Multirate systems

---

1. Influence of the sampling rate
2. Why multirate systems
3. Decreasing the sampling rate
   - ideal decimator; practical decimating filter; realization
4. Increasing the sampling rate
   - ideal interpolator; practical interpolating filter; realization
5. Transposition
6. Applications
   - A/D and D/A converters

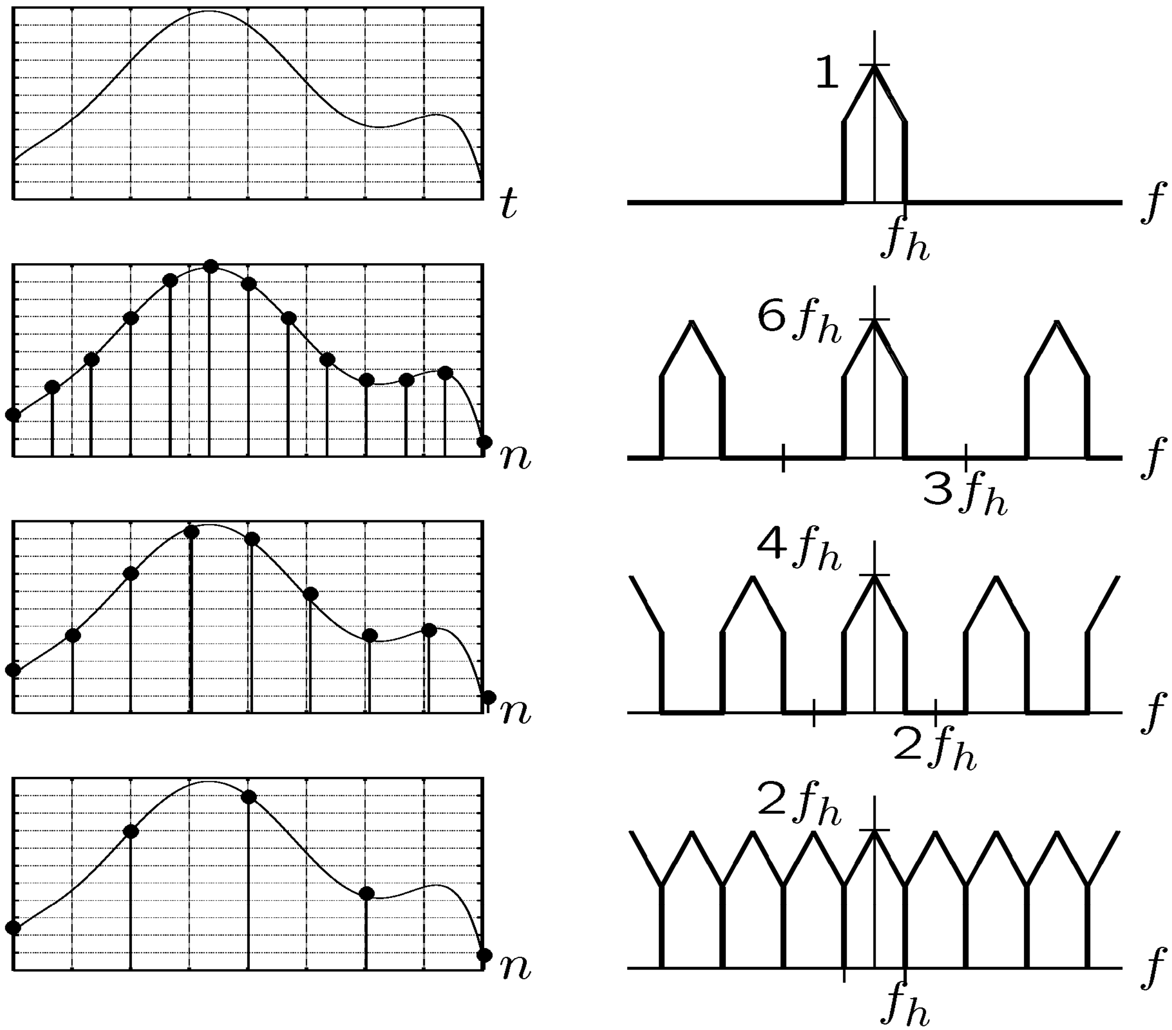
1. Influence F_S

Analog signal $x(t)$:

Sampling with $6f_h$:

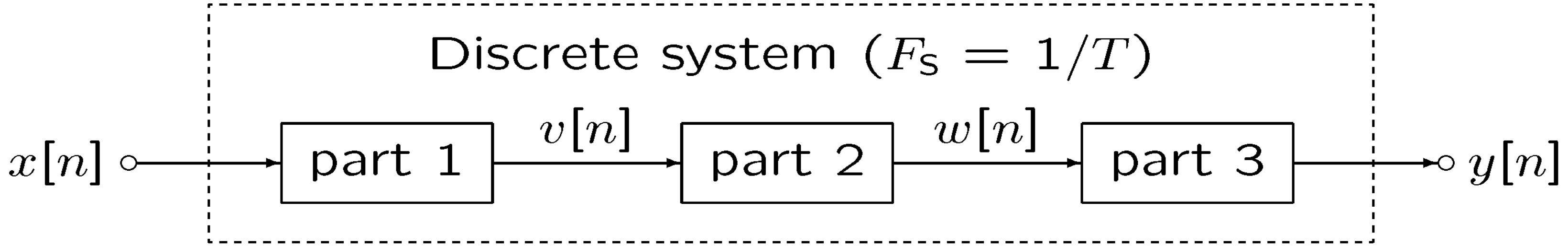
Sampling with $4f_h$:

Sampling with $2f_h$:

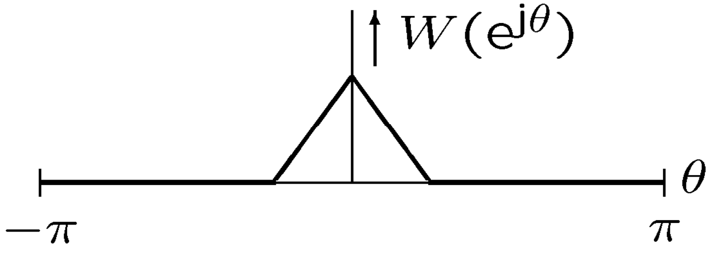


2. Why multirate systems

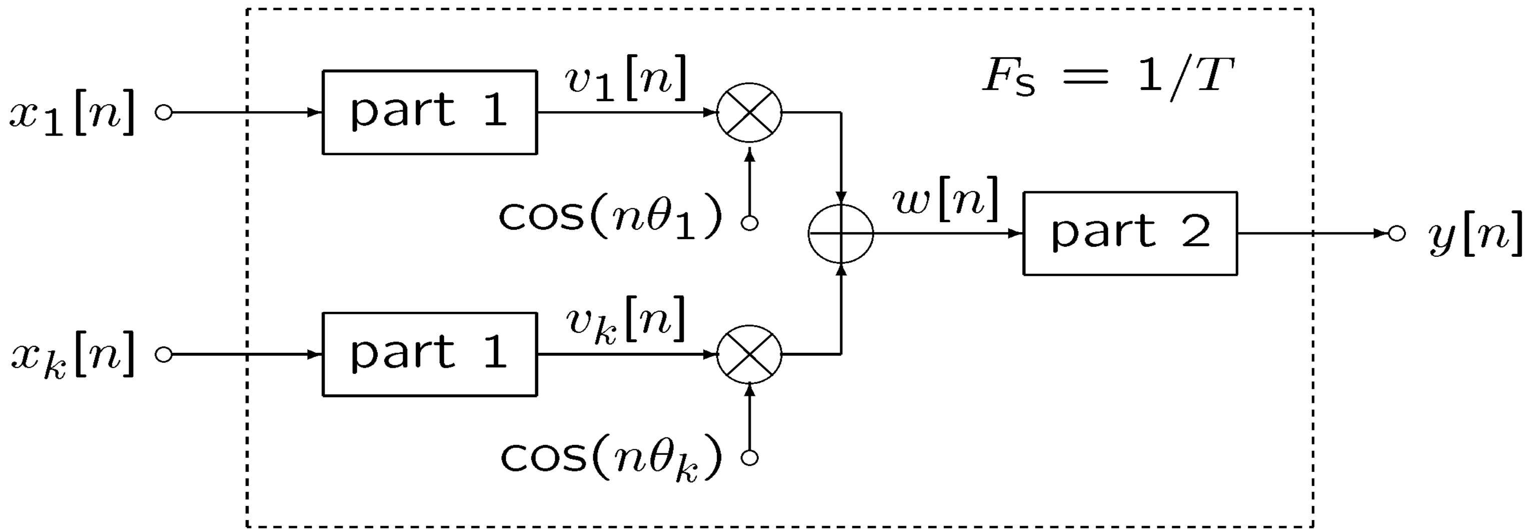
The lower the sampling rate, the cheaper the system. Use the lowest possible sampling frequency. Example for downsampling:



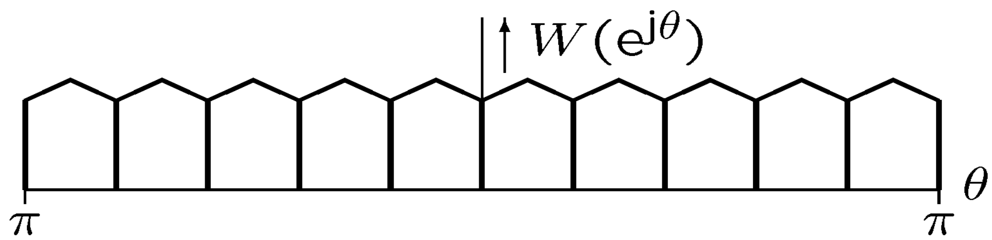
If part 2 is an ideal lowpass filter, signal $w[n]$ can be sampled with a lower sampling frequency. We therefore use downsampling



Example for upsampling:



Input can be sampled at a lower sampling frequency: upsampling



Examples of multirate systems:

- oversampled A/D and D/A converters
- bitstream converters
- Compact Disc and subband coding in MPEG-audio
- 100 Hz television and DCT coding for video
- filtering in receivers like FM, OFDM, TV, $\cdots$

We now use absolute values for time and frequency; so we use:

- $x[nT_1]$ and $y[nT_2]$ instead of $x[n]$ and $y[n]$
- $X(\mathrm{e}^{\mathrm{j}\omega T_1})$ and $Y(\mathrm{e}^{\mathrm{j}\omega T_2})$ instead of $X(\mathrm{e}^{\mathrm{j}\theta})$ and $Y(\mathrm{e}^{\mathrm{j}\theta})$

3. Sampling rate decrease with a factor R

Downsampling

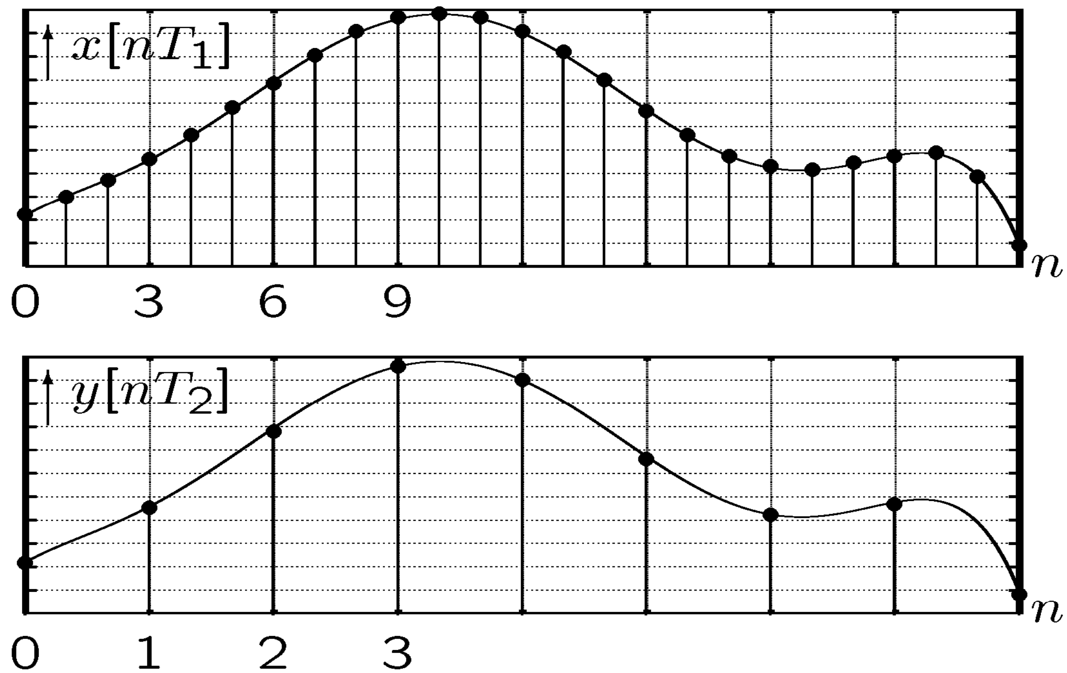
with $R = 3$:

