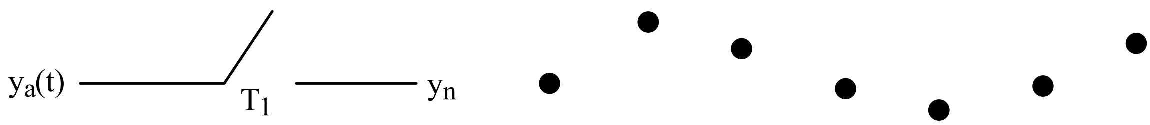
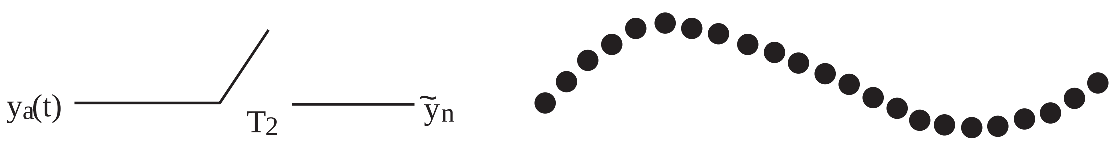
ECE 410 **DIGITAL SIGNAL PROCESSING** **D. Munson**
University of Illinois **Chapter 13**

Digital Interpolation

Suppose have $y_n = y_a(nT_1)$ as pictured:



and want $\tilde{y}_n = y_a(nT_2)$ where $T_2 = \frac{T_1}{L}$, and L is integer:



Thus, $\{\tilde{y}_n\}$ are denser samples of $y_a(t)$. How do we get $\{\tilde{y}_n\}$ from $\{y_n\}$? Could use

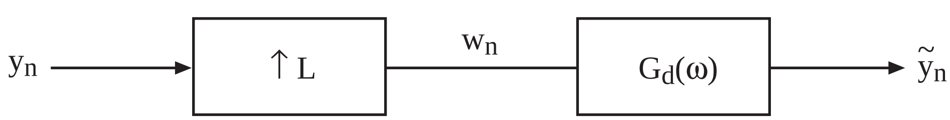
$$y_a(t) = \sum_{k=-\infty}^{\infty} y_k \ \text{sinc}\left[\frac{\pi}{T_1}(t - kT_1)\right]$$

to get

$$\tilde{y}_n = y_a(nT_2) = \sum_{k=-\infty}^{\infty} y_k \ \text{sinc}\left[\frac{\pi}{T_1}(nT_2 - kT_1)\right]$$

But, this involves an infinite sum (which must be truncated in practice) and evaluation of the sinc function. Alternatively, we might try something simpler, such as a piecewise linear or polynomial approximation to $y_a(t)$, but these methods are not particularly accurate.

Alternative Digital Approach

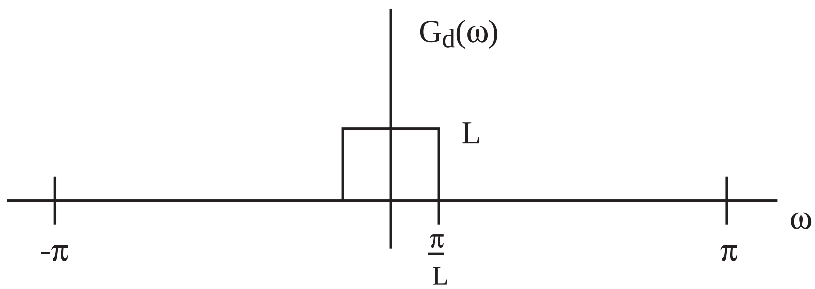


where the first box is an up-sampler that inserts L-1 zeros between each pair of inputs:

13.2

$$w_n = \{0, 0, \dots, 0, y_{-1}, \underbrace{0, 0, \dots, 0}_{\text{L-1 zeros}}, y_0, 0, 0, \dots, 0, y_1, 0 \dots\}$$

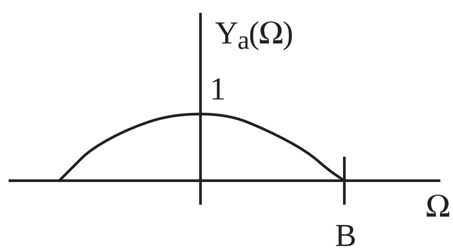
and $G_d(\omega)$ is an ideal LPF with cutoff π/L and passband gain L:



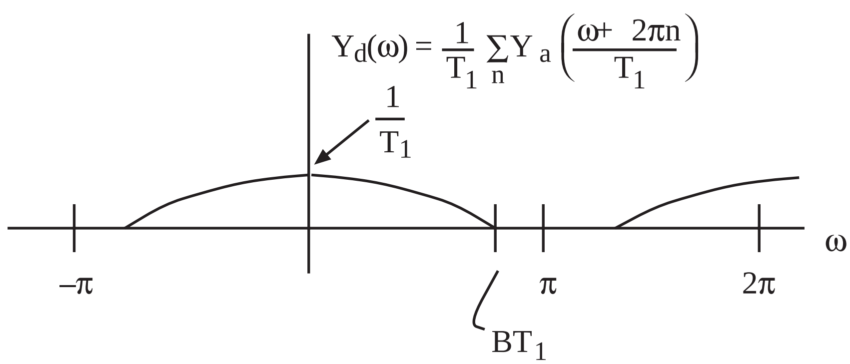
Why does this work??

Analyze the problem in the Fourier domain.

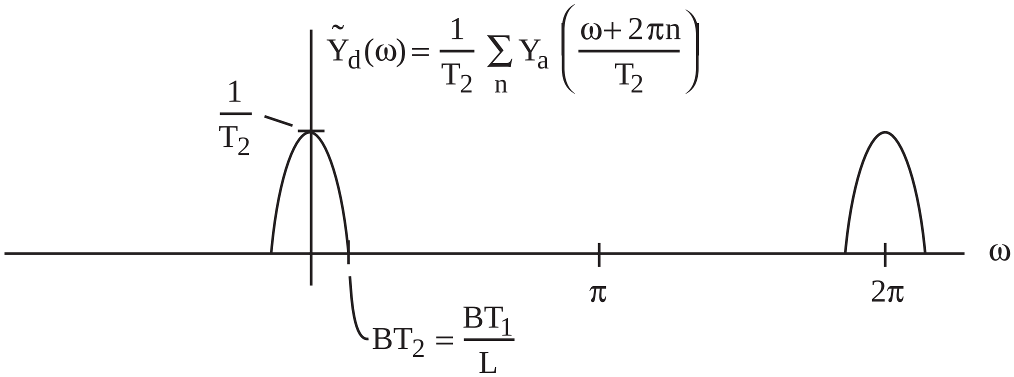
First, note that if



then sampling $y_a(t)$ at times nT_1 $\left(\text{with } T_1 < \frac{\pi}{B}\right)$ would give $\{y_n\}$ with



Sampling $y_a(t)$ on the denser grid nT_2 would give $\{\tilde{y}_n\}$ with



(Sampling at a higher frequency <u>shrinks</u> the DTFT of the A/D output).

