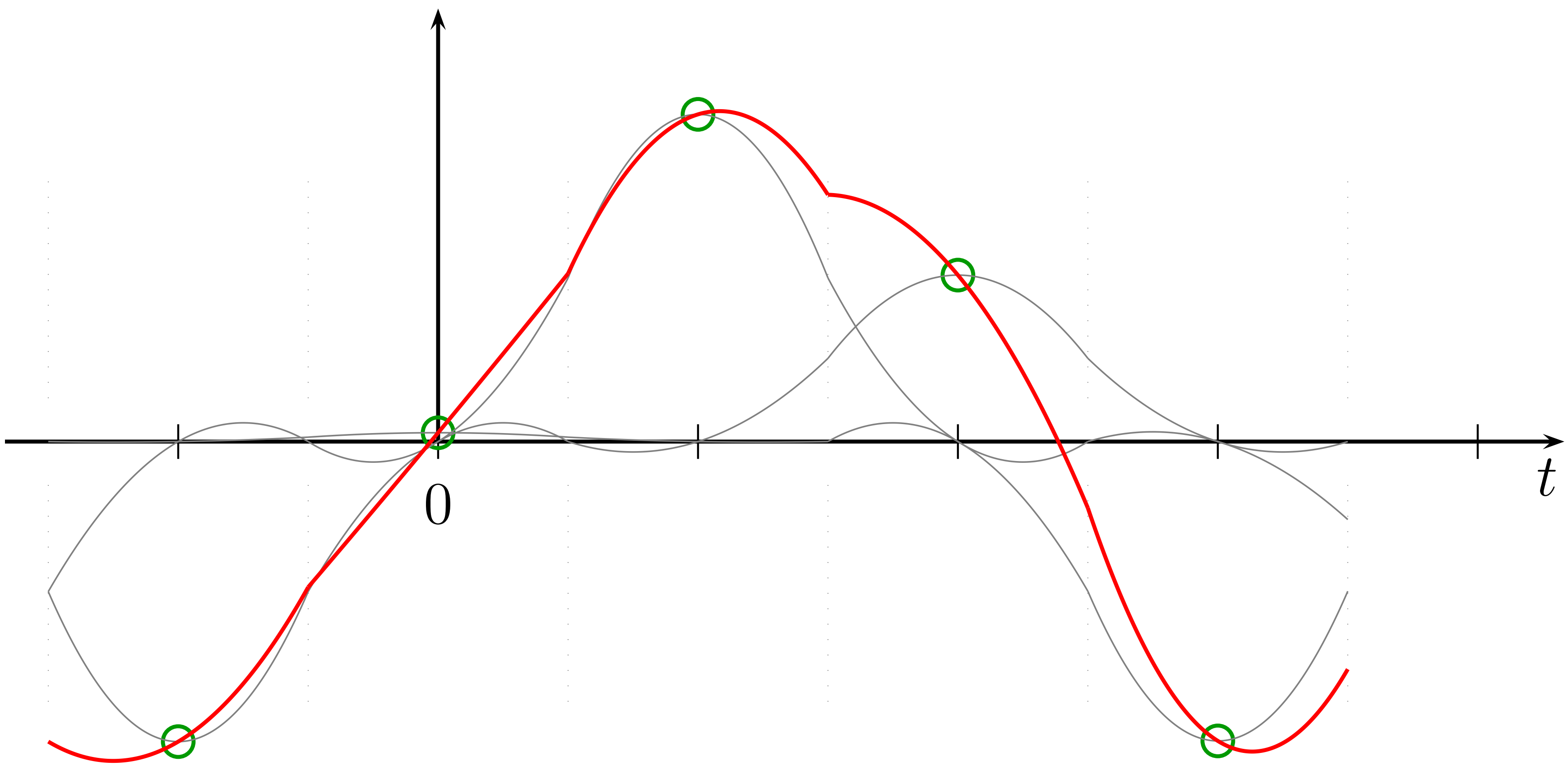
$$y(t) = \sum_{n=-\infty}^{\infty} x[n]\, h(t-n)$$



Filter Application

How did I do?

Not very well. Maybe I can design a better filter?



Continuity

Theorem: If $h(t)$ is continuous, $y(t)$ will also be continuous.

Proof: $y(t)$ is a sum of continuous functions. Thus, by closure, $y(t)$ is continuous. Or, consider,

$$\begin{aligned}
\lim_{t \to t_o} y(t) &= \lim_{t \to t_o} \sum_{n=-\infty}^{\infty} x[n]\, h(t-n) \\
&= \sum_{n=-\infty}^{\infty} x[n] \lim_{t \to t_o} h(t-n)
\end{aligned}$$

And, if $h(t)$ is continuous, then

$$\begin{aligned}
&= \sum_{n=-\infty}^{\infty} x[n]\, h(t-n) \\
&= y(t)
\end{aligned}$$

Some Characteristics

Theorem: If $h(t-n)$ is a polynomial of degree k for all $n \in \mathbb{Z}$, then $y(t)$ will also be a polynomial of degree k near t.

Proof: Identical to the proof of continuity.

Interpolation

Definition: A continuous filter, $h(t)$, is called an *interpolating filter* if the filtered output, $y(t)$, passes through $x[n]$ whenever $t = n$.

Interpolation

Theorem: If $h(t)$ is an interpolating filter, then $h(t)$ must be constrained by,

$$h(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \quad t \in \mathbb{Z} \\ \text{Something else otherwise} \end{cases}$$

Proof: By direct substitution, we have,

$$y(n) = \sum_{k=-\infty}^{\infty} x[k]\, h(n-k).$$

This will only equal $x[n]$ if $h(n-k)$ is one when $n = k$, and zero otherwise. (Otherwise you get other terms from $x[n]$ mixed in...) $\mathcal{Q}.\mathcal{E}.\mathcal{D}.$

Fourier Transform

The operation of $h(t)$ on $x[n]$ may be examined in frequency. To see this, take the Fourier transform of $y(t)$. By the definition,

$$Y(f) = \int_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} x[n]\, h(t-n) \right) e^{-j2\pi f t} dt.$$

We then reorder the operations,

$$= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} h(t-n)\, e^{-j2\pi f t} dt,$$

and adjust to integrate over $t-n$,

$$= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} h(t-n)\, e^{-j2\pi f (t-n)} e^{-j2\pi f n} dt.$$

Fourier Transform

Examining $y(t)$ in frequency: (continued ...)

We continue by applying a u substitution, $u = t - n$, and then shuffle the results into a product of two terms:

$$Y(f) = \left(\sum_{n=-\infty}^{\infty} e^{-j2\pi f n} x[n]\right)\left(\int_{-\infty}^{\infty} h(u)\, e^{-j2\pi f u} du\right).$$

By the definitions of the Fourier transform,

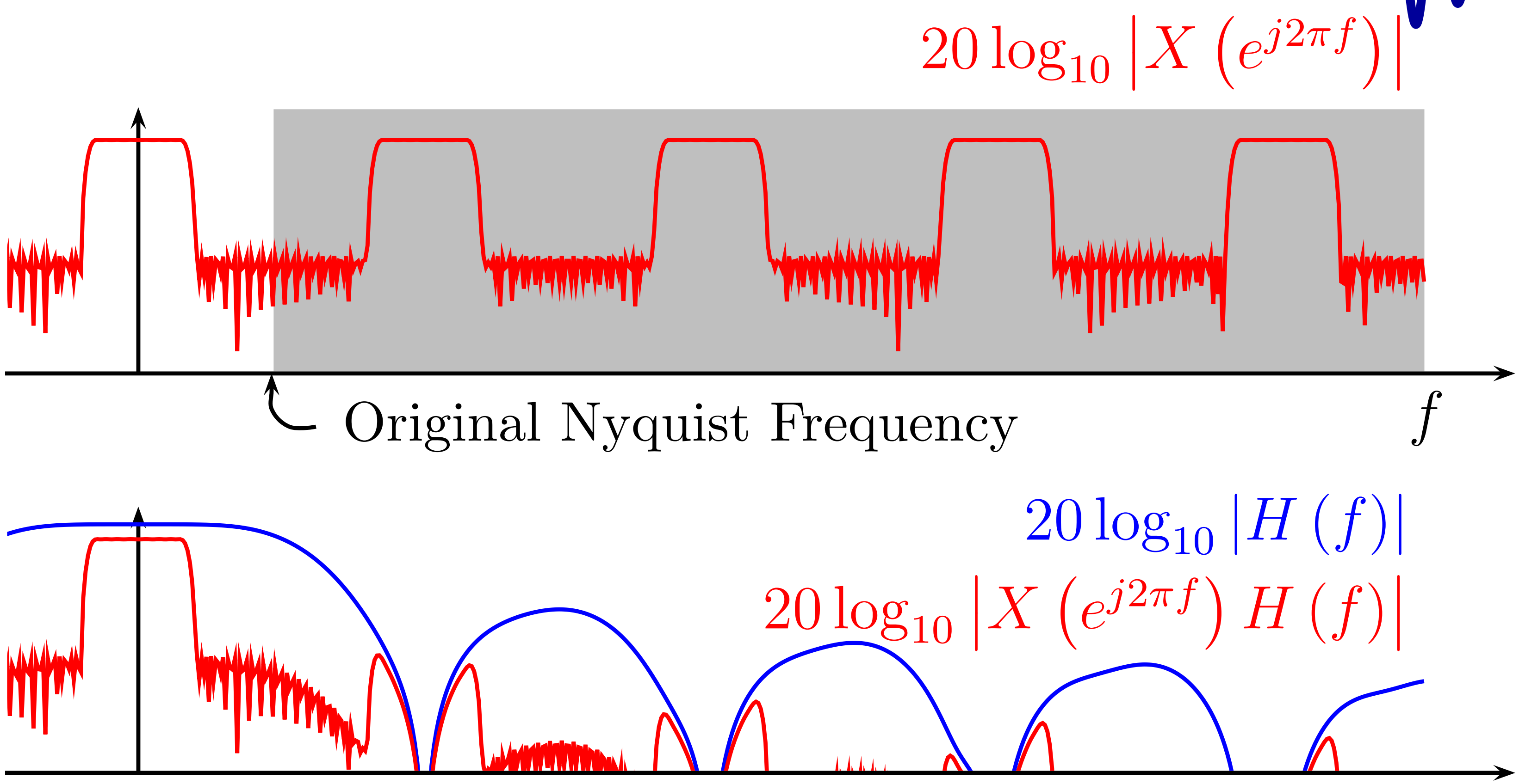
$$= \left(\sum_{n=-\infty}^{\infty} e^{-j2\pi f n} x[n]\right) H(f),$$

and the Discrete Fourier transform,

$$= X\left(e^{j2\pi f}\right) H(f).$$

Conclusion: $h(t)$ acts as a filter upon the input data. Therefore $h(t)$ may be designed as such.

Aliasing



No Nyquist frequency applies. This signal is now aperiodic in frequency.

Design Criteria

How shall we design our interpolating filter, $h(t)$? I know of no "optimal" method, however you may find the following criteria useful:

Primary

1. Piecewise polynomial, of degree k.
2. Lowpass filter
3. Interpolator

Secondary

1. Continuous
2. Constant inputs$\Rightarrow$ constant output
3. Linear input $\Rightarrow$ linear output
4. Continuous derivative

Resampling

So far we have only described the operator that upsamples a discrete signal and makes it continuous. What happens when this signal is resampled?

The answer is that anything above the Nyquist frequency aliases. To see this, let's look at the Inverse fourier transform of $y\left(nT_r\right)$,

$$y_r\left[n\right] = y\left(nT_r\right) = \int_{-\infty}^{\infty} X\left(e^{j2\pi f}\right) H\left(f\right) e^{j2\pi f\left(nT_r\right)} df.$$

Separating the infinite integral into bands,

$$= \int_{-\frac{1}{2T_r}}^{\frac{1}{2T_r}} \sum_{k=-\infty}^{\infty} X\left(e^{j2\pi\left(f+\frac{k}{T_r}\right)}\right) H\left(f + \frac{k}{T_r}\right) e^{j2\pi f\left(nT_r\right)} df$$

From here on out, we just simplify.

Resampling

$$\begin{aligned} \text{Let} \quad u &= fT_r & f &= \frac{u}{T_r} \\ du &= T_r df & df &= \frac{1}{T_r} du, \text{ then} \end{aligned}$$

$$y_r[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} \underbrace{\left[\frac{1}{T_r} \sum_{k=-\infty}^{\infty} X\left(e^{j2\pi\left(\frac{u+k}{T_r}\right)}\right) H\left(\frac{u+k}{T_r}\right) \right]}_{Y_r\left(e^{j2\pi u}\right)} e^{j2\pi un} du$$

Here, you can find $Y_r\left(e^{j2\pi f}\right)$ defined inside an inverse discrete Fourier transform,

$$Y_r\left(e^{j2\pi f}\right) = \frac{1}{T_r} \sum_{k=-\infty}^{\infty} X\left(e^{j2\pi\left(\frac{f+k}{T_r}\right)}\right) H\left(\frac{f+k}{T_r}\right)$$

Of the terms in this infinite summation, only one is desired.

Resampling

Here you can see vividly the result or downsampling upon the Frequency domain:

$$
\begin{aligned}
Y_r\left(e^{j2\pi f}\right) &= \overbrace{\frac{1}{T_r} X\left(e^{j2\pi\left(\frac{f}{T_r}\right)}\right)}^{\text{Desired response}} H\left(\frac{f}{T_r}\right) \\
&+ \underbrace{\frac{1}{T_r}\sum_{k\neq 0} X\left(e^{j2\pi\left(\frac{f+k}{T_r}\right)}\right) H\left(\frac{f+k}{T_r}\right)}_{\text{Undesired aliasing terms}}
\end{aligned}
$$

The entire performance of this system depends upon the design of $H\left(f\right)$.

Filter Design

Looking at the passband of $X\left(e^{j2\pi f}\right)$,

$$Y_r\left(e^{j2\pi f}\right) = \frac{1}{T_r} X\left(e^{j2\pi\left(\frac{f}{T_r}\right)}\right) H\left(\frac{f}{T_r}\right) + \cdots$$

- Good signal should not alias, thus

$$T_r f_x < \frac{1}{2}$$

- Neither should it be distorted

$$H\left(e^{j2\pi f}\right) = T_r, \text{ whenever } |f| < f_x$$

Filter Design

Next, consider the alias terms. We want,

$$0 \;=\; \frac{1}{T_r}\sum_{k\neq 0} X\left(e^{j2\pi\left(\frac{f+k}{T_r}\right)}\right) H\left(\frac{f+k}{T_r}\right).$$

As before, we can either design a filter with multiple stopbands, or one aggregated stopband. $H\left(\frac{f+k}{T_r}\right)$ must be zero any time $\left|\frac{f}{T_r}\right| < f_x$. That means,

$$\begin{aligned} -f_x < \quad & \frac{f}{T_r} \quad < f_x \\ \frac{k}{T_r} - f_x < \quad & \frac{k+f}{T_r} \quad < \frac{k}{T_r} + f_x \end{aligned}$$

The least of these regions starts at $f_s = \dfrac{1}{T_r} - f_x$, giving us our stopband frequency.

Filter Design

We now have our criteria for success. Our goal will have been achieved when,

- We don't downsample too far, $T_r < \frac{1}{2f_x}$, and
- The interpolation filter is good,

$$H(f) = \begin{cases} T_r & |f| < f_x \\ 0 & |f| > \frac{1}{T_r} - f_x \end{cases}$$

Quadratic Interpolators

Quadratics

Let's build a piecewise quadratic interpolating filter.

Why piecewise? Because our resulting signal will then be a quadratic between sample points.

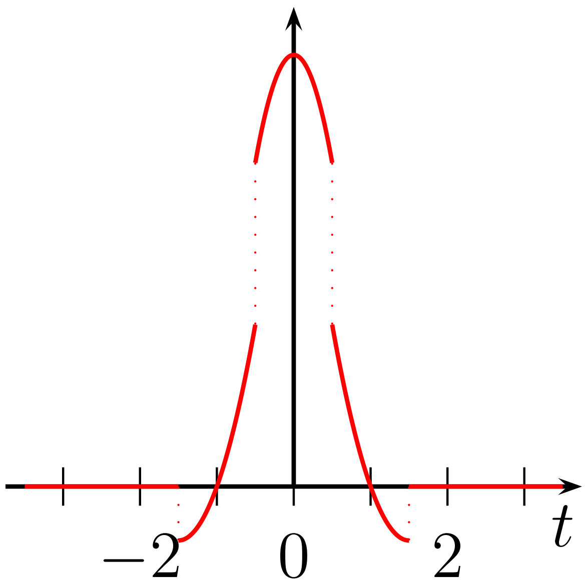
Why quadratic? Two reasons: First, it's simple. Second, finding maxima is easy. Could we use higher order polynomials? Certainly.

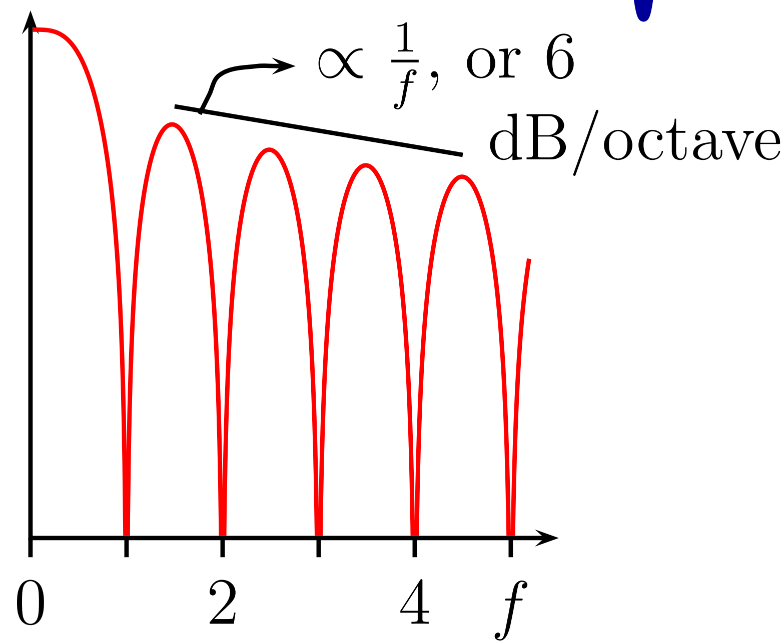
Overview

In this section, we'll build and look at:

- A straight quadratic interpolator
- A continuous quadratic
- Two longer quadratics, and
- The smoothest quadratic of all.

Original Quadratic





$$
\begin{aligned}
a[n] &= x[n] \otimes \{ \tfrac{1}{2} \quad -1 \quad \tfrac{1}{2} \} \\
b[n] &= x[n] \otimes \{ \tfrac{1}{2} \quad 0 \quad -\tfrac{1}{2} \} \\
c[n] &= x[n] \otimes \{ 0 \quad 1 \quad 0 \} \\
y(t) &= a[n](t-n)^2 + b[n](t-n) + c[n]
\end{aligned}
$$

Homework

1. What quadratic function interpolates the sequence(s):
   - $[-2, -1, 0, 1, 2, 3, \ldots]$, from $x[n] = n$.
   - $[4, 1, 0, 1, 4, 9, \ldots]$, from $x[n] = n^2$.
   - $[1, -1, 1, -1, 1, -1, \ldots]$, from $x[n] = \cos\left(2\pi \frac{n}{2}\right)$.
   - $[1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, \ldots]$, from $x[n] = \cos\left(2\pi \frac{n}{4}\right)$.
   - $\left[1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 1, \ldots\right]$, from $x[n] = \cos\left(2\pi \frac{n}{8}\right)$.
2. Using matlab, plot the signals above together with their continuous–time counterparts.