We can write:

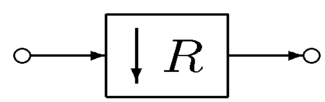
$$y[nT_2] = x[3nT_1]$$

In general:

$$y[nT_2] = x[RnT_1]$$

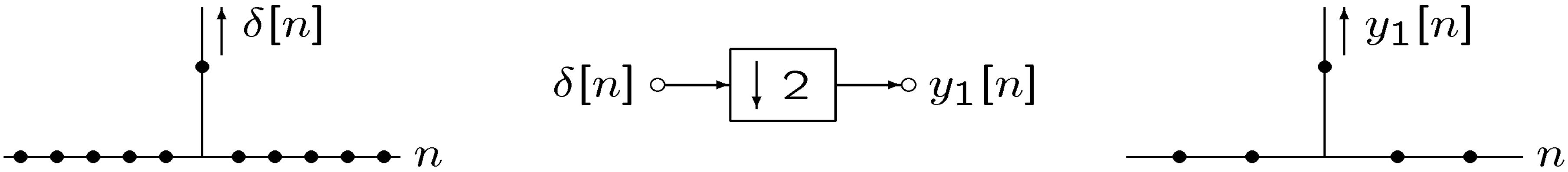


Sampling Rate Decreaser (SRD):

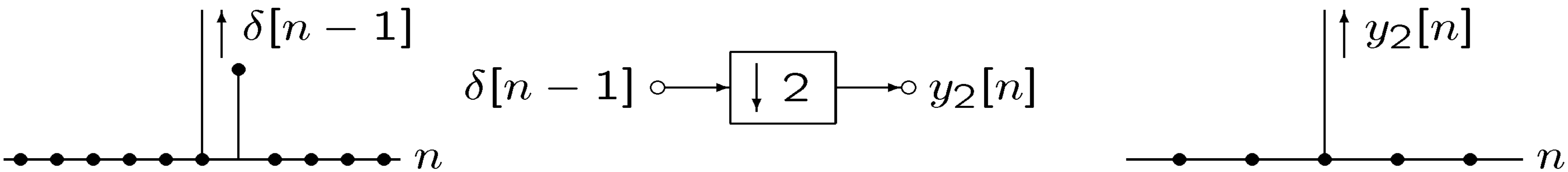


$$T_2 = RT_1$$

An SRD is a time-varying system; because:



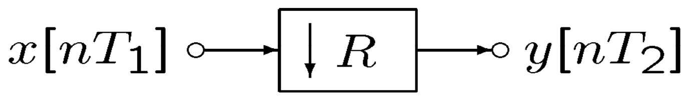
and a shifted pulse gives:



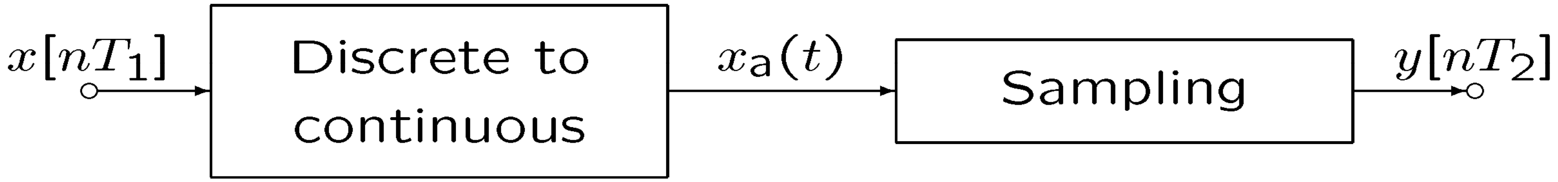
An SRD cannot be described by: $h[n]$, $H(e^{j\theta})$, $H(z)$ or by poles and zeros.

Find the relation between the spectra of the input and the output

Downsampling by a factor R:

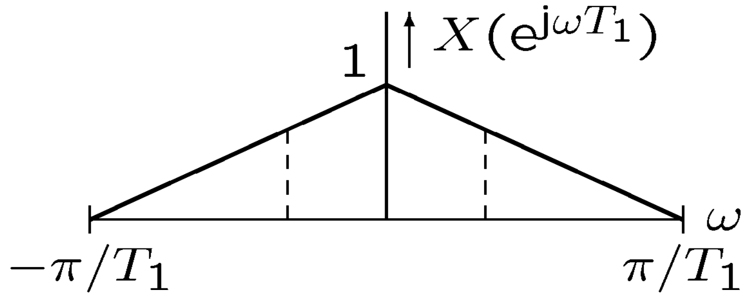


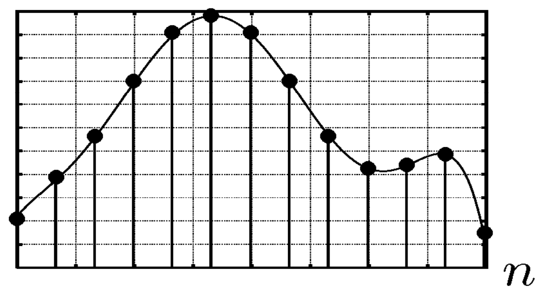
An SRD is equivalent with this unpractical system:



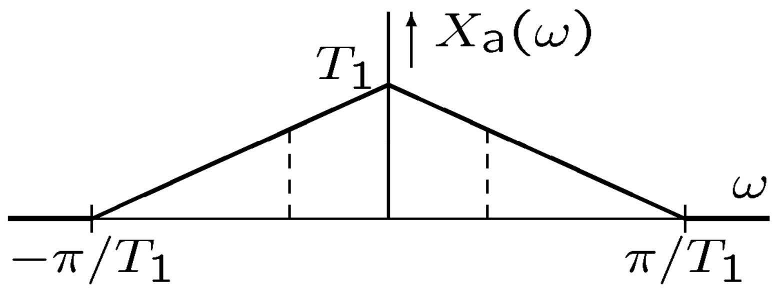
Use the lower circuit to determine the spectrum of the output signal $y[nT_2]$

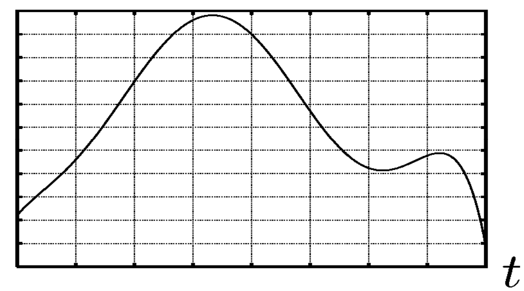
Original signal:



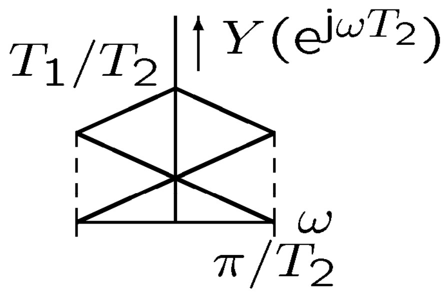


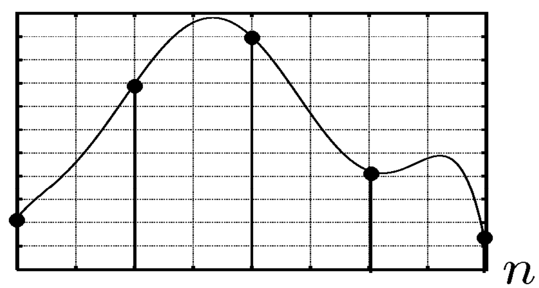
Analog signal:





Downsampled signal:





Spectrum $Y(e^{j\omega T_2})$ equals T_1/T_2 times the sum of 3 components of $X(e^{j\omega T_1})$:

$$Y(e^{j\omega T_2}) = \frac{T_1}{T_2}\Big(X(e^{j\omega T_1}) + X(e^{j(\omega-2\pi/T_2)T_1}) + X(e^{j(\omega-4\pi/T_2)T_1})\Big)$$

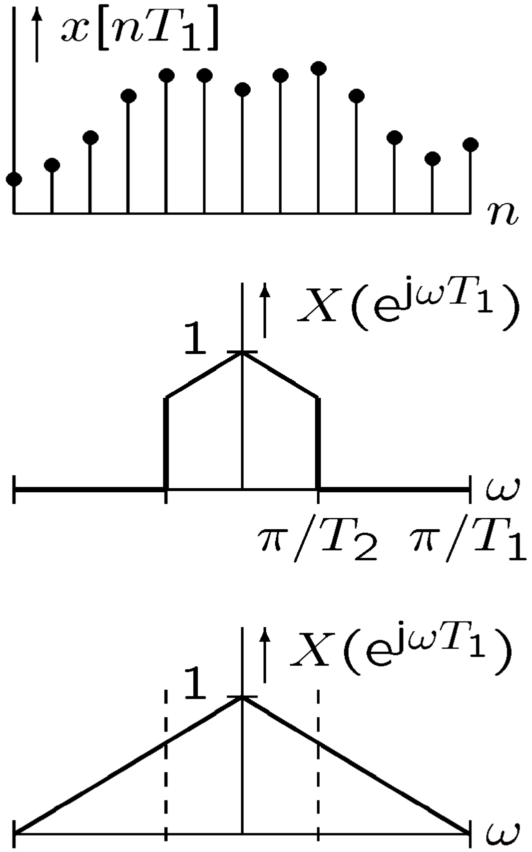
$$= \frac{1}{3}\sum_{i=0}^{2} X(e^{j(\omega T_1 - i2\pi/3)})$$

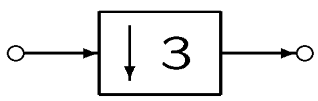
In general: $$Y(e^{j\omega T_2}) = \frac{1}{R}\sum_{i=0}^{R-1} X\left(e^{j(\omega T_1 - i2\pi/R)}\right)$$

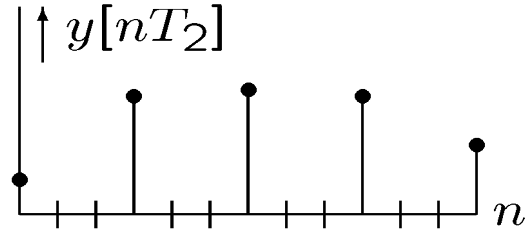
If the input spectrum is zero for $\pi/T_2 \leq |\omega| \leq \pi/T_1$, we get:

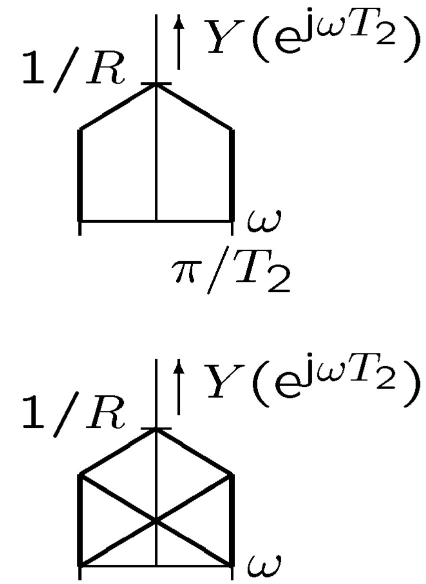
$$Y(e^{j\omega T_2}) = \frac{1}{R}X(e^{j\omega T_1})$$

No aliasing if the input spectrum is limited to $|\omega| \leq \pi/T_2$:



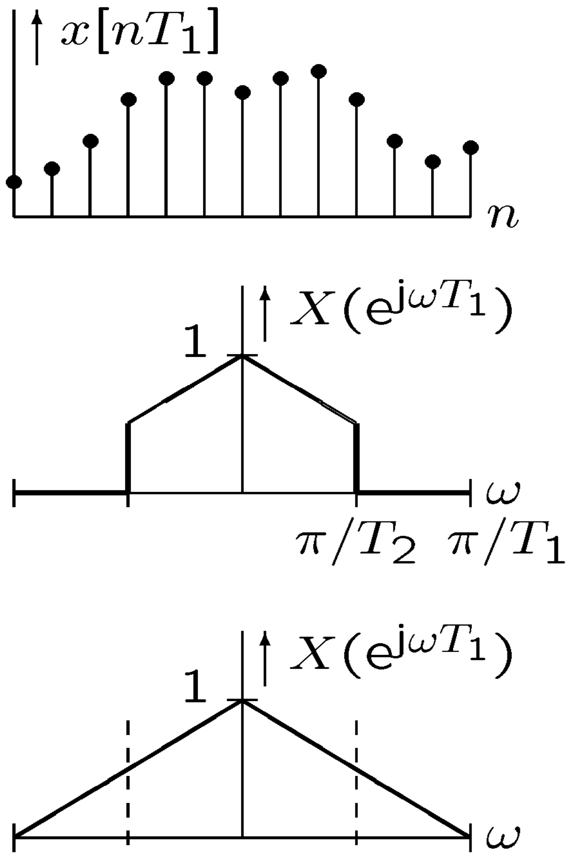


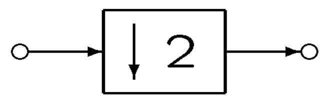


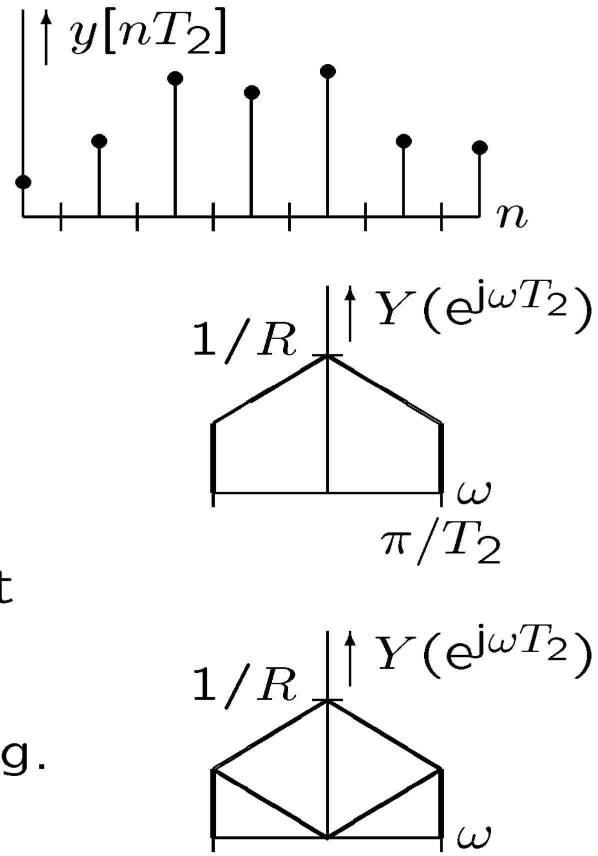


If spectrum is not band-limited, the SRD gives aliasing. Use prefilter

The same for downsampling with $R = 2$:

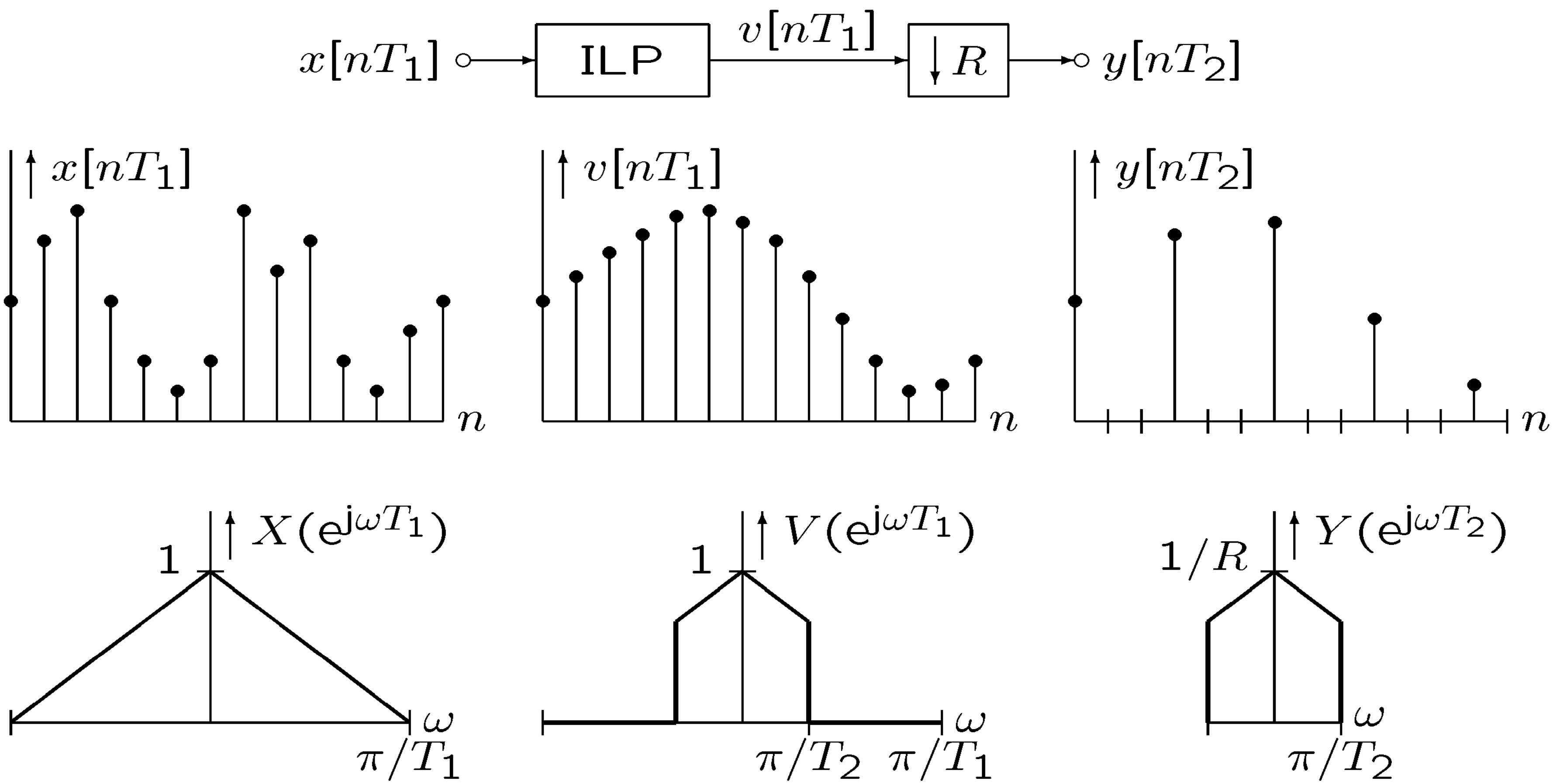




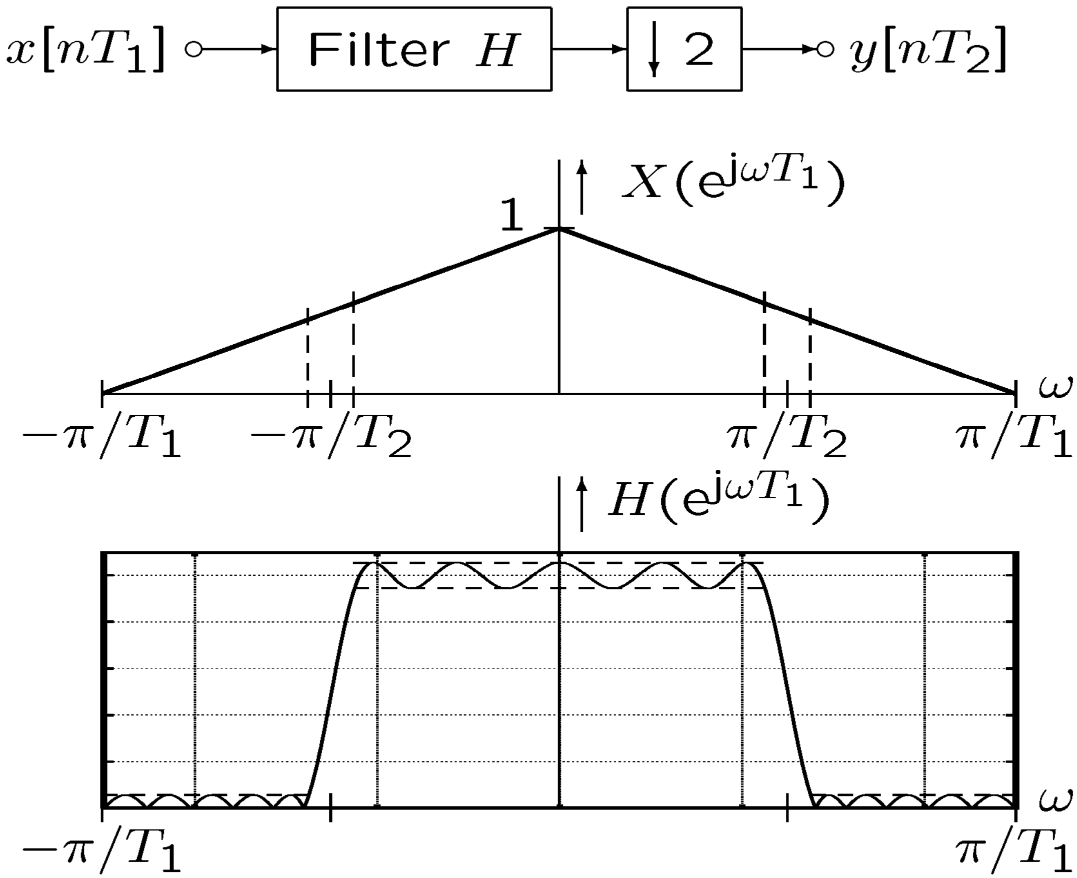


If spectrum is not band-limited, the SRD gives aliasing. Use prefilter

A decimator is an ideal lowpass filter (ILP) followed by an SRD:

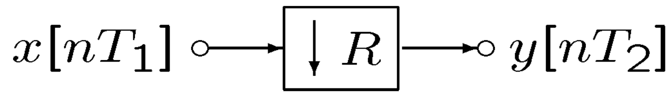


A decimating filter is a digital lowpass filter followed by an SRD:



Summary downsampling

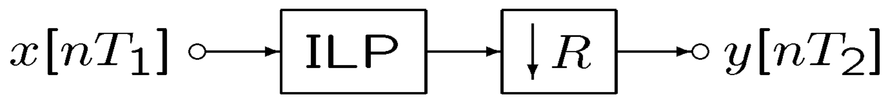
Sampling rate decreaser or SRD:



If input signal is not band-limited, then SRD introduces aliasing

---

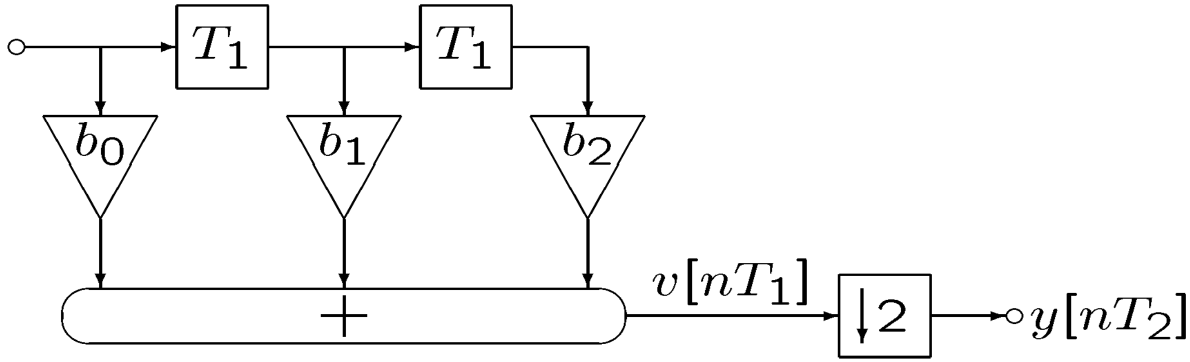
The ideal decimator uses an ideal lowpass filter ILP:



The decimator does not give any aliasing

The practical decimating filter uses a realizable FIR or IIR filter instead of a non-realizable ideal filter. Always some aliasing

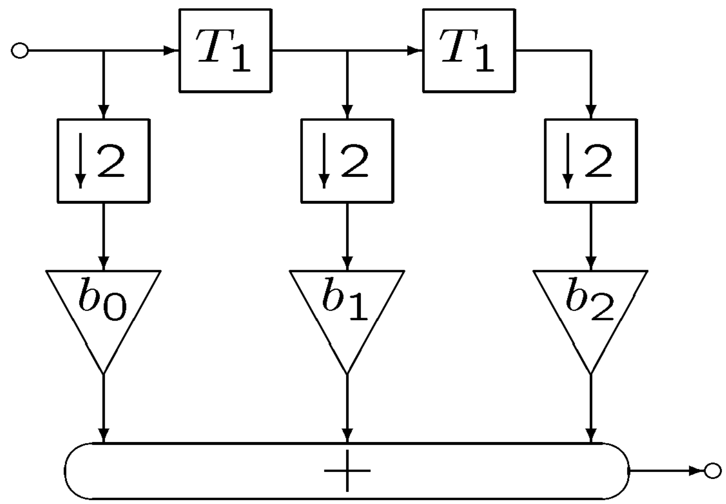
Decimating FIR filter:



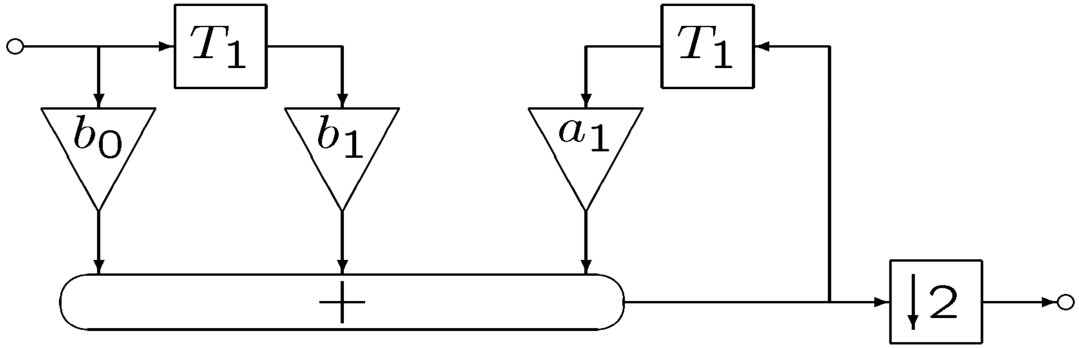
We write:

$$y[nT_2] = v[2nT_1] = b_0x[2nT_1] + b_1x[2nT_1 - T_1] + b_2x[2nT_1 - 2T_1]$$

Half the number of multiplications. Also half number of delays; see later



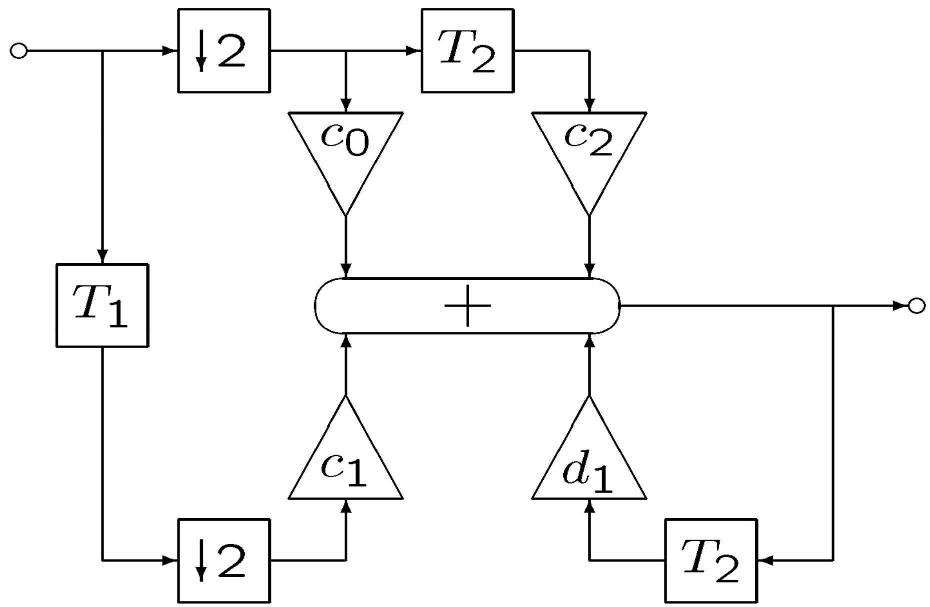
A decimating recursive filter:



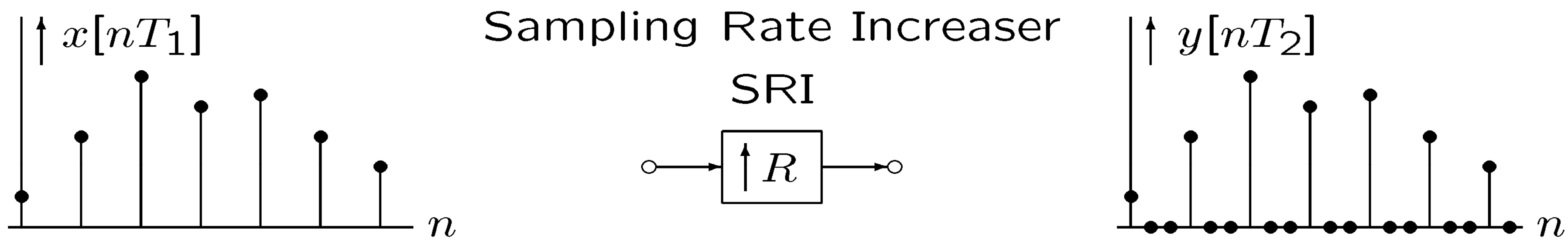
Equivalent circuit; to be shown later:

$$c_0 = b_0$$
$$c_1 = b_1 + a_1 b_0$$
$$c_2 = a_1 b_1$$
$$d_1 = a_1^2$$

Less multiplications 2/3



4. Sampling rate increase with integer factor R



$$y[nT_2] = \begin{cases} x[nT_1/R] & \text{for } n = kR; \\ 0 & \text{elsewhere} \end{cases}$$

What is the relation between $Y(e^{j\omega T_2})$ and $X(e^{j\omega T_1})$:

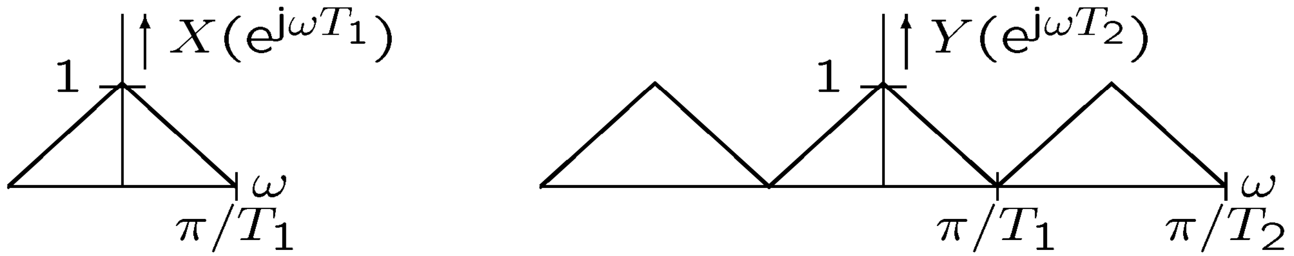
$$Y(e^{j\omega T_2}) = \sum_{n=-\infty}^{\infty} y[nT_2]e^{-jn\omega T_2} = \sum_{n=kR} x[nT_1/R]e^{-jn\omega T_2}$$

Substitute $n = iR$: $\quad Y(e^{j\omega T_2}) = \sum_{i=-\infty}^{\infty} x[iT_1]e^{-jiR\omega T_2}$

From the foregoing slide:

$$Y(e^{j\omega T_2}) = \sum_{i=-\infty}^{\infty} x[iT_1]e^{-ji R\omega T_2} = \sum_{i=-\infty}^{\infty} x[iT_1]e^{-ji\omega T_1} = X(e^{j\omega T_1})$$

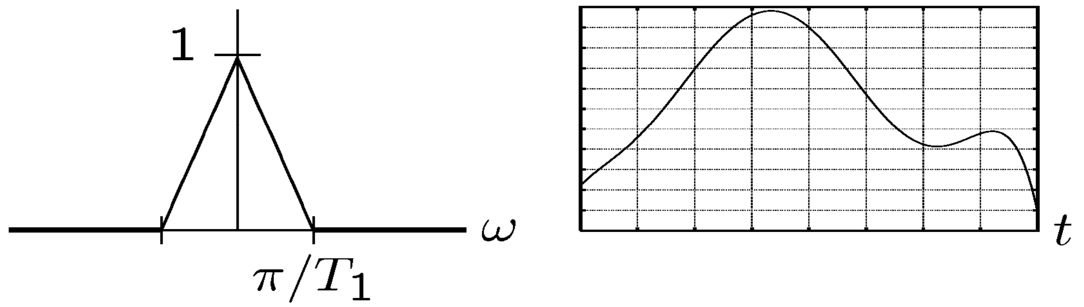
Introducing zeros gives the same spectrum; the fundamental interval is R times larger. For $R = 3$:



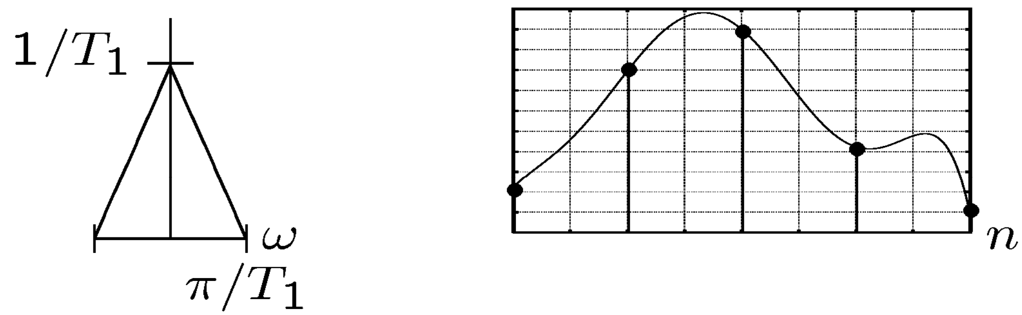
Cascade SRI by lowpass filter to suppress images. We get an interpolator:

$$H(e^{j\omega T_2}) = \begin{cases} R & \text{for } |\omega| \leq \pi/T_1 \\ 0 & \text{for } \pi/T_1 \leq |\omega| \leq \pi/T_2 \end{cases}$$

The analog signal:

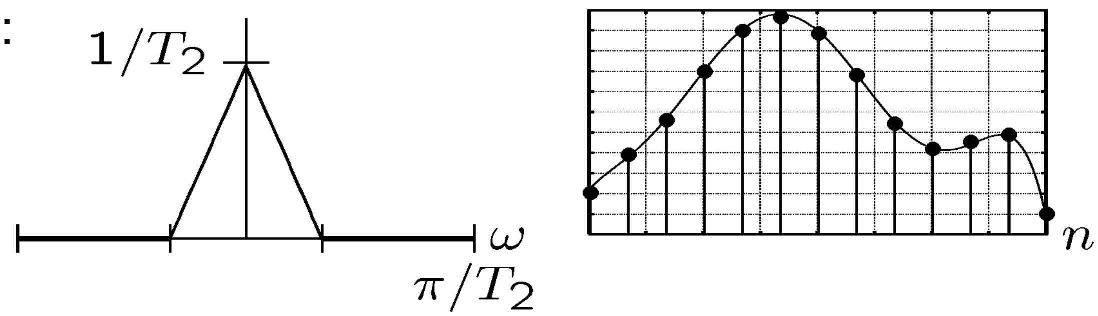


Sampling by $1/T_1$:

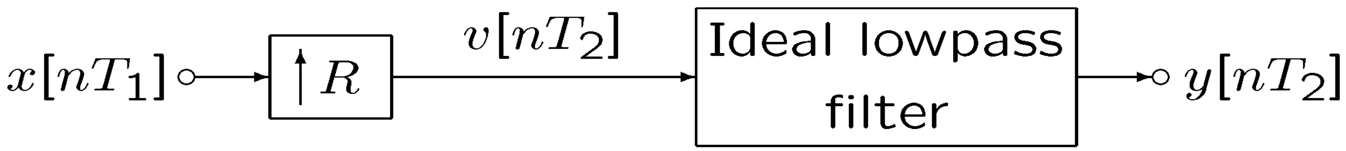


Sampling by $1/T_2 = 3/T_1$:

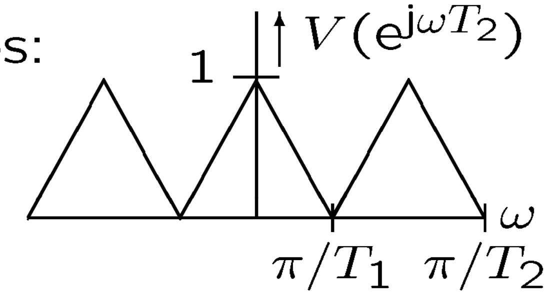
Interpolation: insert zeros and pass signal through lowpass filter