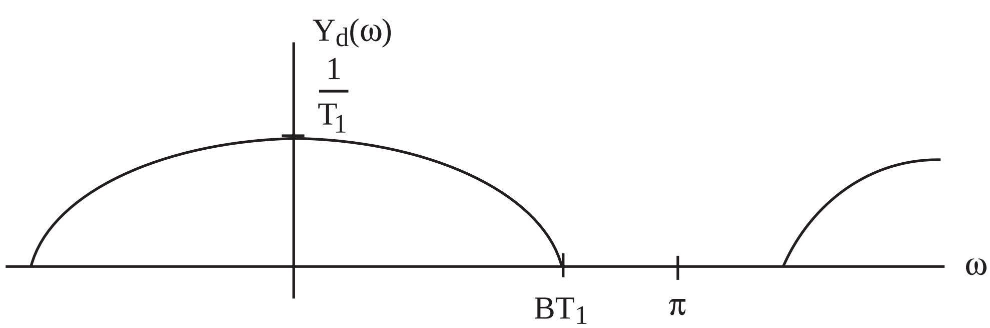
Now, show that the above digital interpolation approach gives $\tilde{Y}_d$ from Y_d (and therefore $\tilde{y}_n$ from y_n).

We have

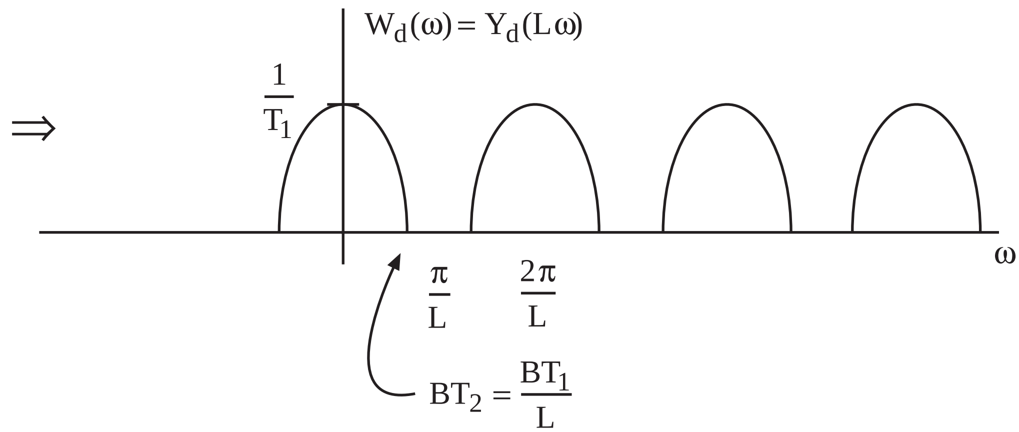
$$W_d(\omega) = \sum_n w_n \; e^{-j\omega n} = \sum_n y_n \; e^{-j\omega L n}$$

$$\Rightarrow \quad \boxed{W_d(\omega) = Y_d(L\omega)}$$

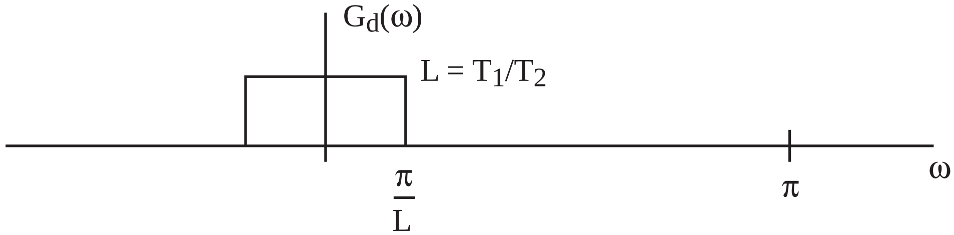
So,



13.4

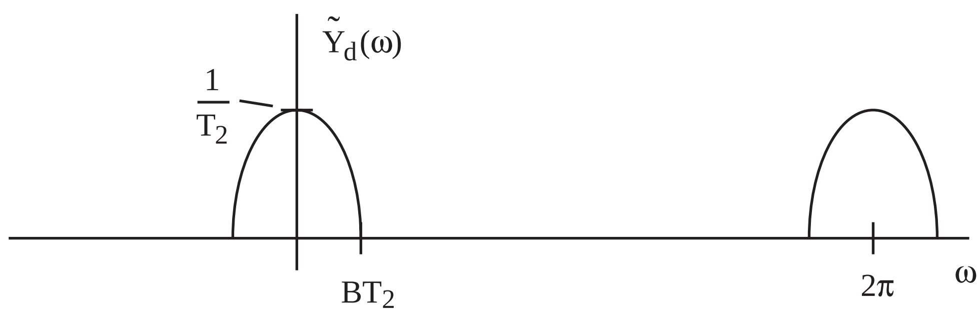


Now, since



we have

$\tilde{Y}_d(\omega) = G_d(\omega)\, W_d(\omega)$ given by



as desired. Thus, in principle, the digital interpolator will compute $\{\tilde{y}_n\}$ from $\{y_n\}$ exactly. In practice, the quality of the interpolator depends on the quality of $G_d(\omega)$), i.e., on how close $G_d(\omega)$ is to the ideal low-pass shape.

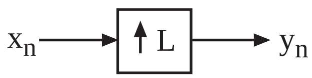
Comments:

1) L-1 out of every L inputs to $G_d(\omega)$ are zero. This saves many multiplications for L large! This is readily apparent for nonrecursive $G_d(\omega)$, but is also true for some recursive $G_d(\omega)$.

2) There exist efficient digital interpolation schemes for $T_2 = \alpha\, T_1$, where α is any real number (doesn't have to be $\frac{1}{L}$).

A Further Look at Up-Sampler

A digital interpolator uses an up-sampler as one of its components.



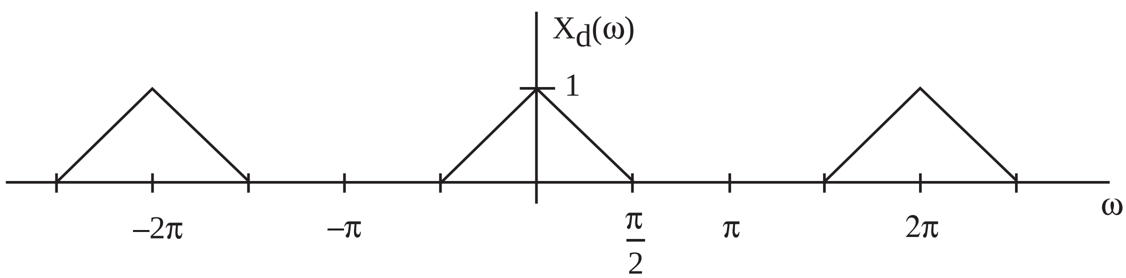
We have shown that $Y_d(\omega)$ is a <u>squashed</u> version of $X_d(\omega)$, namely

$$Y_d(\omega) = X_d(L\omega).$$

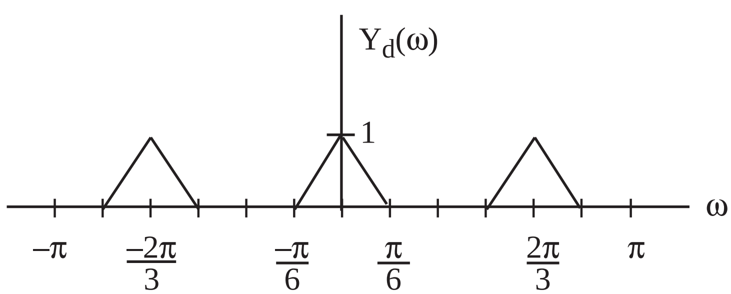
Notice that the amplitude of Y_d is the same as the amplitude of X_d. This makes intuitive sense since the energy in the y_n sequence is the same as that of the x_n sequence, because the up-sampler inserts just zeros between the x_n elements.

