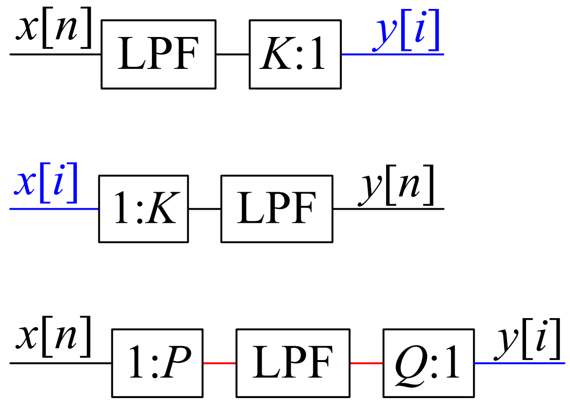
13: Resampling Filters

Resampling

Suppose we want to change the sample rate while preserving information:
e.g. Audio 44.1 kHz↔48 kHz↔96 kHz



Downsample:
LPF to new Nyquist bandwidth: $\omega_0 = \frac{\pi}{K}$

Upsample:
LPF to old Nyquist bandwidth: $\omega_0 = \frac{\pi}{K}$

Rational ratio: $f_s \times \frac{P}{Q}$
LPF to lower of old and new Nyquist bandwidths: $\omega_0 = \frac{\pi}{\max(P,Q)}$

- Polyphase decomposition reduces computation by $K = \max(P, Q)$.
- The transition band centre should be at the Nyquist frequency, $\omega_0 = \frac{\pi}{K}$
- Filter order $M \approx \frac{d}{3.5\Delta\omega}$ where d is stopband attenuation in dB and $\Delta\omega$ is the transition bandwidth (Remez-exchange estimate).
- Fractional semi-Transition bandwidth, $\alpha = \frac{\Delta\omega}{2\omega_0}$, is typically fixed.
  e.g. $\alpha = 0.05 \quad \Rightarrow \quad M \approx \frac{dK}{7\pi\alpha} = 0.9dK \quad$ (where $\omega_0 = \frac{\pi}{K}$)

Halfband Filters

If $K = 2$ then the new Nyquist frequency is $\omega_0 = \frac{\pi}{2}$.

We multiply ideal response $\frac{\sin \omega_0 n}{\pi n}$ by a Kaiser window. All even numbered points are zero except $h[0] = 0.5$.

If $4 \mid M$ and we make the filter causal ($\times z^{-\frac{M}{2}}$),
$H(z) = 0.5 z^{-\frac{M}{2}} + z^{-1} \sum_{r=0}^{\frac{M}{2}-1} h_1[r] z^{-2r}$

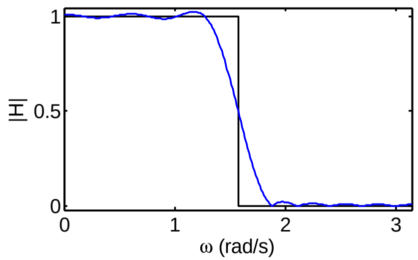
where $h_1[r] = h[2r + 1 - \frac{M}{2}]$

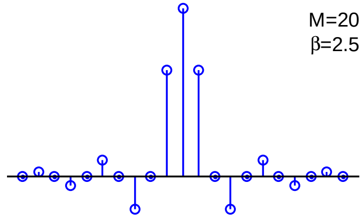
Half-band upsampler:

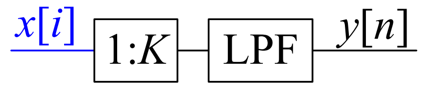
We interchange the filters with the 1:2 block and use the commutator notation.