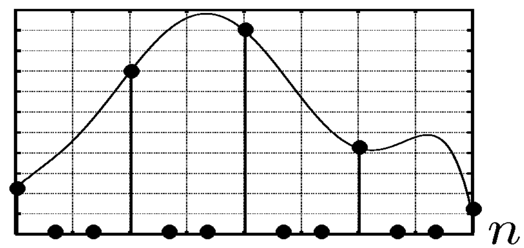


The ideal interpolator

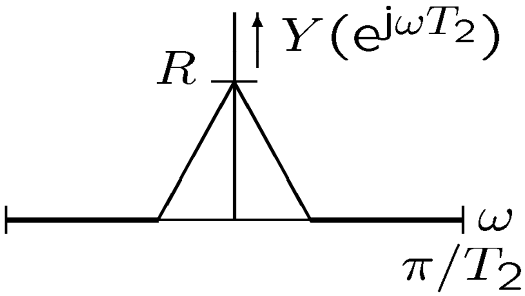


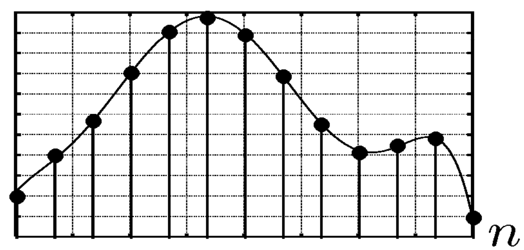
Introduce zeros:



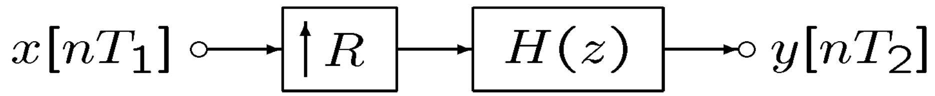


Ideal filter:

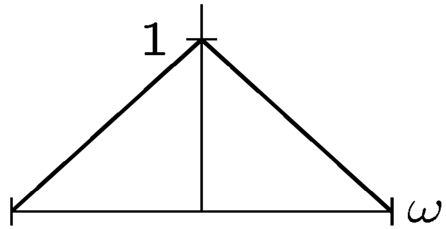




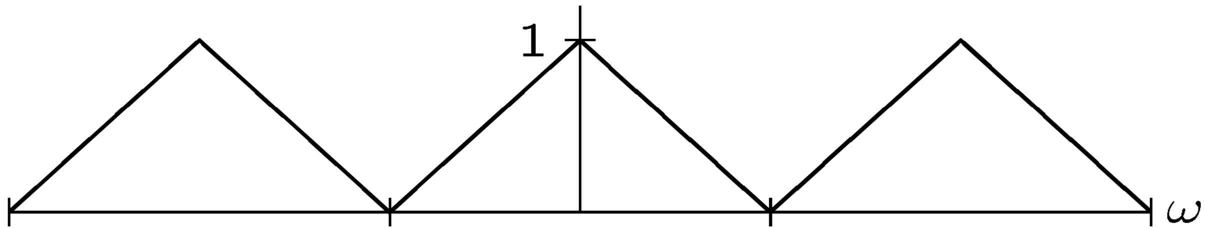
Interpolating filter:



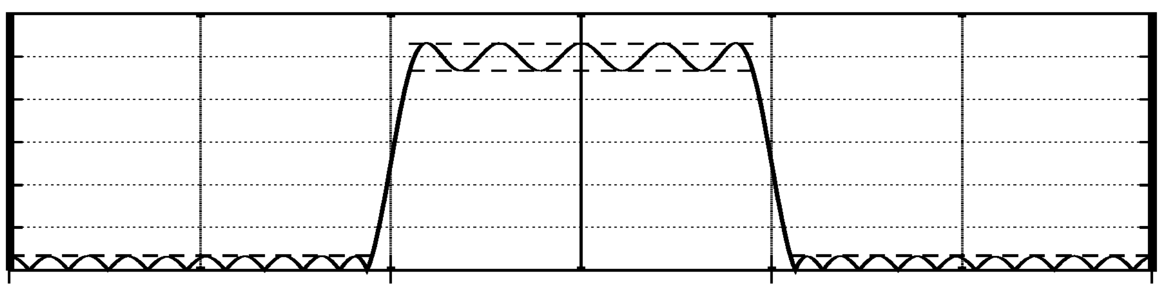
Input:



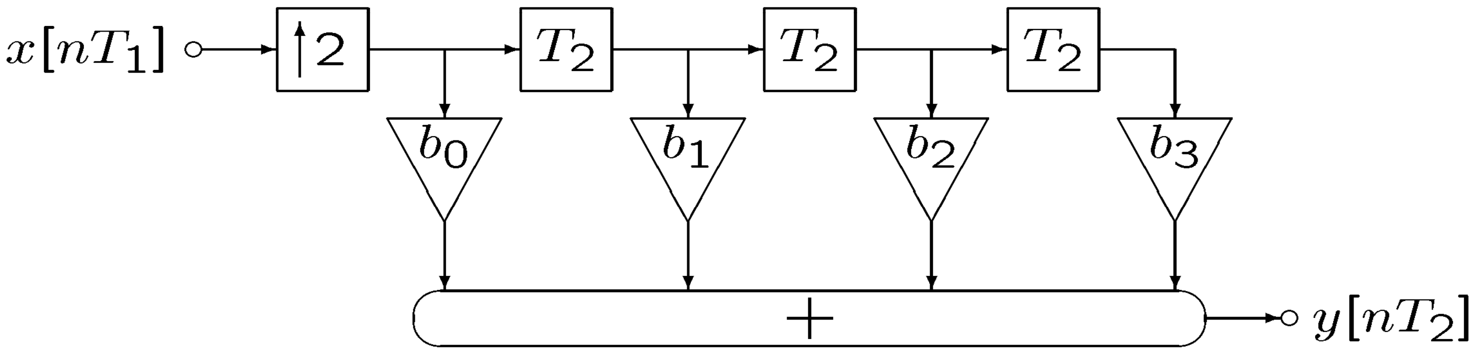
Introduce zeros:



Non-ideal filter:



Interpolating transversal filter ;



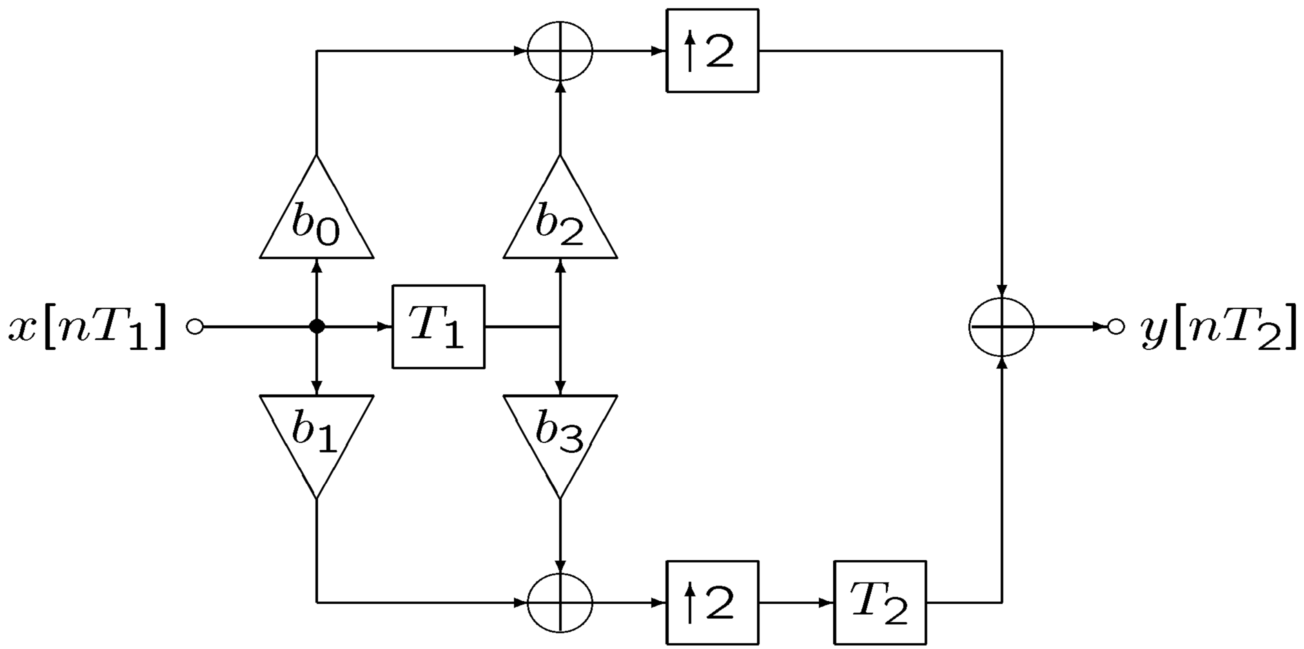
For the output signal $y[nT_2]$, we find:

$$y[2nT_2] = b_0 x[nT_1] + b_2 x[(n-1)T_1]$$

$$y[(2n+1)T_2] = b_1 x[nT_1] + b_3 x[(n-1)T_1]$$

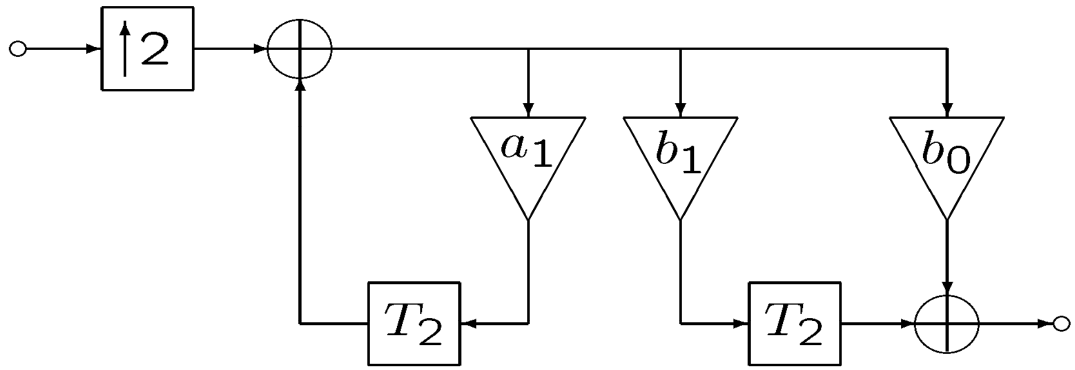
Later, we will formalize this using the noble and prime identities and the polyphase decomposition

The foregoing two equations can be implemented by the following digital structure:

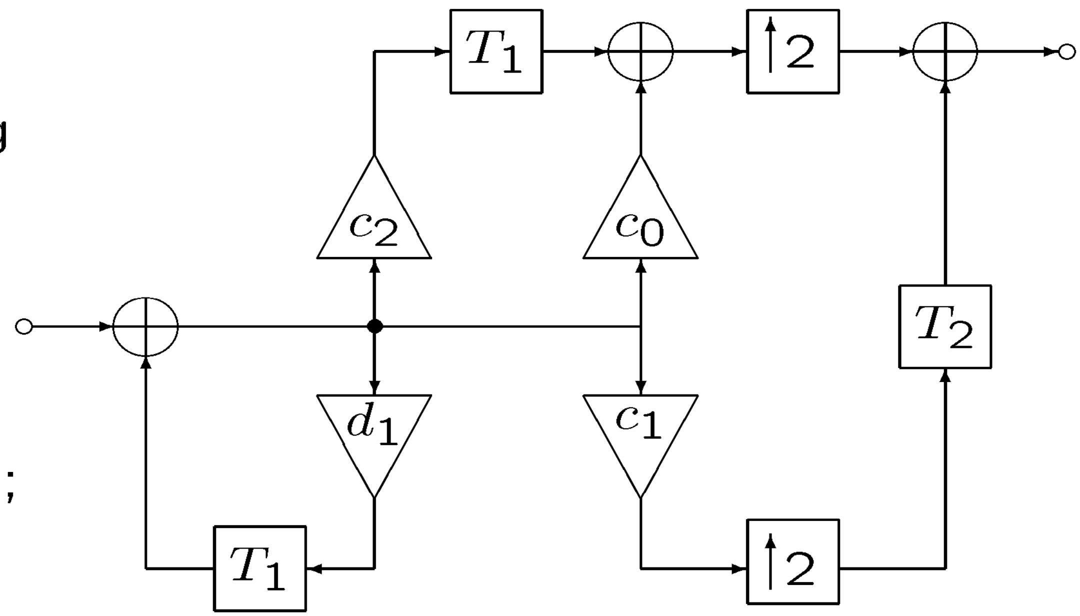


On the next slide we give some information about recursive interpolating filters

Interpolating first order recursive filter with $6F_s$ multiplications



Alternative circuit found by transposing a foregoing circuit; this gives $4F_s$ multiplications