Example (Up-Sampler)

Suppose L = 3. Sketch $Y_d(\omega)$, assuming

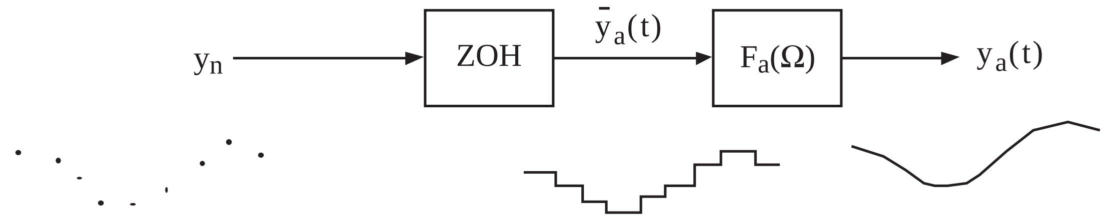


The <u>entire</u> ω axis is squashed by a factor of 3 to give

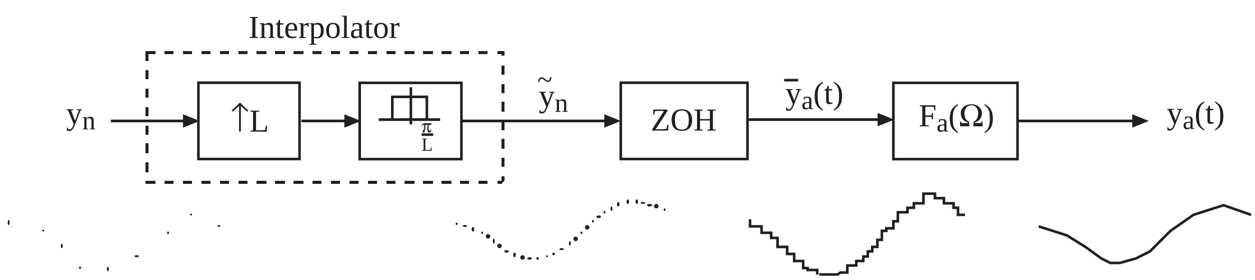


Oversampling D/A

Used in C-D players, for example. Idea is to simplify analog filter in D/A by using interpolation prior to the D/A. Interpolating $\{y_n\}$ prior to the D/A permits the use of a ZOH with a smaller step-size. This ZOH puts out a finer staircase approximation to $y_a(t)$, which relaxes the requirements on $F_a(\Omega)$. So, instead of this:

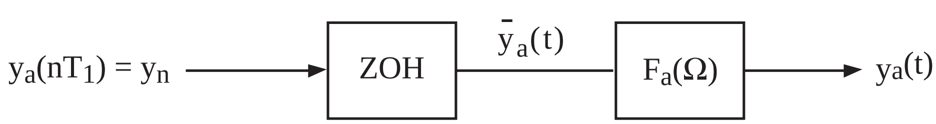


do this:



As you can imagine, a far simpler filter $F_a(\Omega)$ can be used in the second system to produce $y_a(t)$ from $\bar{y}_a(t)$, since $\bar{y}_a(t)$ is much smoother in the second system than in the first system. We gain considerable insight into this via the following analysis.

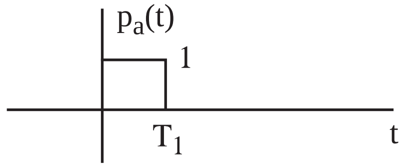
Our analysis of the oversampling D/A is facilitated by <u>first</u>, considering a usual D/A, assuming sampling period of T_1. The standard way to reconstruct $y_a(t)$ from $y_n = y_a(nT_1)$ is:



where

$$\bar{y}_a(t) = \sum_n y_n \; p_a(t - nT_1)$$

with



and

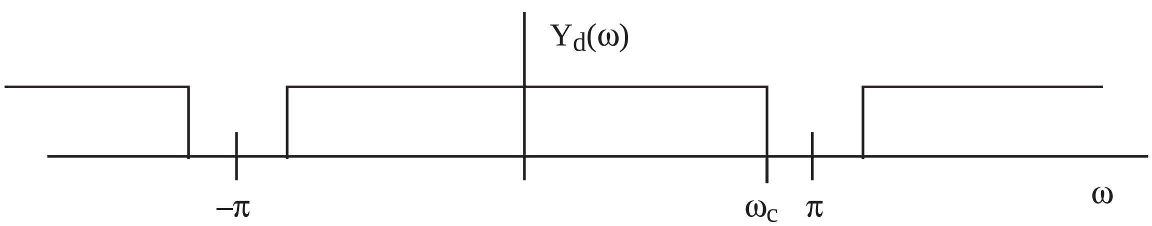
$$\bar{Y}_a(\Omega) = P_a(\Omega)\, Y_d(\Omega T_1)$$

↑ from analysis of general D/A

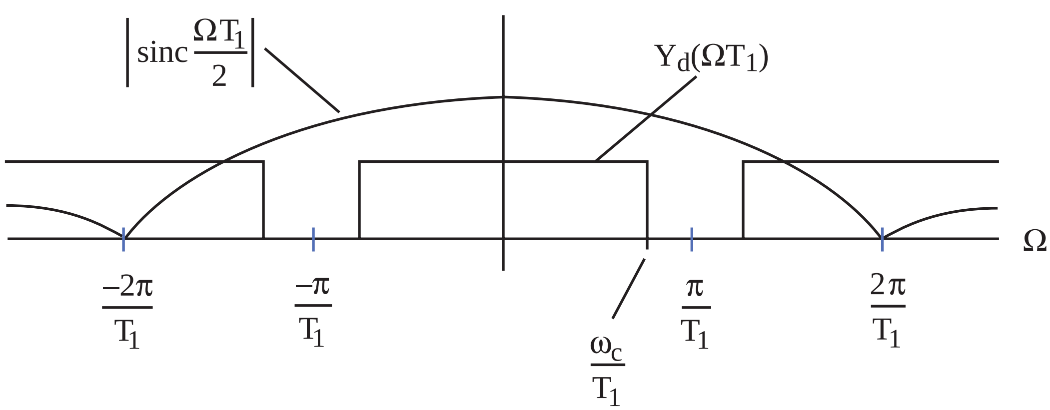
so that

$$\bar{Y}_a(\Omega) = T_1 \operatorname{sinc} \frac{\Omega T_1}{2}\, e^{-j\frac{\Omega T_1}{2}}\, Y_d(\Omega T_1) \qquad (❑)$$