Answers

1. What quadratic function interpolates the sequence(s):

   - $[-2, -1, 0, 1, 2, 3, \ldots]$, from $x[n] = n$.

$$
\begin{aligned}
a[n] &= x[n] \otimes \left\{ \frac{1}{2}, -1, \frac{1}{2} \right\} \\
&= \frac{1}{2}(n-1) - n + \frac{1}{2}(n+1) = 0 \\
b[n] &= x[n] \otimes \left\{ \frac{1}{2}, 0, -\frac{1}{2} \right\} \\
&= -\frac{1}{2}(n-1) + \frac{1}{2}(n+1) = 1 \\
c[n] &= x[n] \otimes \{0, 1, 0\} = n \\
y(t) &= (t - \lfloor t \rfloor) + \lfloor t \rfloor = t
\end{aligned}
$$

Answers

1. What quadratic function interpolates the sequence(s):
   - $[4, 1, 0, 1, 4, 9, \ldots]$, from $x[n] = n^2$.

$$
\begin{aligned}
a[n] &= x[n] \otimes \left\{ \frac{1}{2}, -1, \frac{1}{2} \right\} \\
&= \frac{1}{2}(n-1)^2 - n^2 + \frac{1}{2}(n+1)^2 = 1 \\
b[n] &= x[n] \otimes \left\{ \frac{1}{2}, 0, -\frac{1}{2} \right\} \\
&= -\frac{1}{2}(n-1)^2 + \frac{1}{2}(n+1)^2 = -2n \\
c[n] &= x[n] \otimes \{0, 1, 0\} = n^2 \\
y(t-n) &= t^2 - 2tn + n^2 = (t-n)^2
\end{aligned}
$$

Answers

1. What quadratic function interpolates the sequence(s):

   - $x[n] = \cos(2\pi f n)$.

$$
\begin{aligned}
a[n] &= \frac{1}{2}\cos(2\pi f(n-1)) - \cos(2\pi f n) \\
&\quad + \frac{1}{2}\cos(2\pi f(n+1)) \\
&= \frac{1}{2}\cos(2\pi f n)\cos(2\pi f) + \frac{1}{2}\sin(2\pi f n)\sin(2\pi f) \\
&\quad - \cos(2\pi f n) \\
&\quad + \frac{1}{2}\cos(2\pi f n)\cos(2\pi f) - \frac{1}{2}\sin(2\pi f n)\sin(2\pi f) \\
&= [\cos(2\pi f) - 1]\cos(2\pi f n)
\end{aligned}
$$

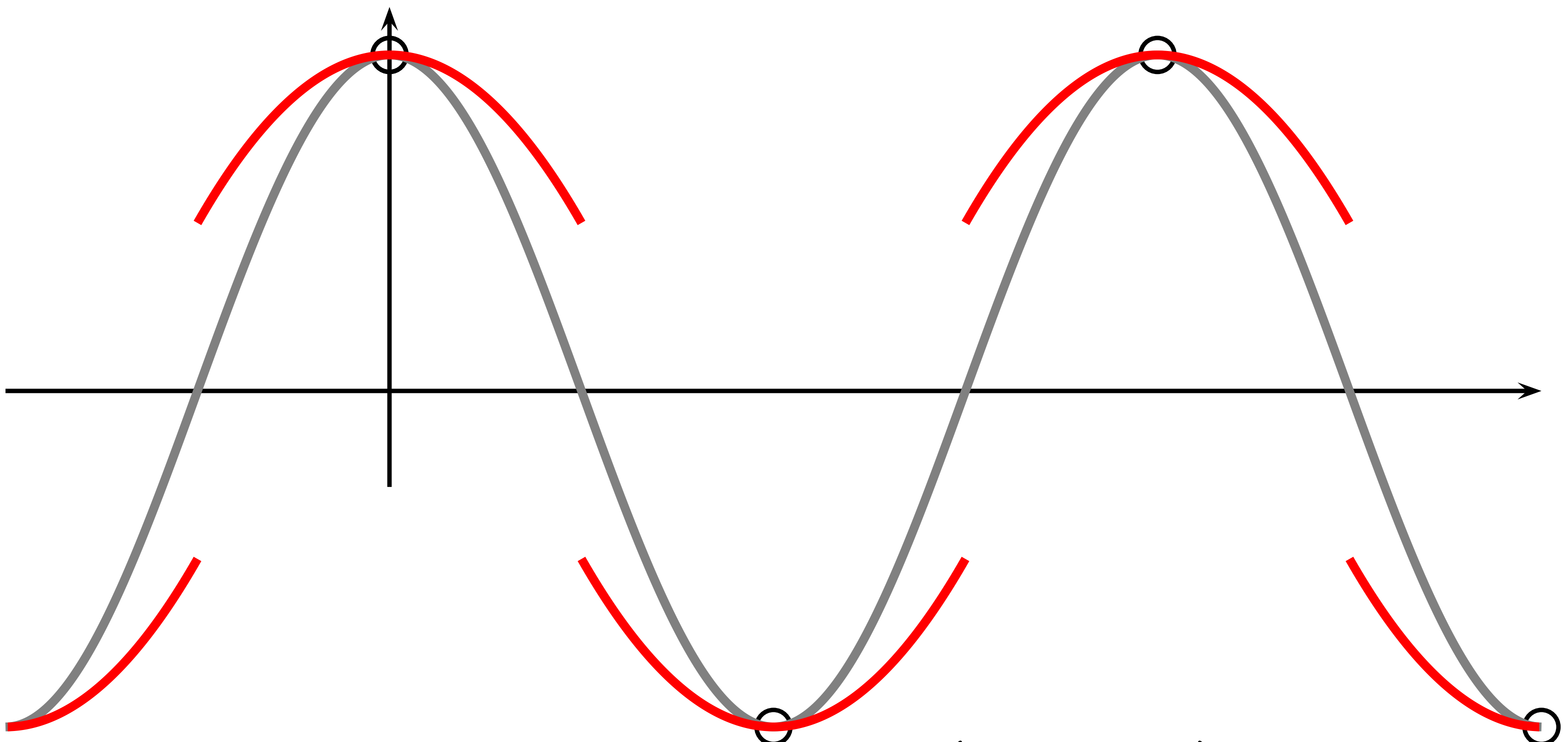
Answers

1. What quadratic function interpolates the sequence(s):

- $x[n] = \cos(2\pi f n)$.

$$
\begin{aligned}
a[n] &= [\cos(2\pi f) - 1]\cos(2\pi f n) \\
b[n] &= -\frac{1}{2}\cos(2\pi f(n-1)) + \frac{1}{2}\cos(2\pi f(n+1)) \\
&= -\frac{1}{2}\cos(2\pi f n)\cos(2\pi f) - \frac{1}{2}\sin(2\pi f n)\sin(2\pi f) \\
&\quad +\frac{1}{2}\cos(2\pi f n)\cos(2\pi f) - \frac{1}{2}\sin(2\pi f n)\sin(2\pi f) \\
&= -\sin(2\pi f)\sin(2\pi f n) \\
c[n] &= \cos(2\pi f n)
\end{aligned}
$$

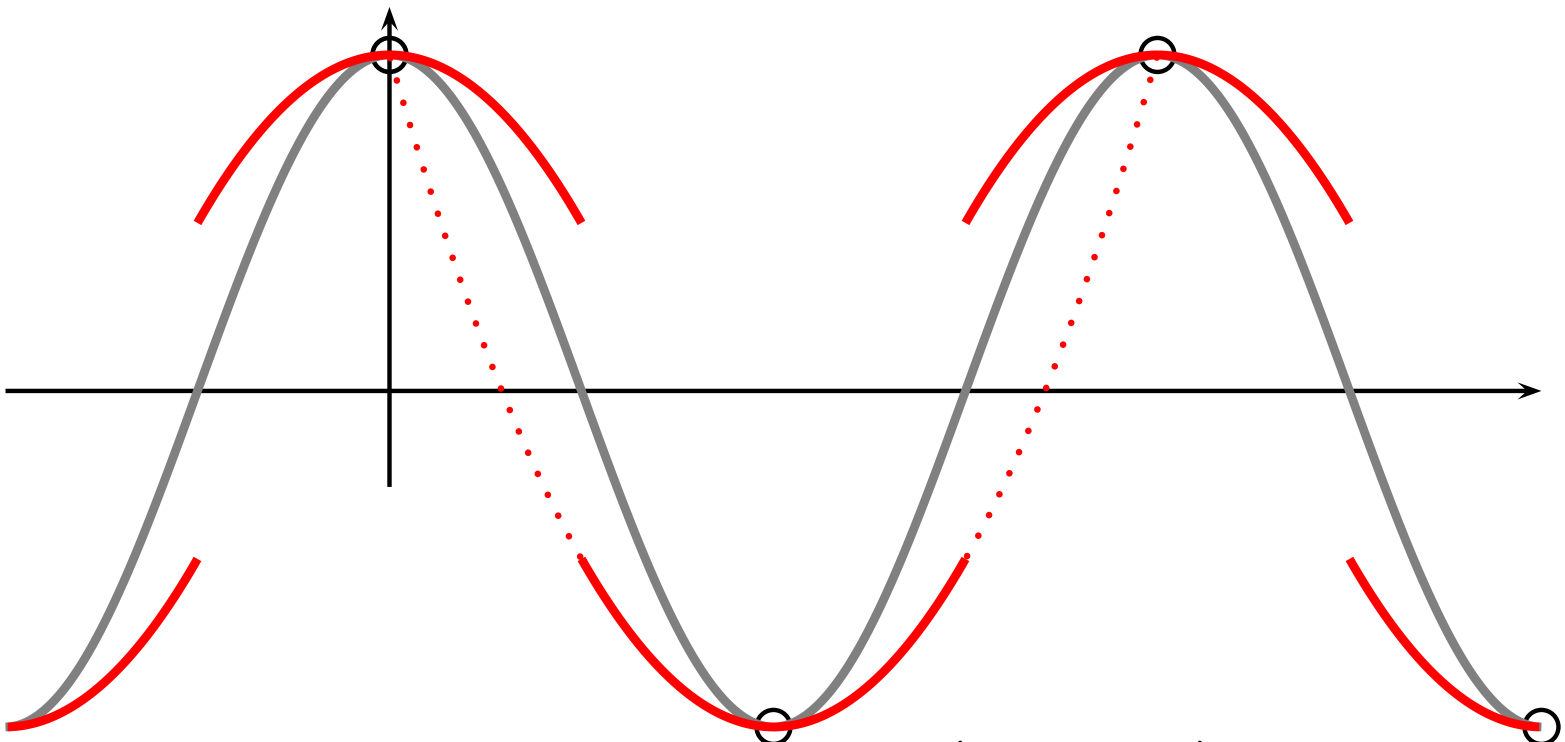
Examples



$$x[n] = \cos\left(2\pi \frac{n}{2}\right)$$

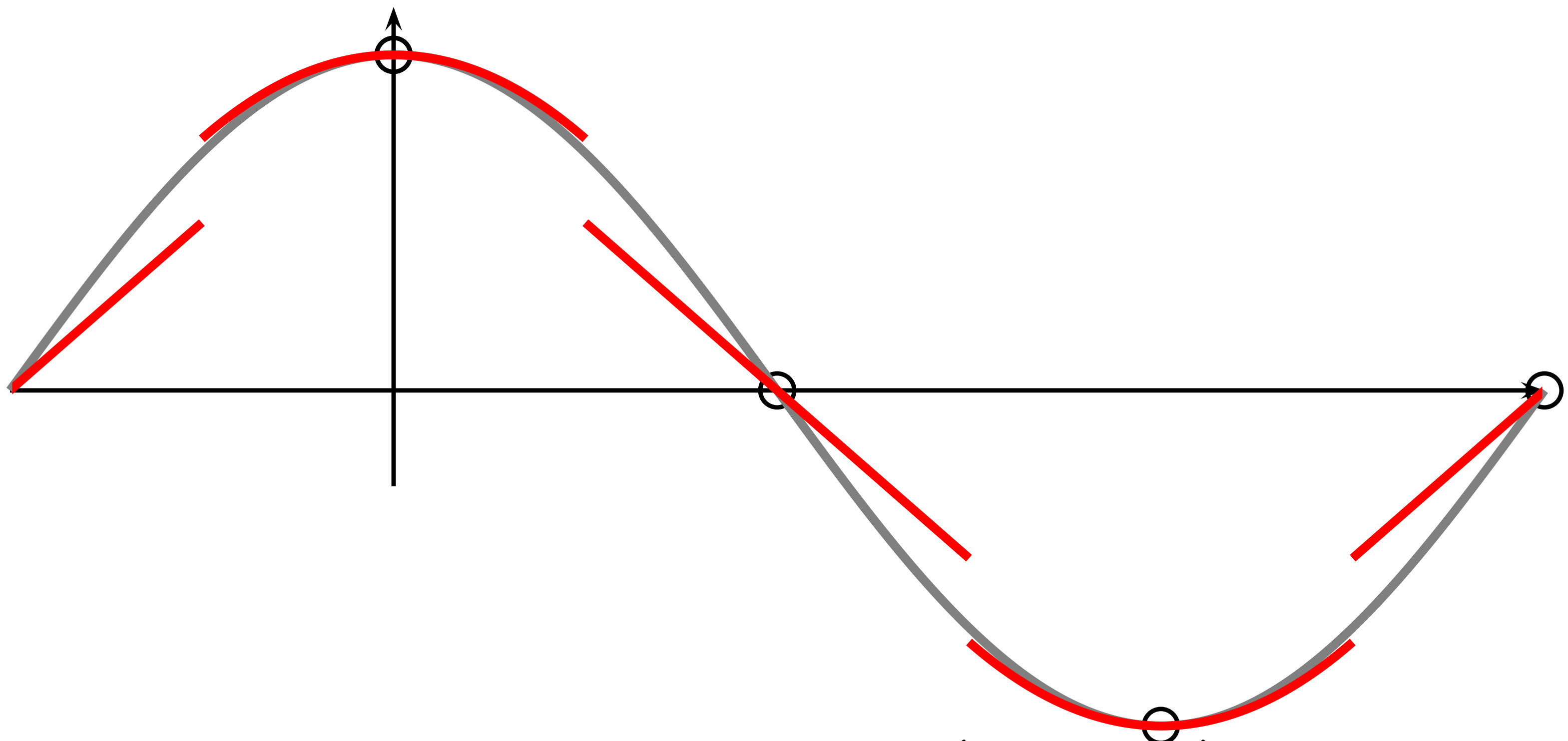
What happened here?

Examples



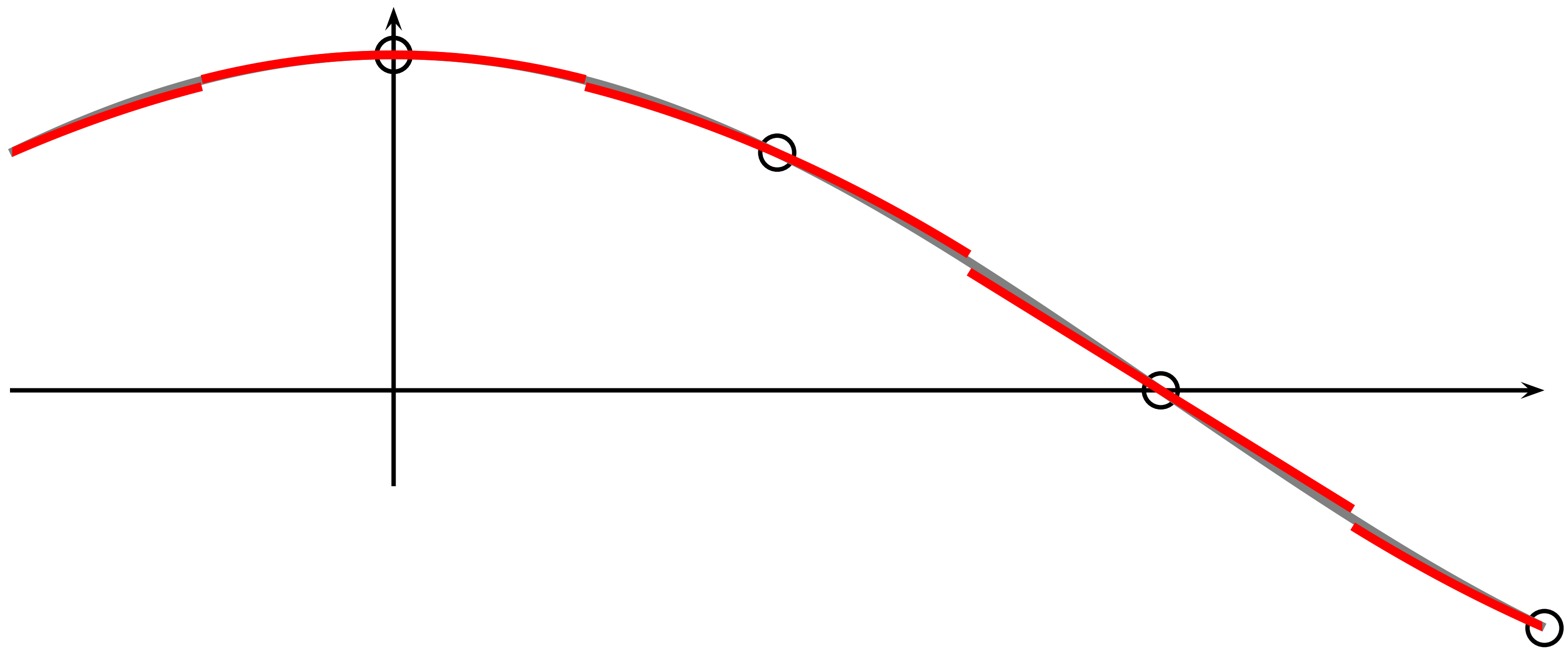
$$x[n] = \cos\left(2\pi \frac{n}{2}\right)$$

Examples



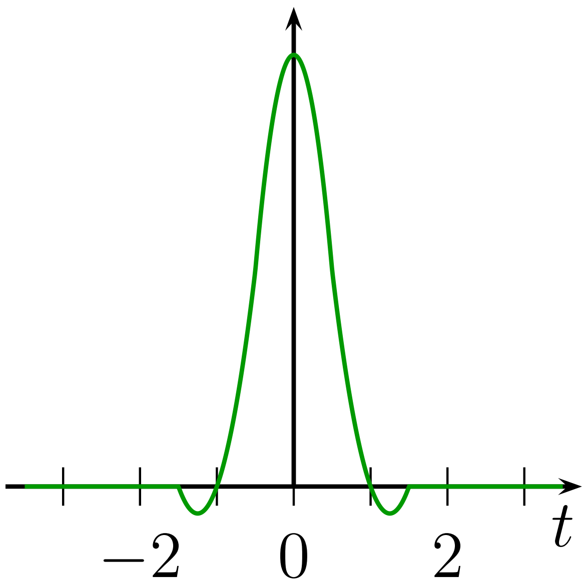
$$x[n] = \cos\left(2\pi \frac{n}{4}\right)$$

Examples



$$x[n] = \cos\left(2\pi \frac{n}{8}\right)$$

Continuous Quadratic



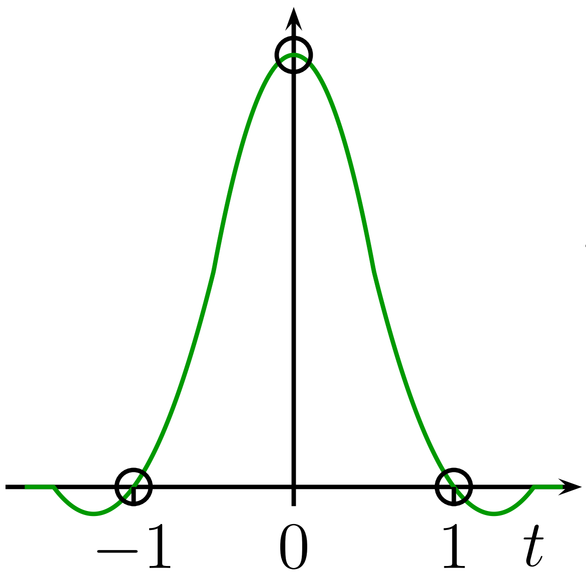
How shall I go about designing a better quadratic?