$c_0 = b_0; \quad c_1 = b_1 + a_1 b_0;$

$c_2 = a_1 b_1; \quad d_1 = a_1^2$

Decreasing the sampling rate by an integer factor R:

- SRD, decimator and decimating filter

Increasing the sampling rate by an integer factor R:

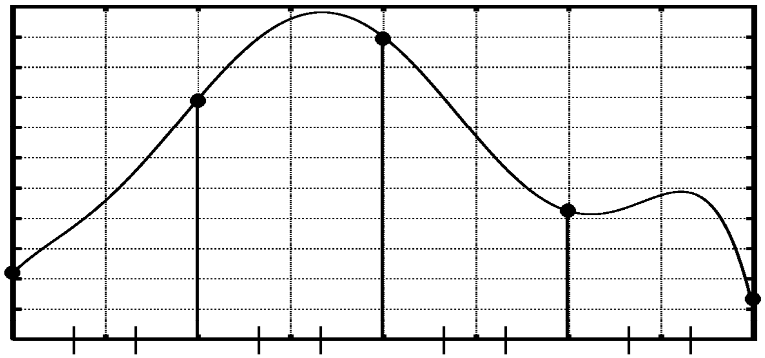
- SRI, interpolator and interpolating filter

Changing the sampling rate by a non-integer factor: R_1/R_2

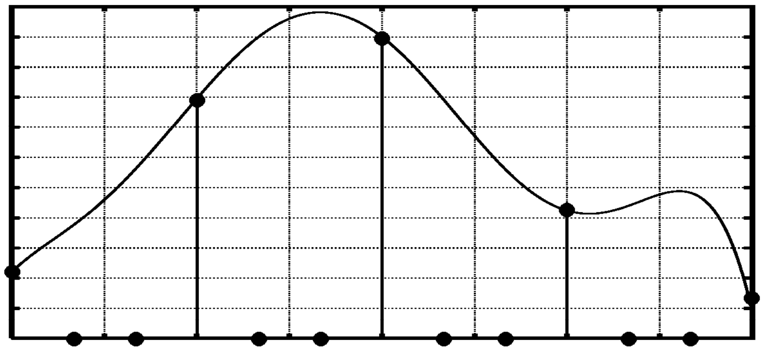
Introduce $R_1 - 1$ zeros; interpolate the signal by a low pass filter. For the bandwidth of this filter; see the examples. Decimate the interpolated signal. Economic structures are given later

Increase the sampling rate by 3/2

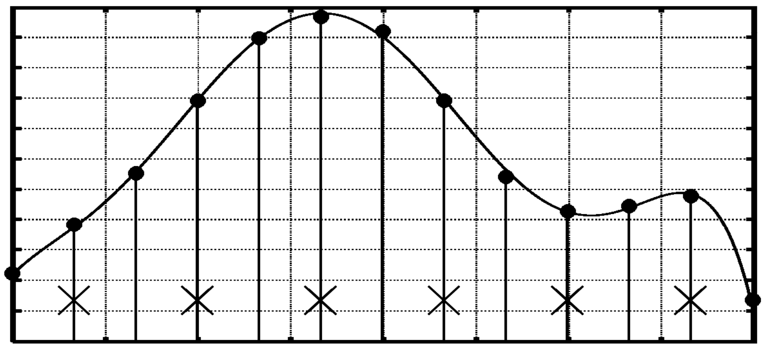
Input:



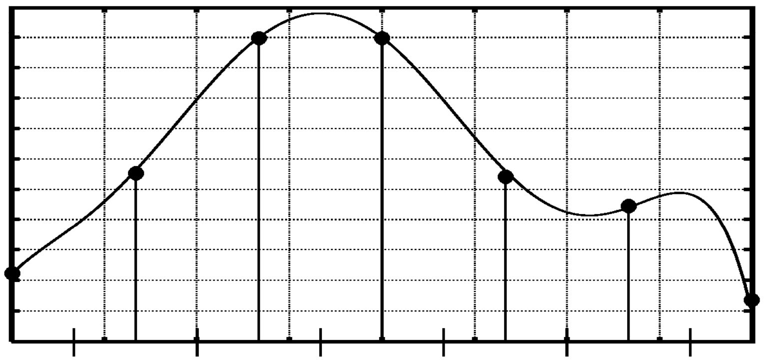
Introduce 2 zeros ($R_1 = 3$):



Filter:



Decimate by 2 ($R_2 = 2$):

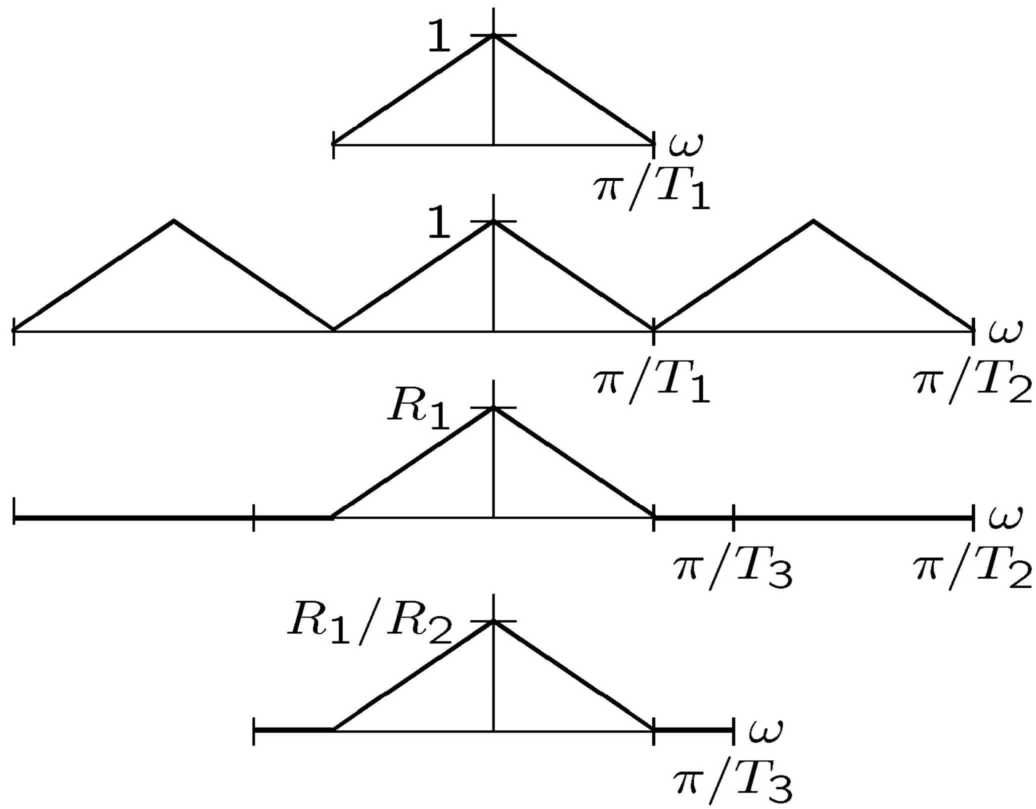


input spectrum:

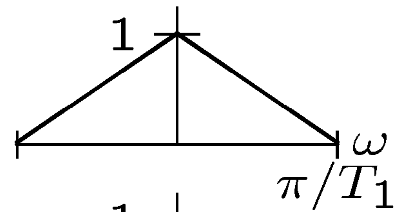
interpolate by 3:

ideal lowpass filter:

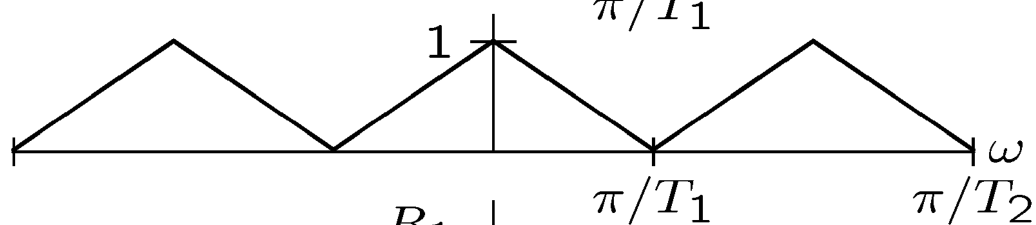
decimate by 2:



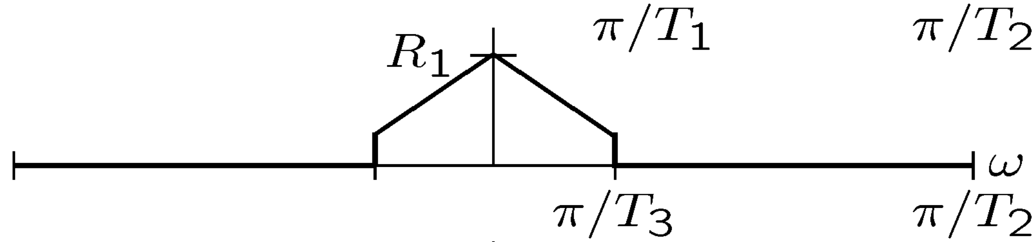
input spectrum:



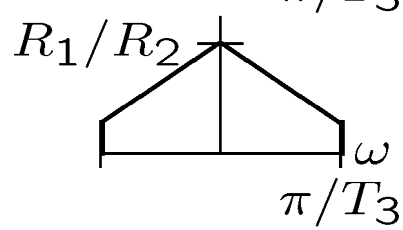
interpolate by 3:



ideal lowpass filter:

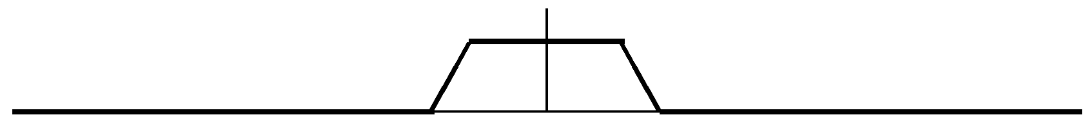


decimate by 4:

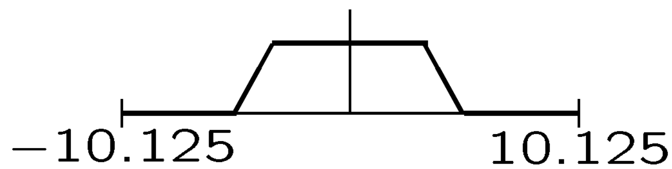


Video sampling rate conversion by 2/3; from 20.25 to 13.5 MHz:

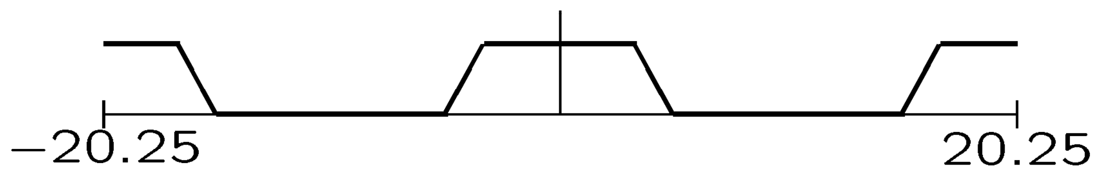
analog input spectrum:



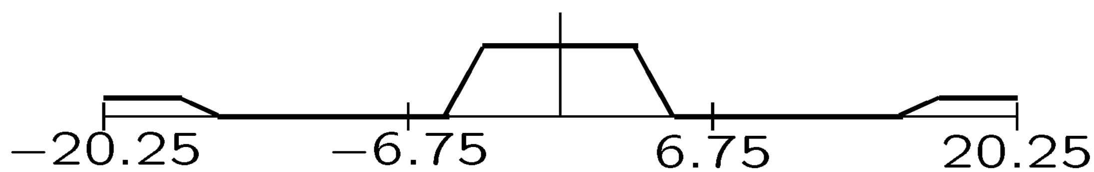
A/D with 20.25 MHz:



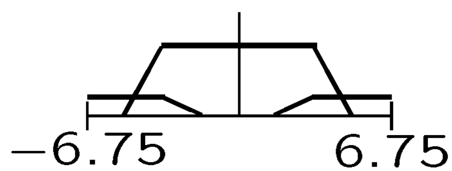
SRI with $R_1 = 2$:



the interpolating filter:

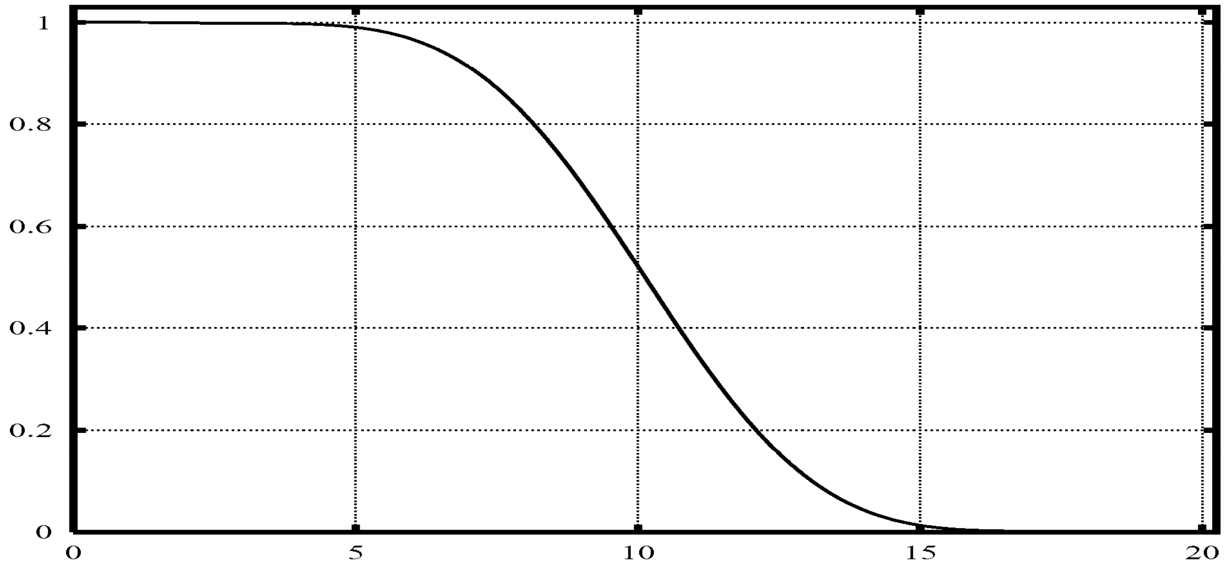


decimation by 3 gives some aliasing:



The impulse response for a time domain interpolation is:

$$h[0] = h[14] = -5 \qquad h[4] = h[10] = -245$$
$$h[1] = h[13] = 0 \qquad h[5] = h[9] = 0$$
$$h[2] = h[12] = 49 \qquad h[6] = h[8] = 1225$$
$$h[3] = h[11] = 0 \qquad h[7] = 2048$$



14 delay elements

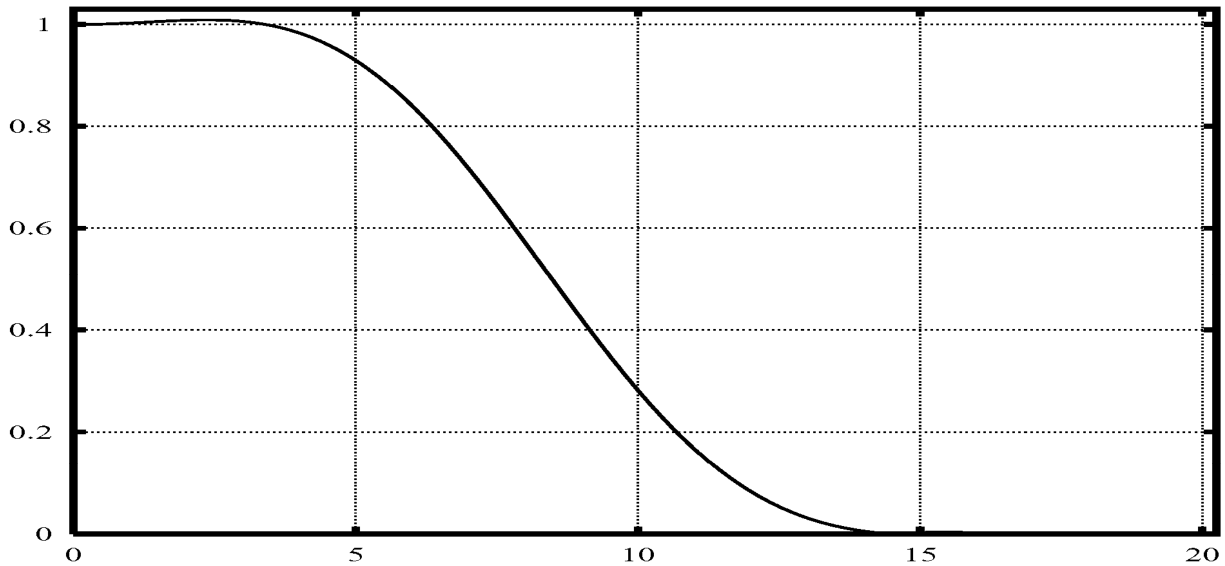
5 multiplications

8 additions

12 bits coefficients

Design in the frequency domain:

$$h[0] = h[8] = -5$$
$$h[1] = h[7] = -10$$
$$h[2] = h[6] = 15$$
$$h[3] = h[5] = 74$$
$$h[4] = 108$$



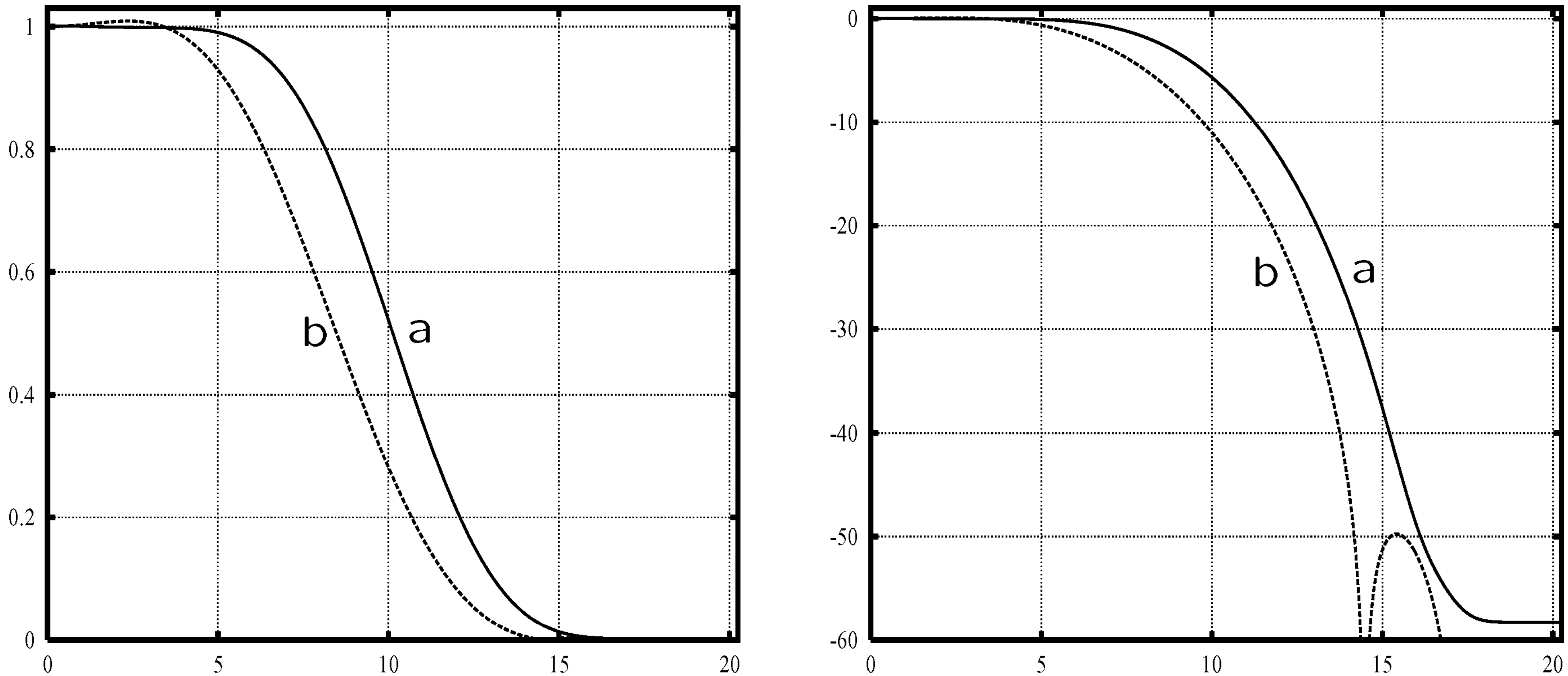
8 delay elements

5 multiplications

8 additions

8 bits coefficients

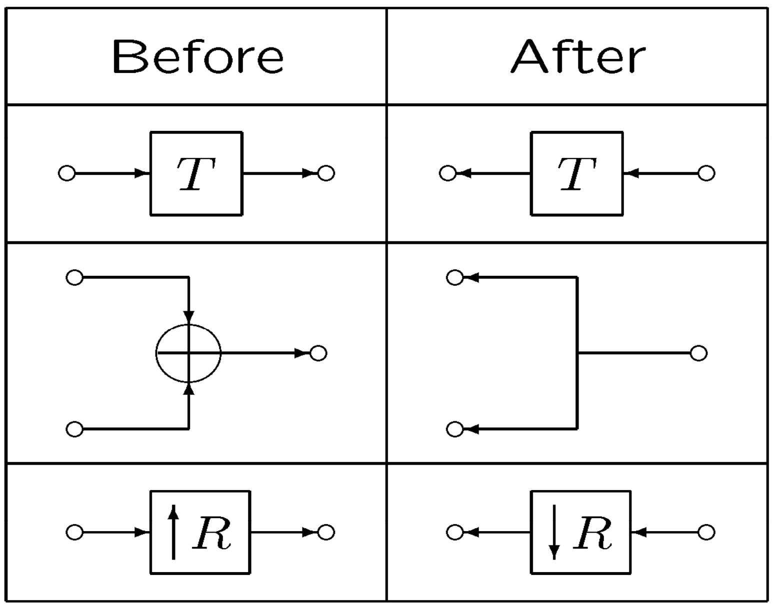
A comparison between a time domain design (a) and a frequency domain design (b)

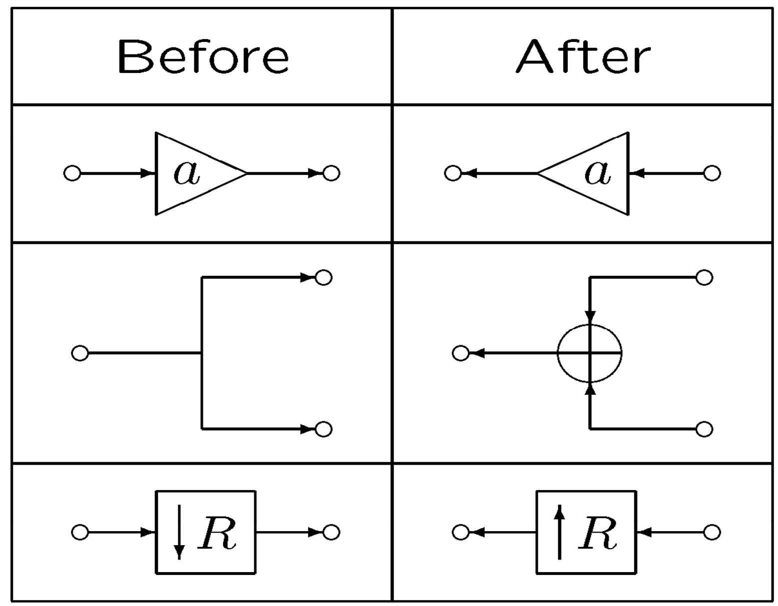


5. The transposition theorem

Three transposition rules:

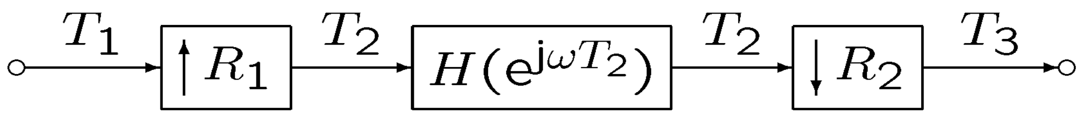
1. reverse signal flow
2. interchange nodes and adders
3. interchange SRIs and SRDs



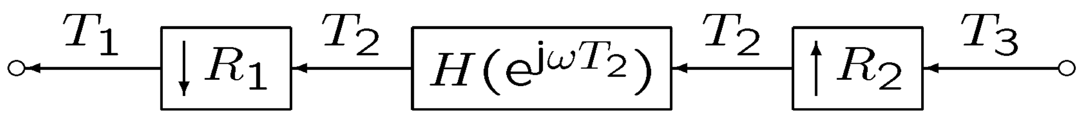


What does this mean for interpolating and decimating filters?