As a specific example, assume

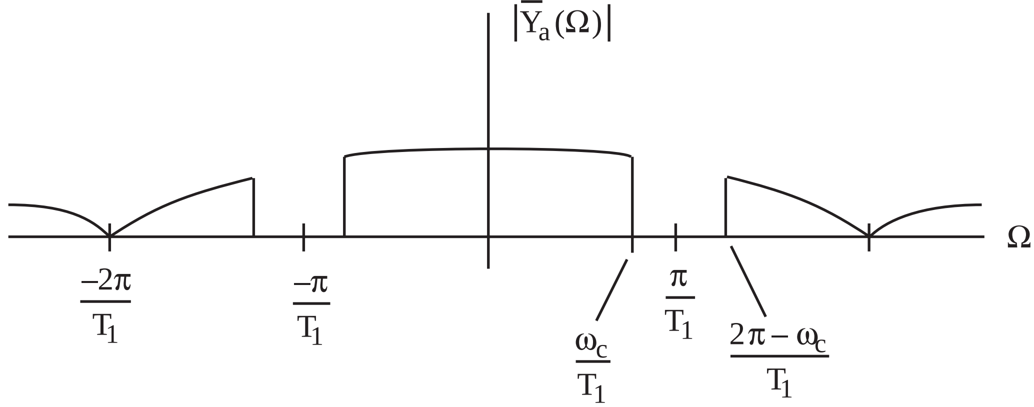


Then $|\bar{Y}_a(\Omega)|$ is the <u>product</u> of the following two curves:



giving

13.8

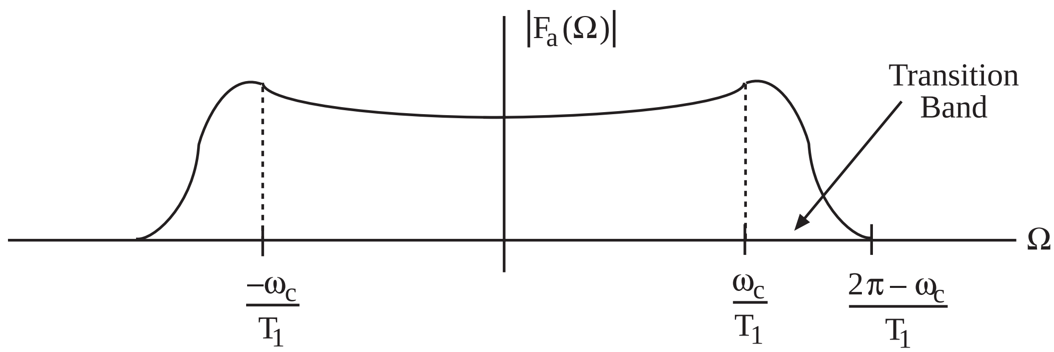


Now, as we know, $F_a(\Omega)$ should be a LPF with a

$$\frac{1}{\operatorname{sinc}\dfrac{\Omega T_1}{2}}$$

shape in its passband. For the situation above, with $\omega_c < \pi$, there is room for a transition band of $F_a(\Omega)$ on the interval $\dfrac{\omega_c}{T_1} < |\Omega| < \dfrac{2\pi - \omega_c}{T_1}$. A finite-order (realizable) $F_a(\Omega)$ needs room for a transition band (transition cannot be infinitely sharp). A wider transition band permits a lower order (less complicated) $F_a(\Omega)$.

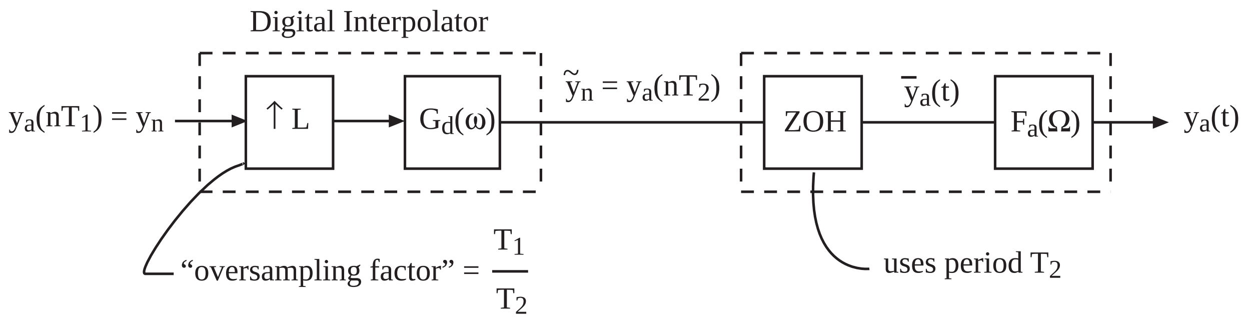
A realizable $F_a(\Omega)$ might look like:



This filter is permitted a transition bandwidth of

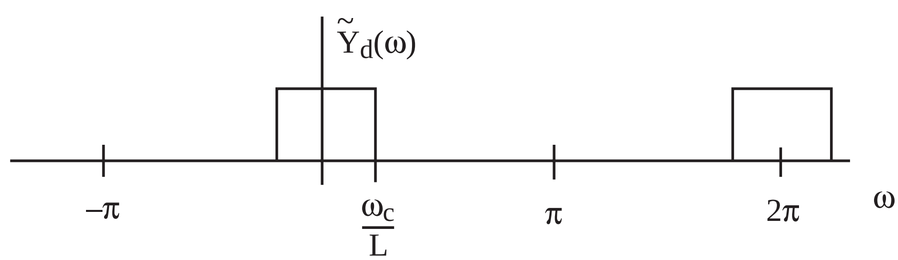
$$\frac{2\pi - \omega_c}{T_1} - \frac{\omega_c}{T_1} = \frac{2(\pi - \omega_c)}{T_1}$$

Now, consider oversampling D/A:

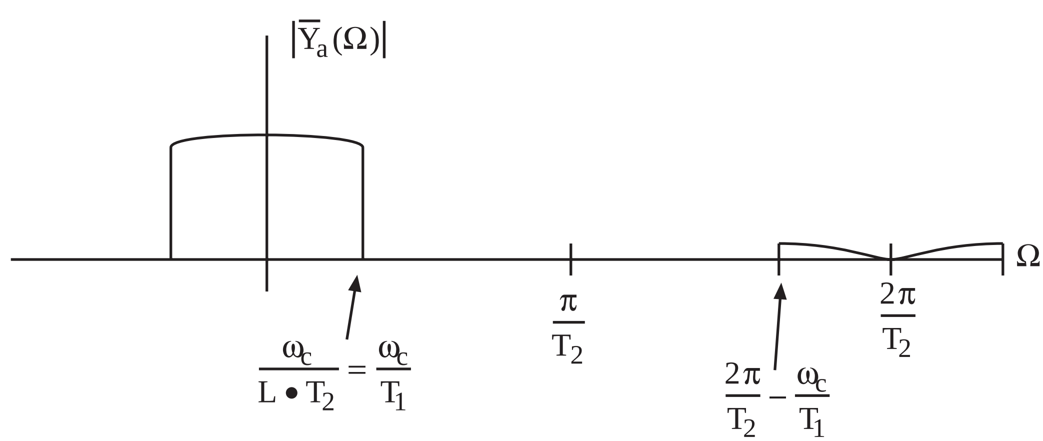


Due to the interpolation, the above ZOH puts out a finer staircase approximation with narrow steps (width T_2). Thus, we expect that $F_a(\Omega)$ can be simpler in this scheme. Let's analyze this in the frequency domain:

The interpolator squashes the DTFT of y_n:



So, $|\overline{Y}_a(\Omega)|$ now looks like the curve below (use eqn. (❑)) except with T_2 instead of T_1 and $\tilde{Y}_d$ instead of Y_d):



Thus, the transition band of $F_a(\Omega)$ can now be much wider.

13.10

$$\text{Transition BW} = \frac{2\pi}{T_2} - \frac{\omega_c}{T_1} - \frac{\omega_c}{T_1}$$

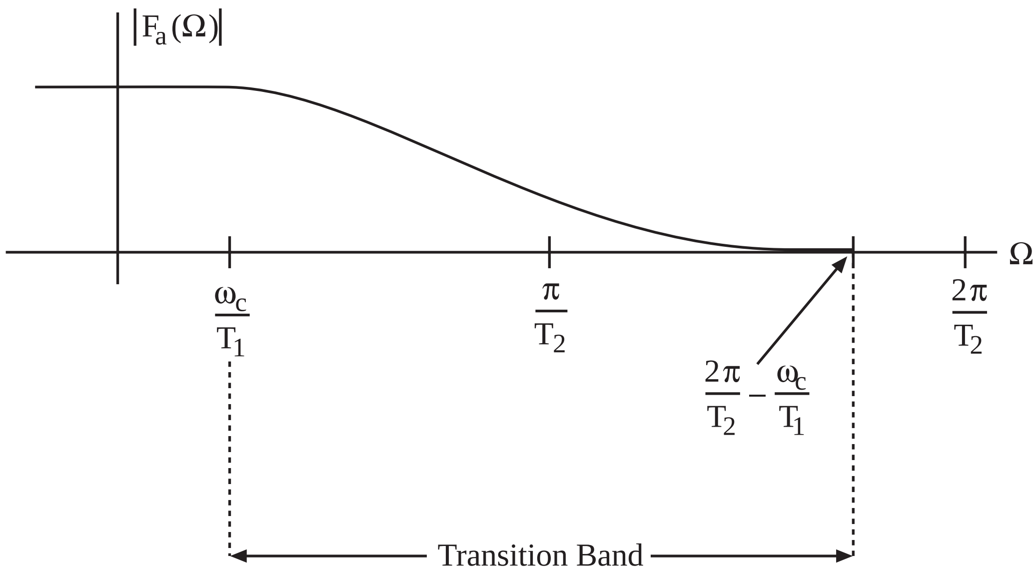
$$= \frac{2(L\pi - \omega_c)}{T_1} >> \frac{2(\pi - \omega_c)}{T_1}$$

↑ from before for regular D/A

so that implementation of $F_a(\Omega)$ can be far simpler.

Also, from the picture above we see that the center pulse of $\overline{Y}_a(\Omega)$ is almost flat and that the artifact centered at $\frac{2\pi}{T_2}$ is nearly zero, so even a fairly crude $F_a(\Omega)$ will do a good job. $F_a(\Omega)$ should have a nearly flat response in its passband, can have a huge transition band, and needs only moderate attenuation in its stopband.

$F_a(\Omega)$ in an oversampling D/A can look like:

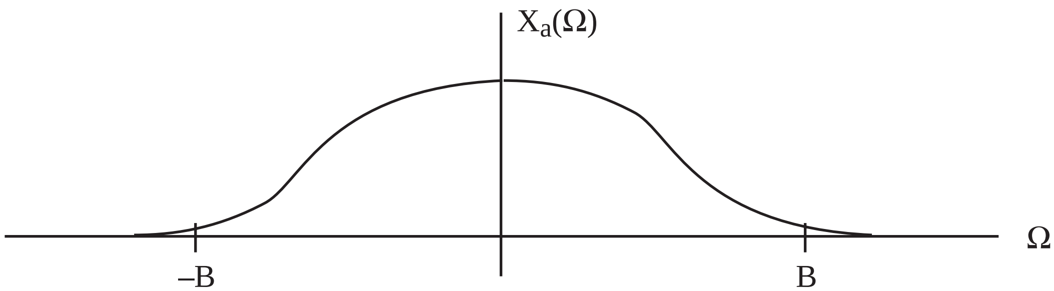


Oversampling A/D

A different type of oversampling is sometimes used to limit aliasing in the A/D. We will examine this as the second method, described below, for preventing aliasing at the A/D.

Prevention of Aliasing at A/D

Suppose $x_a(t)$ is nearly (not exactly) BL to B rad/sec.



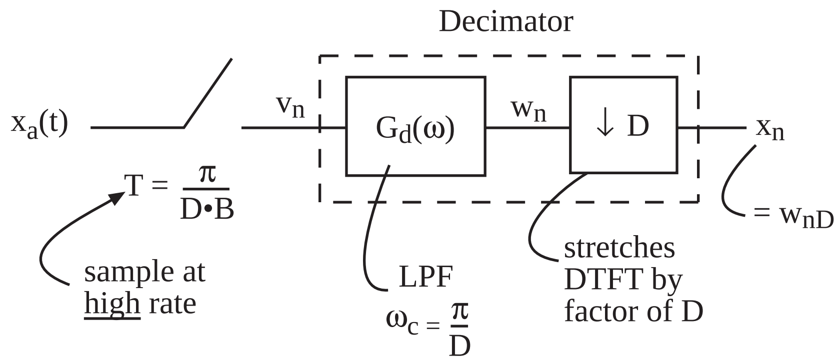
Here, B is an "effective band limit," but sampling with $T = \frac{\pi}{B}$ will still cause measurable aliasing.

How do we prevent aliasing at the sampler? Two possibilities:

1) Precede the A/D with an analog "antialiasing" LPF with cutoff B rad/sec. Then sample using $T < \frac{\pi}{B}$. This approach is very common.

2) Alternatively, sample at a high rate with $T = \frac{\pi}{B \bullet D}$ where D is an integer and is large enough to virtually prevent aliasing (choose D so that $X_a(\Omega)$ is virtually limited to $D \bullet B$ rad/sec). Then digitally LPF with cutoff $\omega_c = \frac{\pi}{D}$. Then decimate by a factor of D (discard D–1 of every D samples). This is called an oversampling A/D:

13.12

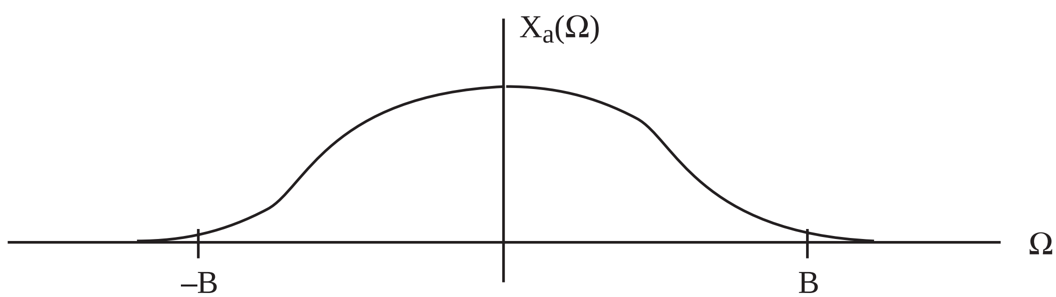


Ordinarily, sampling at such a high rate would be an expensive proposition, since this could create a very high data rate. The decimator, however, reduces the sampling rate back down by a factor of D. Note that implementation of $G_d(\omega)$ is not nearly so complicated as you might expect. Since D-1 of every D outputs of $G_d(\omega)$ will be discarded, only every Dth output need be computed!

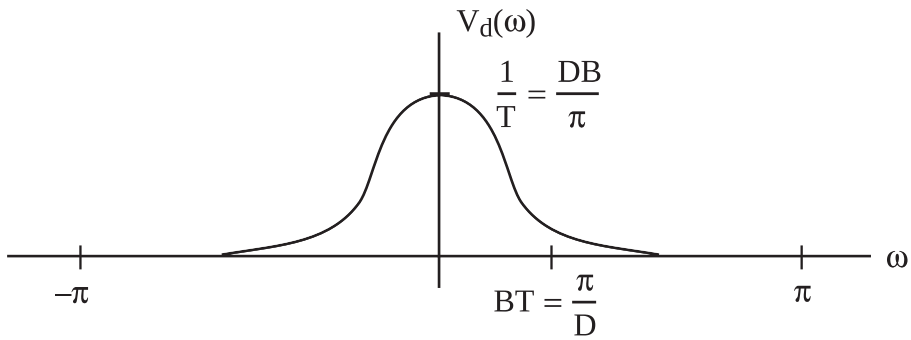
Choosing between 1) and 2) is simply an issue of whether you put the complexity in the analog or digital part of your system.

<u>Analysis of Oversampling A/D</u>

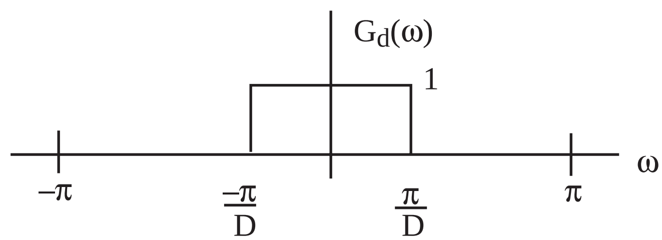
We will show that Option 2 (oversampling approach) produces exactly the same output $\{x_n\}$ as does Option 1. Suppose:



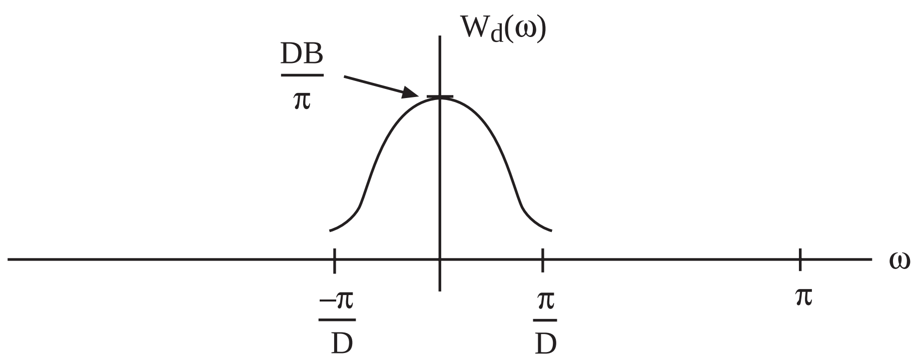
If $T = \dfrac{\pi}{D \bullet B}$ then



We have



So:



Now, what is the relationship between $X_d(\omega)$ and $W_d(\omega)$?

<u>Digression</u>

Note

$$\frac{1}{D}\sum_{k=0}^{D-1} e^{j\frac{2\pi}{D}kn} = \begin{cases} 1 & n = mD \\ \frac{1}{D}\,\frac{1-e^{j\frac{2\pi}{D}nD}}{1-e^{j\frac{2\pi n}{D}}} & n \neq mD \end{cases}$$

$$= \begin{cases} 1 & n = mD \\ 0 & n \neq mD \end{cases}$$

13.14

Now,

$$X_d(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega n}$$

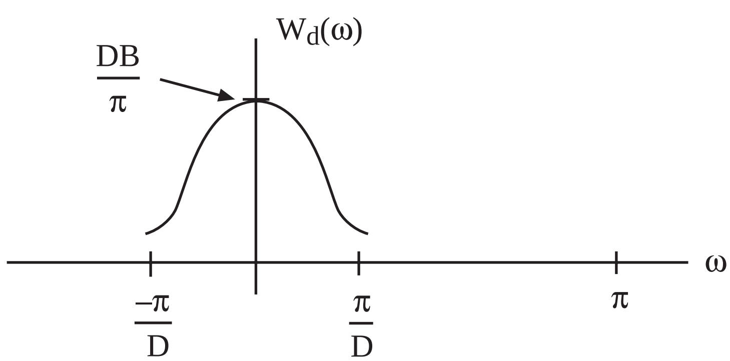
$$= \sum_{n=-\infty}^{\infty} w_{nD} e^{-j\omega n} = \sum_{\substack{n=mD \\ m=-\infty}}^{\infty} w_n e^{-j\omega \frac{n}{D}}$$

$$\underset{\substack{\uparrow \\ \text{trick from} \\ \text{digression}}}{=} \sum_{n=-\infty}^{\infty} w_n \underbrace{\frac{1}{D}\sum_{k=0}^{D-1} e^{j\frac{2\pi}{D}kn}}_{=\,0 \text{ unless } n\,=\,mD} e^{-j\omega\frac{n}{D}}$$

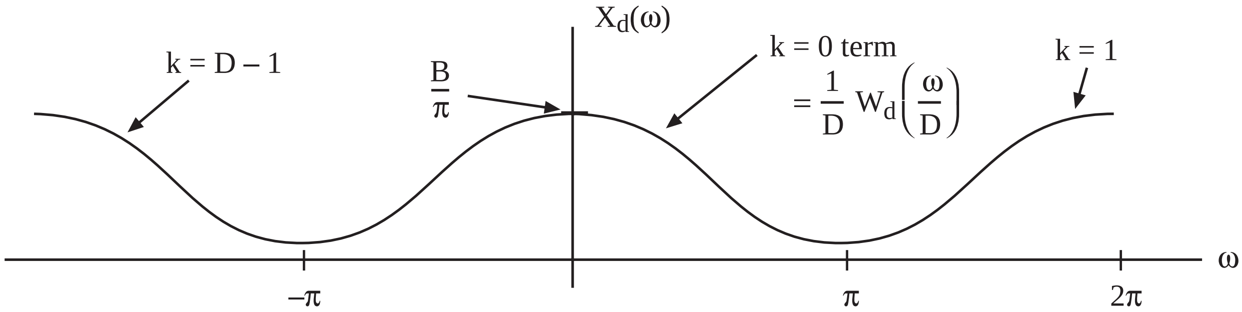
$$= \frac{1}{D}\sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} w_n e^{-jn\left(\frac{\omega-2\pi k}{D}\right)}$$

$$\Rightarrow \quad \boxed{X_d(\omega) = \frac{1}{D}\sum_{k=0}^{D-1} W_d\left(\frac{\omega-2\pi k}{D}\right)} \qquad (\Delta)$$

Now, had



Using this $W_d(\omega)$ in (Δ) gives



Note: This X_d is just what we would have obtained if we had analog low-pass filtered $x_a(t)$ to B rad/sec and then sampled with period $T = \frac{\pi}{B}$!

Thus, 2) does an equivalent job to 1).

Note:

How can (Δ) produce a periodic $X_d(\omega)$? (Δ) has only a finite number of terms in its sum. Answer: Each term is a periodic DTFT, not a FT as in Eq. (◊).

k=0 term in (Δ) has pulses centered at 0, $\pm 2\pi D$, $\pm 4\pi D$, etc.

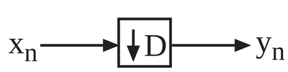
k=1 term has pulses centered at 2π, $2\pi \pm 2\pi D$, $2\pi \pm 4\pi D$, etc.

⋮

k = D – 1 term has pulses centered at $(D–1)2\pi$, $(D–1)2\pi \pm 2\pi D$, $(D–1)2\pi \pm 4\pi D$, etc.

A Further Look at Down-Sampler

A decimator uses a down-sampler as one of its components:



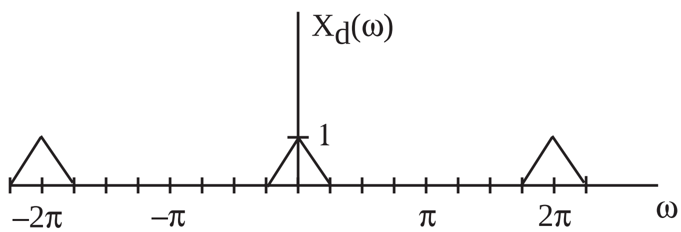
The down-sampler essentially stretches X_d. However, if $X_d(\omega)$ is not limited to $|\omega| < \frac{\pi}{D}$, then aliasing also occurs. Specifically,

$$Y_d(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X_d\left(\frac{\omega - 2\pi k}{D}\right). \qquad (\Delta)$$

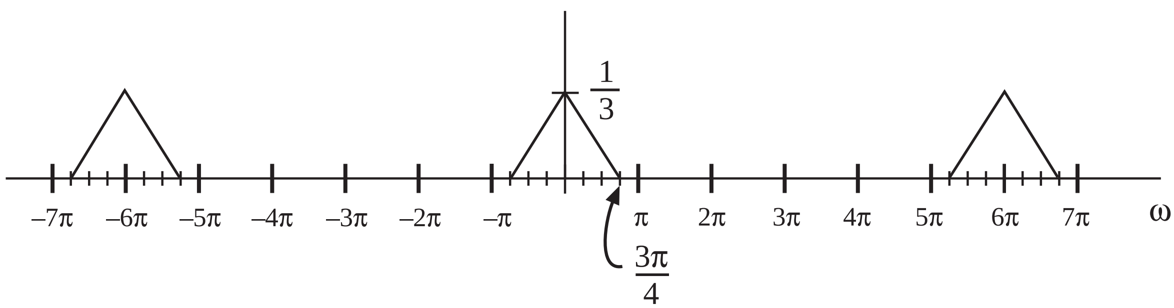
Notice the scaling in amplitude by $\frac{1}{D}$. This factor is not surprising, given that in the time domain, the down-sampler discards D–1 out of every D samples. By contrast, the up-sampler does not discard any samples, and inserts only zero-valued samples, so that there is no amplitude scaling in the Fourier domain for the up-sampler.

Example (Down-Sampler)

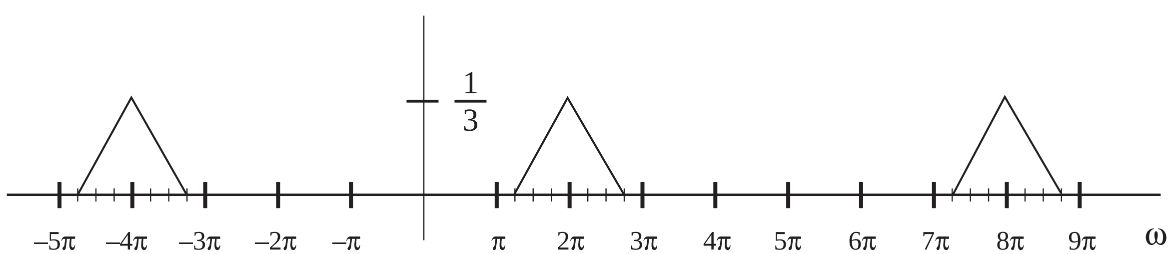
Suppose D = 3. Sketch $Y_d(\omega)$, assuming



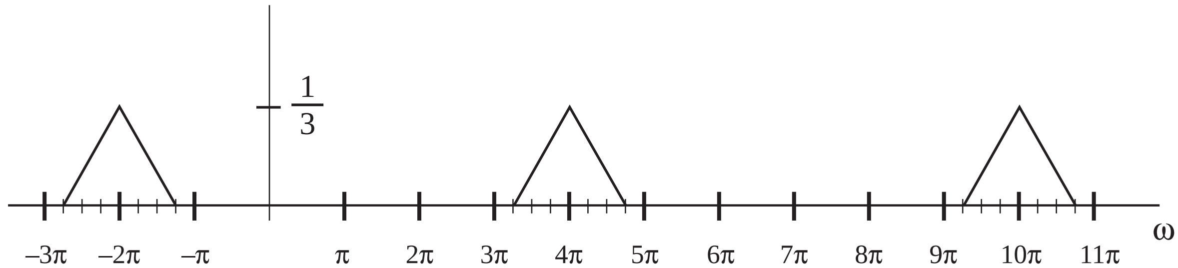
Then the k = 0 term in (Δ) is



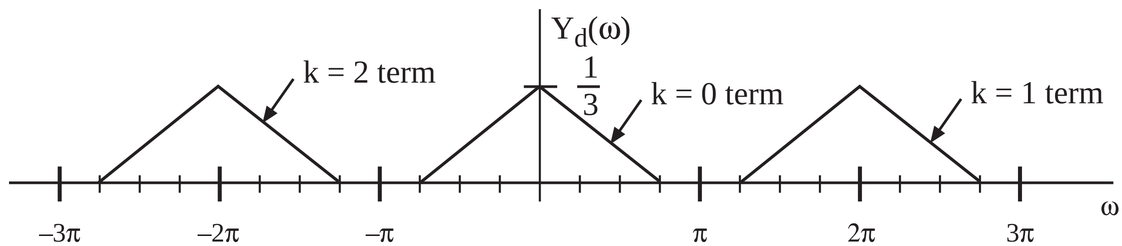
The k = 1 term in (Δ) is a 2π-shifted version of the above, namely