Criteria

- Continuous interpolator
- Constants $\Rightarrow$ constants
- Lines $\Rightarrow$ lines

How do we go from this criteria to the polynomial above?

Interpolation

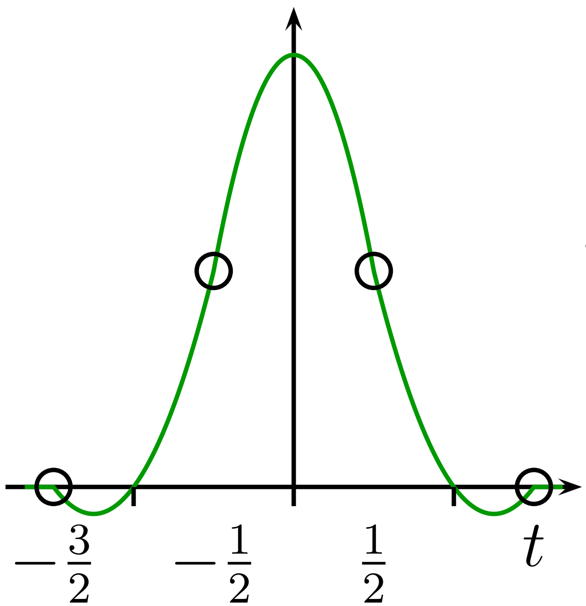


$$h(t) = \begin{cases} 0 & t < -1.5 \\ a_1 (t+1)^2 - b_1 (t+1) + 0 & -1.5 < t < -0.5 \\ a_0 t^2 + 1 & -0.5 < t < 0.5 \\ a_1 (t-1)^2 + b_1 (t-1) + 0 & 0.5 < t < 1.5 \\ 0 & t > 1.5 \end{cases}$$

$$\begin{aligned} 1 &= h(0) \\ 0 &= h(1) = h(-1) \end{aligned}$$

We've chosen a polynomial representation for our filters that guarantees this property. (Shown above.)

Continuity



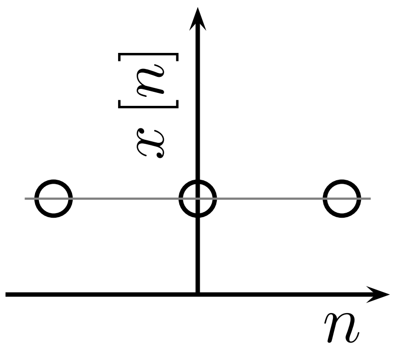
$$h(t) = \begin{cases} 0 & t < -1.5 \\ a_1(t+1)^2 - b_1(t+1) + 0 & -1.5 < t < -0.5 \\ a_0 t^2 + 1 & -0.5 < t < 0.5 \\ a_1(t-1)^2 + b_1(t-1) + 0 & 0.5 < t < 1.5 \\ 0 & t > 1.5 \end{cases}$$

$$a_0\left(\frac{1}{2}\right)^2 + 1 = a_1\left(-\frac{1}{2}\right)^2 + b_1\left(-\frac{1}{2}\right)$$

$$0 = a_1\left(\frac{1}{2}\right)^2 + b_1\left(\frac{1}{2}\right)$$

Here, we examine where the polynomials connect—making sure there are no breaks in our impulse response.

Constants



$$h(t) = \begin{cases} 0 & t < -1.5 \\ a_1 (t+1)^2 - b_1 (t+1) + 0 & -1.5 < t < -0.5 \\ a_0 t^2 + 1 & -0.5 < t < 0.5 \\ a_1 (t-1)^2 + b_1 (t-1) + 0 & 0.5 < t < 1.5 \\ 0 & t > 1.5 \end{cases}$$

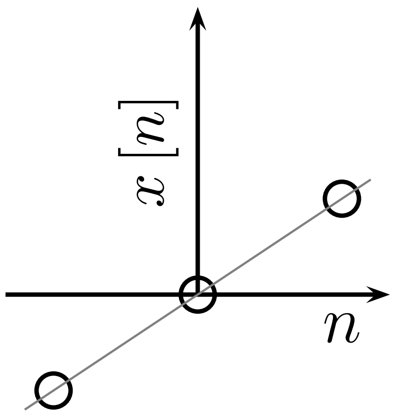
If the input is a constant, the output should be too.

$$\begin{aligned} y(t) &= \sum_k c h(t-k) \\ &= c a_1 t^2 - c b_1 t + 0 \\ &\quad + c a_0 t^2 + c \\ &\quad + c a_1 t^2 + c b_1 t + 0 \\ &= c + c(2a_1 + a_0) t^2 \end{aligned}$$

This yields a linear constraint.

$$0 = 2a_1 + a_0$$

Lines



$$h(t) = \begin{cases} & & & & 0 & & t & < -1.5 \\ a_1 (t+1)^2 & - & b_1 (t+1) & + & 0 & -1.5 < & t & < -0.5 \\ a_0 t^2 & & & + & 1 & -0.5 < & t & < 0.5 \\ a_1 (t-1)^2 & + & b_1 (t-1) & + & 0 & 0.5 < & t & < 1.5 \\ & & & & 0 & & t & > 1.5 \end{cases}$$

$$\begin{aligned} y(t) &= \sum_k k h(t-k) \\ &= \begin{array}{ll} & (1)\, h(t-1) \\ + & (0)\, h(t) \\ + & (-1)\, h(t+1) \end{array} = \begin{array}{rlrlrl} & a_1 t^2 & - & b_1 t & + & 0 \\ + & 0 & + & 0 & + & 0 \\ - & a_1 t^2 & - & b_1 t & - & 0 \end{array} \\ &= (-2b_1)\, t + 0 \\ -\frac{1}{2} &= b_1 \end{aligned}$$

Likewise with a line.

Solving

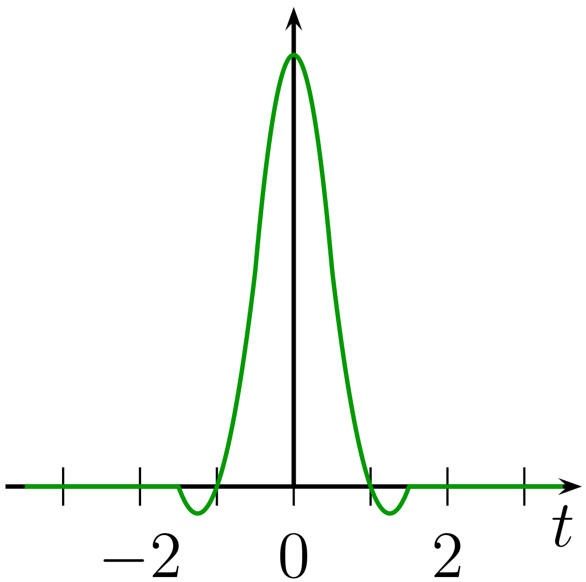
We now have four equations, in three unknowns:

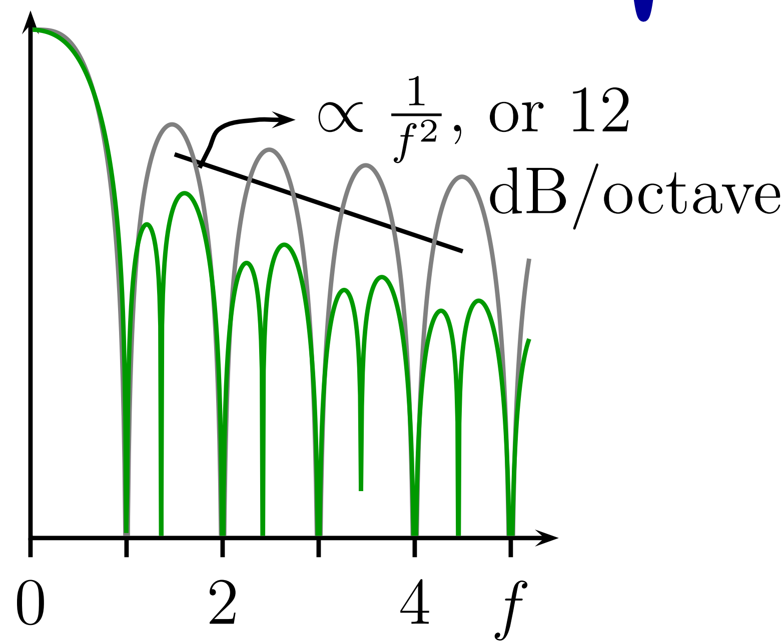
$$
\begin{array}{rcrcrcrl}
1 & = & -\frac{1}{4}a_0 & + & \frac{1}{4}a_1 & - & \frac{1}{2}b_1 & \text{(Continuity)} \\
0 & = & & & \frac{1}{4}a_1 & + & \frac{1}{2}b_1 & \text{(Continuity)} \\
0 & = & a_0 & + & 2a_1 & & & \text{(Constants)} \\
-\frac{1}{2} & = & & & & & b_1 & \text{(Lines)}
\end{array}
$$

The good news is that one solution fits all,

$$
\begin{aligned}
a_0 &= -2 \\
a_1 &= 1 \\
b_1 &= -\frac{1}{2}
\end{aligned}
$$

Continuous Quadratic





Criteria

- Continuous interpolator
- Constants $\Rightarrow$ constants
- Lines $\Rightarrow$ lines

How about other quadratics and criteria?

Homework

1. Let's apply our new quadratic filter to the following functions. What's the result of upsampling via this quadratic filter?
   - $[-2, -1, 0, 1, 2, 3, \ldots]$
   - $[4, 1, 0, 1, 4, 9, \ldots]$
   - $[1, -1, 1, -1, 1, -1, \ldots]$
   - $[1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1, \ldots]$
   - $\left[1, \frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, -1, -\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 1, \ldots\right]$
2. Using matlab, plot the root function and the results for each of these. How close did we come?
3. Is this filter better or worse than the last one?
4. What set of convolutions defines this filter?

Answers

1. Let's apply our new quadratic filter to the following functions. What's the result of upsampling via this quadratic filter?
   - $[-2, -1, 0, 1, 2, 3, \ldots]$
   - $[4, 1, 0, 1, 4, 9, \ldots]$

   These two are left as an exercise for the student.

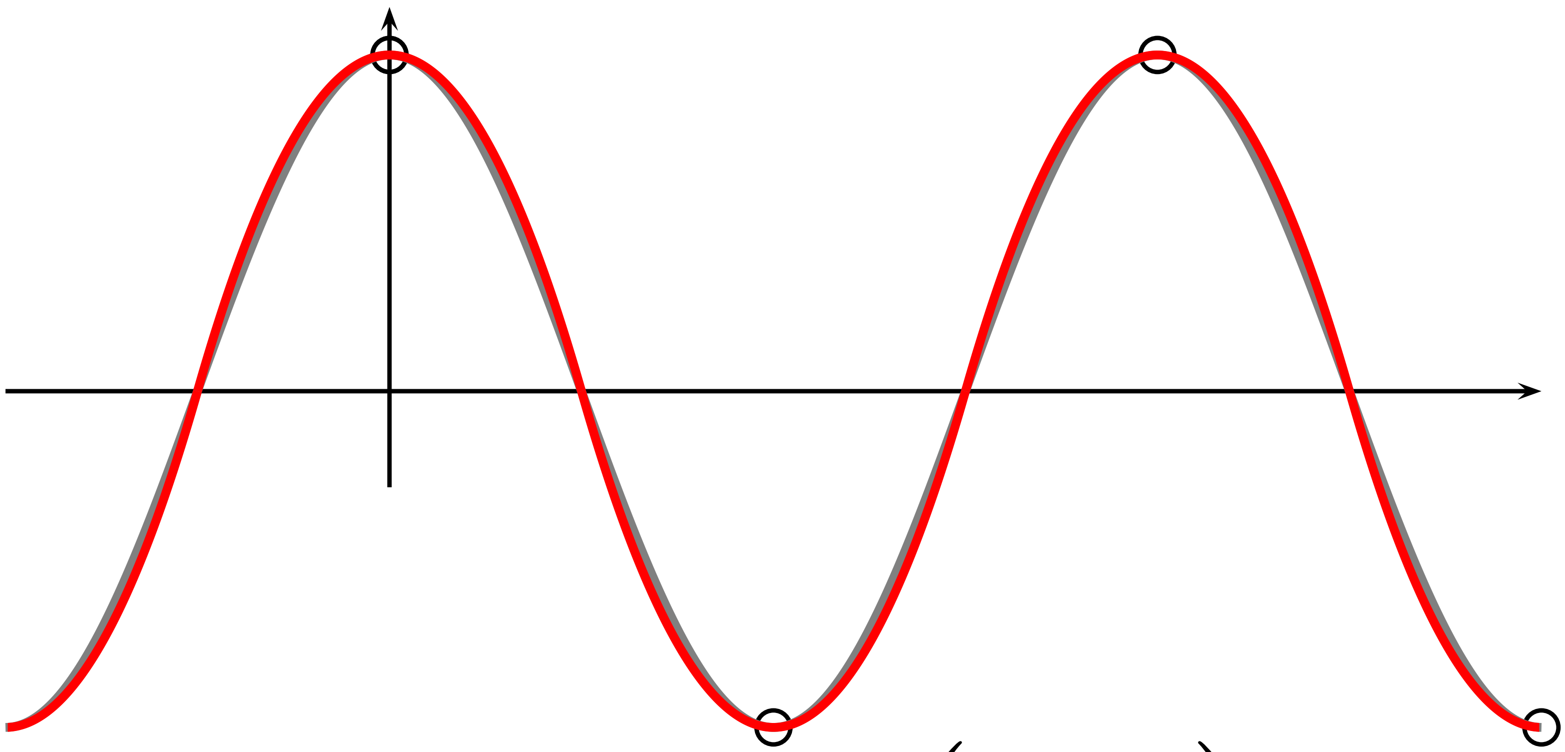
Answers

1. Let's apply our new quadratic filter to the following functions. What's the result of upsampling via this quadratic filter?

   - $x[n] = \cos(2\pi f n)$

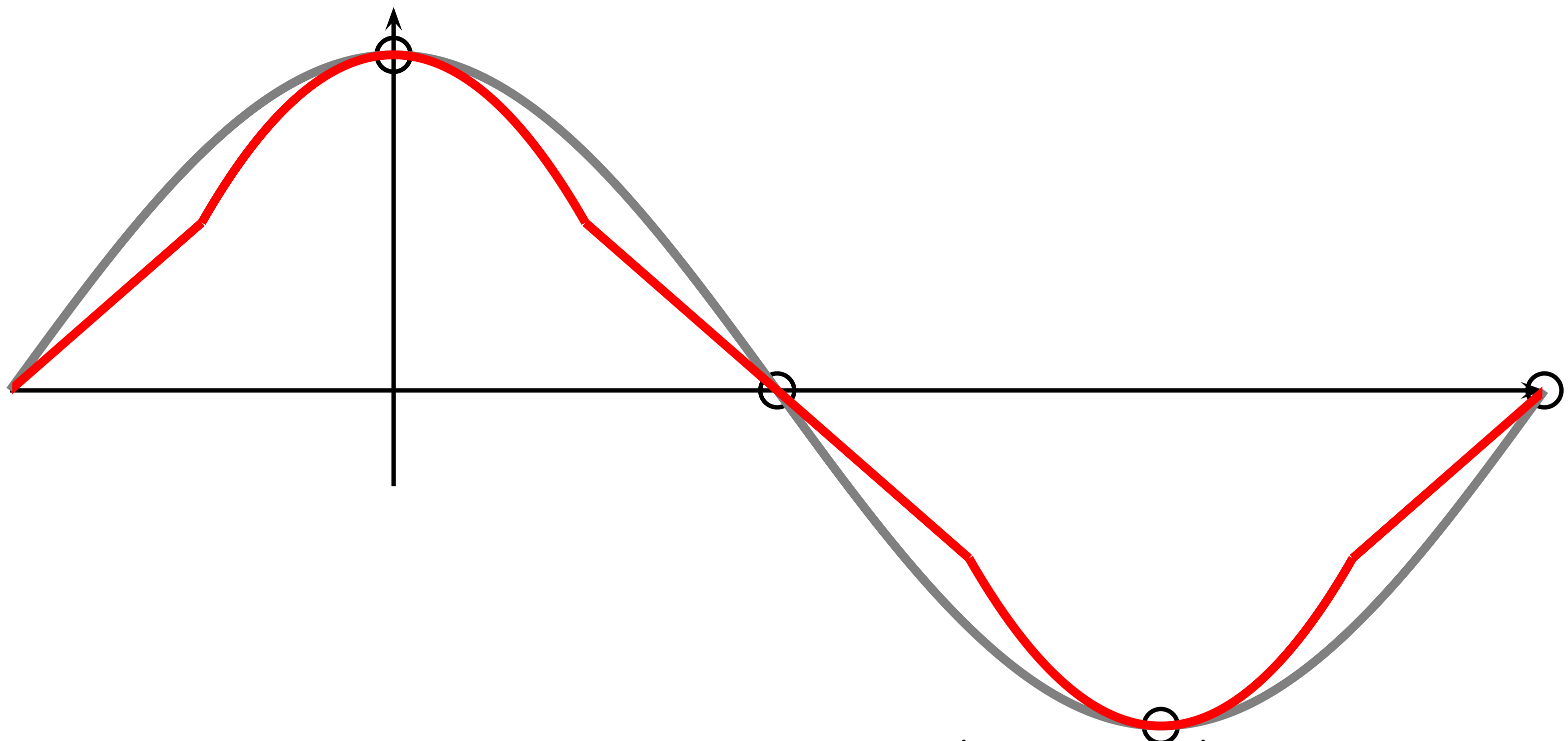
$$
\begin{aligned}
a[n] &= 2\left[\cos(2\pi f) - 1\right]\cos(2\pi f n) \\
b[n] &= -\sin(2\pi f)\sin(2\pi f n) \\
c[n] &= \cos(2\pi f n)
\end{aligned}
$$

Examples



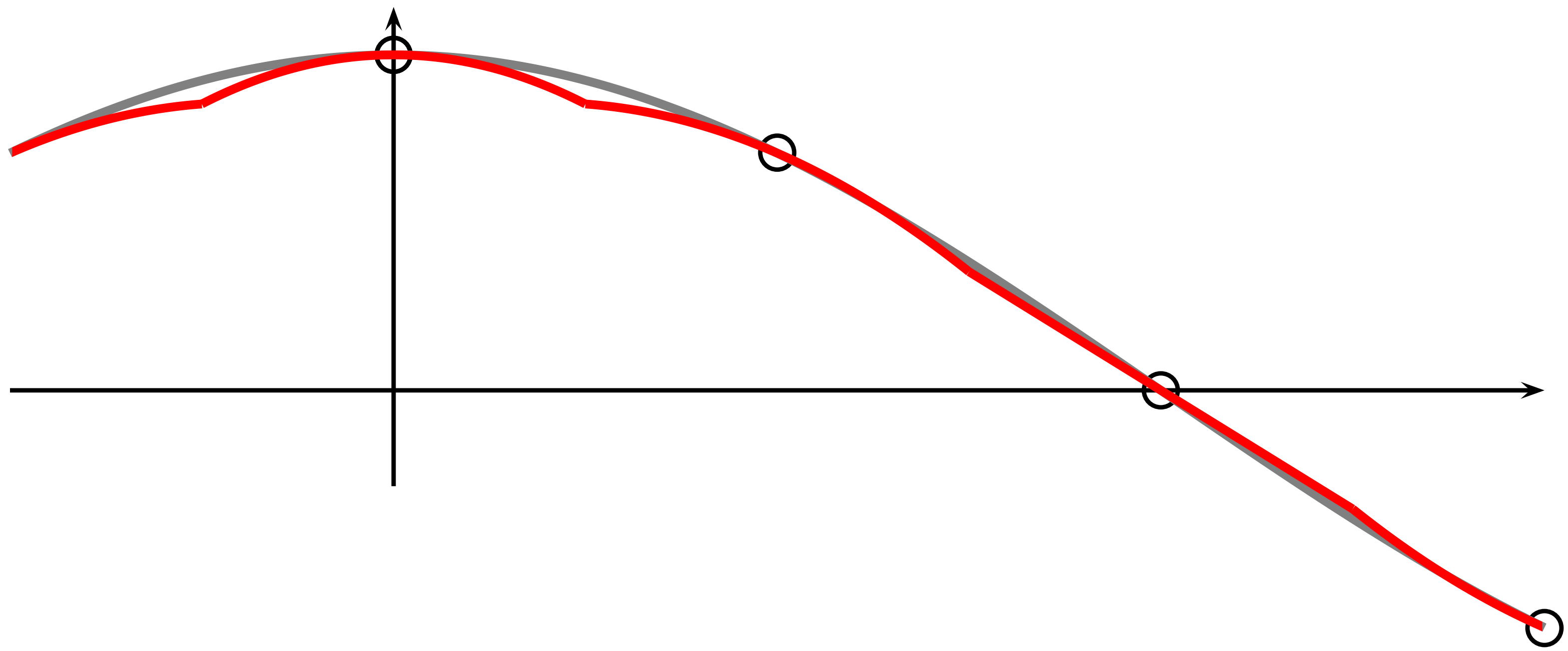
$$x[n] = \cos\left(2\pi\frac{n}{2}\right)$$

Examples



$$x[n] = \cos\left(2\pi \frac{n}{4}\right)$$

Examples



$$x[n] = \cos\left(2\pi \frac{n}{8}\right)$$

Answers

4. What set of convolutions defines this filter?

Start with our definition for applying an interpolator,

$$y(t) = \sum_n x[n]\, h(t-n),$$

and expand it out near $t = 0$,

$$\begin{aligned} = \quad & x[-1] & h(t+1) \\ + \quad & x[0] & h(t) \\ + \quad & x[1] & h(t-1). \end{aligned}$$

Fill in the definition of our quadratic,

$$\begin{aligned} = \quad & x[-1]\,[& a_1 t^2 & + & b_1 t & & 0 &] \\ + \quad & x[0]\,[& a_0 t^2 & & & & +1 &] \\ + \quad & x[1]\,[& a_1 t^2 & - & b_1 t & & 0 &], \ldots \end{aligned}$$

Answers

4. What set of convolutions defines this filter? (Continued . . .)

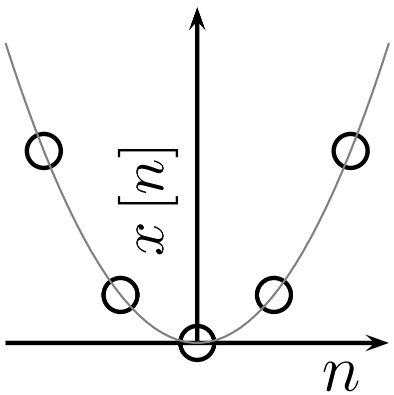
. . . and apply the numbers we have for our coefficients.

$$\begin{array}{rcrl} y(t) & = & x[-1] & \left[\begin{array}{ccccc} t^2 & - & \frac{1}{2}t & & 0 \end{array}\right] \\ & + & x[0] & \left[\begin{array}{ccccc} -2t^2 & & & & +1 \end{array}\right] \\ & + & x[1] & \left[\begin{array}{ccccc} t^2 & + & \frac{1}{2}t & & 0 \end{array}\right]. \end{array}$$

At this point, you can read the convolutions off of the columns. They are,

$$\begin{array}{rcccl} a[n] & = & x[n] & \otimes & \left\{ \begin{array}{ccc} 1 & -2 & 1 \end{array} \right\}, \\ b[n] & = & x[n] & \otimes & \left\{ \begin{array}{ccc} \dfrac{1}{2} & 0 & -\dfrac{1}{2} \end{array} \right\}, \\ \text{and } c[n] & = & x[n] & \otimes & \left\{ \begin{array}{ccc} 0 & 1 & 0 \end{array} \right\}. \end{array}$$

Longer Quadratic

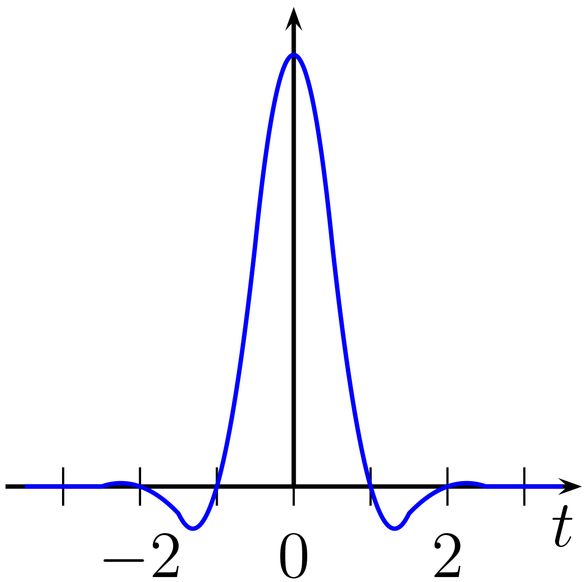


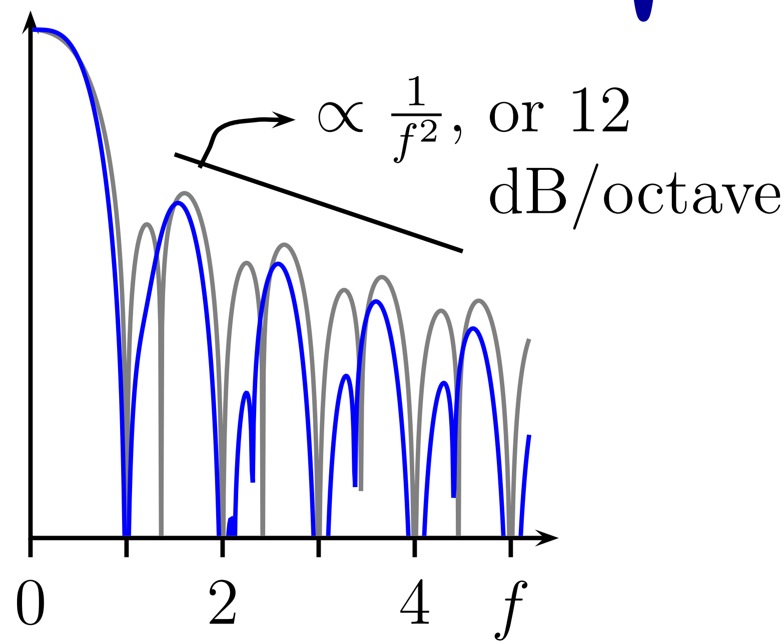
$$h(t) = \begin{cases} 0 & t < -2.5 \\ a_2 (t+2)^2 - b_2 (t+2) + 0 & -2.5 < t < -1.5 \\ a_1 (t+1)^2 - b_1 (t+1) + 0 & -1.5 < t < -0.5 \\ a_0 t^2 + 1 & -0.5 < t < 0.5 \\ a_1 (t-1)^2 + b_1 (t-1) + 0 & 0.5 < t < 1.5 \\ a_2 (t-2)^2 + b_2 (t-2) + 0 & 1.5 < t < 2.5 \\ 0 & t > 2.5 \end{cases}$$

$$\text{Continuity} \left\{ \begin{array}{rcl} 1 & = & -a_0 \left(\frac{1}{2}\right)^2 + a_1 \left(-\frac{1}{2}\right)^2 + b_1 \left(-\frac{1}{2}\right) \\ 0 & = & -a_1 \left(\frac{1}{2}\right)^2 - b_1 \left(\frac{1}{2}\right) + a_2 \left(-\frac{1}{2}\right)^2 + b_2 \left(-\frac{1}{2}\right) \\ 0 & = & a_2 \left(\frac{1}{2}\right)^2 + b_2 \left(\frac{1}{2}\right) \end{array} \right.$$

$$\begin{array}{rrcl} \text{Constants} & 0 & = & a_0 + 2a_1 + 2a_2 \\ \text{Lines} & 1 & = & -2b_1 - 4b_2 \\ \text{Quadratic} & 1 & = & 2a_1 + 8a_2 \end{array}$$

Longer Quadratic



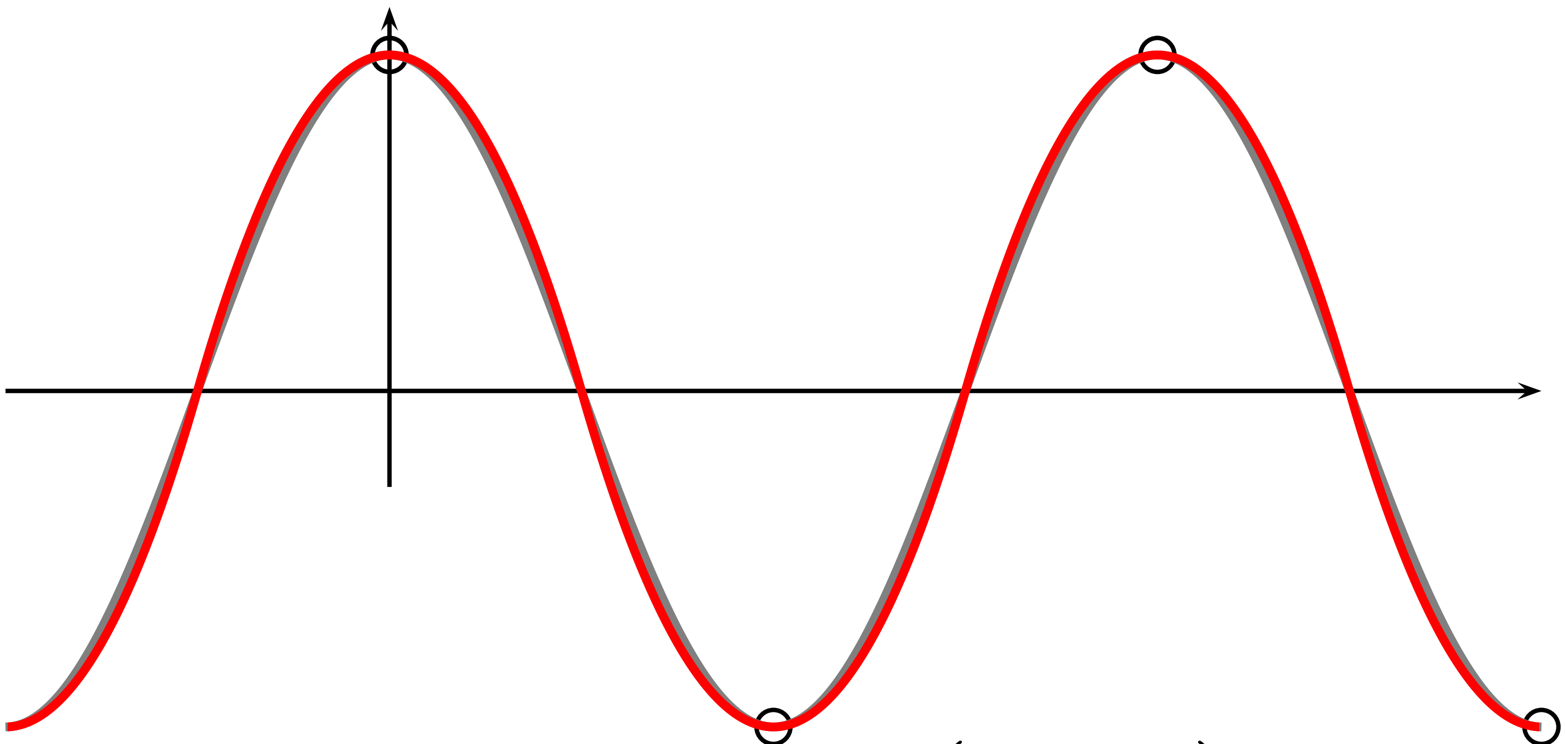


Criteria

- Continuous interpolator
- Constants $\Rightarrow$ constants
- Lines $\Rightarrow$ lines
- Quadratics $\Rightarrow$ quadratics

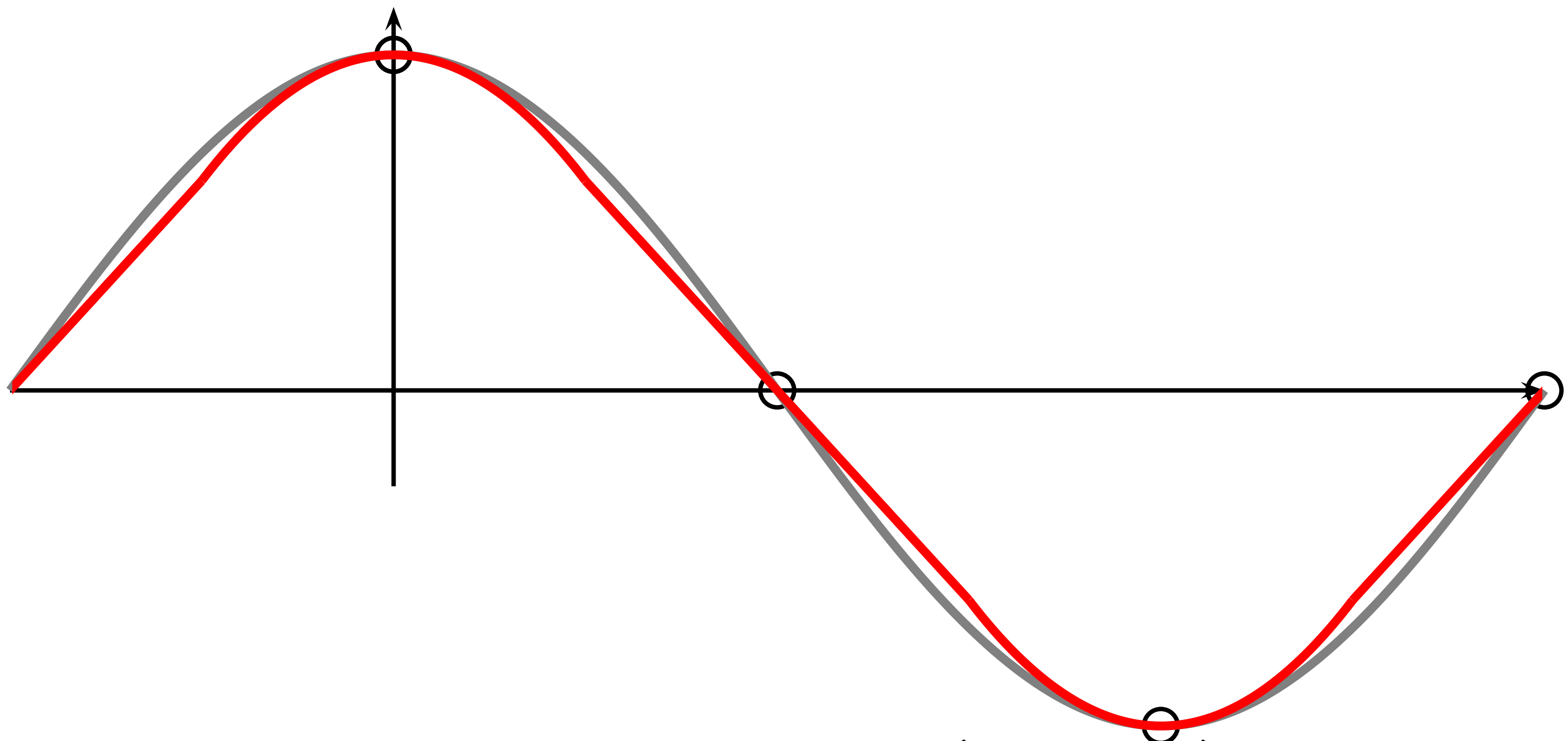
$$\begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} -28 \\ 16 \\ 10 \\ -2 \\ -1 \end{bmatrix}$$

Examples



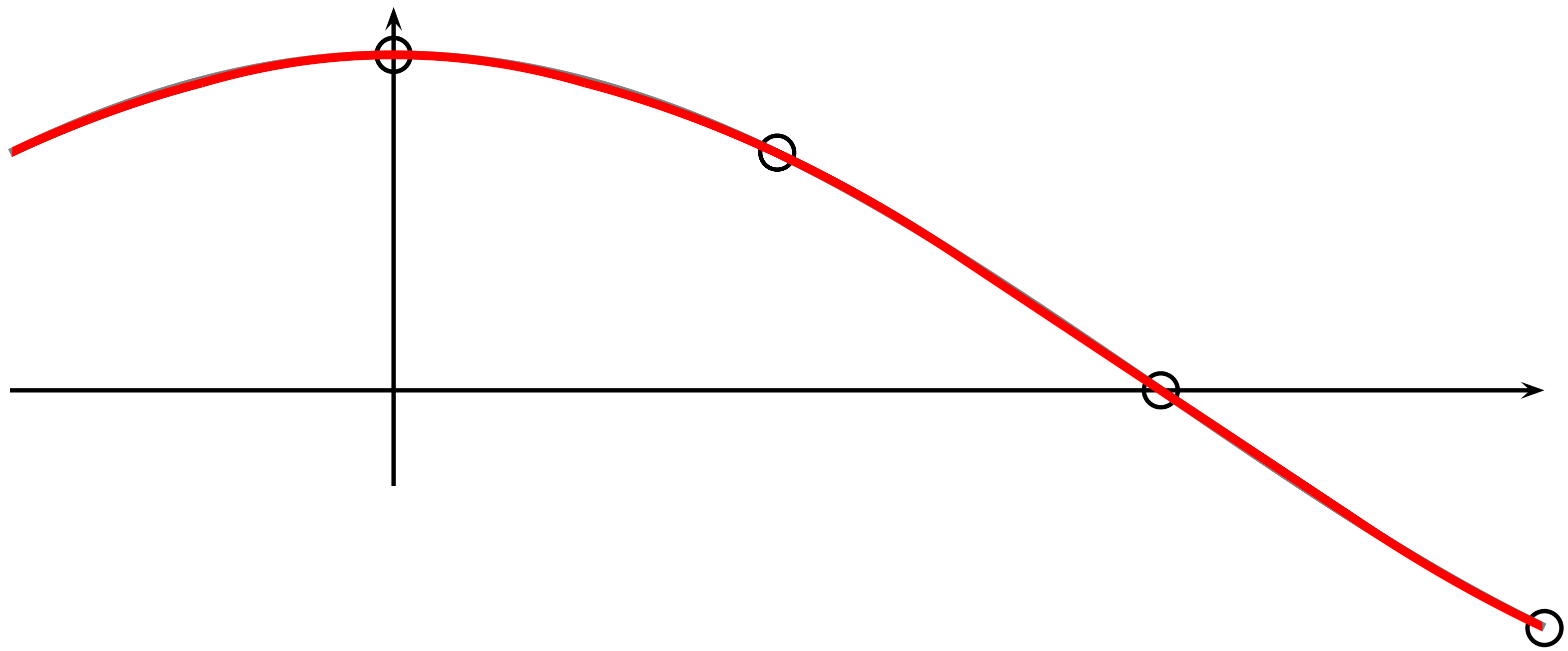
$$x[n] = \cos\left(2\pi\frac{n}{2}\right)$$