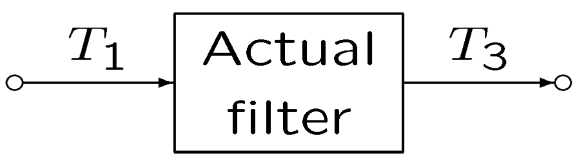
The original filter:



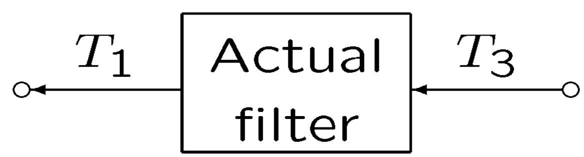
transforms into:



Ratio changes from R_1/R_2 to R_2/R_1: decimation transforms to interpolation and vice versa. The same for an efficient circuit.

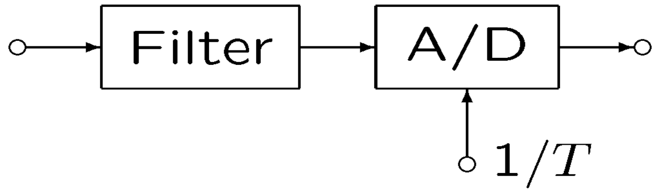


transforms to:



The frequency responses of both filters have to be equal

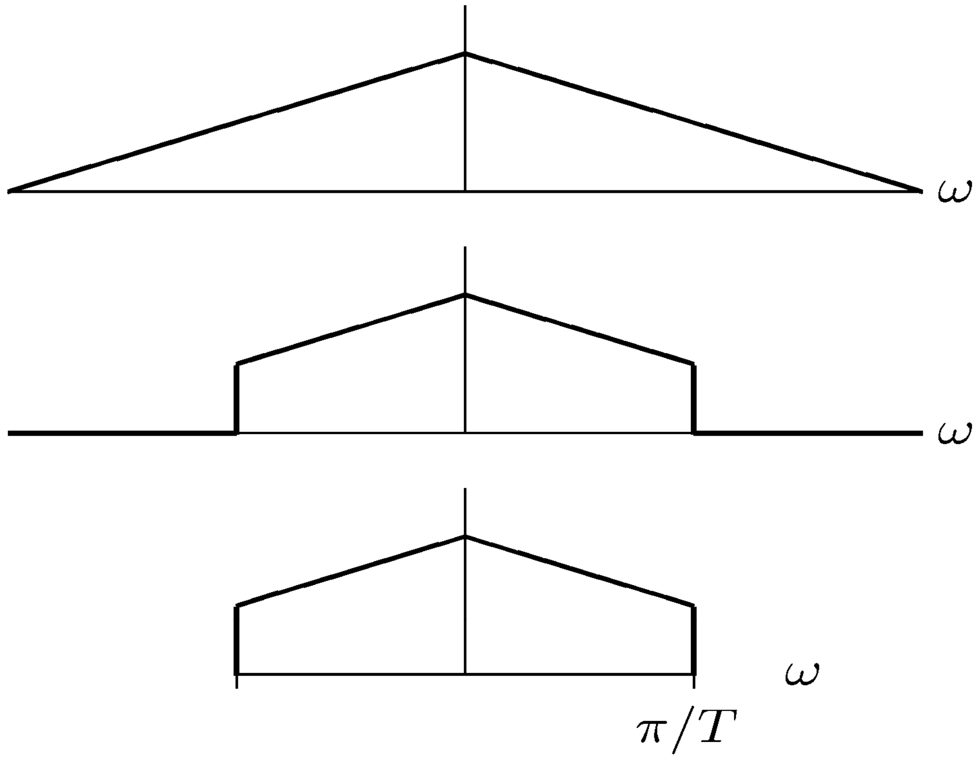
6. A conventional A/D converter:



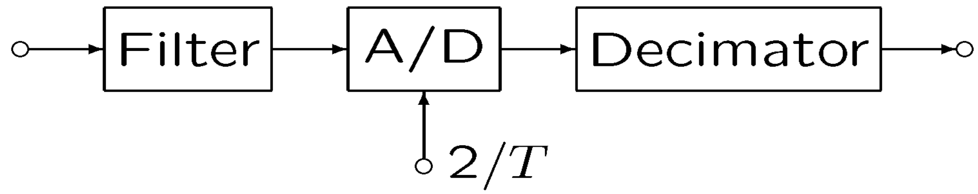
input spectrum:

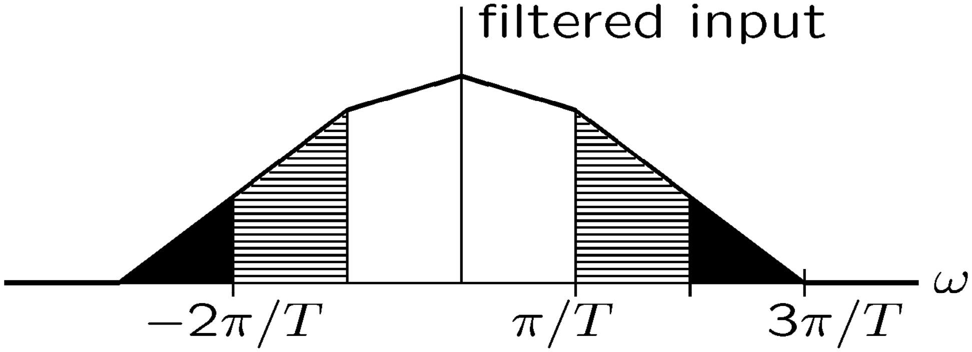
filtered spectrum:

output spectrum:

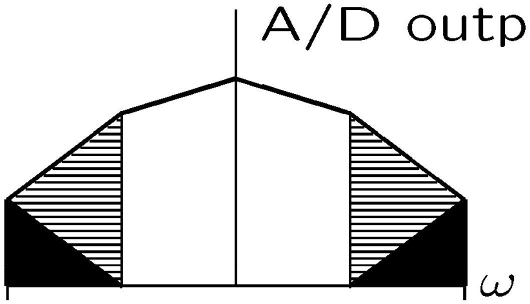


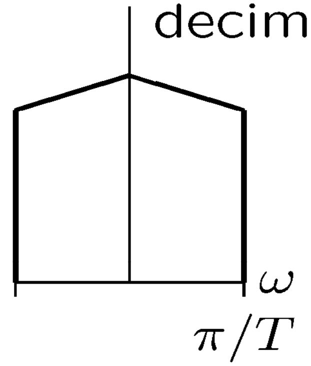
Alternative A/D converter:



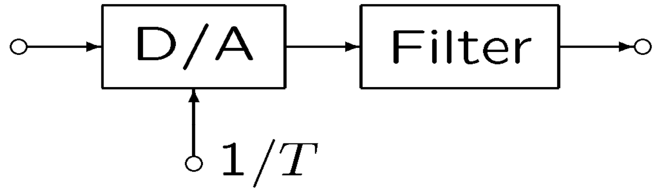


Transition band from $\pi/T \leq |\omega| \leq 3\pi/T$:

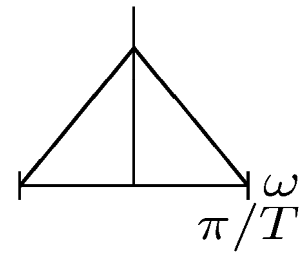




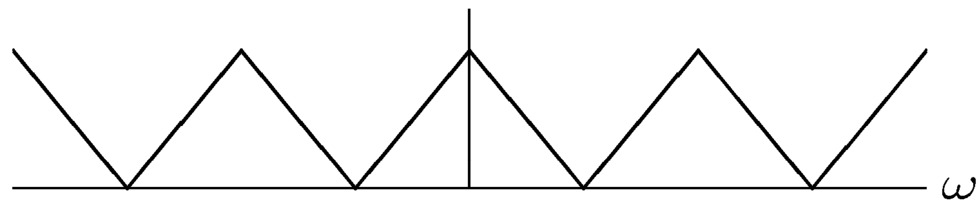
Conventional D/A converter:



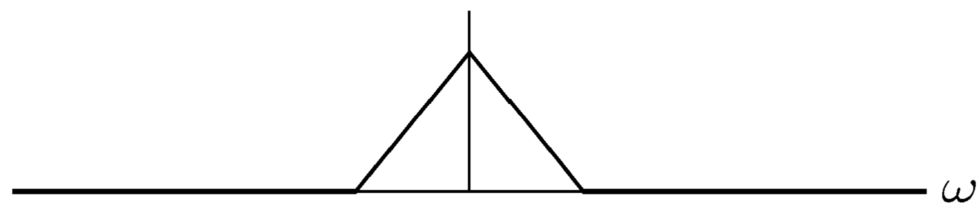
input spectrum:



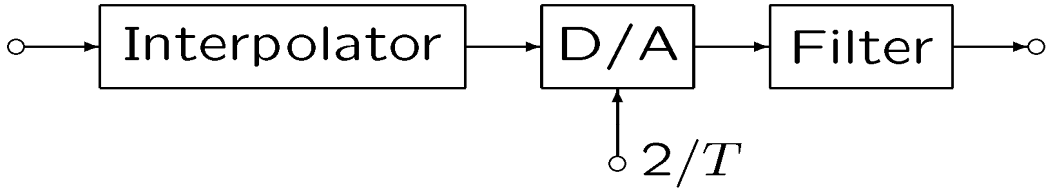
unfiltered spectrum:

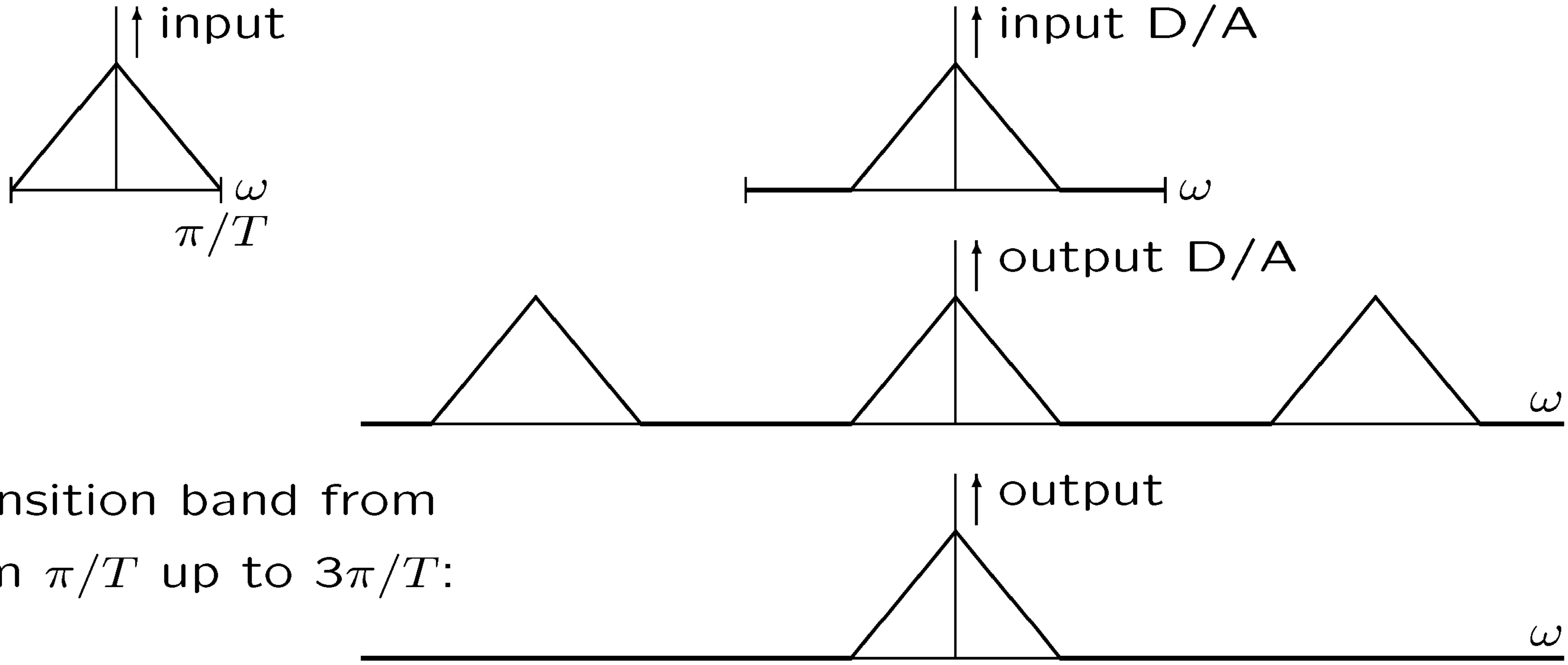


output spectrum:



Alternative D/A:





Transition band from

from π/T up to $3\pi/T$: