MAS 160/510 Notes

Sampling, Decimation, and Interpolation

Fall 1997

1 Motivation

Sampling can be thought of as taking "snapshots" in time. When you sample a signal, you keep values of it only at particular time instances. The samples take up less space, but haven't you thrown away important information? We will see that under certain conditions the samples completely describe the original signal. The key issues will be the frequency content of the signal, and the rate at which the samples are taken.

This topic contains some remarkable principles. We'll cover the most important ones from a practical standpoint in these notes. But we won't take space here to cover some other exciting ones which are active research topics in their own right: Can you reconstruct an entire image from just the edges of the objects in the picture? Can you recover a signal of length T when all you have is the first third of it? What amplitude and position accuracy is necessary when sampling? Like in the rest of the course, we will ignore issues related to amplitude quantization in these notes.

However there are still practical questions you can answer based on what we present here. Consider an analog speech signal – how fast do you need to sample it so it will sound the same when you play back just the samples? What causes car wheels to look like they're turning backwards on the highway at night? How do you explain water droplets appearing to defy gravity over in Doc Edgerton's Strobe Alley? How can you scale down an image from HDTV to fit on your wristwatch TV? If you play back speech faster than the rate at which it was recorded, why does the pitch go up? These questions can be answered by answering the following four general questions, the focus of these notes:

1. What sampling rate do we use to insure we can later recover the signal from its samples?
2. Given a signal of length N samples, what's the best way to "shrink" it to a signal of length M samples ("downsampling")?
3. What's the best way to "stretch" it to a signal of length L samples ("upsampling")?
4. What's the best way to change the rate of sampling by a fractional number?

One can quibble over the precise usage of the words "recover" and "best" in these questions, but these discussions involve Ph.D.-sized ideas that are not a part of this class. In these notes, we'll stick to the most typical ways to process signals.

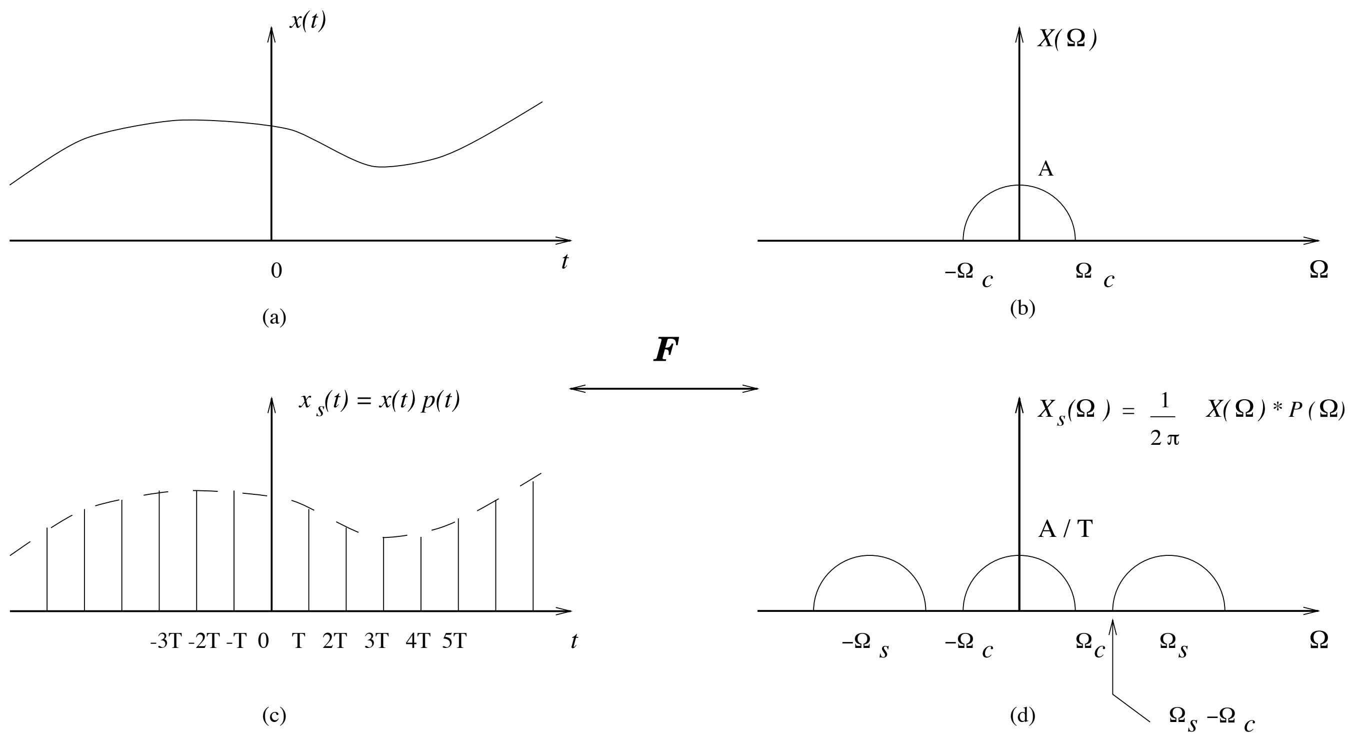

Figure 1: Original signal $x(t)$ is sampled by multiplication with pulse train $p(t)$. Sampling results in replication of $X(\Omega)$.

2 Nyquist sampling theorem and Nyquist rate

Consider some continuous time signal, $x(t)$ and its Fourier transform, $X(\Omega)$, illustrated in Fig. 1 (a) and (b). Assume that $X(\Omega)$ is "bandlimited" to ω_c, i.e., $X(\Omega) = 0$ when $|\Omega| > \Omega_c$. We can sample $x(t)$ by multiplying with a periodic impulse train,

$$\begin{aligned} x_s(t) &= x(t)p(t) \\ &= x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \\ &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \\ &= \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT) \end{aligned}$$

where $x[n]$ is the new discrete sequence. Alternatively, we can convolve the Fourier transforms,

$$\begin{aligned} X_s(\Omega) &= \frac{1}{2\pi}\{X(\Omega) * P(\Omega)\} \\ &= \frac{1}{2\pi}\left\{X(\Omega) * \Omega_s \sum_{n=-\infty}^{\infty} \delta(\Omega - n\Omega_s)\right\} \end{aligned}$$

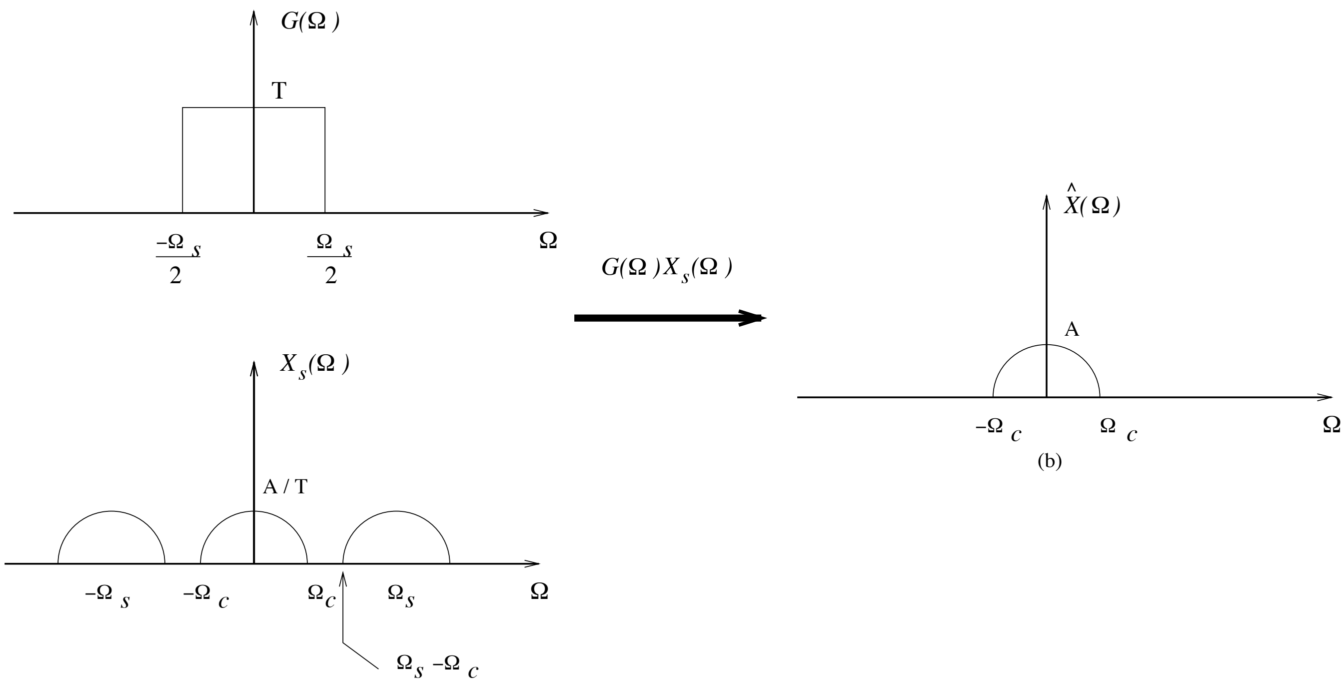

Figure 2: Recovery of $X(\Omega)$, by multiplying an unaliased $X_s(\Omega)$ with the Fourier transform of an ideal low-pass filter.

$$= \frac{1}{T}\sum_{n=-\infty}^{\infty} X(\Omega - n\Omega_s), \quad \Omega_s = \frac{2\pi}{T}$$

Sampling in time has made the signal periodic in frequency. Earlier in the course we saw that a periodic time signal has a discrete set of Fourier series coefficients. This brings up a general and useful principle:

Sampling in one domain corresponds to replication in the other domain.

So far we have converted a continuous time signal $x(t)$ to a discrete sequence $x[n]$. Let's consider how much information was lost in the sampling process, when we threw away everything but the samples. Can the original signal be recovered?

Recall that if we have $X(\Omega)$ then $x(t)$ is uniquely recovered by an inverse Fourier transform. But after sampling we have $X_s(\Omega)$, not $X(\Omega)$. Under what conditions can we get $X(\Omega)$ from $X_s(\Omega)$? We can get $X(\Omega)$ if the replication is done in such a way that a full copy of it is preserved. A general rule is that there should be no "overlap" during replication.

To see this, consider the signal in Fig. 1 (d). If the values of Ω_s and Ω_c are such that $\Omega_s - \Omega_c > \Omega_c$, then there will be no overlap in the frequency domain. In this case, $X(\Omega)$ can be obtained by multiplying $X_s(\Omega)$ with an ideal lowpass filter, $G(\Omega)$ as shown in Fig. 2. Here's the mathematical expression for Fig. 2 in the frequency domain:

$$\hat{X}(\Omega) = X_s(\Omega)G(\Omega),$$

and in the time domain:

$$\begin{aligned}\hat{x}(t) &= x_s(t) * g(t) \\ &= x_s(t) * \text{sinc}\,(\frac{\Omega_s t}{2}) \\ &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) * \text{sinc}\,(\frac{\pi t}{T}) \\ &= \sum_{n=-\infty}^{\infty} x(nT)\text{sinc}\,(\frac{\pi(t-nT)}{T})\end{aligned}$$

The original signal $x(t) = \hat{x}(t)$ is recovered by convolving the discrete sequence $x(nT)$ with a continuous sinc function. When the sinc is centered on top of a sample ($t = nT$), it multiplies that sample by "1" and all other samples by "0". When it is centered between two samples ($t \neq nT$), it forms a weighted sum of all the samples to interpolate the value of $x(t)$ that had been thrown away.

In summary, even after sampling it may be possible to completely recover the original signal. This is somewhat remarkable, especially since it seems that sampling throws away "more" of the time signal than it keeps. Nonetheless, given the following condition, a signal can be completely represented by its samples:

Nyquist Sampling Theorem

Let $x(t)$ be a signal with bandlimited Fourier transform, i.e., $X(\Omega) = 0$ when $|\Omega| > \Omega_c$. If

$$\Omega_s > 2\Omega_c$$

then $x(t)$ is uniquely specified by its samples $x[n] = x(nT)$, $n = 0, \pm1, \pm2, \ldots$ with $T = \frac{2\pi}{\Omega_s}$.

We often paraphrase this as, "you must sample at greater than twice the highest frequency present in the signal." The *Nyquist rate* refers to $2\Omega_c$, the frequency that must be exceeded by the sampling frequency. Hence the time samples must be spaced by $T < \frac{\pi}{\Omega_c}$.

If there is overlap during replication, it is called "aliasing." This case is illustrated in Fig. 3. The name "aliasing" comes from energy at one frequency masquerading as a different frequency. This can lead to bizarre behavior – wheels that spin backwards when the car is moving forward, droplets that appear to fall "up," a fan in your dining room that *looks* like it's spinning slowly even though it is cooling you by spinning quite fast, etc. Feel free to tell us about unexpected new examples of aliasing that you encounter.

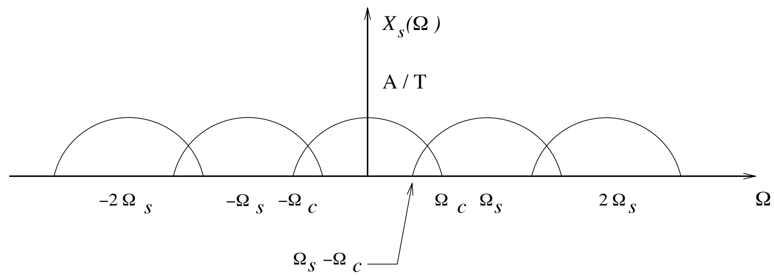

Figure 3: A signal sampled at less than the Nyquist rate yields aliasing in the frequency domain.

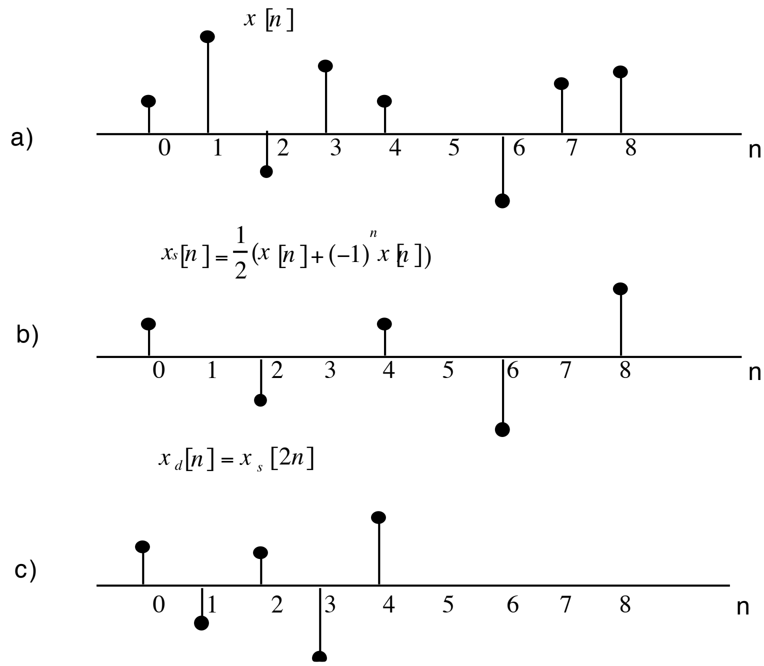

Figure 4: Down-sampling a discrete signal by 2, time domain.

3 "Downsampling" or "decimation" $\boxed{\downarrow M}$

In this and subsequent sections we assume discrete signals, $x[n]$. The words "downsampling," "decimation," and "subsampling" all refer to reducing the number of samples used to specify $x[n]$. Suppose we have samples of speech such as illustrated in Fig. 4 (a). The computer plays the samples out at a fixed rate. Suppose you're in a hurry and want to speed up the play by a factor of two. You decide to cut the number of samples in half. What happens?

Let's be more precise. The simplest way to halve the set of samples is to throw away every other sample and "shrink" the sequence. These two steps are illustrated in Fig. 4 (b) where the odd samples are zeroed, and in Fig. 4 (c) where the samples are squeezed together. When they are played on the computer they will last half as long. But will it sound like the same signal played twice as fast?

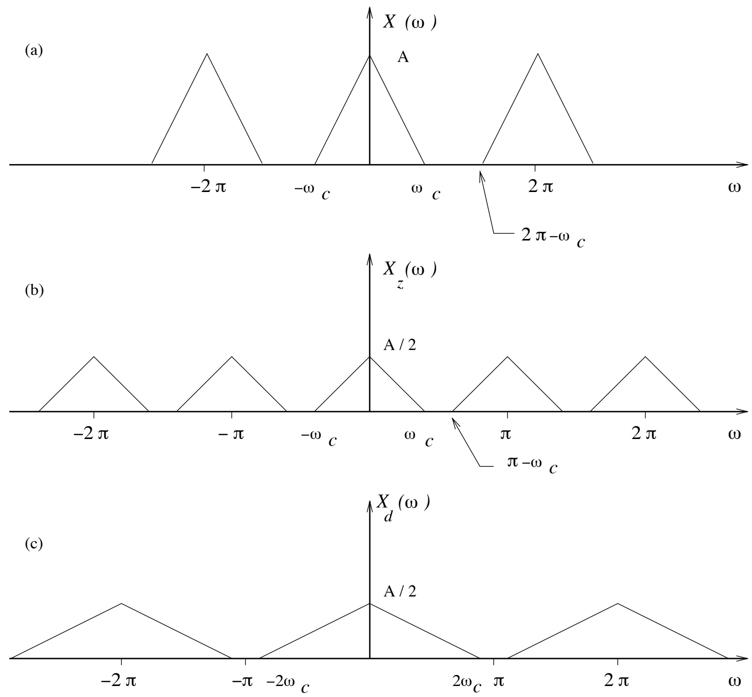

Figure 5: Downsampling a discrete signal by 2, frequency domain.

Experience tells us the pitch will sound higher – but why? Pitch is a frequency domain concept, so let's look at the discrete Fourier transforms of Fig. 4 (a)-(c) sketched in Fig. 5 (a)-(c). The mathematics for these pictures is also helpful. Here's how to zero the odd samples:

$$x_s[n] = \frac{1}{2}(x[n] + (-1)^n x[n])$$

which transforms to,

$$\begin{aligned}
\mathcal{F}\{x_s[n]\} &= \frac{1}{2}(\mathcal{F}\{x[n]\} + \mathcal{F}\{(-1)^n x[n]\}) \\
X_s(\omega) &= \frac{1}{2}(X(\omega) + \sum_{n=-\infty}^{\infty} (-1)^n x[n] e^{-j\omega n}) \\
&= \frac{1}{2}(X(\omega) + \sum_{n=-\infty}^{\infty} e^{j\pi n} x[n] e^{-j\omega n}) \\
&= \frac{1}{2}(X(\omega) + \sum_{n=-\infty}^{\infty} x[n] e^{-j(\omega-\pi)n}) \\
&= \frac{1}{2}(X(\omega) + X(\omega - \pi))
\end{aligned}$$

The second stage yields $x_d[n]$:

$$x_d[n] = x_s[2n],$$

with the transform,

$$\mathcal{F}\{x_d[n]\} = \mathcal{F}\{x_s[2n]\}$$

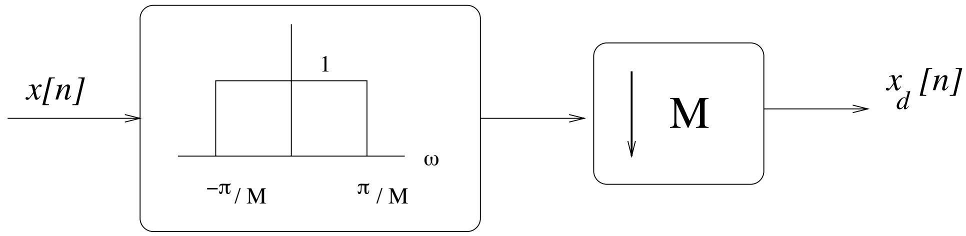

Figure 6: Block diagram: Antialiasing followed by downsampling by M.

$$
\begin{aligned}
X_d(\omega) &= X_s(\omega/2) \\
&= \frac{1}{2}(X(\omega/2) + X(\omega/2 - \pi))
\end{aligned}
$$

The decimation procedure for arbitrary values of M generalizes to:

Downsampling by M

Stage 1:

$$
\begin{aligned}
x_s[n] &= \begin{cases} x[n], & n = 0, \pm M, \pm 2M, \dots \\ 0, & otherwise. \end{cases} \\
&= \tfrac{1}{M} \textstyle\sum_{k=0}^{M-1} e^{j\frac{2\pi kn}{M}} x[n] \qquad\qquad X_s(\omega) = \frac{1}{M} \sum_{k=0}^{M-1} X(\omega - \tfrac{2\pi k}{M})
\end{aligned}
$$

Stage 2:

$$
x_d[n] = x_s[Mn] \qquad\qquad X_d(\omega) = X_s(\tfrac{\omega}{M})
$$

You might wish to verify that the above example is the special case when $M = 2$.

Typically a bandlimiting or "anti-aliasing" filter is included before subsampling to prevent aliasing. The complete process is shown in Fig. 6, where we show a low-pass anti-aliasing filter being applied before the signal is decimated. Before downsampling by an arbitrary integer M, one should prefilter with a low-pass having cutoff frequency π/M in order to ensure that the downsampling will not cause aliasing.

When you see the box with the "M" and the down-arrow in it, remember that this box involves TWO stages: first, zeroing every Mth sample, and second, "squeezing" the signal into $1/M$th the original length in time. The first stage of the process, zeroing half the samples, led to replication of $\frac{1}{M}X(\omega)$. This will cause aliasing if $X(\omega)$ is not already bandlimited to π/M. The second stage of the process, rescaling the axis, stretches every frequency by M. Hence we hear the pitch increase by a factor of M! This technique quickly loses intelligibility for speech. More sophisticated techniques can be employed to simultaneously compensate for the pitch while changing the sampling rate, but these go beyond our discussion.

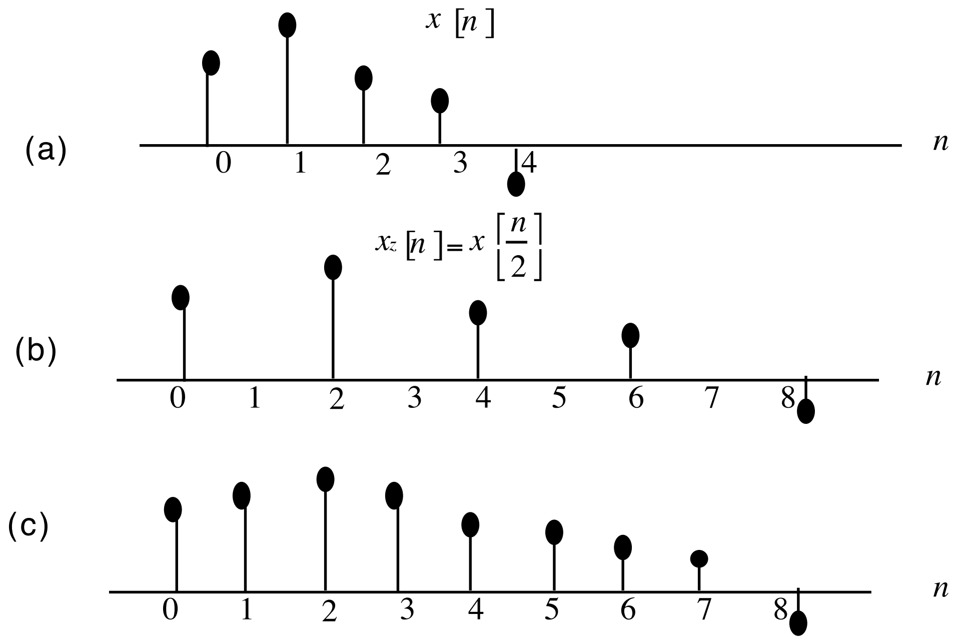

Figure 7: Upsampling by 2 followed by an interpolation filter (low-pass).

4 Upsampling or "interpolation" $\boxed{\uparrow L}$

Suppose you want to slow down a signal, or "zoom in" on it. Either application requires expanding the number of samples. No information will be discarded, and as we shall soon see, there is no risk of aliasing (there is no subsampling). The expressions "upsampling," "interpolation," and "expansion" are used interchangeably to refer to increasing the number of samples in $x[n]$.

The simplest way to upsample is to insert zeros between the existing samples. An example for $L = 2$ is illustrated in Fig. 7. Upsampling, represented by a box with an up-arrow in it, only involves this single-stage operation. However, this operation is rarely done alone, but is almost always followed by an interpolation filter, which "fills in" values for the zeros we inserted. The process is similar to that in Sec. 2, where a low-pass filter was applied to reconstruct a sampled signal, except here the whole process is discrete.

Consider the frequency-domain representation of Fig. 7, shown in Fig. 8. The Fourier transform was already periodic, and inserting zeros has only "shrunk" it by a factor of 2. We always see "shrinking" in the Fourier domain when there is "stretching" in the time domain, and vice versa. The shrinking does NOT involve replication, so there is no aliasing. However, now there are two copies of the signal between 0 and 2π (or in general, L copies). We only need one copy. This can be obtained by low-pass filtering with cutoff π/L.

The final sequence is shown in Fig. 7 (c). It may appear like there is introduction of new information, but this is not the case. The samples which were at multiples of L have not changed, and the newly constructed samples result from *removing* the extra copies of $X_z(L\omega)$

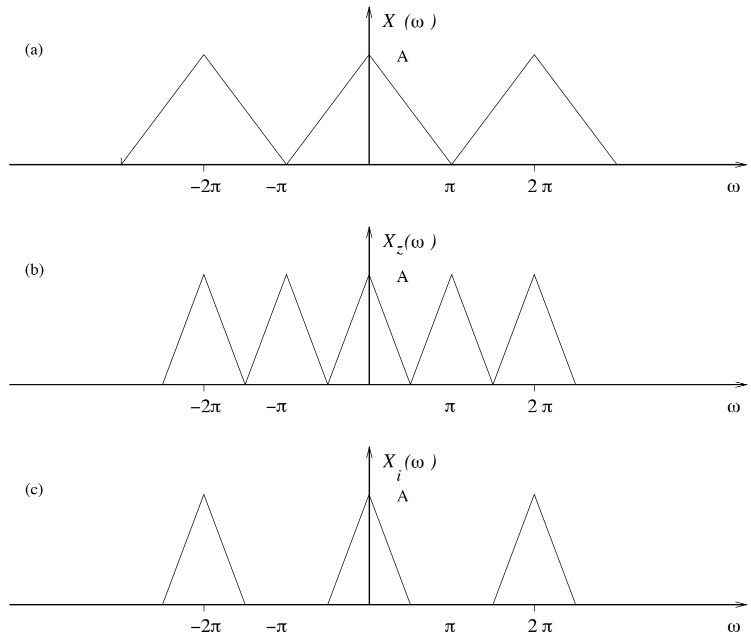

Figure 8: Upsampling by 2, Fourier transforms for Fig. 7.

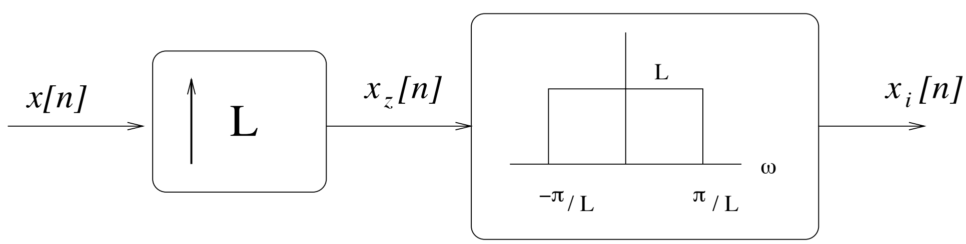

Figure 9: Block diagram: Upsampling by L followed by interpolation.

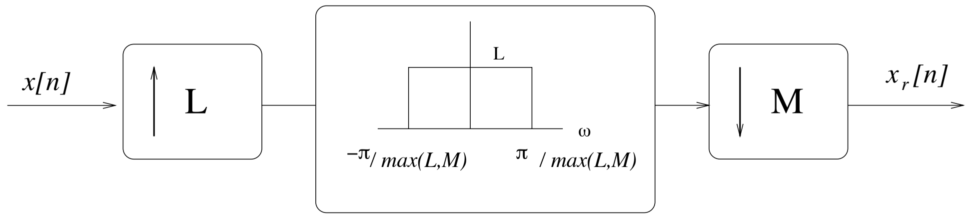

Figure 10: For rational sampling rate changes, L/M, it is better practice to upsample before downsampling.

between $\frac{\pi}{L}$ and $2\pi - \frac{\pi}{L}$.

Upsampling by L

$$x_z[n] \;=\; \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \ldots \\ 0, & otherwise. \end{cases} \qquad X_z(\omega) \;=\; X(L\omega)$$

5 Fractional sampling rate changes

Suppose we wish to increase the sampling rate by a factor of 1.5 (expand the image or sound signal by 1.5). This can be done by decreasing the sampling period $T' = \frac{M}{L}T = \frac{2}{3}T$. This is simple given the last two sections. All we do is combine upsampling by L with downsampling by M. Does it matter which is performed first? Not always, but sometimes. Since upsampling never throws away information, and downsampling usually does, it is a good habit to perform *upsampling before downsampling.* Starting with Figs. 6 and 9 and choosing the more restrictive of their two low-pass filters yields the sampling-rate change system shown in Fig. 10. An example with $M = 2$ and $L = 3$ is illustrated in Fig. 11.

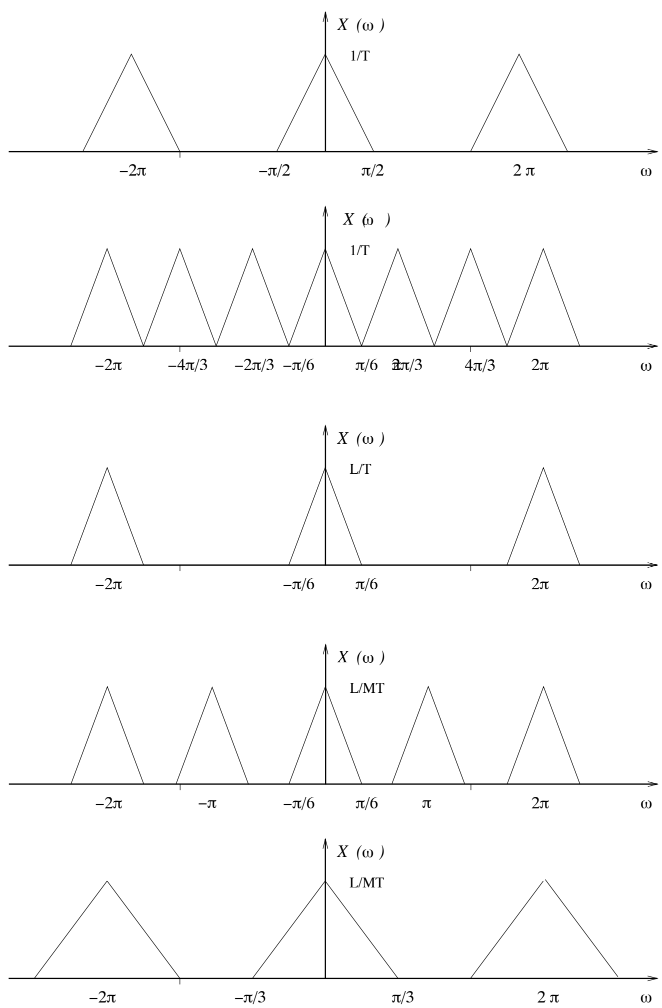

Figure 11: Example of increasing sampling rate by 1.5. The net effect is interpolation.