Examples



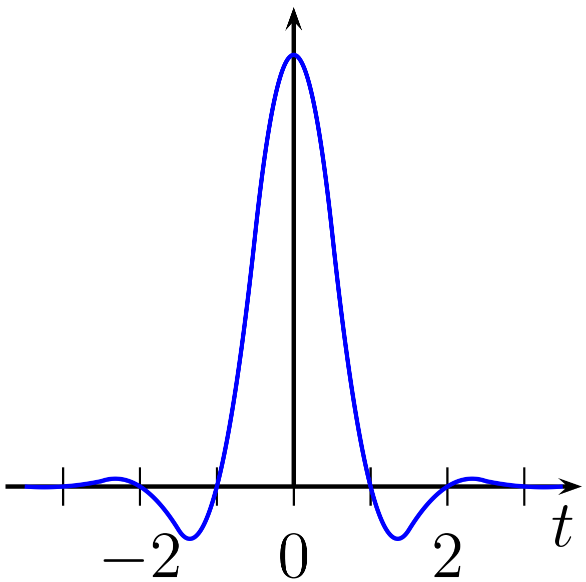
$$x[n] = \cos\left(2\pi \frac{n}{4}\right)$$

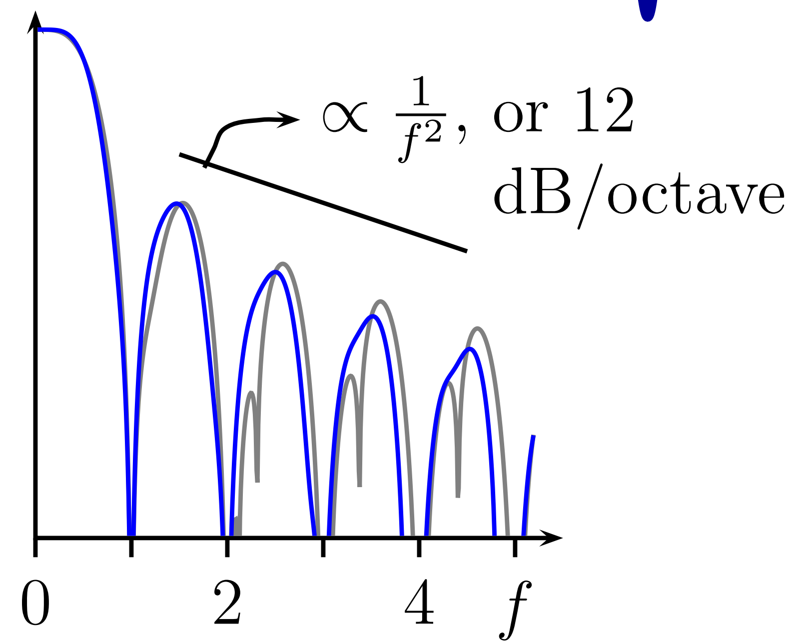
Examples



$$x[n] = \cos\left(2\pi \frac{n}{8}\right)$$

My Favorite



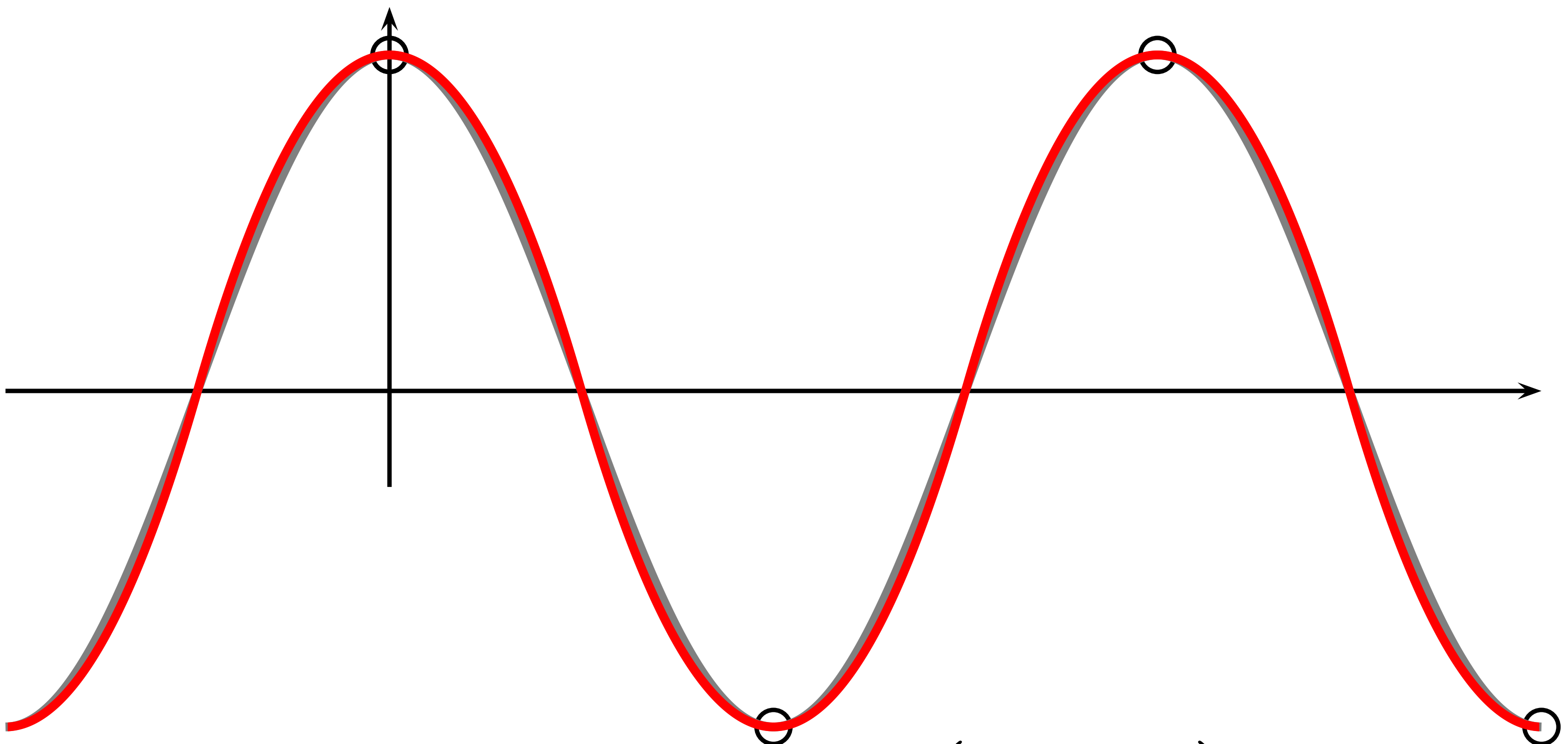


Criteria

- Continuous interpolator
- Constants $\Rightarrow$ constants
- Lines $\Rightarrow$ lines
- Quadratics $\Rightarrow$ quadratics
- Nearly continuous derivative

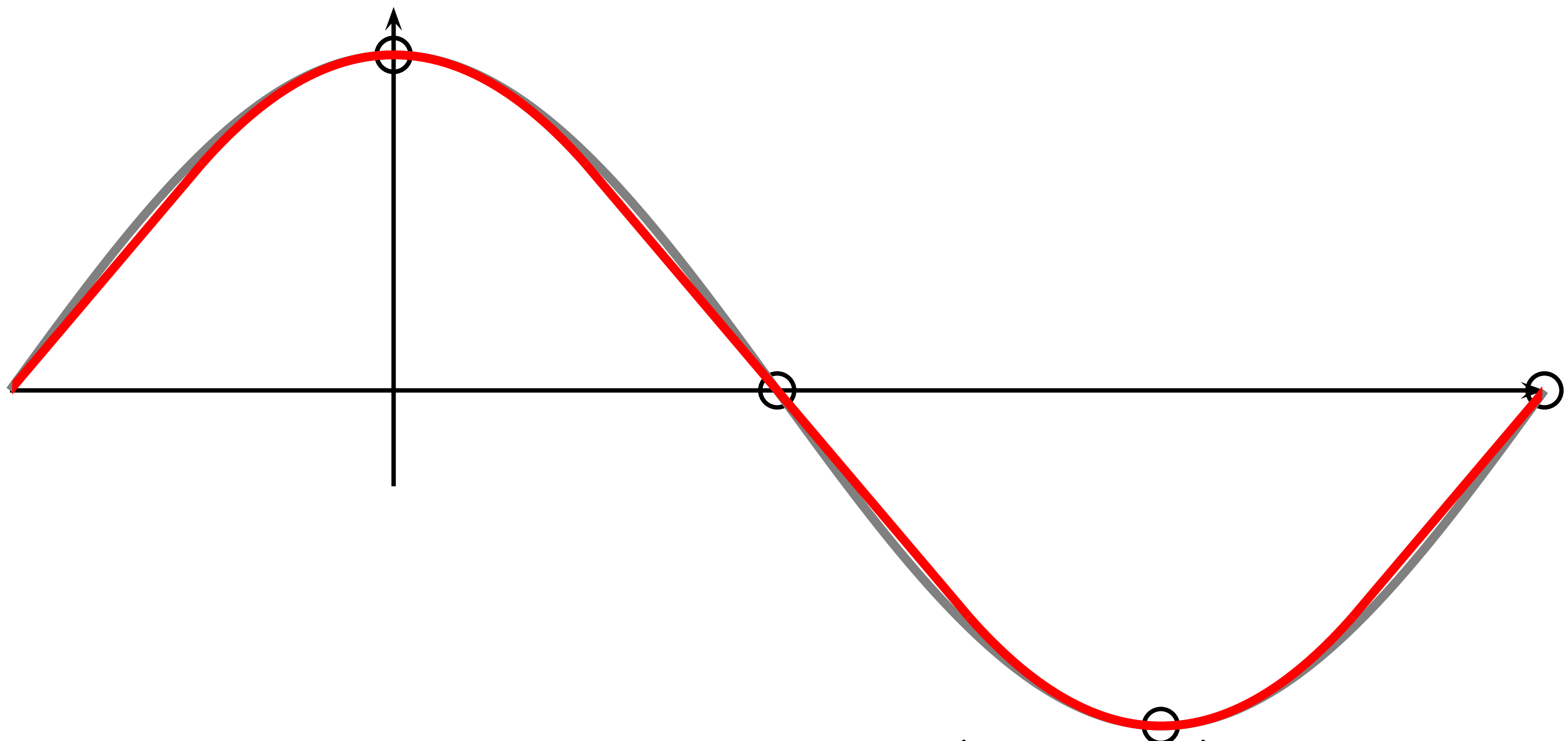
$$\begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \end{bmatrix} = \frac{1}{80} \begin{bmatrix} -132 \\ 78 \\ 55 \\ -14 \\ -9 \\ 2 \\ 1 \end{bmatrix}$$

Examples



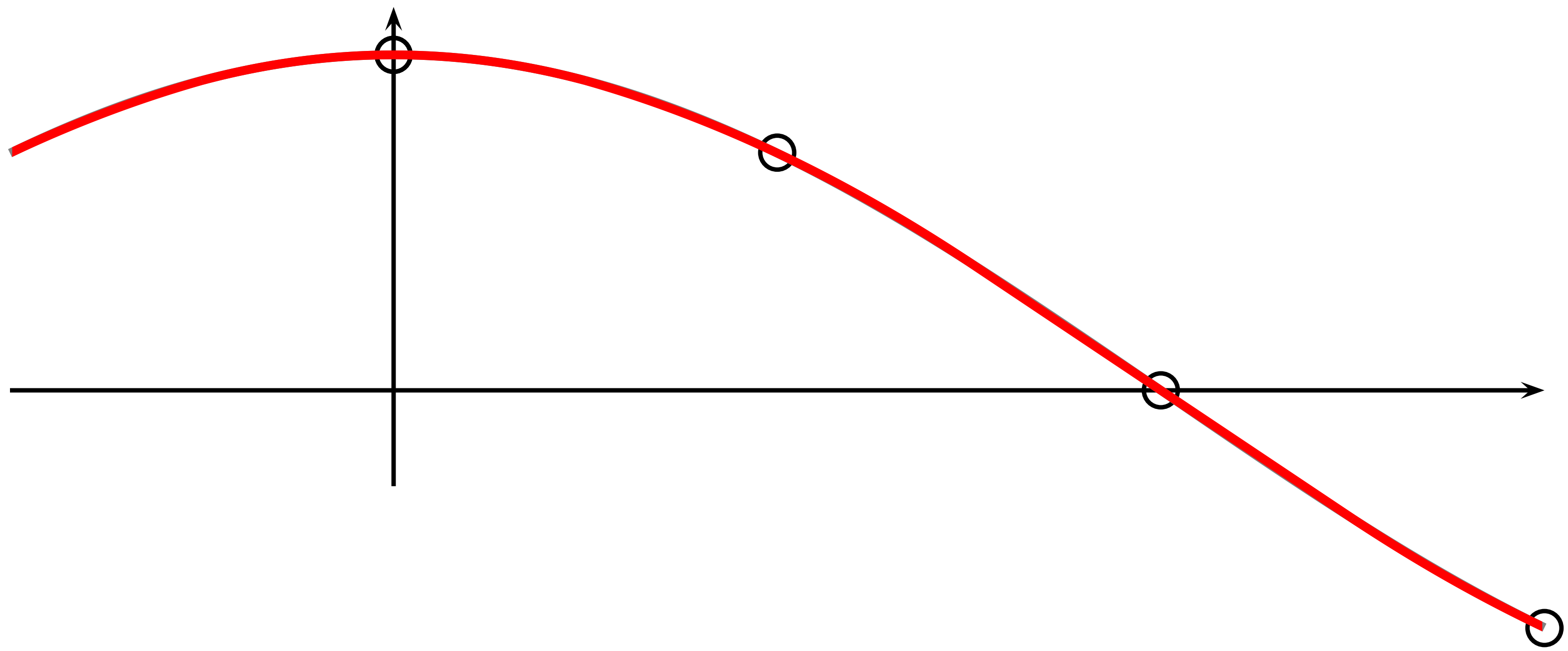
$$x[n] = \cos\left(2\pi \frac{n}{2}\right)$$

Examples



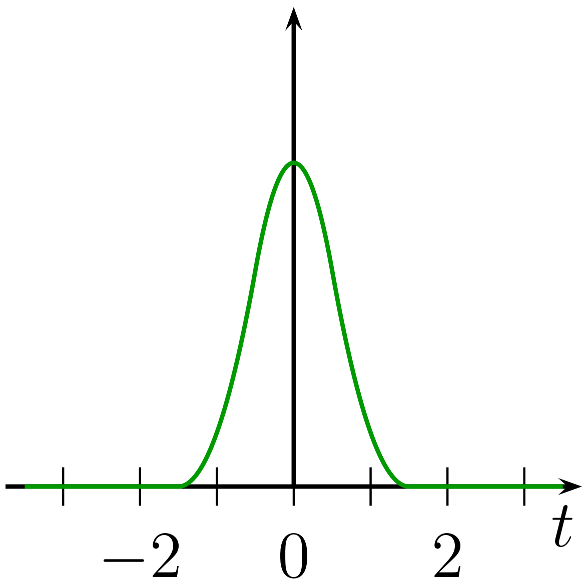
$$x[n] = \cos\left(2\pi \frac{n}{4}\right)$$

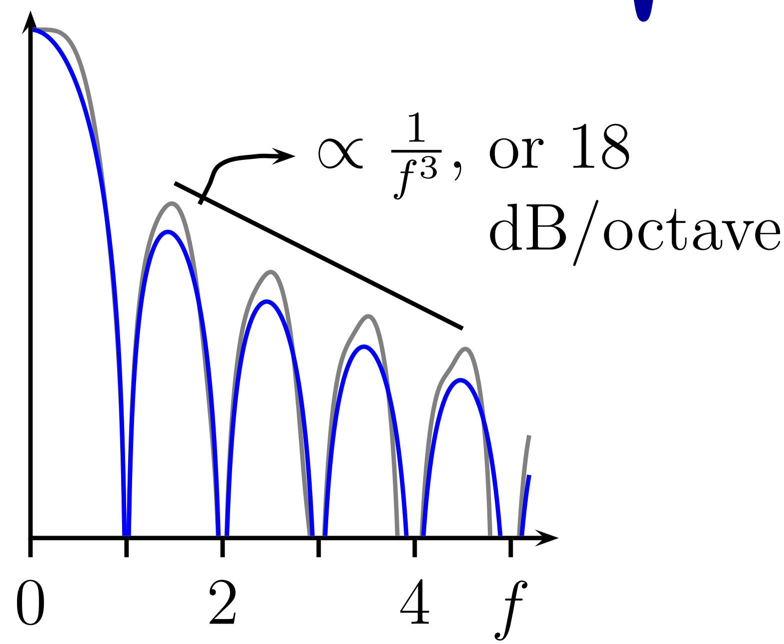
Examples



$$x[n] = \cos\left(2\pi \frac{n}{8}\right)$$

Smoothest





Fastest decaying quadratic!

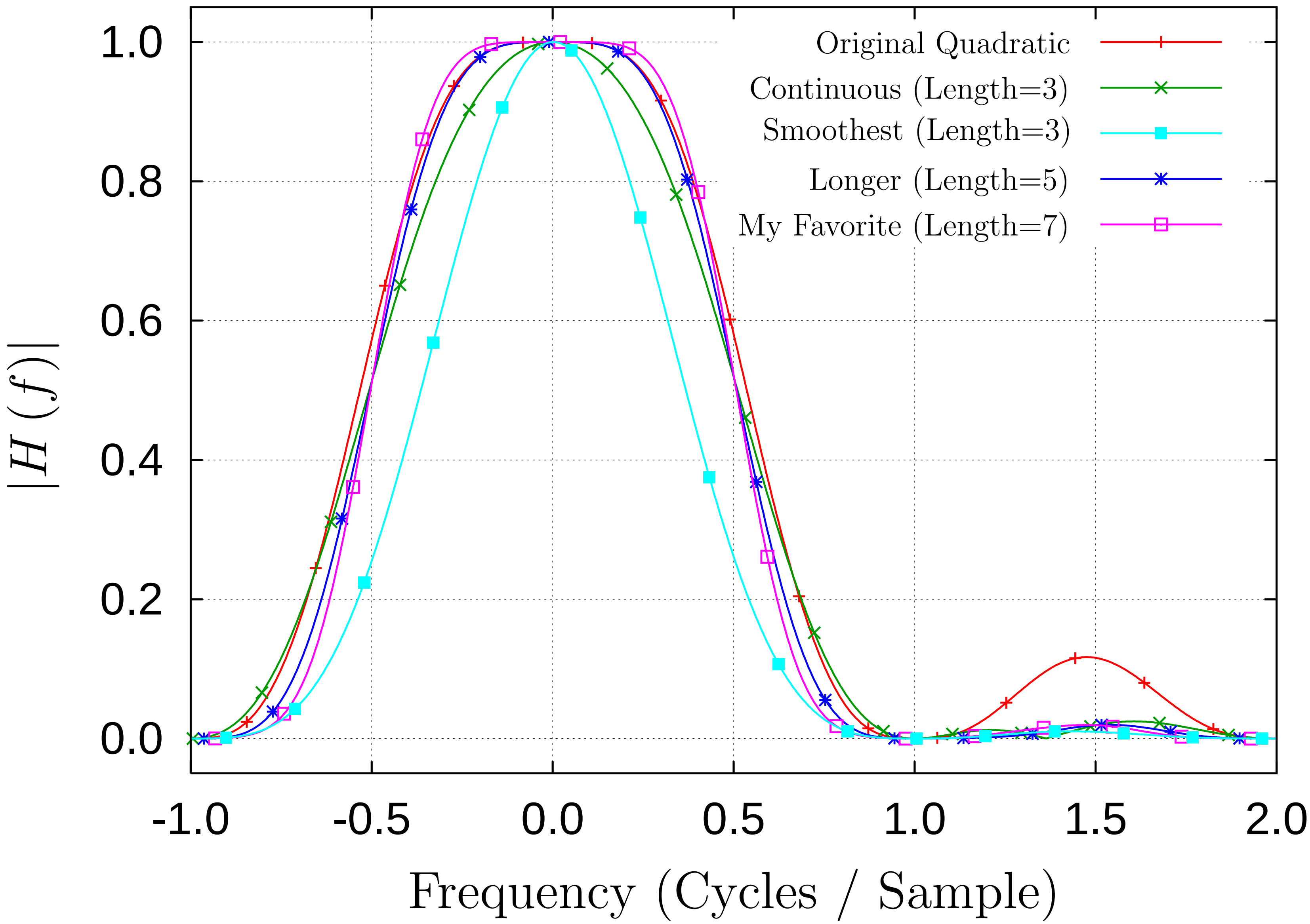
Criteria

- Continuous ~~interpolator~~
- Constants $\Rightarrow$ constants
- Lines $\Rightarrow$ lines
- Continuous derivative

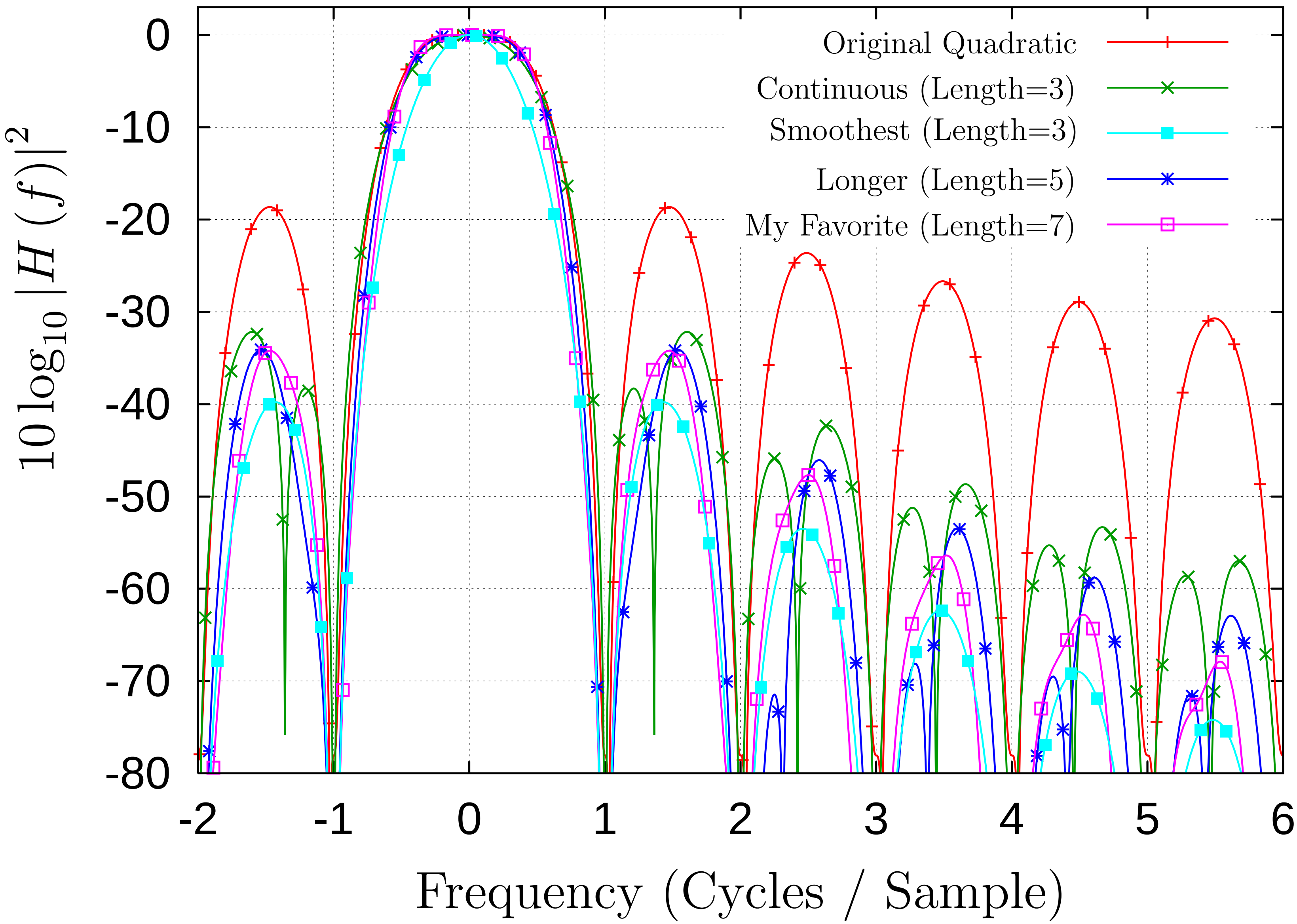
$$h(t) = \begin{cases} -t^2 + \frac{3}{4} & |t| < \frac{1}{2} \\ \frac{1}{2}\left(t - \frac{3}{2}\right)^2 & \frac{1}{2} < |t| < \frac{3}{2} \\ 0 & \text{Otherwise} \end{cases}$$

One minor problem: this quadratic isn't an interpolator.

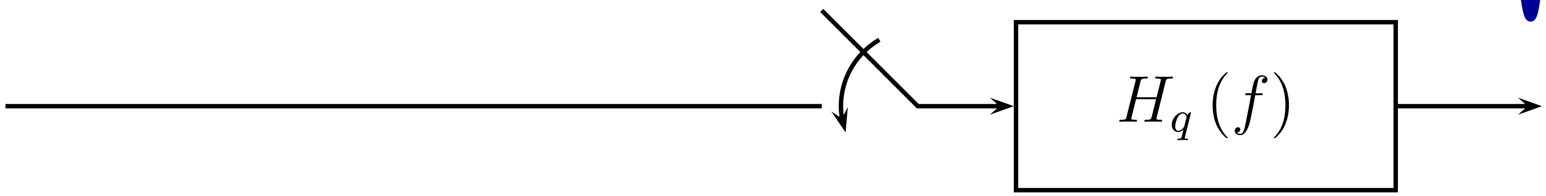
Frequency Responses

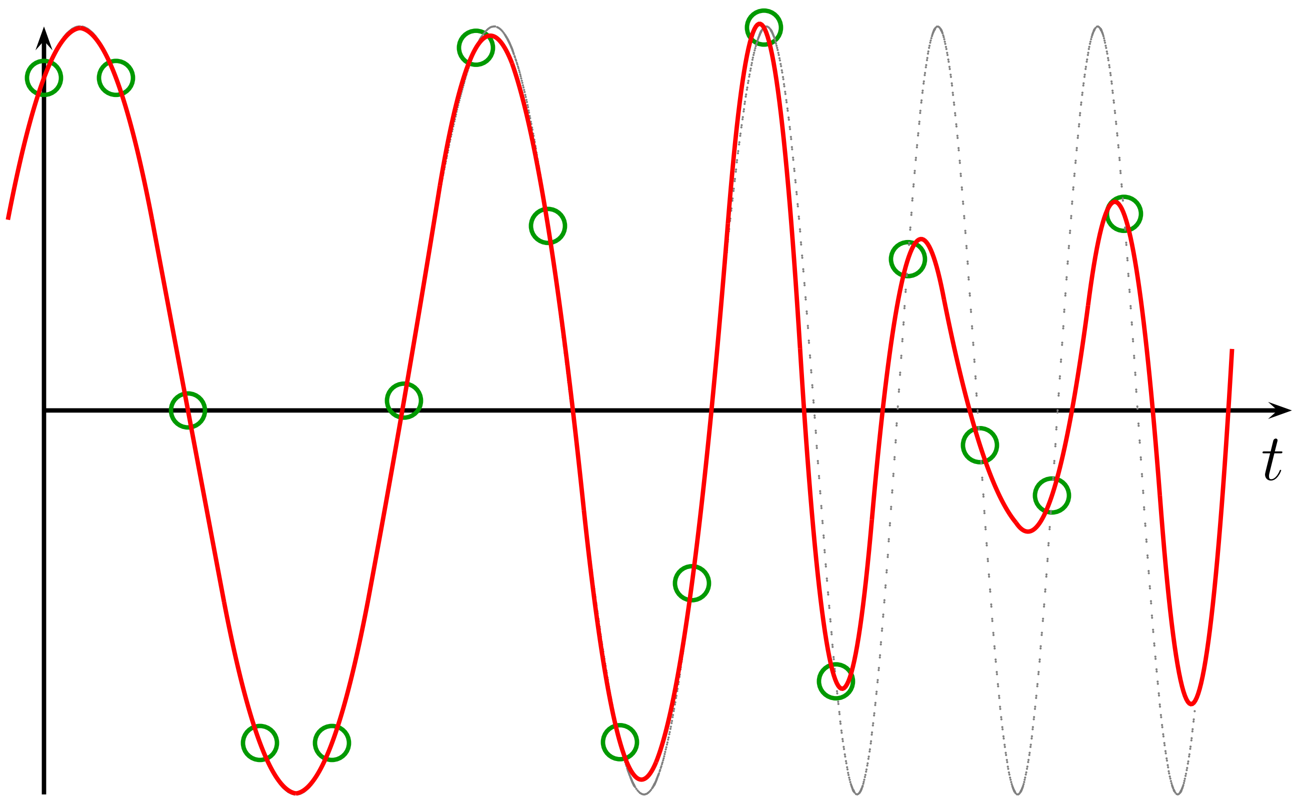


Log Responses



Final Result





Conclusion: This works fine for the low frequency signals. What about higher frequency signal components?

Discussion

1. What's the highest frequency that can be interpolated via a quadratic interpolator?

2. All of our polynomial interpolators have interpolated from the closest point, rather than between pairs of points. Is there a difference? Which is better?

3. What about spline interpolators? How are they different?

4. Are splines better? worse?

5. Which is more appropriate? Spline interpolation or piece–wise polynomial filter interpolators?

Discussion

1. What's the highest frequency that can be interpolated via a quadratic interpolator?

   It depends. What is your quality cutoff?

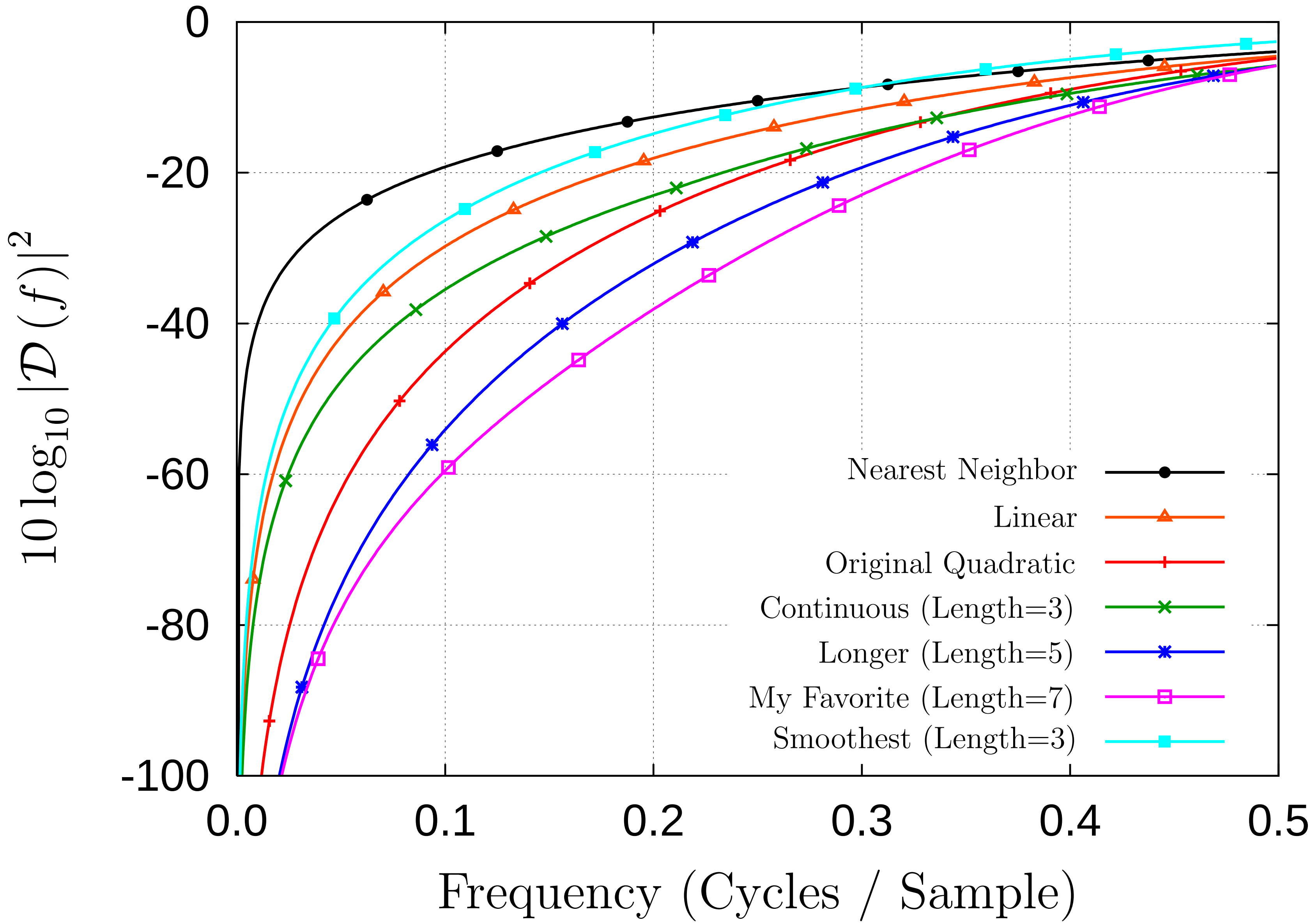
   Let's define a quality measure, and call it the distortion:

   $$\mathcal{D}(f) \triangleq \max_{k=1,\infty} \left[\; |H(f) - 1|, \;\; |H(k-f)|, \;\; |H(k+f)| \; \right]$$

   This measures the maximum absolute error in our measure.

   We can now plot this and compare interpolators.

Distortion



0
-20
-40
-60
-80
-100
0.0
0.1
0.2
0.3
0.4
0.5
$10 \log_{10} |\mathcal{D}(f)|^2$
Frequency (Cycles / Sample)
Nearest Neighbor
Linear
Original Quadratic
Continuous (Length=3)
Longer (Length=5)
My Favorite (Length=7)
Smoothest (Length=3)

Discussion

2. All of our polynomial interpolators have interpolated from the closest point, rather than between pairs of points. Is there a difference? Which is better?

   Left as an exercise to the student .... From my experience, interpolating near the closest point is best for quadratic interpolators, but not necessarily for other orders.

3. What about spline interpolators? How are they different?

   The common spline development depends upon a matrix inverse. The piecewise polynomial interpolators presented here require convolution only. Is it possible to do spline's via convolution? Not really, since a spline's impulse response is infinite.

Discussion

4. Are splines better? worse?

   They are different. The typical spline development requires full knowledge of all of the data. These interpolators require only local knowledge. As such, polynomial interpolators are much more appropriate for infinite length signals.

5. Which is more appropriate? Spline interpolation or piece–wise polynomial filter interpolators? Piecewise polynomials are more appropriate for signal processing.

Optimal Interpolators

Optimal Interpolators

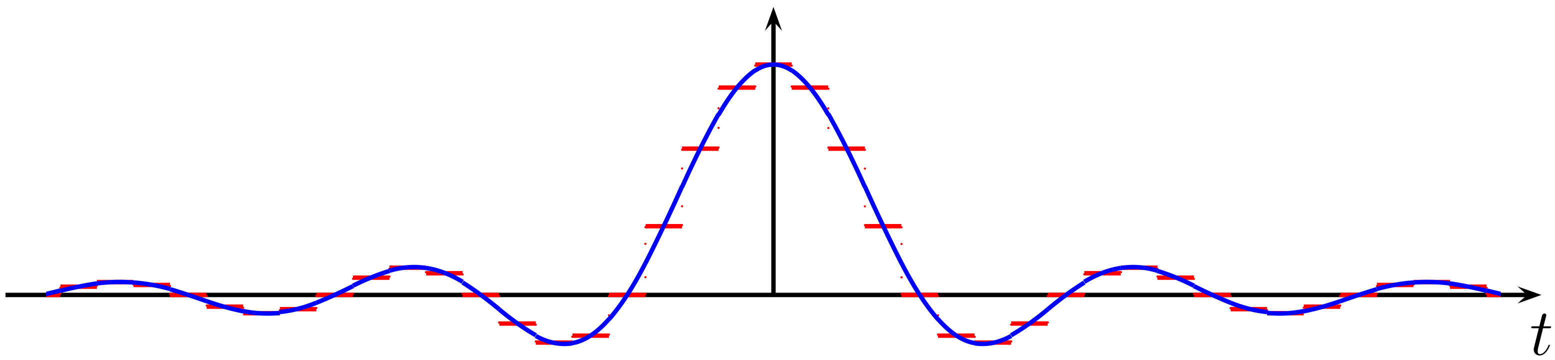
Recap:

- Want a continuous time interpolator
- Quadratics only seem to handle low frequencies well
- Rational resamplers handle all frequencies up to the Nyquist well

Let's combine these:

- Upsample with a rational resampler
- And then interpolate with a quadratic.
- We can then downsample as we see fit.

Impulse Response



- Before, the effective impulse response was only sampled.
- When mixed with a sample and hold, it produced the red result above.
- Now, the effective impulse response has been interpolated with a quadratic.
- No sample and hold circuit is required, producing the blue filter above.

Frequency Response

$$Y(f) \quad = \quad X\left(e^{j2\pi f}\right) H_u\left(e^{j2\pi \frac{f}{L}}\right) H_q\left(\frac{f}{L}\right)$$

If $f = f_o + k$, $0 < k < L$,

then $X\left(e^{j2\pi f}\right)$ aliases,

but $H_u\left(e^{j2\pi \frac{f_o}{L}}\right) \approx 0$,

so $Y(f) \approx 0$.

If $f = f_o + kL$, $k \neq 0$,

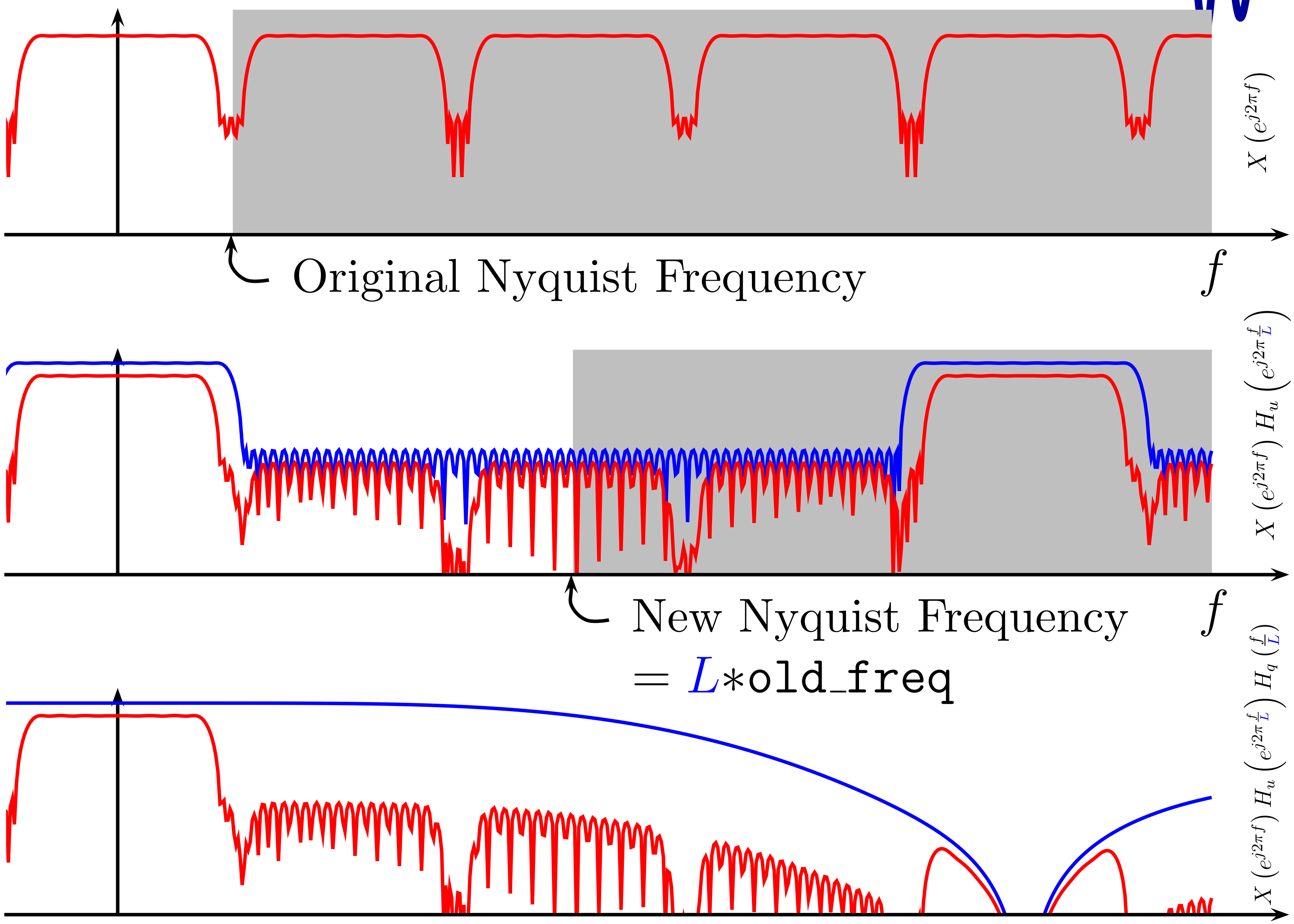
then $X\left(e^{j2\pi f}\right)$ and $H_u\left(e^{j2\pi \frac{f}{L}}\right)$ alias,

but $H_q\left(\frac{f_o}{L} + k\right) \approx 0$,

so $Y(f) \approx 0$.

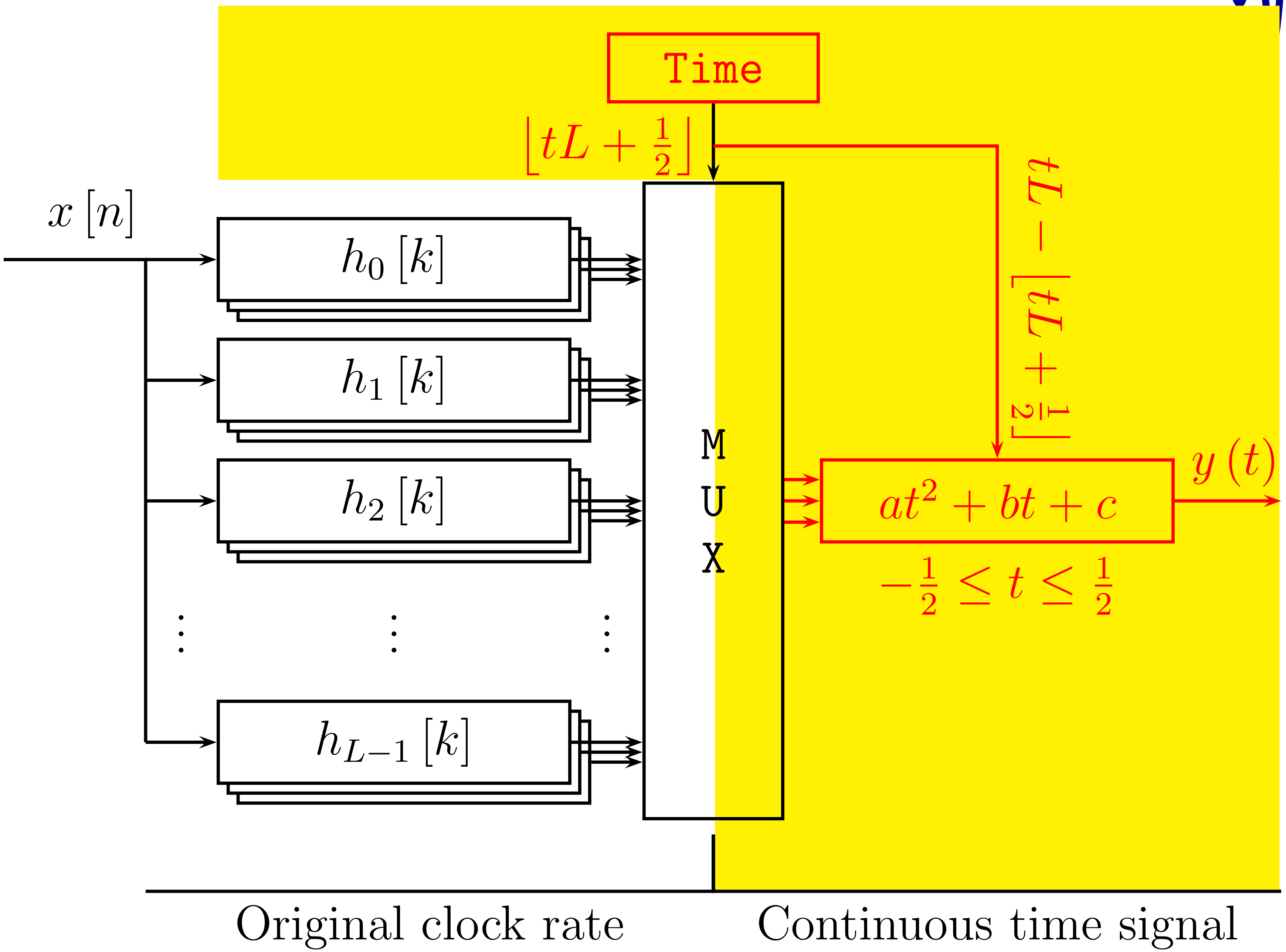
Or, at least that's the goal.

Aliasing

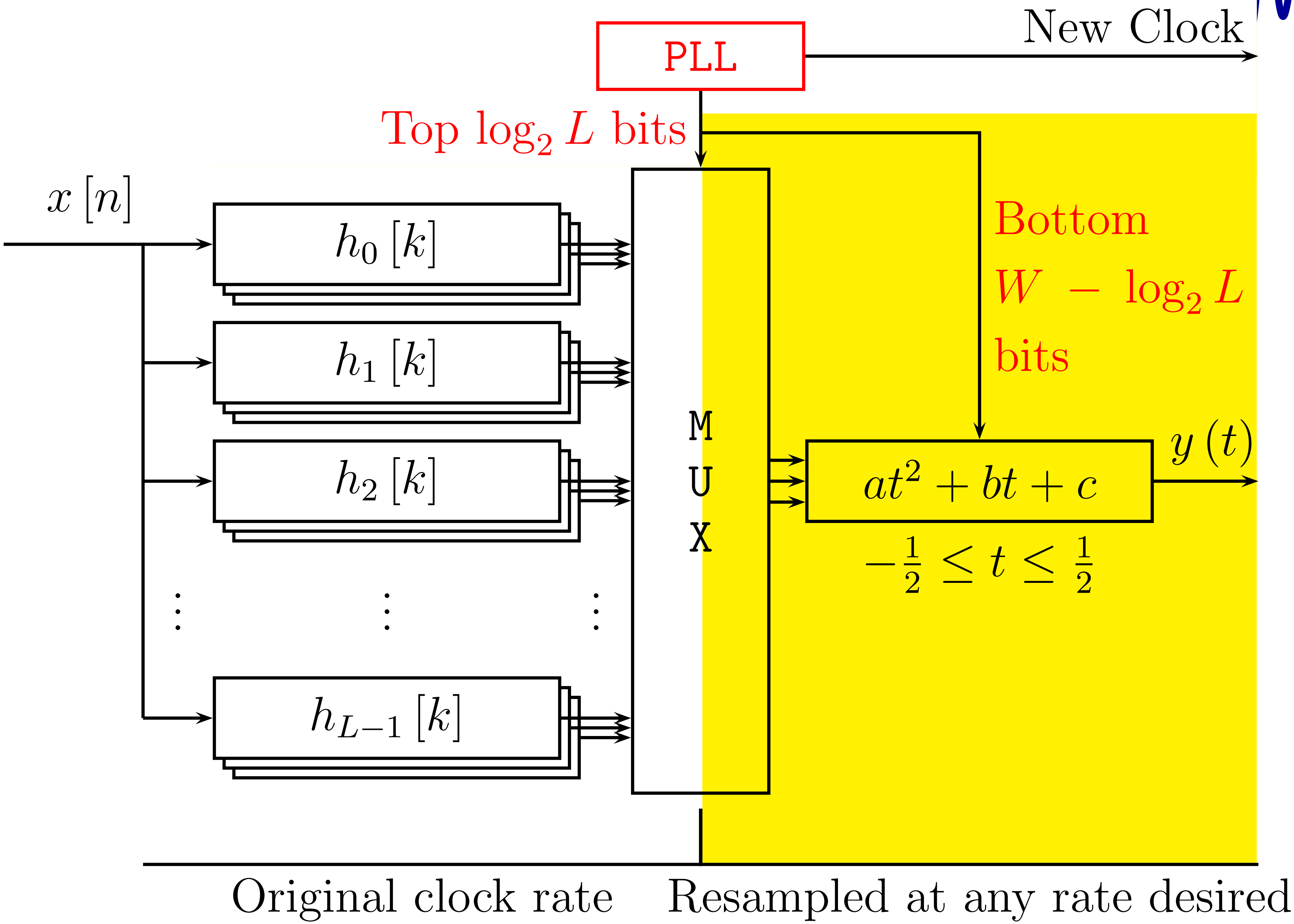


No Nyquist frequency applies. This signal is now aperiodic in frequency.

Polynomial Regenerator



Implementation



Filter Design

$H_q(f)$

- Design it to a good, smooth, lowpass filter

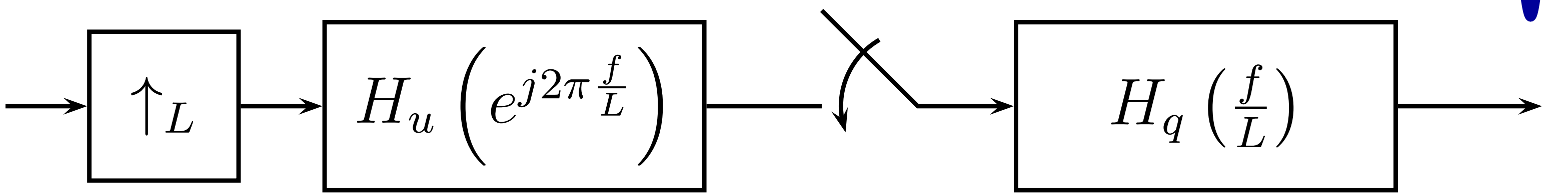
$H_u(f)$

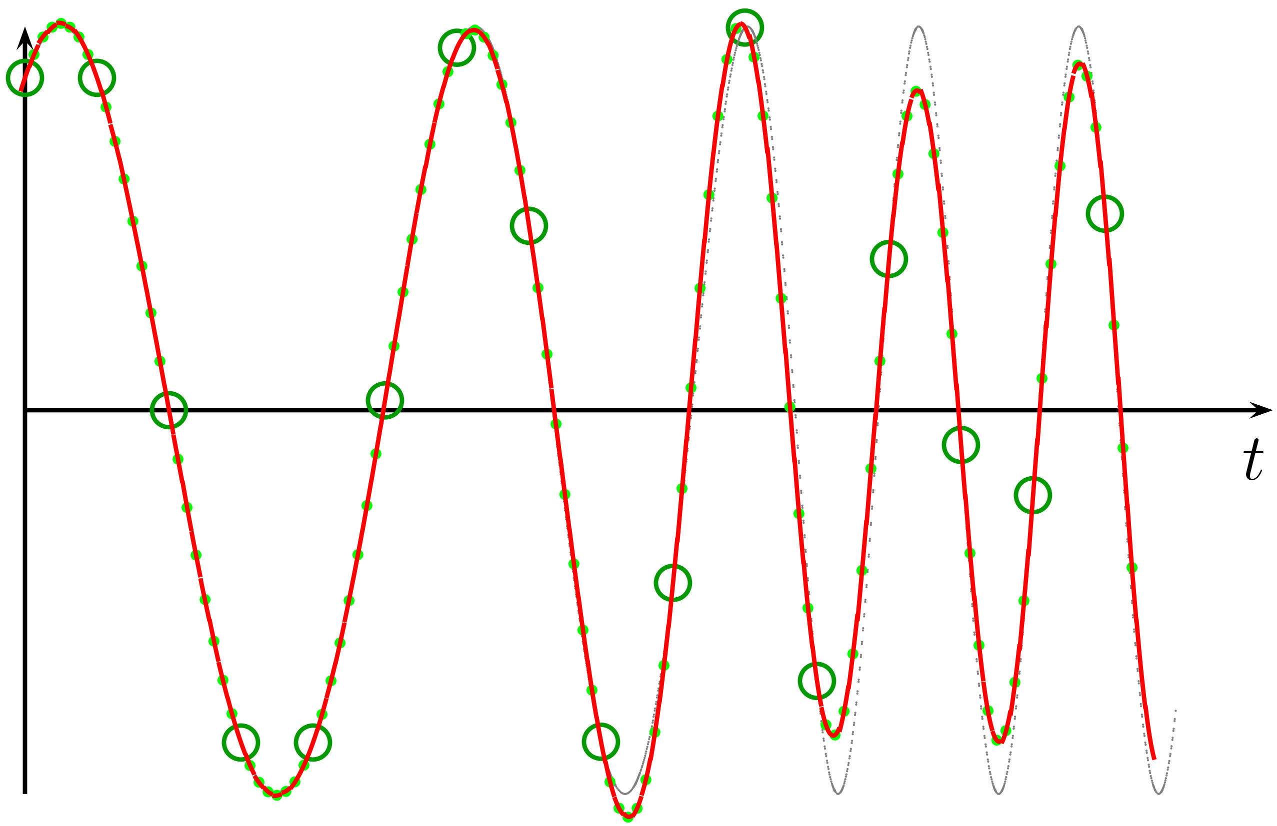
- Design the passband to equal $\frac{1}{H_q(f)}$
- Design the stopband so that $\left|H_u\left(e^{j2\pi f}\right)\right| < \delta$
- Upsample by L such that, for all passband frequencies f_p and integers $k > 1$, $\left|H_q\left(\frac{f_p}{L} + k\right) H_u\left(e^{j2\pi \frac{f_p}{L}}\right)\right| < \delta$

Conclusion

- Arbitrary resampling is possible
- Arbitrary filter design and implementation is realizable
- Upsampling filters are guaranteed optimal if so desired
- You can matched filter and resample in one step

Final Result

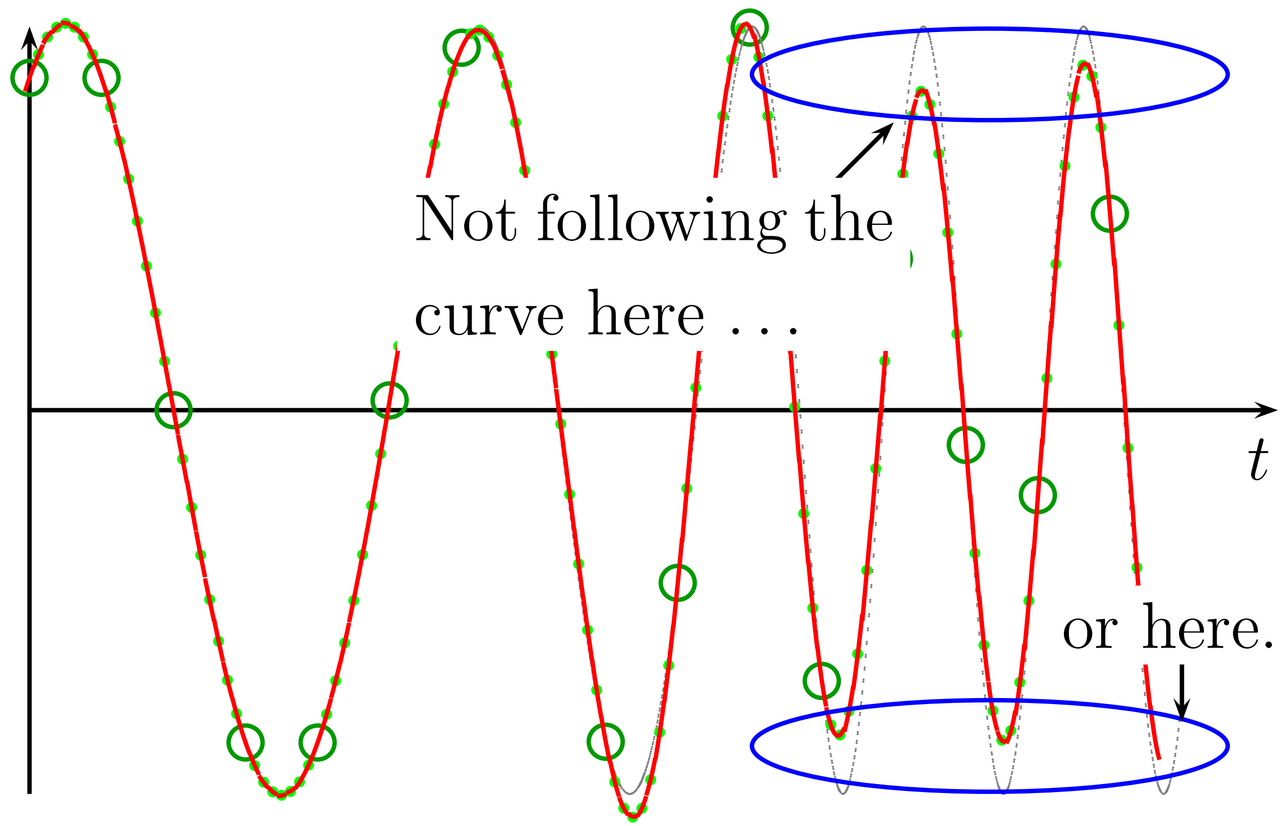




Conclusion: Not bad . . .

