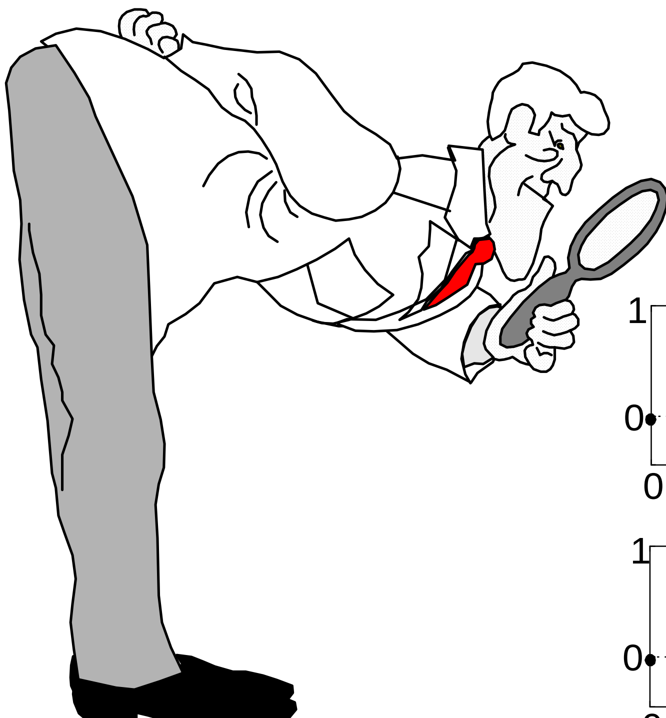
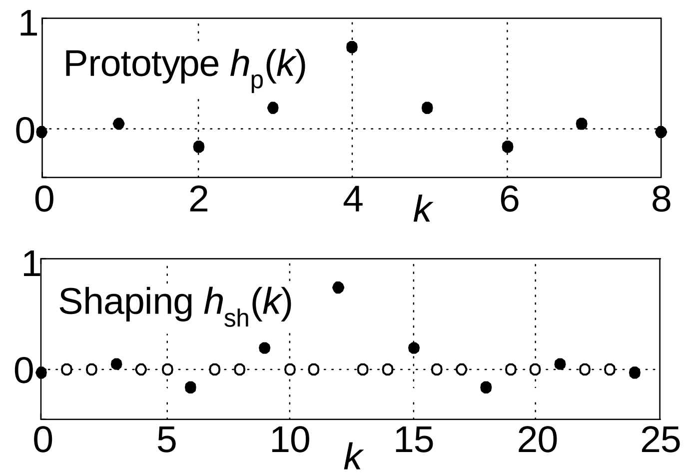
2004 COMP.DSP Conference; Cannon Falls, MN, July 29-30, 2004

Interpolated Lowpass FIR Filters

Speaker: Richard Lyons
Besser Associates
E-mail: r.lyons@ieee.com





Interpolated FIR Filters

► ***Interpolated FIR filters*** **are <u>used to build narrowband lowpass FIR filters</u>,**

- **possibly more computationally efficient than traditional Parks-McClellan-designed FIR filters.**

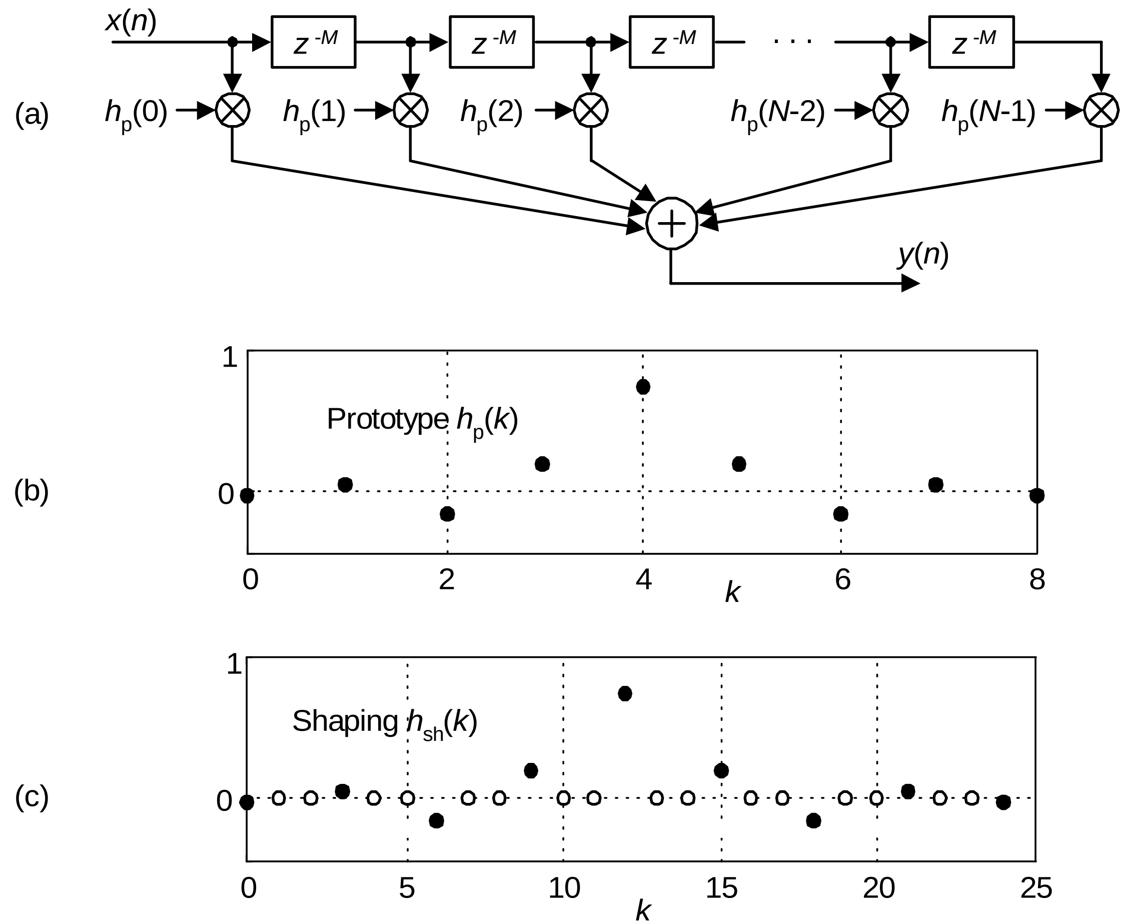
► **Interpolated FIR (IFIR) filters are based upon the behavior of an *N*-tap nonrecursive linear-phase FIR filter,**

- **when each of its single-unit delays are replaced with *M*-unit delays,**
- **where *M* is an integer.**

For example:

$h_p(k)$ **impulse response of a 9-tap FIR *prototype* filter.**

$h_{sh}(k)$ **impulse response of an expanded FIR filter, where** $M = 3$**. We the expanded filter the *shaping filter*.**



► **Prototype FIR filter's transfer function as**

$$H_p(z) = \sum_{k=0}^{N_p-1} h_p(k) z^{-k}$$

- **where N_p is the length of $h_p(k)$, and k is the filter coefficient index.**

► **Transfer function of a general shaping FIR filter [z in $H_p(z)$ replaced with z^M] is**

$$H_{sh}(z) = \sum_{k=0}^{N_p-1} h_p(k) z^{-kM}.$$

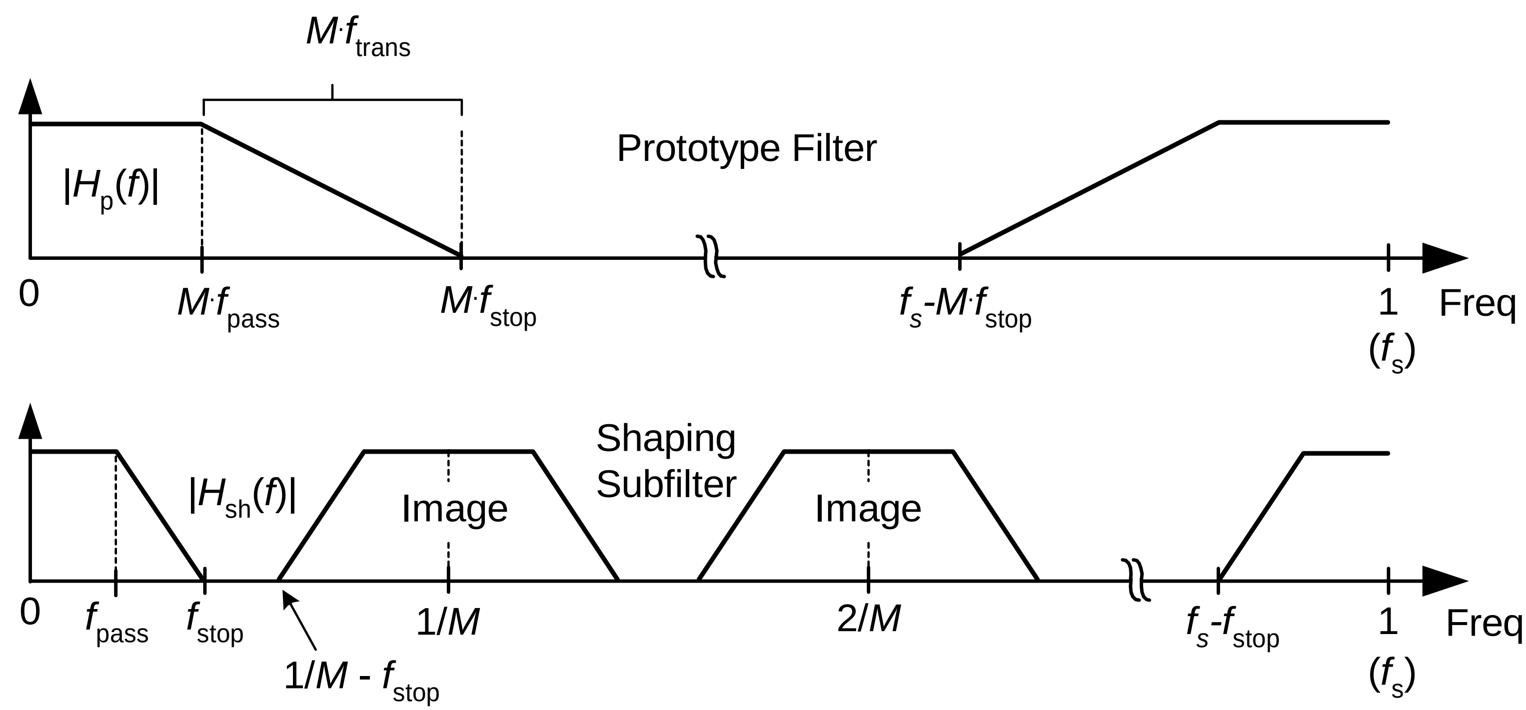
► **If the number of coefficients in the prototype filter is N_p,**

- **expanded impulse response length of shaping filter is**

$$K_{sh} = M(N_p - 1) + 1.$$

- **An *M*-fold expansion of the impulse response causes an *M*-fold compression (and repetition) of $|H_p(f)|$ frequency magnitude response.**

- **There are *M* repetitive passbands in $|H_{sh}(f)|$,**
  - **centered at integer multiples of 1/*M* (f_s/*M* Hz),**
  - **called *images*.**

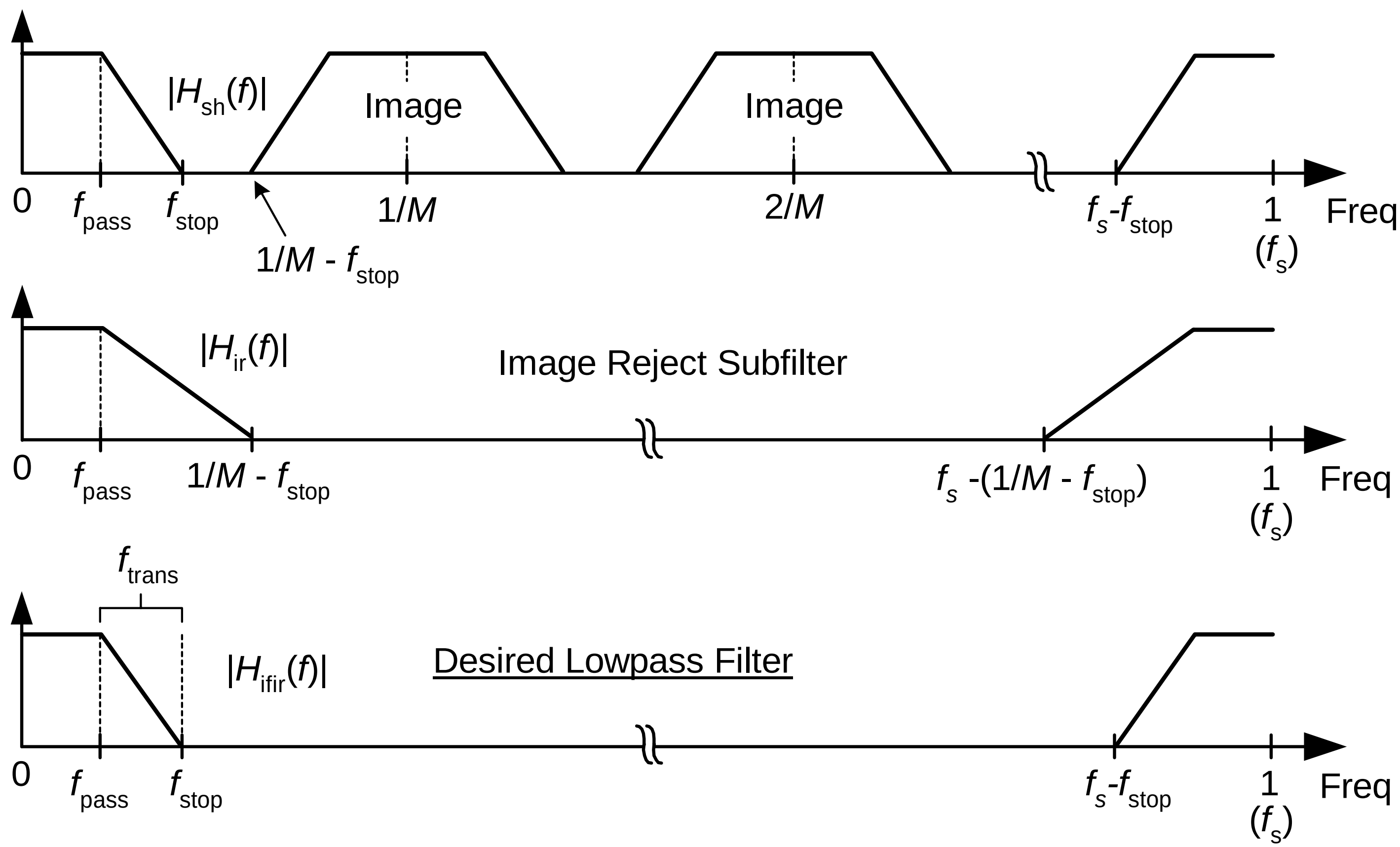


► **Next, we follow the shaping subfilter with a lowpass *image-reject* subfilter,**

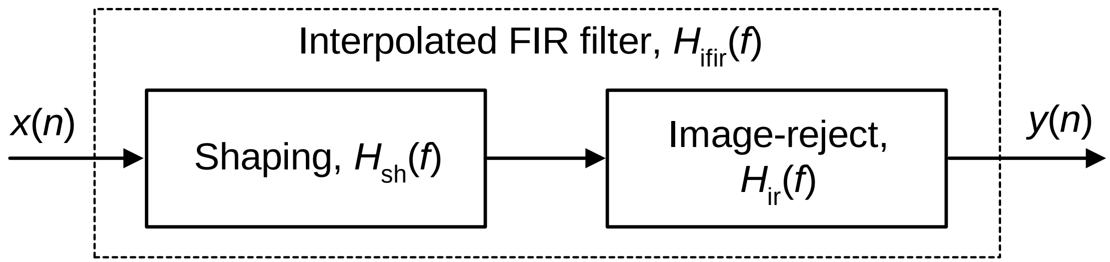
- **whose task is to attenuate the *image* passbands,**

► **The resultant $|H_{\text{ifir}}(f)|$ frequency magnitude response is, of course, the product**

$$|H_{\text{ifir}}(f)| = |H_{\text{sh}}(f)|\,|H_{\text{ir}}(f)|.$$

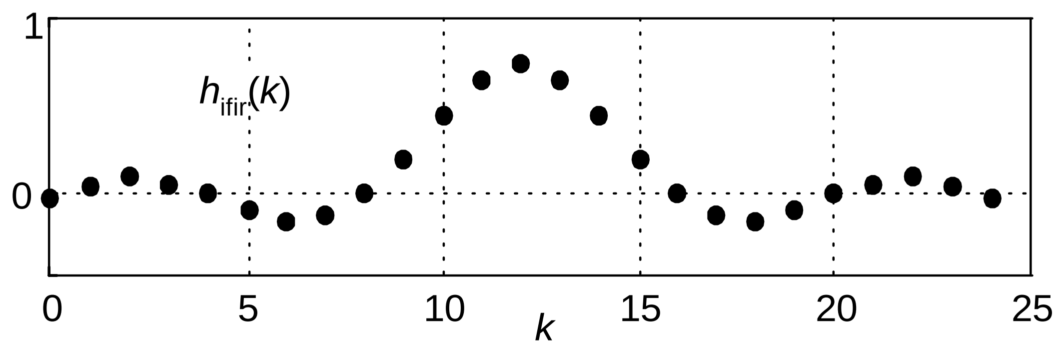


Cascaded subfilters is called an Interpolated FIR (IFIR) filter.



IFIR filter interpolated impulse response.

(Interpolated version of $h_p(k)$.)



- **Original desired lowpass filter's passband width is f_{pass},**
  - **its stopband begins at f_{stop}, and**
  - **Its transition region width is $f_{trans} = f_{stop}$-f_{pass},**
- **Then the prototype subfilter's normalized frequency parameters are defined as**

$$f_{p\text{-}pass} = Mf_{pass}, \quad f_{p\text{-}stop} = Mf_{stop}, \quad \text{and} \quad f_{p\text{-}trans} = Mf_{trans} = M(f_{stop} - f_{pass}).$$

► **The image-reject subfilter's frequency parameters are**

$$f_{\text{ir-pass}} = f_{\text{pass}}, \text{ and } \quad f_{\text{ir-stop}} = \frac{1}{M} - f_{\text{stop}}.$$

► **Stopband attenuations of the prototype filter and image-reject subfilter are identical,**

- **set equal to the desired IFIR filter stopband attenuation.**

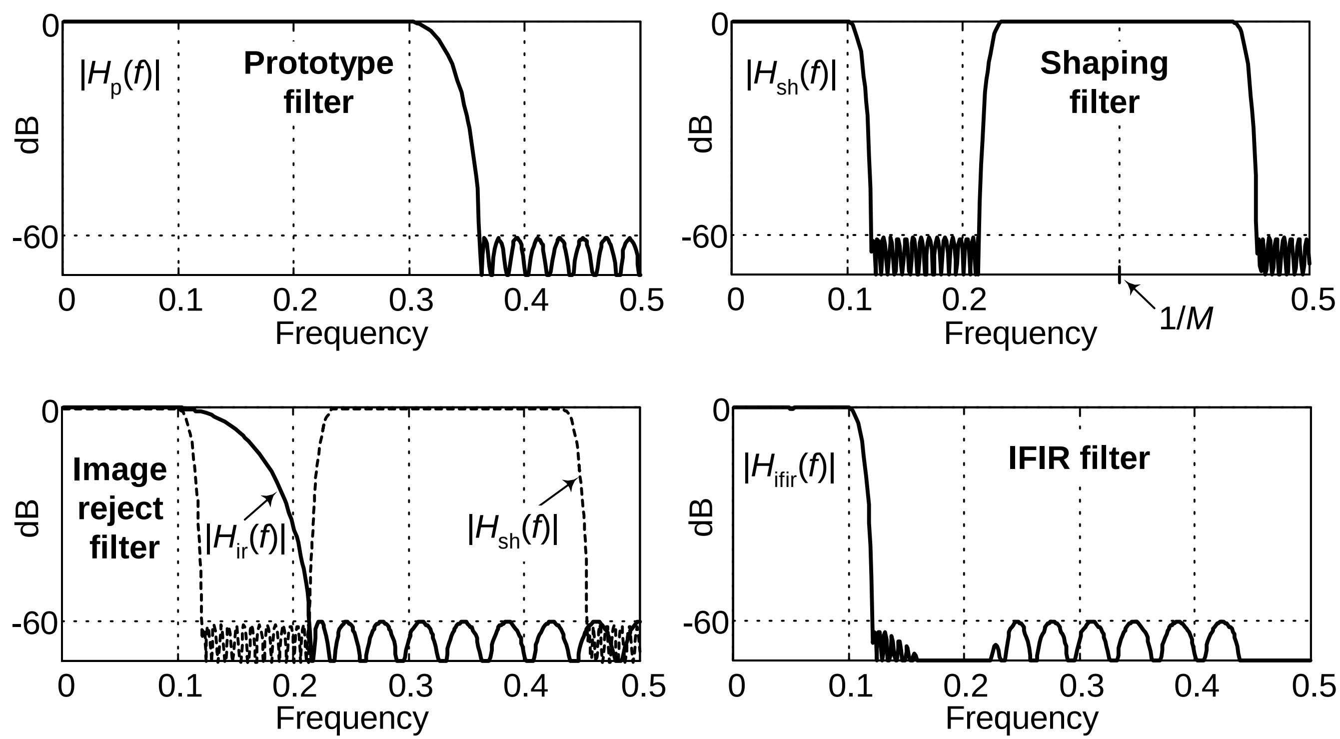
► **Let's look at a design example:**

► **Consider the design of a *desired* linear-phase FIR filter:**

- **normalized passband width is $f_{\text{pass}} = 0.1$,**
- **passband ripple is 0.1 dB, (peak-peak)**
- **transition region width is $f_{\text{trans}} = 0.02$, and**
- **stopband attenuation is 60 dB.**

► **Expansion factor of $M = 3$.**

► **Here's what we have:**



► **Satisfying the original desired filter specifications would**

- **require a traditional single-stage FIR filter with N_{tfir} = 137 taps,**
- **'tfir' subscript means *traditional FIR*.**

► **Shape of $|H_{ifir}(f)|$ determined by $|H_{sh}(f)|$ "shaping subfilter".**

- ► **IFIR's shaping and the image-reject subfilters require $N_p = 45$ and $N_{ir} = 25$ taps respectively,**
  - **for a total of $N_{ifir} = 70$ taps.**

- ► **We define the percent reduction in computational workload as**

$$\%\ \text{computation reduction} = 100\,\frac{N_{tfir} - N_p - N_{ir}}{N_{tfir}}. \qquad (1)$$

- ► **IFIR filter computational workload reduction:**

$$\%\ \text{computational reduction} = 100\,\frac{137 - 70}{137} = 49\%.$$ ☺

Choosing the Optimum Expansion Factor M

► **Expansion factor *M* has a profound effect on the computational efficiency of IFIR filters.**

► **To show this, consider other values of expansion factor *M*.**

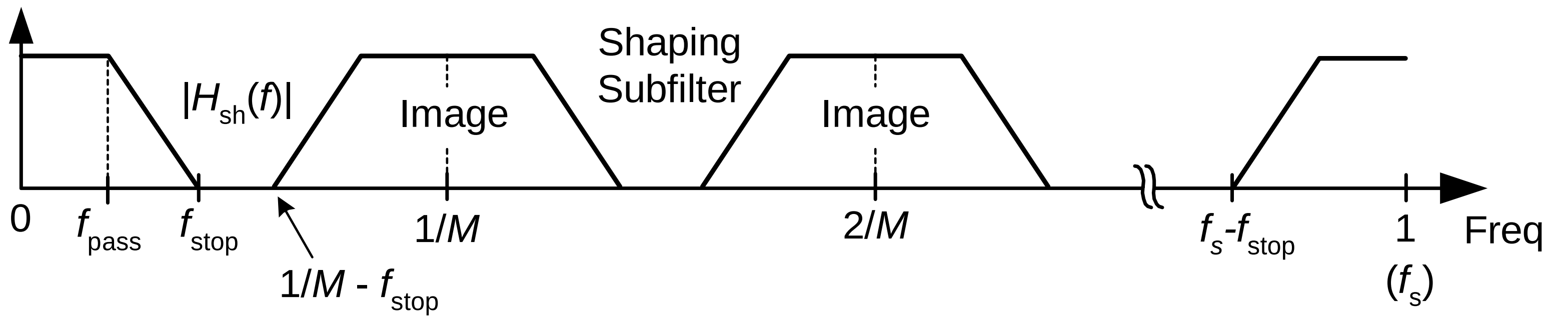
Expansion factor M	Number of taps			Computation reduction
	$h_{sh}(k)$	$h_{ir}(k)$	IFIR total	
2	69	8	77	43%
3	45	25	70	49%
4	35	95	130	8%

► **As so often happens in signal processing designs, there is a trade off to be made.**

- **Smaller M, reduced frequency compression in $H_{sh}(f)$, increases necessary N_p taps,**
- **Larger M, reduces transition region width of $H_{ir}(f)$, increases necessary N_{ir} taps.**

► As indicated in the following figure,

- max M is the largest integer satisfying $1/M - f_{stop} \geq f_{stop}$, (or $1/M \geq 2f_{stop}$),
- ensuring no passband image overlap.



► This yields an upper bound on M of

$$M_{max} = \left\lfloor \frac{1}{2f_{stop}} \right\rfloor$$

- where $\lfloor x \rfloor$ indicates truncation of x to an integer.

► Thus the acceptable expansion factors are integers in the range $2 \leq M \leq M_{max}$.

► For our above IFIR filter design example:

$$M_{max} = \left\lfloor \frac{1}{2(0.1 + 0.02\ )} \right\rfloor = 4.$$

Estimating the Number of FIR Filter Taps

► **To estimate the computation reduction of IFIR filters,**

- **we need an algorithm to compute N_{tfir},**
- **the number of taps, in a traditional nonrecursive FIR filter.**

► **A particularly simple expression for N_{tfir} is**

$$N_{tfir} \approx \frac{Atten}{22(f_{stop} - f_{pass})} . \qquad (2)$$

- **Where *Atten* = stopband attenuation in dB**

► **Likewise, the number of taps in the prototype and image-reject subfilters are**

$$N_{p} \approx \frac{Atten}{22(M)(f_{stop} - f_{pass})} , \quad \text{and} \qquad (2')$$

$$N_{ir} \approx \frac{Atten}{22(1/M - f_{stop} - f_{pass})} . \qquad (2'')$$

Modeling IFIR Filter Performance

► **We want to model "% computation reduction" in terms of desired filter parameters.**

► **If we substitute the expressions from Eq. (2) into Eq. (1),**

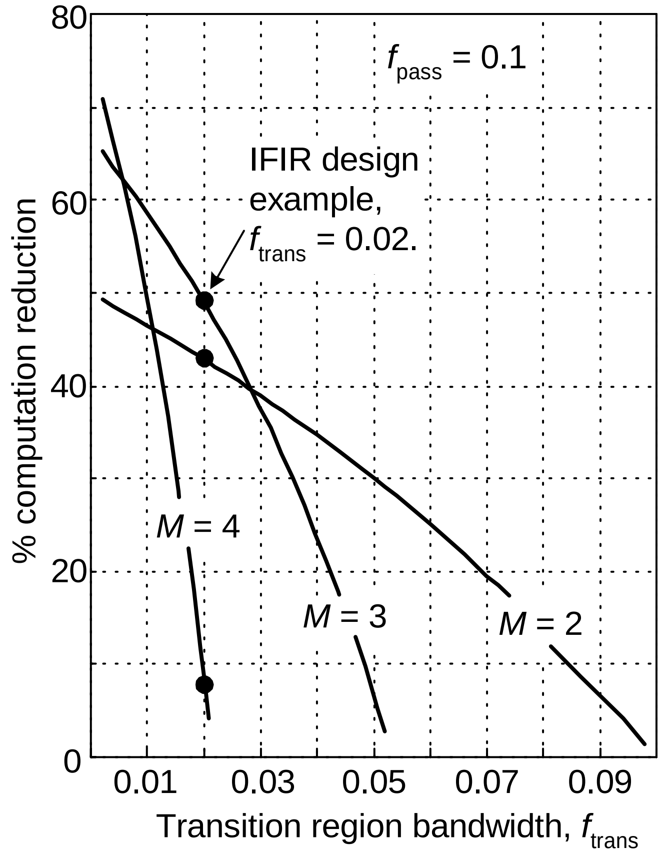
- **we can write the important IFIR filter design equation:**

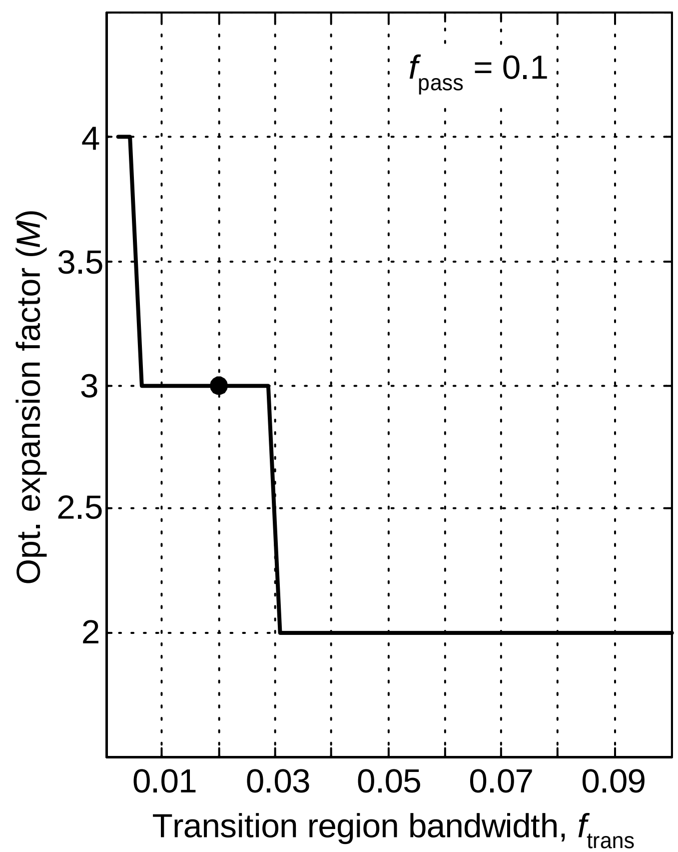
$$\text{\% computation reduction} = 100\left[\frac{M - 1}{M} - \frac{Mf_{\text{trans}}}{1 - Mf_{\text{trans}} - 2Mf_{\text{pass}}}\right]. \qquad (3)$$

- **where $f_{\text{trans}} = f_{\text{stop}} - f_{\text{pass}}$.**

► **Equation (3) is plotted below, for f_{pass} = 0.1**

- **showing % computation reduction vs. f_{trans}.**

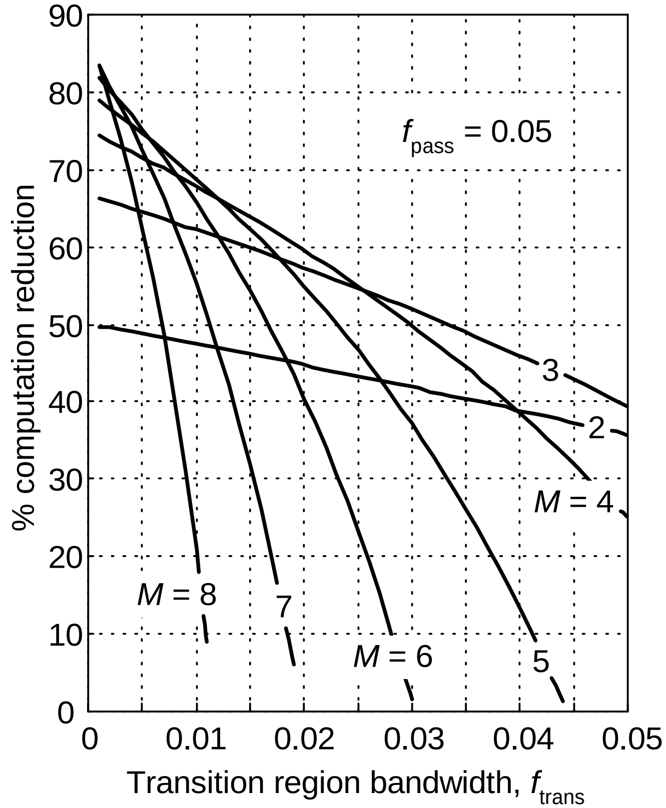


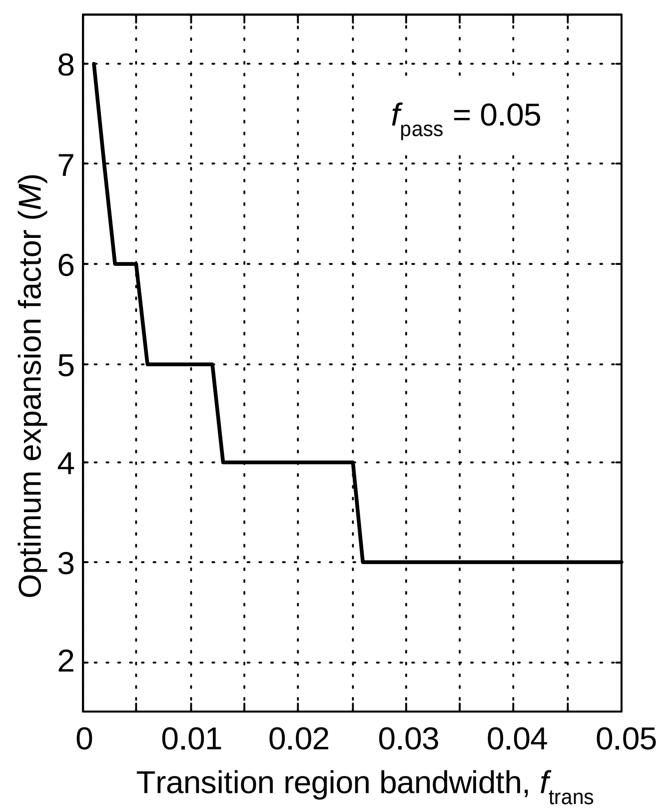


► **When the transition region width is large, only a small *M* will avoid passband image overlap.**

► **At smaller transition region widths, larger expansion factors are possible.**

► **Here's IFIR filter performance when the f_{pass} = 0.05.**

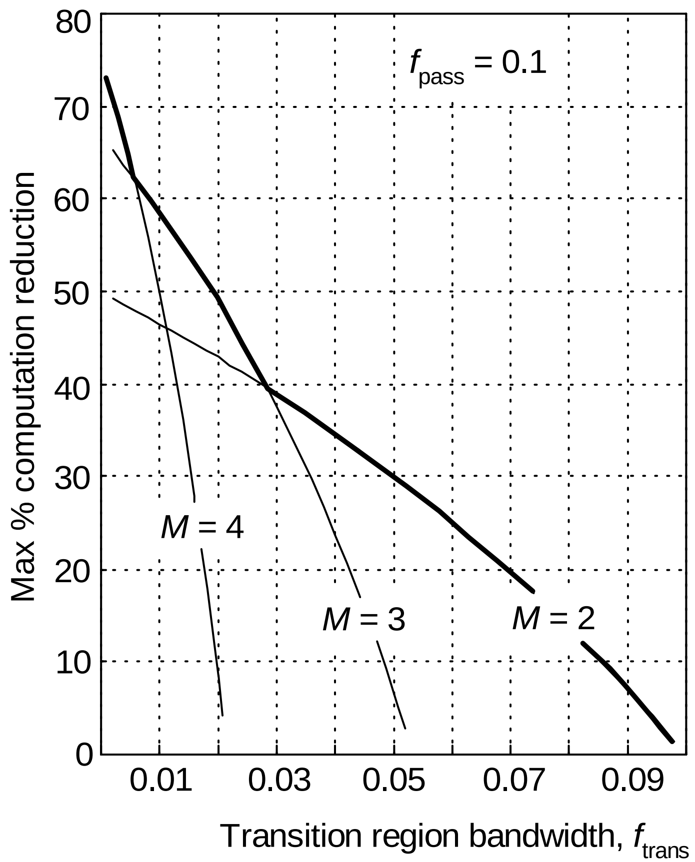




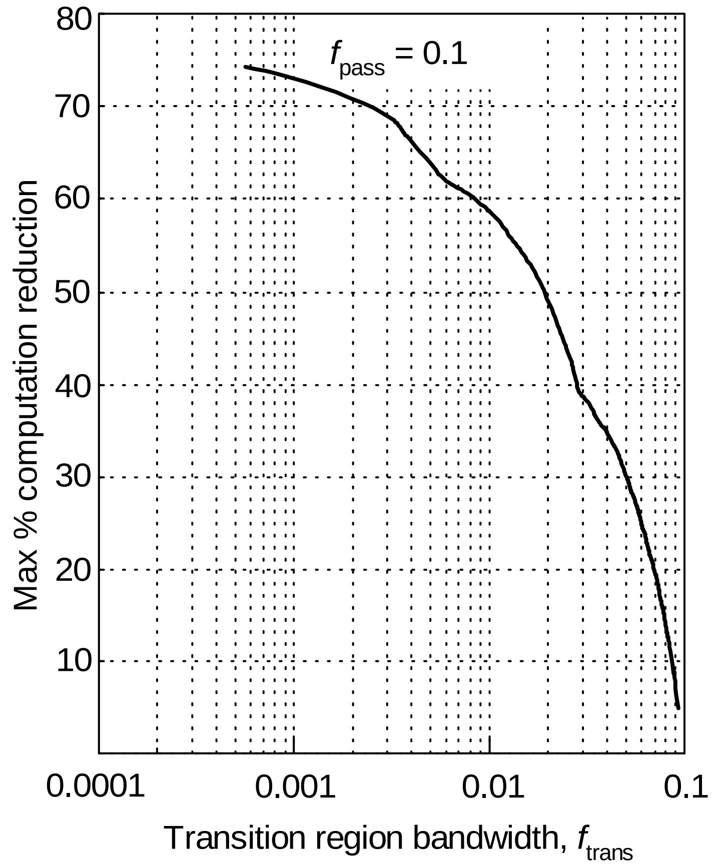
► **As f_{trans} approaches zero, % computation reduction approaches 100(M-1)/M.**

► **Here we plot max % computation reduction as a function of f_{trans} for f_{pass} = 0.1**

- **on a logarithmic frequency axis.**



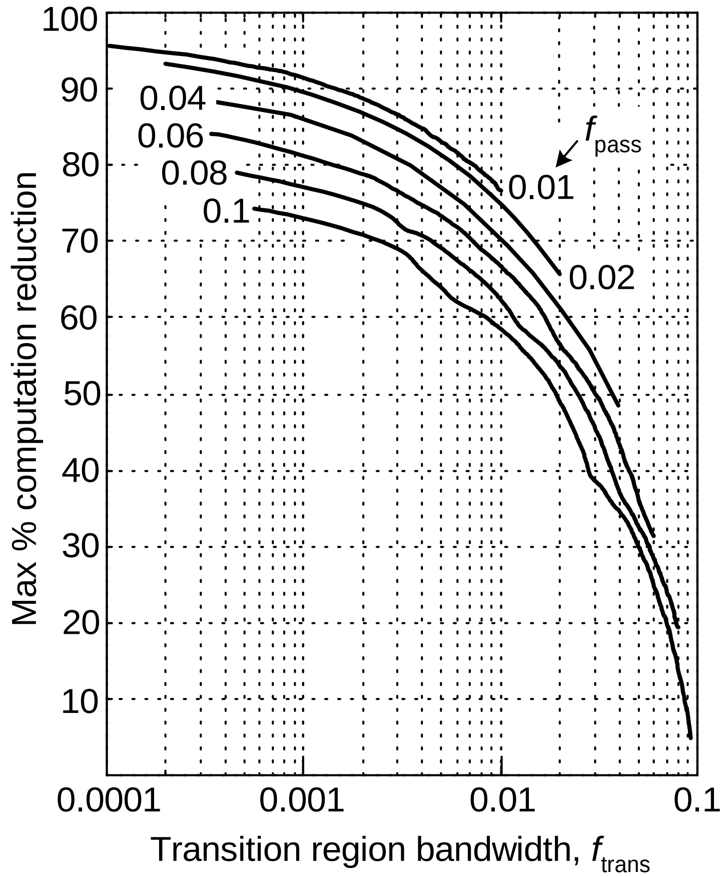
(a)

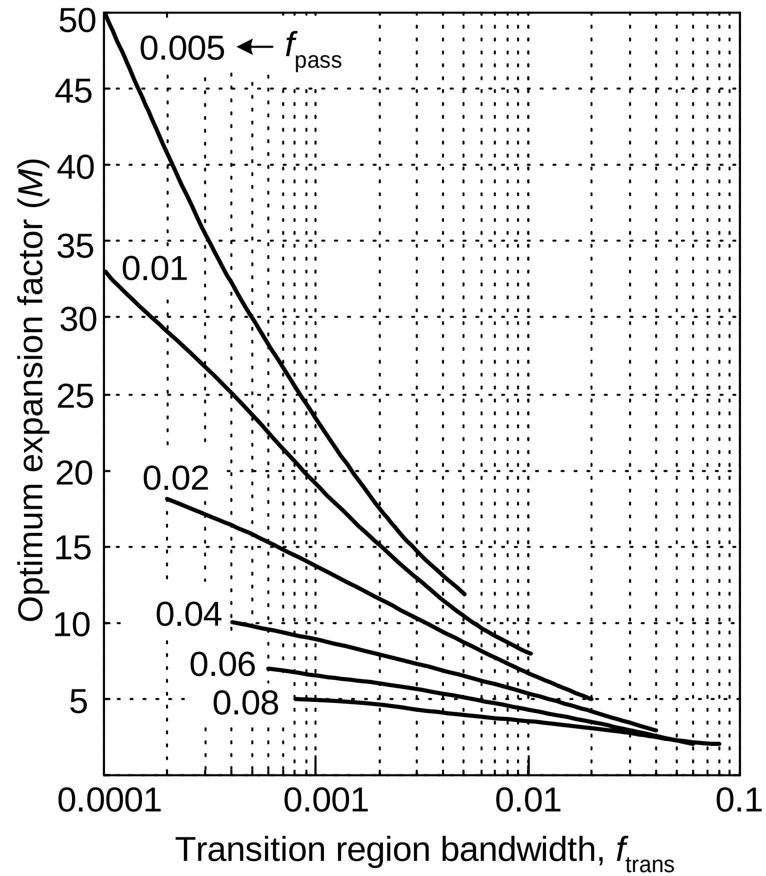


(b)

► **Next, we include other f_{pass} curves to show max % computation reduction vs. f_{trans},**

- **and optimum M used to compute the max % computation reduction curves.**





► **These are our IFIR filter design curves. ☺**

IFIR Filter Implementation Issues

► **Please resist the temptation to combine the two subfilters into a single filter**

- **whose coefficients are the convolution of the subfilters' impulse responses.**
- **With such a maneuver would we'd lose all computation reduction.**

► **When using programmable DSP chips, larger values of *M* require a larger block of hardware *data memory*, in the form of a *circular buffer*, be available for the shaping subfilter.**

► **The size of this data memory must be at least**

$$K_{sh} = M(N_p - 1) + 1.$$

► **When implementing an IFIR filter with a programmable DSP chip,**

- **you must loop through the circular signal data buffer using an increment equal to *M*.**