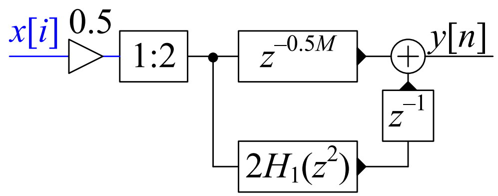
$H_1(z)$ is symmetrical with $\frac{M}{2}$ coefficients so we need $\frac{M}{4}$ multipliers in total (input gain of 0.5 can usually be absorbed elsewhere).

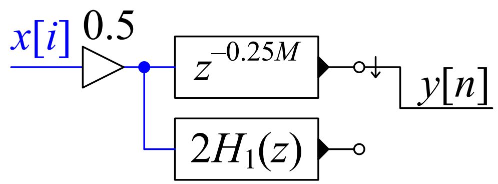
Computation: $\frac{M}{4}$ multiplies per input sample



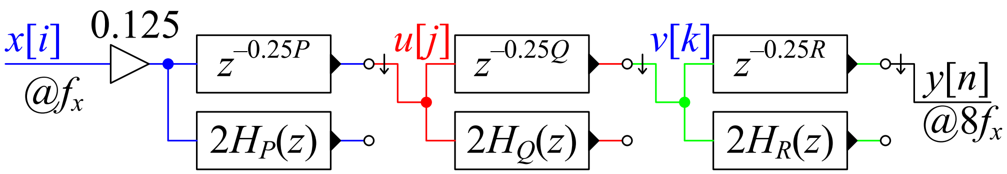








Dyadic 1:8 Upsampler



Suppose $X(z)$: BW $= 0.8\pi \Leftrightarrow \alpha = 0.2$

Upsample 1:2 $\rightarrow U(z)$:

Filter $H_P(z)$ must remove image: $\Delta\omega = 0.2\pi$

For attenuation $= 60$ dB, $P \approx \frac{60}{3.5\Delta\omega} = 27.3$

Round up to a multiple of 4: $P = 28$

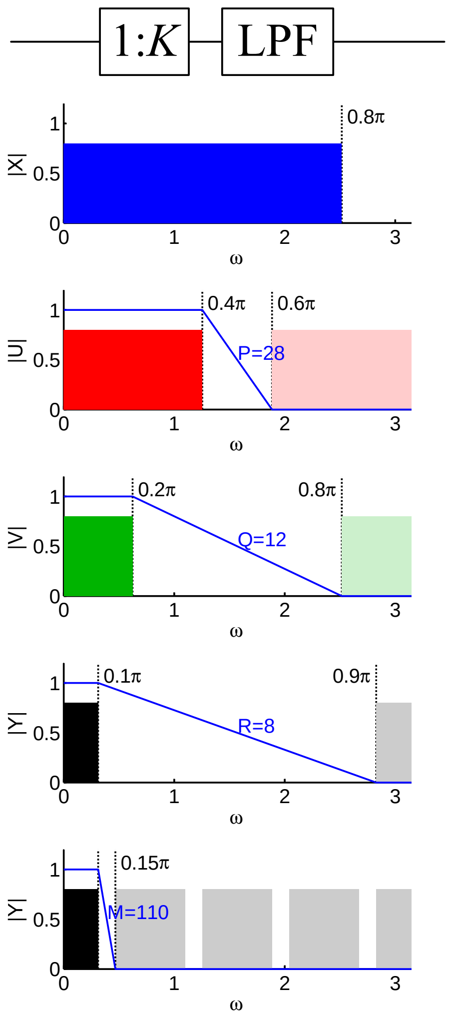
Upsample 1:2 $\rightarrow V(z)$: $\Delta\omega = 0.6\pi \Rightarrow Q = 12$

Upsample 1:2 $\rightarrow Y(z)$: $\Delta\omega = 0.8\pi \Rightarrow R = 8$

[diminishing returns + higher sample rate]

Multiplication Count:

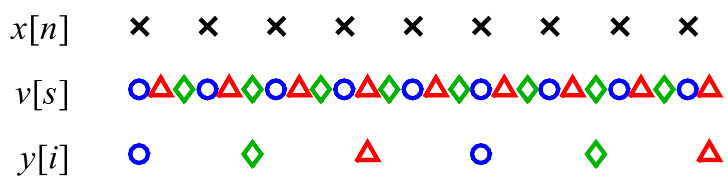
$$\left(1 + \frac{P}{4}\right) \times f_x + \frac{Q}{4} \times 2f_x + \frac{R}{4} \times 4f_x = 22f_x$$



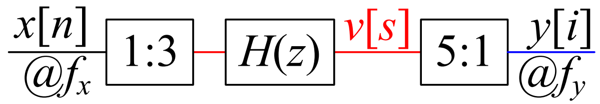
Alternative approach using direct 1:8 upsampling:

$\Delta\omega = 0.05\pi \Rightarrow M = 110 \Rightarrow 111f_x$ multiplications (using polyphase)

Rational Resampling

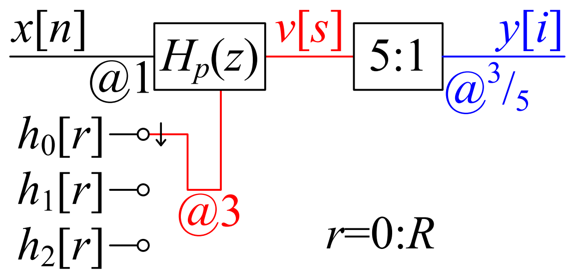


To resample by $\frac{P}{Q}$ do 1:P then LPF, then Q:1.



Resample by $\frac{P}{Q} \Rightarrow \omega_0 = \frac{\pi}{\max(P,Q)}$

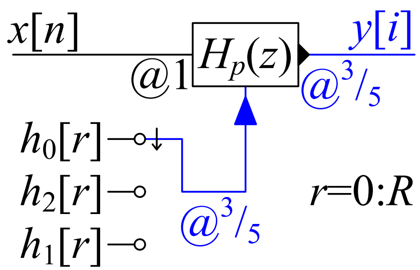
$$\Delta\omega \triangleq 2\alpha\omega_0 = \frac{2\alpha\pi}{\max(P,Q)}$$



Polyphase: $H(z) = \sum_{p=0}^{P-1} z^{-p} H_p(z^P)$

Commutate coefficients:

$v[s]$ uses $H_p(z)$ with $p = s \bmod P$



Keep only every Q^{th} output:

$y[i]$ uses $H_p(z)$ with $p = Qi \bmod P$

Multiplication Count:

$$H(z):\; M+1 \approx \frac{60\text{ [dB]}}{3.5\Delta\omega} = \frac{2.7\max(P,Q)}{\alpha}$$

$$H_p(z):\; R+1 = \frac{M+1}{P} = \frac{2.7}{\alpha}\max\left(1, \frac{Q}{P}\right)$$

$M+1$ coeficients in all

Multiplication rate: $\frac{2.7}{\alpha}\max\left(1, \frac{Q}{P}\right) \times f_y = \frac{2.7}{\alpha}\max\left(f_y, f_x\right)$

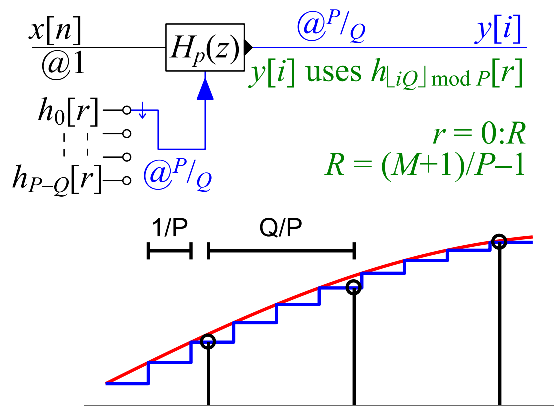
Arbitrary Resampling

Sometimes need very large P and Q:

e.g. $\frac{44.1\text{ kHz}}{48\text{ kHz}} = \frac{147}{160}$

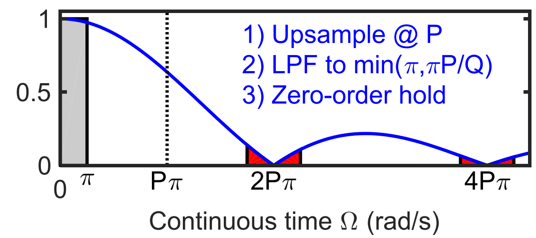
Multiplication rate OK: $\frac{2.7\max(f_y, f_x)}{\alpha}$

However # coefficients: $\frac{2.7\max(P, Q)}{\alpha}$



Alternatively, use any large integer P and round down to the nearest sample:

E.g. for $y[i]$ at time $i\frac{Q}{P}$ use $h_p[r]$ where $p = (\lfloor iQ \rfloor)_{\bmod P}$



Equivalent to converting to analog with zero-order hold and resampling at $f_y = \frac{P}{Q}$.

Zero-order hold convolves with rectangular $\frac{1}{P}$-wide window $\Rightarrow$ multiplies periodic spectrum by $\frac{\sin\frac{\Omega}{2P}}{\frac{\Omega}{2P}}$. Resampling aliases Ω to $\Omega_{\bmod \frac{2P\pi}{Q}}$.

Unit power component at Ω_1 gives alias components with total power:

$$\sin^2\frac{\Omega_1}{2P}\sum_{n=1}^{\infty}\left(\frac{2P}{2nP\pi+\Omega_1}\right)^2 + \left(\frac{2P}{2nP\pi-\Omega_1}\right)^2 \approx \frac{\omega_1^2}{4P^2}\frac{2\pi^2}{6\pi^2} = \frac{\Omega_1^2}{12P^2}$$

For worst case, $\Omega_1 = \pi$, need $P = 906$ to get -60 dB ☹

[Arbitrary Resampling]

Suppose we wish to upsample by an irrational factor, $\sqrt{2} = \frac{P}{Q}$. We choose a integer value for $P \gg \frac{P}{Q}$, say $P = 25$. Conceptually, we will upsample by $P = 25$ to obtain $v[s]$ and then downsample by $Q = \frac{P}{\sqrt{2}} = 17.6\ldots$. Taking the input sample rate to be 1, the output sample number i will be at time $\frac{i}{\sqrt{2}} = \frac{iQ}{P}$ which corresponds to the sample $n' = \frac{iQ}{P}$ of $x[n]$ and to sample $s' = iQ$ of $v[s]$.
Unfortunately, s' is not an integer and so we will instead use sample $s = \lfloor s' \rfloor = \lfloor iQ \rfloor$ of $v[s]$ instead where $\lfloor\ \rfloor$ denotes the "floor" function which rounds down to the nearest integer. To calculate this, we use the sub-filter $h_p[r]$ where $p = s \mod P$. The input samples used by the filter will be the $R+1$ most recent samples of $x[n]$ namely $x[\lfloor n' \rfloor - R]$ to $x[\lfloor n' \rfloor]$.

i	$n' = iQ/P$	$s' = iQ$	$s = \lfloor s' \rfloor$	$p = s \mod P$	$\lfloor n' \rfloor - R : \lfloor n' \rfloor$
0	0	0	0	0	$-R : 0$
1	0.71	17.68	17	17	$-R : 0$
2	1.41	35.36	35	10	$1 - R : 1$
3	2.12	53.03	53	3	$2 - R : 2$
4	2.83	70.71	70	20	$2 - R : 2$
5	3.54	88.39	88	13	$3 - R : 3$

The table shows the values of everything for the first six samples of $y[i]$. Since we only use every 17^{th} or 18^{th} value of $v[s]$, the subfilter that is used, p, increases by 17 or 18 (modulo P) each time.

[Alias Components]

Ignoring the polyphase implementation, the low pass filter operates at a sample rate of P and therefore has a periodic spectrum that repeats at intervals of $2P\pi$. Therefore, considering positive frequencies only, a signal component in the passband at Ω_1 will have images at $\Omega = 2nP\pi \pm \Omega_1$ for all positive integers n.

These components are multiplied by the $\frac{\sin 0.5P^{-1}\Omega}{0.5P^{-1}\Omega}$ function and therefore have amplitudes of

$$\frac{\sin 0.5P^{-1}(2nP\pi\pm\Omega_1)}{0.5P^{-1}(2nP\pi\pm\Omega_1)} = \frac{\sin\left(n\pi\pm0.5P^{-1}\Omega_1\right)}{\left(n\pi\pm0.5P^{-1}\Omega_1\right)} = \frac{\sin\left(\pm1^n 0.5P^{-1}\Omega_1\right)}{\left(n\pi\pm0.5P^{-1}\Omega_1\right)}.$$

When we do the downsampling to an output sample rate of $\frac{P}{Q}$, these images will be aliased to frequencies $\Omega_{\mathrm{mod}\ \frac{2P\pi}{Q}}$. In general, these alias frequencies will be scattered throughout the range $(0, \pi)$ and will result in broadband noise.

We need to sum the squared amplitudes of all these components:

$$\sum_{n=1}^{\infty} \frac{\sin^2\left(\pm1^n 0.5P^{-1}\Omega_1\right)}{\left(n\pi\pm0.5P^{-1}\Omega_1\right)^2} = \sin^2\left(0.5P^{-1}\Omega_1\right)\sum_{n=1}^{\infty} \frac{1}{\left(n\pi\pm0.5P^{-1}\Omega_1\right)^2}$$

If we assume that $n\pi \gg 0.5P^{-1}\Omega_1$ and also that $\sin\left(0.5P^{-1}\Omega_1\right) \approx 0.5P^{-1}\Omega_1$, then we can approximate this sum as

$$\left(0.5P^{-1}\Omega_1\right)^2\sum_{n=1}^{\infty} \frac{2}{(n\pi)^2} = \frac{\Omega_1^2}{4P^2} \times \frac{2}{\pi^2}\sum_{n=1}^{\infty} n^{-2}$$

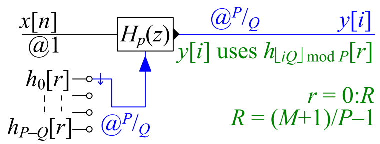
The summation is a standard result and equals $\frac{\pi^2}{6}$.

So the total power of the aliased components is $\frac{\Omega_1^2}{12P^2}$.

Polynomial Approximation

Suppose $P = 50$ and $H(z)$ has order $M = 249$
$H(z)$ is lowpass filter with $\omega_0 \approx \frac{\pi}{50}$
Split into 50 filters of length $R + 1 = \frac{M+1}{P} = 5$:

$h_p[0]$ is the first P samples of $h[m]$
$h_p[1]$ is the next P samples, etc.
$h_p[r] = h[p + rP]$

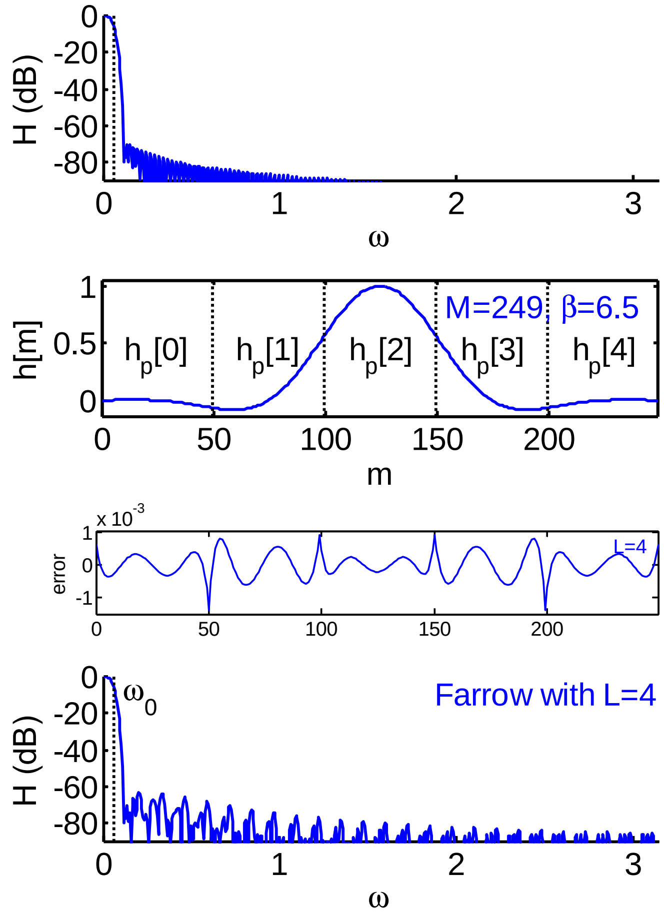


Use a polynomial of order L to approximate each segment:

$$h_p[r] \approx f_r\left(\frac{p}{P}\right) \text{ with } 0 \le \frac{p}{P} < 1$$

$h[m]$ is smooth, so errors are low.
E.g. error $< 10^{-3}$ for $L = 4$

- Resultant filter almost as good
- Instead of $M + 1 = 250$ coefficients we only need $(R + 1)(L + 1) = 25$ where $R + 1 = \frac{2.7}{\alpha} \max\left(1, \frac{Q}{P}\right)$



Farrow Filter

Filter coefficients depend on fractional part of $i\frac{Q}{P}$:

$$\Delta[i] = i\frac{Q}{P} - n \text{ where } n = \left\lfloor i\frac{Q}{P} \right\rfloor$$

$$y[i] = \sum_{r=0}^{R} f_r(\Delta[i])x[n-r]$$

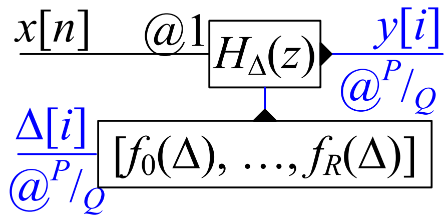
where $f_r(\Delta) = \sum_{l=0}^{L} b_l[r]\Delta^l$

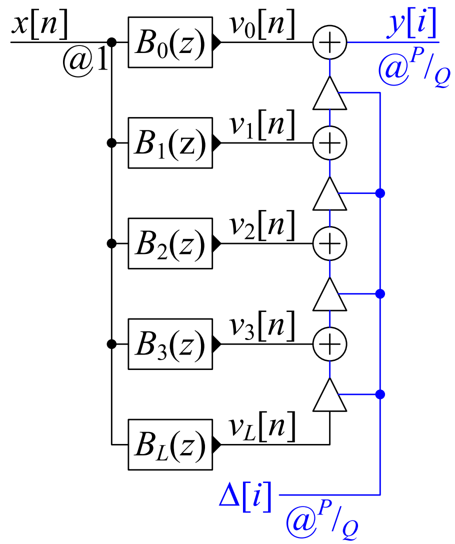
$$\begin{aligned} y[i] &= \sum_{r=0}^{R} \sum_{l=0}^{L} b_l[r]\Delta[i]^l x[n-r] \\ &= \sum_{l=0}^{L} \Delta[i]^l \sum_{r=0}^{R} b_l[r]x[n-r] \\ &= \sum_{l=0}^{L} \Delta[i]^l v_l[n] \end{aligned}$$

where $v_l[n] = b_l[n] * x[n]$

[like a Taylor series expansion]

$$R+1 = \frac{M+1}{P} = 5$$





$$R+1 \approx \frac{2.7}{\alpha} \max\left(1, \frac{Q}{P}\right)$$

Horner's Rule:

$$y[i] = v_0[n] + \Delta\left(v_1[n] + \Delta\left(v_2[n] + \Delta\left(\cdots\right)\right)\right)$$

Multiplication Rate:

Each $B_l(z)$ needs $R+1$ per input sample

Horner needs L per output sample

Total: $(L+1)(R+1)f_x + Lf_y = \frac{2.7(L+1)}{\alpha} \max\left(1, \frac{f_x}{f_y}\right) f_x + Lf_y$

[Farrow Filter sub-filter indexing]

We assume that the input sample rate is 1 and the output sample rate is $\frac{P}{Q}$. Output sample $y[i]$ is therefore at time $n' = \frac{iQ}{P}$ which will not normally be an integer.

Normal Resampling Method

In the normal resampling procedure, this corresponds to sample $s = iQ$ of $v[s]$ where $v[s]$ is obtained by upampling $x[n]$ by a factor of P. Using a polyphase filter to do the upsampling, we use each of the sub-filters $h_p[n]$ in turn to generate the upsampled samples $v[s]$ where $p = s \mod P$ and the filter acts on the $R+1$most recent input samples, $x[n-R]$ to $x[n]$ where $n = \lfloor n' \rfloor]$. We can write any integer s, as the sum of an exact multiple of P and the remainder when $s \div P$ as $s = P\left\lfloor \frac{s}{P} \right\rfloor + s \mod P$. Substituting the previously defined expressions for n and p into this equation gives $iQ = Pn + p$. We can rearrange this to get $p = Pn' - Pn$ where p lies in the range $[0,\, P-1]$ and determines which of the subfilters we will use.

Farrow Filter

In the normal method (above), the sub-filter than we use is indexed by p which lies in the range $[0,\, P-1]$. In the Farrow filter, the sub-filter that we use is instead indexed by the value of the fractional number $\Delta = \frac{p}{P}$ which always lies in the range $[0,\, 1)$. From the previous paragraph, $\Delta[i] = \frac{p}{P} = n' - n = i\frac{Q}{P} - \left\lfloor i\frac{Q}{P} \right\rfloor$ which is a function only of the output sample number, i and the resampling ratio $\frac{P}{Q}$. The advantage of this is that both P nor Q can now be non-integers.

- Transition band centre at ω_0
  - $\omega_0 =$ the lower of the old and new Nyquist frequencies
  - Transition width $= \Delta\omega = 2\alpha\omega_0$, typically $\alpha \approx 0.1$
- Factorizing resampling ratio can reduce computation
  - halfband filters very efficient (half the coefficients are zero)
- Rational resampling $\times \frac{P}{Q}$
  - # multiplies per second: $\frac{2.7}{\alpha} \max(f_y, f_x)$
  - # coefficients: $\frac{2.7}{\alpha} \max(P, Q)$
- Farrow Filter
  - approximate filter impulse response with polynomial segments
  - arbitrary, time-varying, resampling ratios
  - # multiplies per second: $\frac{2.7(L+1)}{\alpha} \max(f_y, f_x) \times \frac{f_x}{f_y} + L f_y$
    - ▷ $\approx (L+1)\frac{f_x}{f_y}$ times rational resampling case
  - # coefficients: $\frac{2.7}{\alpha} \max(P, Q) \times \frac{L+1}{P}$
  - coefficients are independent of f_y when upsampling

For further details see Mitra: 13 and Harris: 7, 8.

MATLAB routines

gcd(p,q)	Find $\alpha p + \beta q = 1$ for coprime p, q
polyfit	Fit a polynomial to data
polyval	Evaluate a polynomial
upfirdn	Perform polyphase filtering
resample	Perform polyphase resampling