Discussion Questions

1. Look at that last slide again. It seems to do well for the beginning, but cannot seem to follow the curve towards the end of the cut.



Why is this?

Project

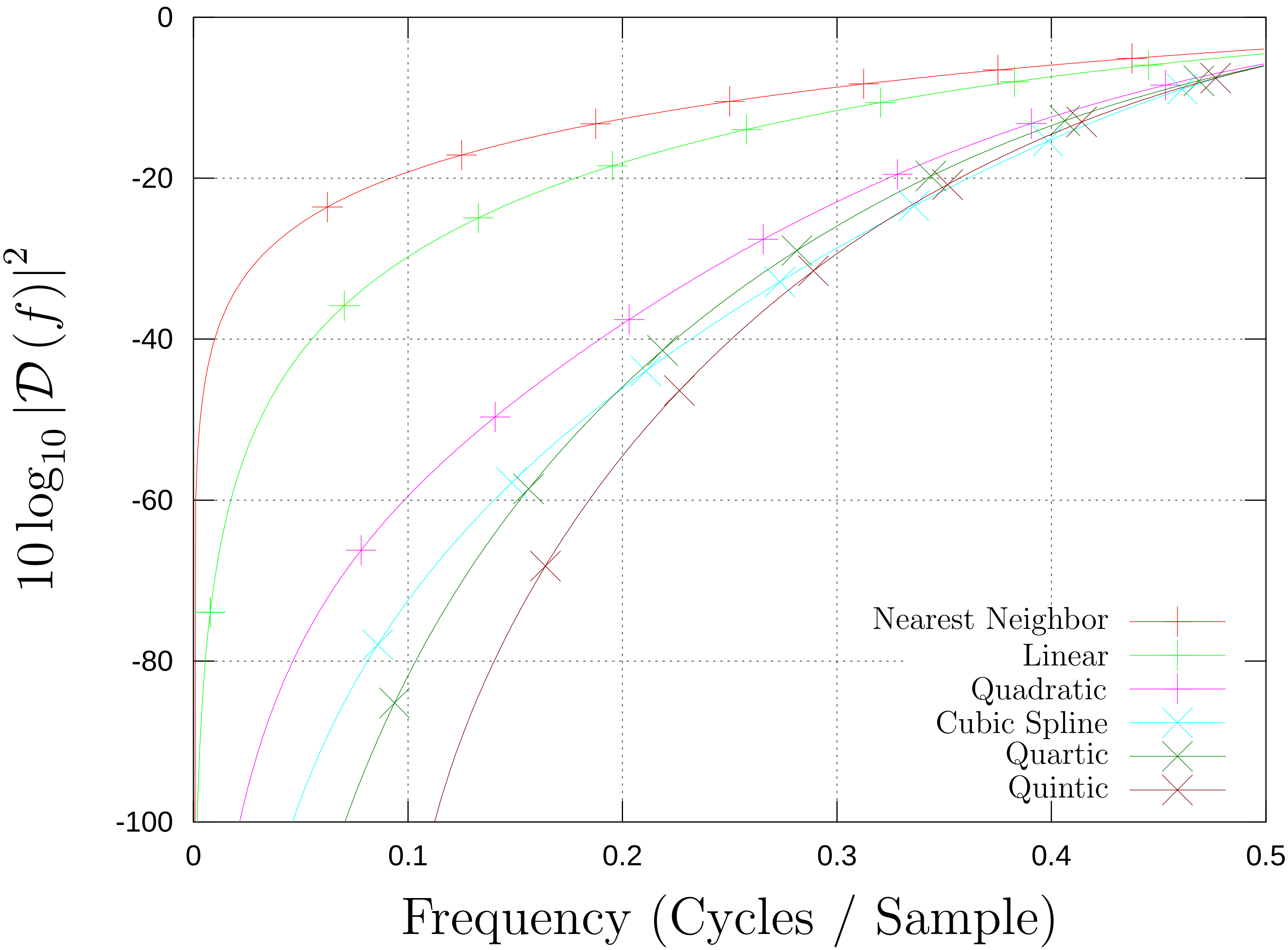
1. Design a polynomial/filter pair.
2. Plot it's impulse response.
3. Plot it's performance as a function of frequency.
4. Plot the distortion function for your filter.
5. Implement this filter to resample an incoming signal by some irrational amount.

Higher Order Interpolators

Not this class

We'll save these for another day.

Higher Order Distortion



Backup Slides

Continuous Slope

Theorem: A quadratic interpolator of finite length will never have a continuous derivative.

Proof: Consider the final non–zero section from $n - \frac{1}{2} < t < n + \frac{1}{2}$. In order for the entire quadratic to be continuous, it must be zero at $t = n + \frac{1}{2}$. Likewise, in order for this to be an interpolator, the quadratic must be zero at all integer samples, such as $t = n$. Having two roots, this section must have the form,

$$h_n(t) = a_n(t-n)\left(t - n - \frac{1}{2}\right).$$

Further, if this is the last quadratic, a_n must be non–zero.

Continuous Slope

Theorem: A quadratic interpolator of finite length will never have a continuous derivative.

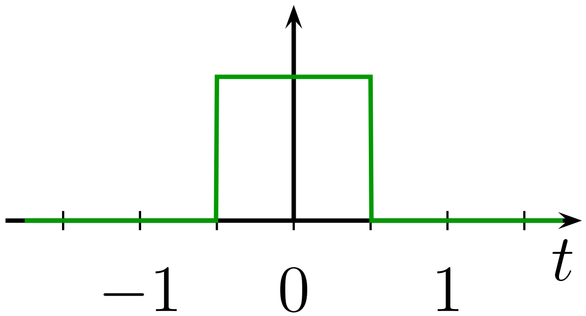
Proof: *... continued ...*

Examining the derivative,

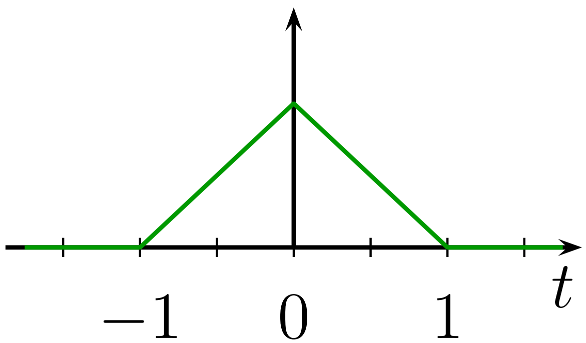
$$\begin{aligned}
\frac{d}{dt}h_n(t) &= a_n\left[(t-n)+\left(t-n-\frac{1}{2}\right)\right] \\
&= a_n\left(2t-2n-\frac{1}{2}\right) = 2a_n\left(t-n-\frac{1}{4}\right) \\
\text{Thus } h'\left(n+\frac{1}{2}\right) &= \frac{1}{2}a_n
\end{aligned}$$

will only be zero if $h_n(t)=0$, making this no longer the last polynomial in the chain. $\mathcal{Q.E.D}$.

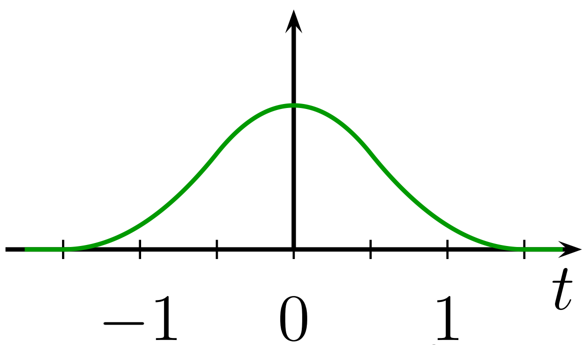
Sinc Sections



$$\frac{\sin(\pi f)}{\pi f}$$



$$\frac{\sin^2(\pi f)}{(\pi f)^2}$$



$$\frac{\sin^3(\pi f)}{(\pi f)^3}$$

Convolving a rectangle function with itself successive times yields the kernel that decays the fastest among all other kernels of the same degree.

Summed Frequency

Theorem: If $h[n]$ is a discrete filter with $\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f-\frac{k}{L}\right)}\right) = 1$, then $h[n]$ is an interpolation filter by L.

Proof: Start with the Fourier transform of the summation:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f-\frac{k}{L}\right)}\right) e^{j2\pi f n} df$$

$$= \sum_{k=0}^{L-1} \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} H\left(e^{j2\pi\left(f-\frac{k}{L}\right)}\right) e^{j2\pi\left(f-\frac{k}{L}\right)n} df\right] e^{j2\pi\frac{k}{L}n}$$

$$= h[n] \cdot \sum_{k=0}^{L-1} e^{j2\pi\frac{k}{L}n} = h[n] \cdot \begin{cases} L & n \text{ is divisible by } L \\ 0 & \text{otherwise} \end{cases}$$

Now, we focus on the other side: the Fourier transform of 1 . . .

Summed Frequency

Theorem: If $h[n]$ is a discrete filter with
$\sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f-\frac{k}{L}\right)}\right) = 1$, then $h[n]$ is an interpolation filter by L.

Proof: (Continued)

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{k=0}^{L-1} H\left(e^{j2\pi\left(f-\frac{k}{L}\right)}\right) e^{j2\pi fn} df \;=\; \int_{-\frac{1}{2}}^{\frac{1}{2}} df = 1$$

$$h[n] \cdot \begin{cases} L & n \text{ is divisible by } L \\ 0 & \text{otherwise} \end{cases} \;=\; 1 \textit{ independent of } n$$

$$h[n] \;=\; \begin{cases} \frac{1}{L} & n = 0 \\ 0 & n \neq 0, \text{ and } n \text{ is divisible by } L \\ & \text{something else otherwise} \end{cases}$$

$\mathcal{Q.E.D.}$

Summed Frequency

Theorem: If $h(t)$ is a continuous interpolation filter, such that $h(n) = 0$ for all integers $n \neq 0$, then $\sum_{k=-\infty}^{\infty} H(f+k) = 1$.

Proof:

$$
\begin{aligned}
\sum_{k=-\infty}^{\infty} H(f+k) &= \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} h(t)\, e^{-j2\pi(f+k)t} dt \\
&= \int_{-\infty}^{\infty} h(t)\, e^{-j2\pi f t} \left[\sum_{k=-\infty}^{\infty} e^{-j2\pi k t} \right] dt \\
&= \int_{-\infty}^{\infty} h(t)\, e^{-j2\pi f t} \left[\sum_{k=-\infty}^{\infty} \delta(t-k) \right] dt \\
&= \sum_{k=-\infty}^{\infty} h(k)\, e^{-j2\pi f k} = h(0) = 1
\end{aligned}
$$

Summed Frequency

Theorem: If $\sum_{k=-\infty}^{\infty} H(f+k) = 1$, then $h(t)$ is a continuous interpolation filter such that $h(n) = 0$ for all integers $n \neq 0$.

Proof: This proof is complicated by the fact that the Fourier transforms of complex exponentials over all infinity do not exist. A separate transform,

$$\lim_{F\to\infty} \frac{1}{F} \int_{-\frac{F}{2}}^{\frac{F}{2}} \sum_{k=-\infty}^{\infty} H(f-k)\, e^{j2\pi f t} df \quad = \quad \lim_{F\to\infty} \frac{1}{F} \int_{-\frac{F}{2}}^{\frac{F}{2}} e^{j2\pi f t} df$$

similar to the Fourier transform, is required. The limit associated with this transform ensures that the integral converges, making the proof possible.

Here, I offer only the following “outline” …

Summed Frequency

Theorem: If $\sum_{k=-\infty}^{\infty} H(f+k) = 1$, then $h(t)$ is a continuous interpolation filter such that $h(n) = 0$ for all integers $n \neq 0$.

Hand–Waive: (Continued)

$$\begin{aligned}
\int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} H(f-k)\, e^{j2\pi ft} df &= \int_{-\infty}^{\infty} e^{j2\pi ft} df \\
\sum_{k=-\infty}^{\infty} e^{j2\pi kt} \int_{-\infty}^{\infty} H(f-k)\, e^{j2\pi (f-k)t} df &= \delta(t) \\
h(t) \sum_{k=-\infty}^{\infty} e^{j2\pi kt} = h(t) \cdot \begin{cases} \delta(t-k) & k \in \mathcal{Z} \\ 0 & \text{otherwise} \end{cases} &= \delta(t)
\end{aligned}$$

Cascaded Interpolators

Theorem: If $h_u[n]$ is a discrete interpolator, with upsample rate L, and $h_q(t)$ is a continuous interpolator, then their convolution will be a continuous interpolator with upsample rate L.

Proof:

$$\begin{aligned}
\sum_{k=-\infty}^{\infty} H\left(f+\frac{k}{L}\right) &= \sum_{k=-\infty}^{\infty} H_q\left(f+\frac{k}{L}\right) H_u\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \\
&= \sum_{k=0}^{L-1} \sum_{n=-\infty}^{\infty} H_q\left(f+\frac{k}{L}+n\right) H_u\left(e^{j2\pi\left(f+\frac{k}{L}+n\right)}\right) \\
&= \sum_{k=0}^{L-1} H_u\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \left[\sum_{n=-\infty}^{\infty} H_q\left(f+\frac{k}{L}+n\right)\right]
\end{aligned}$$

Cascaded Interpolators

Theorem: If $h_u[n]$ is a discrete interpolator, with upsample rate L, and $h_q(t)$ is a continuous interpolator, then their convolution will be a continuous interpolator with upsample rate L.

Proof: (Continued)

$$\begin{aligned}
\sum_{k=-\infty}^{\infty} H\left(f+\frac{k}{L}\right) &= \sum_{k=0}^{L-1} H_u\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) h_q(0) \\
&= h_q(0) \sum_{k=0}^{L-1} H_u\left(e^{j2\pi\left(f+\frac{k}{L}\right)}\right) \\
&= L h_q(0)\, h_u[0] = 1
\end{aligned}$$

Impact of Continuity

Theorem: If $h(t)$ is a piecewise polynomial filter, and

1. if $h(t)$ is continuous then $H(f)$ will asymptotically decay with $\mathcal{O}\left(\frac{1}{f^2}\right)$, and
2. if the first N_k derivatives of $h(t)$ are continuous, then $H(f)$ will asymptotically decay with $\mathcal{O}\left(\frac{1}{f^{(2+N_k)}}\right)$.

Proof: We'll start by breaking $h(t)$ into its component polynomials, such that $h(t) = p_n(t)$ for $t \in \left[n - \frac{1}{2}, n + \frac{1}{2}\right]$. Then, we'll take the Fourier transform of this filter,

$$H(f) = \sum_n \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} p_n(t)\, e^{-j2\pi f t} dt.$$

Impact of Continuity

Theorem: If $h(t)$ is a piecewise polynomial filter, and

1. if $h(t)$ is continuous …

Proof: (Continued …)

$$H(f) = \sum_n \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} p_n(t)\, e^{-j2\pi ft} dt.$$

At this point, we'll integrate by parts, letting $u = p_n(t)$ and $dv = e^{-j2\pi ft} dt$. Thus, $du = p'(t)\, dt$ and $v = \frac{1}{-j2\pi f} e^{-j2\pi ft}$ yielding,

$$H(f) = \sum_n \left[\frac{p_n(t)\, e^{-j2\pi ft}}{-j2\pi f} - \frac{1}{-j2\pi f} \int p'_n(t)\, e^{-j2\pi ft} dt \right]_{n-\frac{1}{2}}^{n+\frac{1}{2}}.$$

Impact of Continuity

Theorem: If $h(t)$ is a piecewise polynomial filter, and

1. if $h(t)$ is continuous ...

Proof: (Continued ...)

We can continue integrating by parts, using (nearly) the same substitution to achieve,

$$H(f) = \sum_{k=0}^{\infty} \left[\sum_n -\frac{p_n^{(k)}(t)\, e^{-j2\pi f t}}{(j2\pi f)^{k+1}} \right]_{t=n-\frac{1}{2}}^{t=n+\frac{1}{2}} .$$

Now, notice: If $h(t)$ was continuous up until it's k^{th} derivative, then adjacent terms in this integration would cancel in the summation over n, leaving only those terms associated with $p_n^{(k)}$ where the polynomial's k^{th} derivative were discontinuous.

Impact of Continuity

Theorem: If $h(t)$ is a piecewise polynomial filter, and

1. if $h(t)$ is continuous then $H(f)$ will asymptotically decay with $\mathcal{O}\left(\frac{1}{f^2}\right)$, and
2. if the first N_k derivatives of $h(t)$ are continuous, then $H(f)$ will asymptotically decay with $\mathcal{O}\left(\frac{1}{f^{(2+N_k)}}\right)$.

Proof: (Continued ...)

Thus, if $h(t)$ is continuous, only the terms associated with $k \geq 1$ would remain. Likewise, if the first N_k derivatives were also continuous, then only the $k \geq N_k$ terms would remain. Thus, the remaining terms would decay at a rate of $\frac{1}{f^{N_k+1}}$ or faster completing our proof. $\mathcal{Q}.\mathcal{E}.\mathcal{D}$.

DC Response

It can be difficult to evaluate $H(f)$ near $f = 0$, as most of the terms have f in the denominator. Thus, we offer the following:

Theorem: The limit of $H(f)$, as $f \to 0$, is given by

$$\lim_{f \to 0} H(f) = \int_{-\infty}^{\infty} h(t)\, dt$$

Proof: The following is rather straightforward:

$$\begin{aligned}
\lim_{f \to 0} H(f) &= \lim_{f \to 0} \int_{-\infty}^{\infty} h(t)\, e^{-j2\pi f t} dt \\
&= \int_{-\infty}^{\infty} h(t) \left[\lim_{f \to 0} e^{-j2\pi f t} \right] dt \\
&= \int_{-\infty}^{\infty} h(t)\, dt
\end{aligned}$$

$\mathcal{Q}.\mathcal{E}.\mathcal{D}.$

DC Response

While the previous theorem allows you to evaluate $H(f)$ at zero, this one will make Taylor series expansions of $H(f)$ near zero possible.

Theorem: The limit of $\frac{d^k}{df^k}H(f)$, as $f \to 0$, is given by

$$\lim_{f\to 0}\frac{d^k}{df^k}H(f) = (-j2\pi)^k\int_{-\infty}^{\infty} t^k h(t)\,dt$$

Proof:

$$\begin{aligned}\lim_{f\to 0}\frac{d^k}{df^k}H(f) &= \lim_{f\to 0}\int_{-\infty}^{\infty} h(t)\left[\frac{d^k}{df^k}e^{-j2\pi ft}\right]dt \\ &= \lim_{f\to 0}\int_{-\infty}^{\infty} h(t)\left[(-j2\pi t)^k e^{-j2\pi ft}\right]dt \\ &= (-j2\pi)^k\int_{-\infty}^{\infty} t^k h(t)\,dt\end{aligned}$$

$\mathcal{Q}.\mathcal{E}.\mathcal{D}.$