► **If possible, use *folded* nonrecursive FIR structures,**

- **to reduce the number of multiplications by a factor of two.**

IFIR Filter Design Example

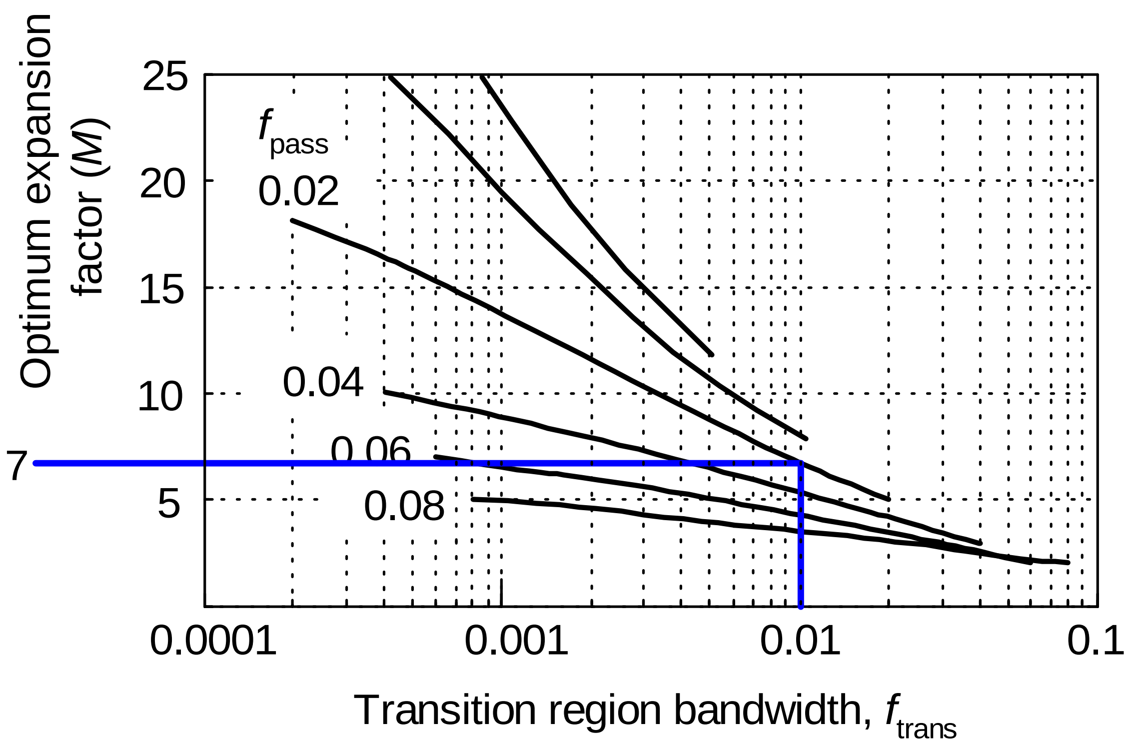
- **The design of practical lowpass IFIR filters is straightforward, and comprises four steps:**
  - **Define the desired lowpass filter performance requirements,**
  - **Determine a candidate value for the expansion factor *M*,**
  - **Design and evaluate the shaping and image-reject subfilters, and**
  - **Investigate IFIR filter performance for alternate expansion factors near the initial *M* value.**

- **As a design example, we'll design a lowpass IFIR filter with:**
  - $f_{pass} = 0.02$,
  - **passband ripple of 0.5 dB (p-p),**
  - $f_{trans} = 0.01$ **(thus** $f_{stop} = 0.03$**), and**
  - **stopband attenuation = 50 dB.**

► **First, we find the f_{trans} = 0.01 point on the abscissa of our design curve and**

- **follow it up to the point where it intersects the f_{pass} = 0.02 curve.**

- **This intersection indicates we should start our design with M = 7.**



► **With $M = 7$, we use our favorite traditional FIR filter design software to design a linear-phase prototype FIR filter with the following parameters:**

$$f_{\text{p-pass}} = Mf_{\text{pass}} = 7(0.02) = 0.14,$$

passband ripple = (0.5)/2 dB = 0.25 dB, (***rule of thumb***)

$$f_{\text{p-stop}} = Mf_{\text{stop}} = 7(0.03) = 0.21, \text{ and}$$

stopband attenuation = 50 dB.

► **Such a prototype FIR filter will have $N_p = 33$ taps and, with $M = 7$,**

- **shaping subfilter has an impulse response length of $K_{sh} = 225$ samples.**

► **Next, we design an image-reject subfilter having the following parameters:**

$$f_{\text{ir-pass}} = f_{\text{pass}} = 0.02,$$

passband ripple = (0.5)/2 dB = 0.25 dB,

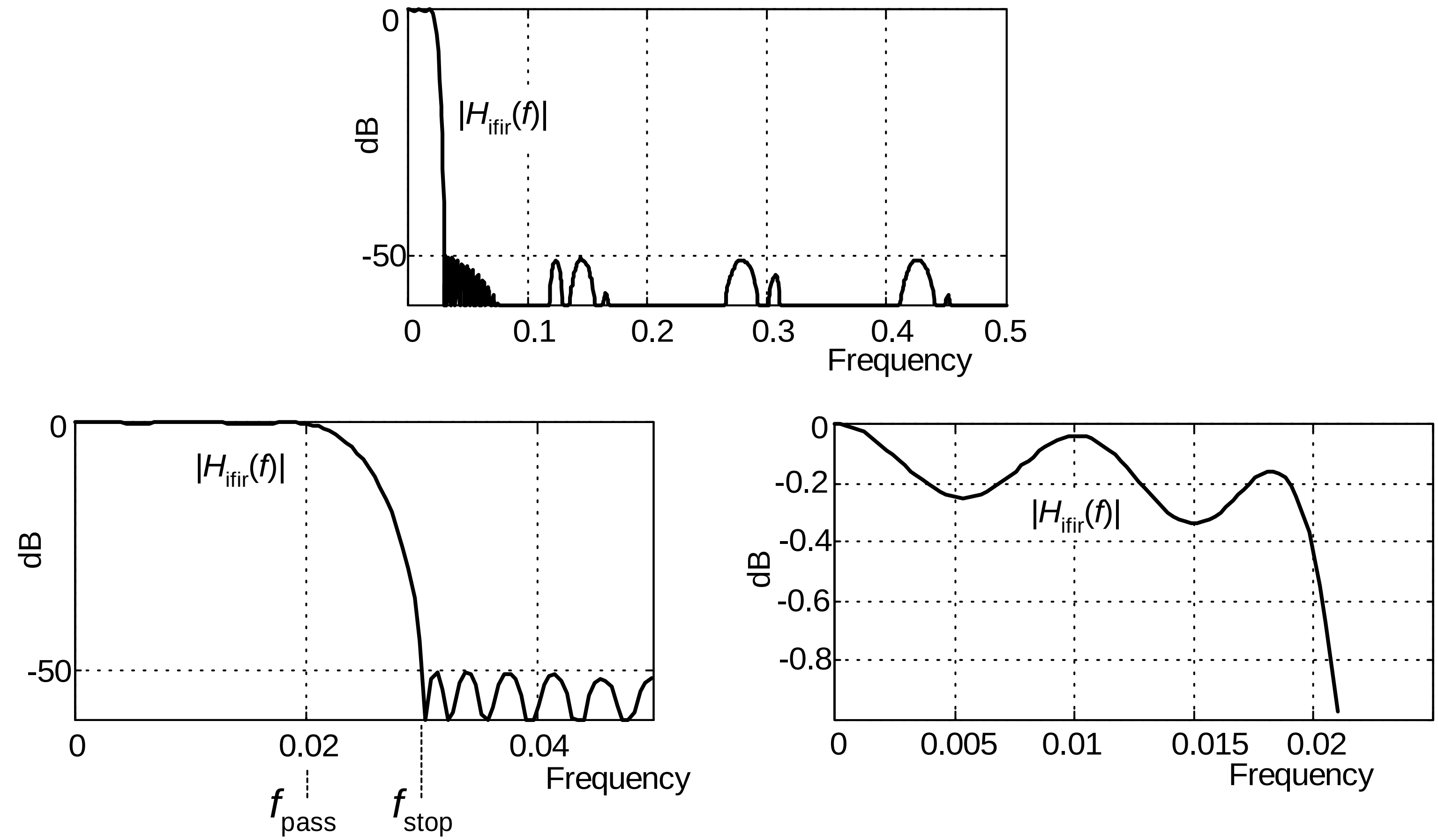
$$f_{\text{ir-stop}} = \frac{1}{M} - f_{\text{stop}} = 1/7 - 0.03 = 0.113, \text{ and}$$

stopband attenuation = 50 dB.