Likewise, the D – 1 = 2 term in (Δ) is a 4π-shifted version of the k = 0 term:



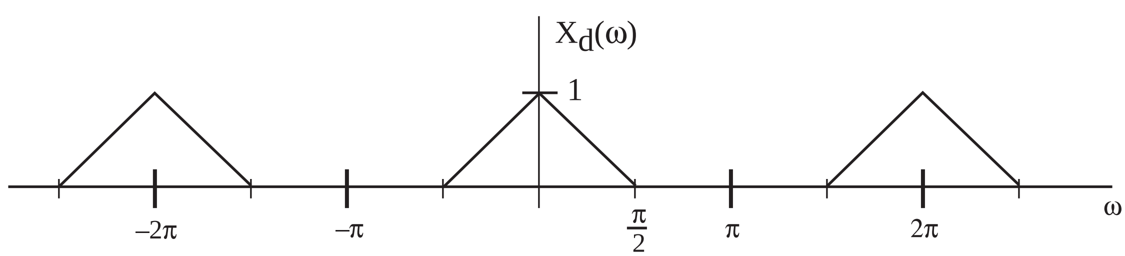
Adding the three previous plots together gives $Y_d(\omega)$:



Note that the various terms in (Δ) interlace to produce a 2π-periodic $Y_d(\omega)$. In this example there was no need to plot the k = 0, 1, 2 terms, since the k = 0 term, alone, determines the shape of $Y_d(\omega)$ for $|\omega| < \pi$. In the next example, the downsampler causes aliasing, so that the terms in (Δ) overlap. This situation is more complicated than in the previous example.

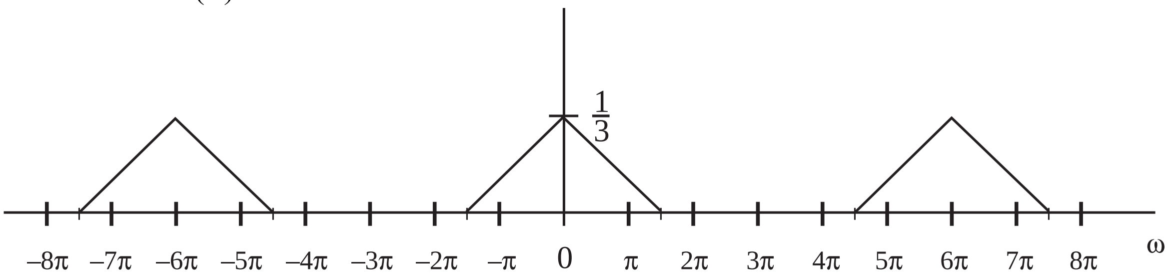
Example (Down-Sampler)

Suppose D = 3 as before, but now with

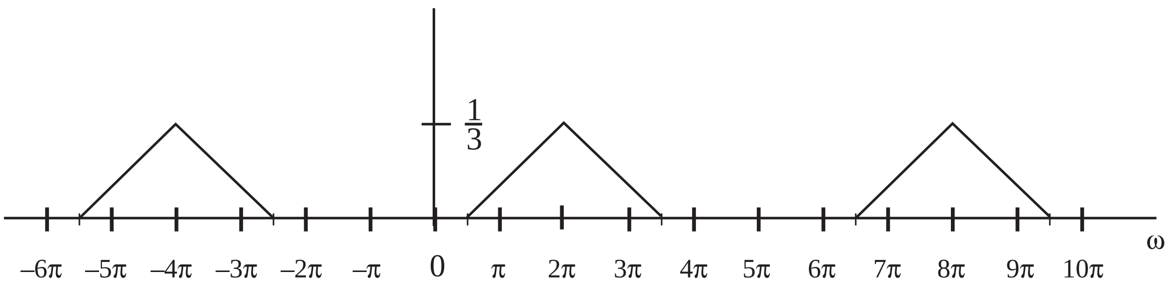


13.18

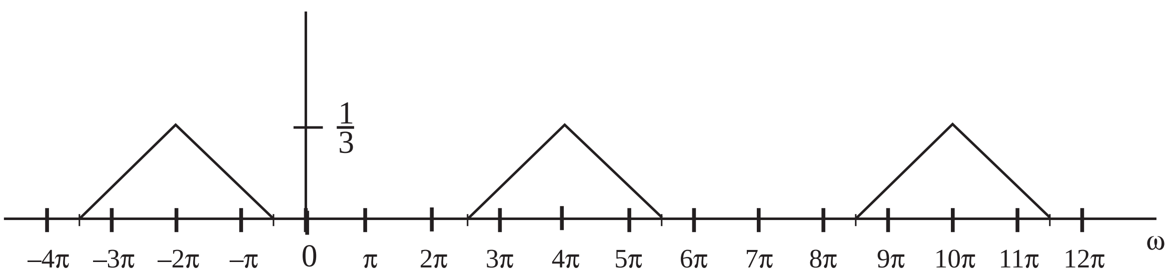
The k = 0 term in (Δ) is:



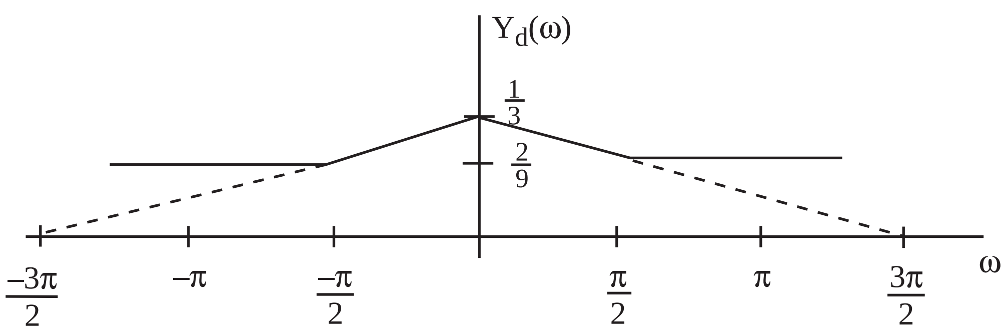
Notice that the center pulse extends beyond $\omega = \pm\,\pi$, which is an indication of aliasing. The k = 1 term in (Δ) is a 2π-shifted version of the above plot, namely:



The k = 2 term in (Δ) is a 4π-shifted version of the k = 0 term:



Adding the k = 0, 1, 2 terms gives $Y_d(\omega)$, which we plot only for $|\omega| \le \pi$:



In this example, we have aliasing because $X_d(\omega)$ extends beyond $\omega = \pm\,\frac{\pi}{D} = \frac{\pi}{3}$. In a decimator, the job of the LPF that precedes the down-sampler is to cut off $X_d(\omega)$ at $\omega = \frac{\pi}{D}$ to prevent this aliasing.