► **This image-reject subfilter will have $N_{ir} = 27$ taps.**

► **Cascaded image-reject and shaping subfilters require 60 multiplications per output sample.**

- **IFIR filter frequency magnitude response is shown below.**



► **A traditional FIR filter requires roughly N_{tfir} = 240 taps.**

- ► **Computational workload reduction is 100x(240 -60)/240 = 75%! ☺**
  - **Final IFIR filter design step is to sit back and enjoy a job well done.**

- ► **Further modeling, using alternate expansion factors, yields the following table.**

Expansion factor M	Number of taps			K_{sh} data storage	Computation reduction:
	$h_{sh}(k)$	$h_{ir}(k)$	IFIR total		
3	76	8	84	226	65%
4	58	12	70	229	71%
5	46	17	63	226	74%
6	39	22	61	229	75%
7	33	27	60	225	75%
8	29	33	62	225	74%
9	26	41	67	226	72%
10	24	49	73	231	70%
11	21	60	81	221	66%

IFIR Filters With Sample Rate Conversion (SRC)

► **IFIR filters useful for signal sample rate change applications,**

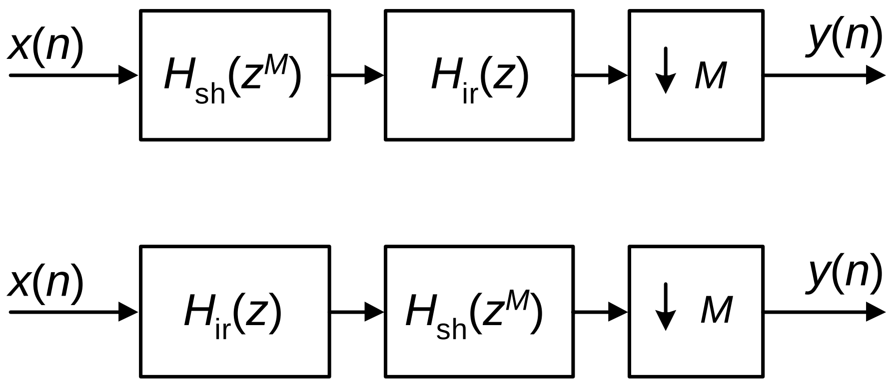
  - decimation or interpolation.

► **Consider an IFIR filter followed by downsampling by integer *M*.**

- Operation '↓*M*' means discard all but every *M*th sample.

► **Because $H_{sh}(z^M)$ and $H_{ir}(z)$ are linear, we can swap their order.**

Decimation

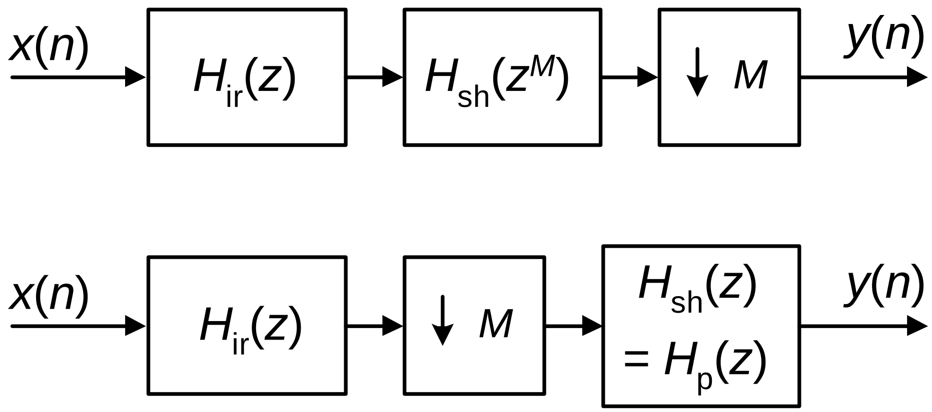


- <u>Here comes the good part</u>.

- We can swap the order of the $H_{sh}(z^M)$ filter with the downsampler.

- Now, where every M-unit delay in $H_{sh}(z^M)$ is replaced by a unit delay.

Decimation



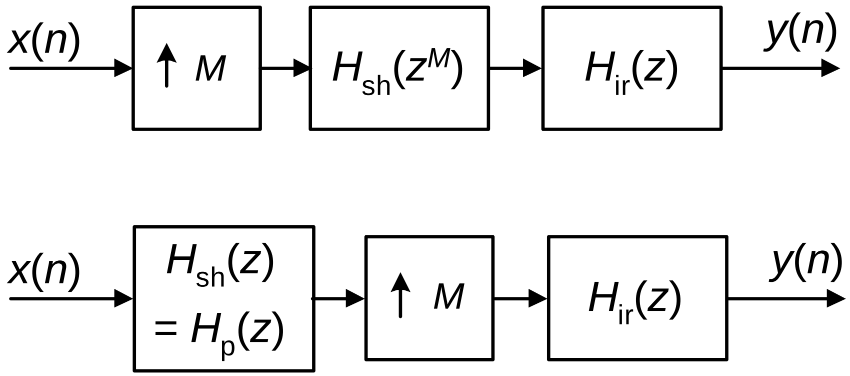
- This takes use back to using our original low-order prototype filter $H_p(z)$,

  - with its reduced signal data storage requirements. ☺

- Also, the $H_{ir}(z)$ and M downsampler combination can use polyphase filtering to reduce computational workload [1].

- Similarly, IFIR filters can be used for interpolation (upsampling).
  - The upsampling (interpolation) operation '↑M' means insert M-1 zero-valued samples between each $x(n)$ sample.

Interpolation



- We swap the order of filter $H_{sh}(z^M)$ with the upsampler,
- Now every M-unit delay in $H_{sh}(z^M)$ is replaced by a unit delay.
- This takes use back to using our original low-order prototype filter $H_p(z)$,
  - with its reduced signal data storage requirements. ☺
- The M upsampler and $H_{ir}(z)$ combination can use polyphase filtering to reduce computational workload.

IFIR Filter Summary

- **We've introduced the structure and performance of IFIR filters.**

- **IFIR filters they can achieve significant computational workload reduction relative to traditional nonrecursive FIR filters,**

  - **reductions as large as 90%.**

- **IFIR filter implementation is a cascade of filters simple tapped-delay line FIR filters,**

  - **designed using readily-available nonrecursive FIR filter design software.**

► **More IFIR filter details,**

- **math derivations**
- **design guidelines, and**
- **additional literature references are provided in:**

Reference [1]:

***Understanding Digital Signal Processing*, 2nd Ed., by R. Lyons, Prentice Hall, Upper Saddle River, New Jersey, 2004**