SIGNALS AND SYSTEMS LABORATORY 10:
Sampling, Reconstruction, and Rate Conversion

INTRODUCTION

Digital signal processing is preferred over analog signal processing when it is feasible. Its advantages are that the quality can be precisely controlled (via wordlength and sampling rate), and that changes in the processing algorithm are made in software. Its disadvantages are that it may be more expensive and that speed or *throughput* is limited. Between samples, several multiplications, additions, and other numerical operations need to be performed. This limits sampling rate, which in turn limits the bandwidth of signals that can be processed. For example, the standard for high fidelity audio is 44,100 samples per second. This limits bandwidth to below the half-sampling frequency of about $f_0 / 2$ = 22 kHz, and gives the processor only 22.7 microseconds between samples for its computations. It should be noted here that the half-sampling frequency $f_0/2$ may also be referred to as the Nyquist frequency and the sampling rate f_0 may be referred to as either the Nyquist sampling rate, Shannon sampling rate, or critical sampling rate.

In order to do digital signal processing of analog signals, one must first sample. After the processing is completed, a continuous time signal must be constructed. The particulars of how this is done are addressed in this lab. We will examine

i. aliasing of different frequencies separated by integer multiples of the sampling frequency,

ii. techniques of digital interpolation to increase the sampling rate (upsampling) and convert to continuous time,

iii. sample rate conversion using upsampling followed by downsampling.

The highest usable frequency in digital signal processing is one half the sampling frequency. In practice, however, undesirable effects creep in at frequencies below the half sampling frequency. (This typically happens near 90% of $f_0/2$. This is why the standard for audio is 44kHz instead of 40 kHz.) We shall observe this phenomenon and explain it.

SAMPLING IN TIME, ALIASING IN FREQUENCY

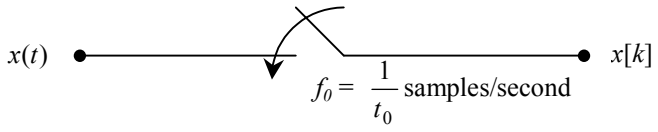


Figure One Sampling

Figure One is a representation for the sampling operation. The input signal is analog and continuous time. The output signal is typically digital, i.e. quantized, but we will ignore the fact that the output samples are finite words of digital data and not real numbers. The part that interests us is that the signal $x[k]$ is *discrete time*. The input signal $x(t)$ is continuous time and has a Fourier transform.

$$(1) \qquad x(t) \quad \xleftrightarrow{Fourier} \quad X(j\omega)$$

The discrete time output signal $x[k]$ has a DTFT. The time and frequency representations for this signal are

$$(2) \qquad x[k] = x(kt_0) \quad \xleftrightarrow{DTFT} \quad X(e^{j\omega t_0}) = \frac{1}{t_0}\sum_{n=-\infty}^{\infty} X\big(j(\omega - n\omega_0)\big), \quad \omega_0 = \frac{2\pi}{t_0} = 2\pi f_0 .$$

The time domain relation on the left-hand side of equation (2) is very simple, but the frequency domain relation on the right hand side is not. It is far more interesting. All the spectral components of $X(j\omega)$ at frequencies that differ by an integer multiple of the sampling frequency are added together. Thus one cannot distinguish these spectral components from the samples $x[k]$. If the sampling frequency is f_0, then for every sinusoid of frequency f there is another sinusoid of frequency $f + f_0$, which has the same samples. The same is true for $f + 2f_0$, $f + 3f_0$, and so on. All of these sinusoids go by the same alias, namely the sinusoid of frequency f. We will observe this phenomenon of *aliasing* experimentally. This is demonstrated in Figure Two below.

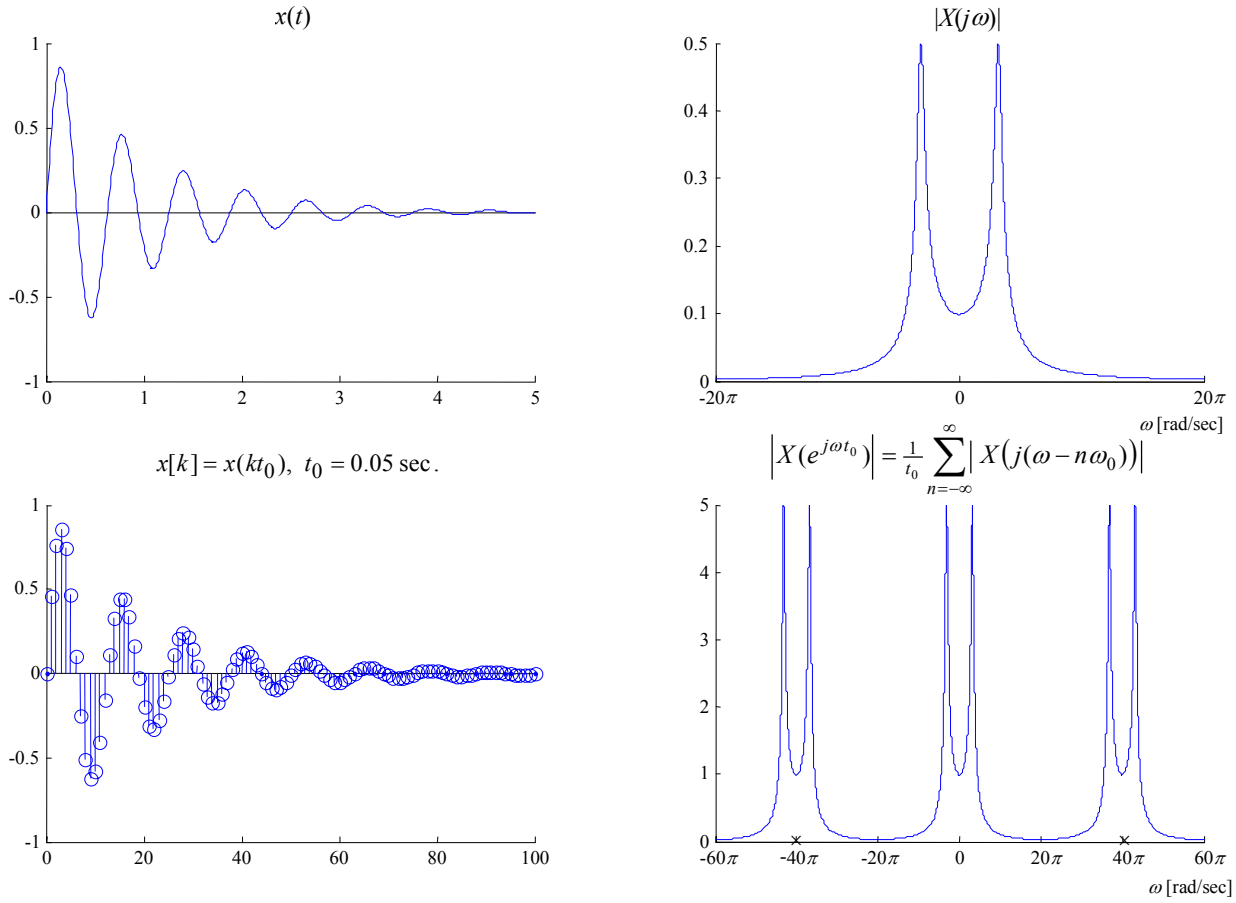


Figure Two Sampling In Time, Aliasing In Frequency

A few comments about Figure Two. The continuous-time signal $x(t)$ is a dampened sinusoid defined by

(3) $$x(t) = e^{-t}\sin(10t)u(t),$$

were $u(t)$ is the unit step function. This signal could be an example of a natural response of an underdamped parallel RLC circuit. The spectrum of $x(t)$ is a commonly known Laplace transform given by

(4) $$X(j\omega) = X(s)\Big|_{s=j\omega} = \frac{10}{s^2 - 2s + 101}\Big|_{s=j\omega}$$

which is shown in the upper right panel of Figure Two. From this graph, it can be seen that $X(j\omega)$ is essentially bandlimited to the frequency band $-20\pi \le \omega \le 20\pi$. This is to say that the frequency components $|\omega| > 20\pi$ contain much less energy than the frequency components of $|\omega| \le 20\pi$. In practice, the signal $x(t)$ will be guaranteed to be

bandlimited by an analog lowpass filter called an *anti-aliasing filter*. The Shannon sampling theorem then says that we need to sample at a rate twice the maximum frequency component of our bandlimited signal. In our case, this means that our sampling frequency will be $\omega_0 = 40\pi$. (Also, $f_0 = \omega_0/2\pi = 20\ Hz,\ and\ t_0 = \frac{2\pi}{\omega_0} = \frac{1}{f_0} = 0.05\,\text{sec}.$) In accordance with the Shannon sampling theorem, $x(t)$ was sampled at rate $\omega_0 = 40\pi$ rad/sec. The sample data signal is shown in the lower left panel of Figure Two and the accompanying discrete-time frequency response is shown in the lower right panel. Notice the marks on the frequency axis of the lower right panel. These marks show where the *aliases* of the original signal are centered. As the sampling rate increases, the center frequencies of the $X(j\omega)$ aliases will move farther away from the original signal's spectrum. In Figure Two, the sampling frequency ω_0 prevents a noticeable overlap of aliases, leaving $X(e^{j\omega t_0})$ looking essentially like $X(j\omega)$. Thus *aliasing effects* are minimized. Aliasing effects due to undersampling amount to a set of samples that appear to be the samples of a signal with a lower frequency, hence the term *alias*. This is to be avoided since these effects cannot be removed in digital processing. To prevent this, one should sample at the Nyquist sampling rate or higher.

THE DISCRETE-TIME TO CONTINOUS TIME RECONSTRUCTION PROCESS

Let us state an unwelcome fact. Given the input $x(t)$, one can easily get the output samples $x[k]$, but the reverse is not true. Given the output samples, we cannot reconstruct the input without knowing extra information, because we have thrown out everything between the samples. Obviously there are infinitely many ways to fill in the gaps between samples. The sampling operation is *many to one*, and is not generally invertible. However, if the input is bandlimited to the half sampling frequency, then there will be exactly one input signal $x(t)$ for the given set of samples. This is the *Shannon sampling theorem*, and the bandlimited reconstruction is

$$(5) \qquad x(t) = \sum_{k=-\infty}^{\infty} x[k]h(t - kt_0)\ \text{, where}\ h(t) = \text{sinc}(\pi t / t_0).$$

Equation (5) represents the ideal. It is the goal of a good analog reconstruction design to approximate this relation. Therefore, make note of two things: first, equation (5) is a *pulse amplitude modulation* (PAM) equation, and second, the PAM pulseshape is that of the *sinc* function, or Shannon wavelet. The *sinc* function in question is the impulse response of an *ideal lowpass filter* with gain equal to the sampling period t_0, and bandwidth one half the sampling frequency, hence the term *bandlimited reconstruction*. One departure from the ideal is allowed in audio systems. Time delay is of no consequence, and therefore one can use a delayed *sinc* pulse. In digital feedback control systems however, time delay can present stability problems and is to be avoided. The time domain process of reconstructing $x(t)$ from its samples $x[k]$ is demonstrated in Figure Three below.

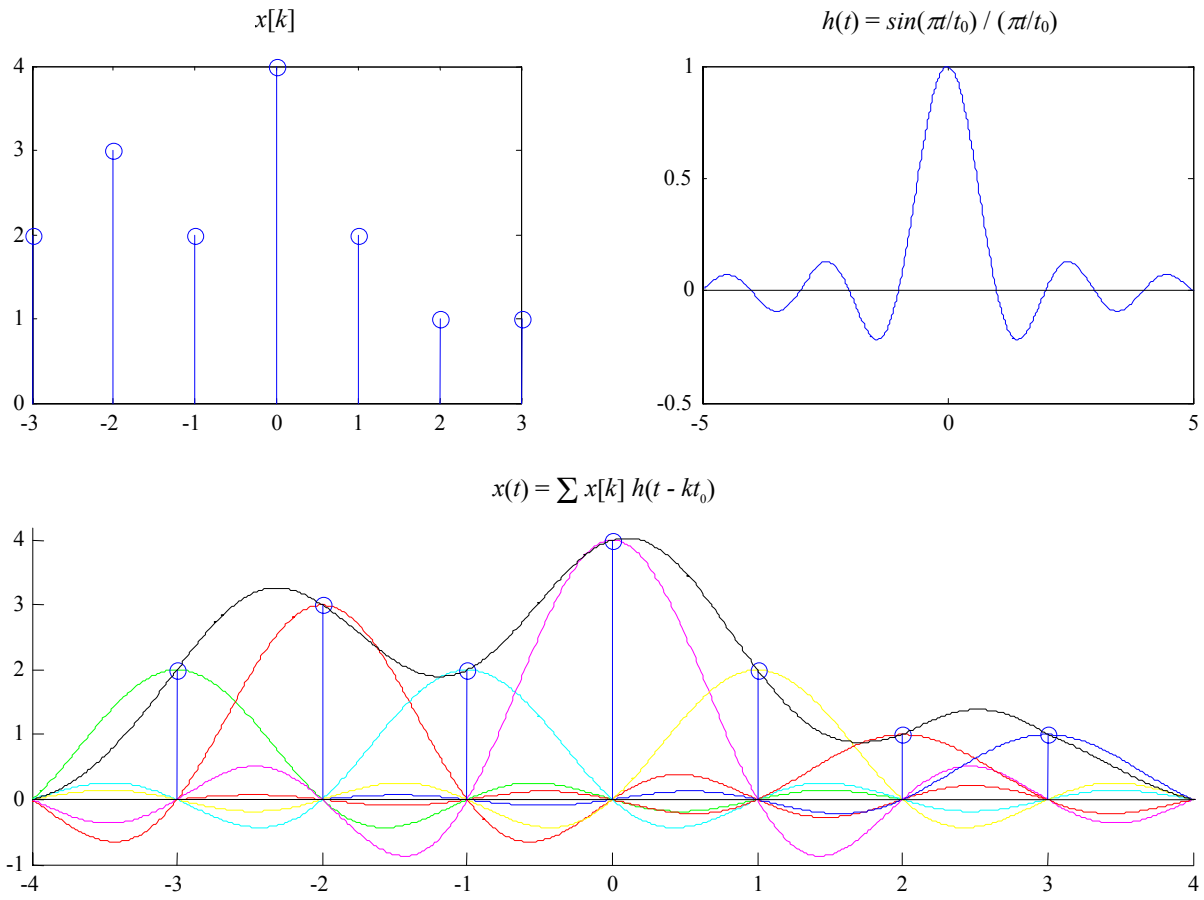


Figure Three: A PAM implementation of Bandlimited Reconstruction

In Figure Three above, a random discrete time sequence $x[k]$ was created that is seven samples long. This finite sequence was then reconstructed using Shannon reconstruction. This might be called ideal construction. Notice in the upper right panel that $t_0 = 1$ sec, since the zero crossings of $h(t)$ occur at 1 second intervals. From this, it can be inferred that a continuous time signal $x(t)$ was sampled at a rate of one sample per second ($f_0 = 1$ Hz and $\omega_0 = 2\pi$ rad/sec) to generate $x[k]$.

PRACTICAL RECONSTRUCTION METHODS: THE DIGITAL PIECE

The basic electronic element used in converting a discrete time digital signal to a continuous time analog voltage signal is called a *digital to analog converter* or *DAC*. If the conversion is very fast, then the output of the DAC will be essentially constant between sample times, and look like a sequence of up/down stair steps. We shall see that this signal has undesirable high frequency artifacts. For a sample rate of 4 kHz, for example, the DAC output will sound artificial with high frequency tinny sounds that aren't in the right harmonic relation to the original signal. These high frequency components are at the aliases of the baseband signal. Therefore they move up in frequency with the sample rate.

The process of quality reconstruction of signals involves a sample rate increase before using a DAC. If the samples, which are inserted between the known samples, $x[k]$ are smoothly interpolated, then the original high frequency artifacts can be completely suppressed.

It is important to understand how this whole process works. An incoming continuous signal, $x(t)$, is bandlimited by an anti-aliasing filter and then sampled (analog to digital conversion) to form $x[n]$. If the samples $x[n]$ were sampled

at or above the Nyquist sampling rate, they will contain enough information to reconstruct the analog signal $\hat{x}(t)$, which is an estimate of the original $x(t)$. These samples, $x[n]$, may then be stored on a CD (compact disc) or transmitted wirelessly from one cellular phone user to another for example. Once this digital signal $x[n]$ needs to be recovered back into an analog signal, it is processed digitally and then converted back to an analog signal. This is shown below in Figure Four. Since ideal reconstruction is not practical, digital signal processing is required before the analog reconstruction process can take place. Let us discuss the elements of Figure Four.

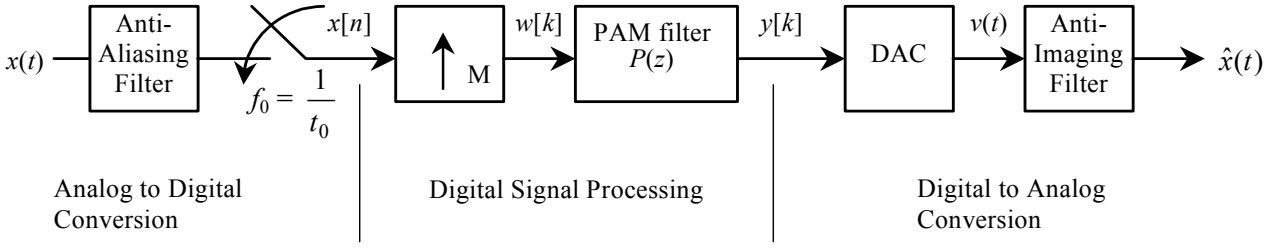


Figure Four Digital Signal Processing block diagram

Upsampling or Zero-Fill

The first block in the Digital Signal Processing section of Figure Four above is a rate conversion of the digital signal $x[k]$. The sampling rate has been artificially increased by the factor M, by placing $M-1$ zeros between every two samples $x[k]$. This is known as *upsampling* (or *zero-fill*). The output of the upsampler is

$$(5) \qquad w[k] = \sum_{n=-\infty}^{\infty} x[n]\delta[k-Mn] \xleftrightarrow{DTFT} W(e^{j\omega t_0/M}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j(Mn)\omega t_0/M} = X(e^{j\omega t_0})$$

The integer M is the *upsampling ratio* and M/t_0 is the new sampling rate after the insertion of $M-1$ zeros. We say the sampling period for $w[k]$ is t_0/M, and the sampling frequency is Mf_0. What effect does this have in the frequency domain? Since the non-zero samples $w[Mn] = x[n]$ occur only at multiples of M, the terms in the DTFT of $w[k]$ are exactly the same as the terms in the DTFT of $x[n]$. Thus the Nyquist band of W is $\frac{-M\pi}{t_0} < \omega \le \frac{M\pi}{t_0}$, and on this band

$$(6) \qquad W(e^{j\omega t_0/M}) = X(e^{j\omega t_0}),$$

The graph of this as a function of ω is unchanged except that it extends M times as far to get to the new half-sampling frequency. This is illustrated in Figure Five below. There is another way to look at this. By filling the gaps with zeros, we have introduced no information that wasn't already there. This demonstrated in Figure Five below.

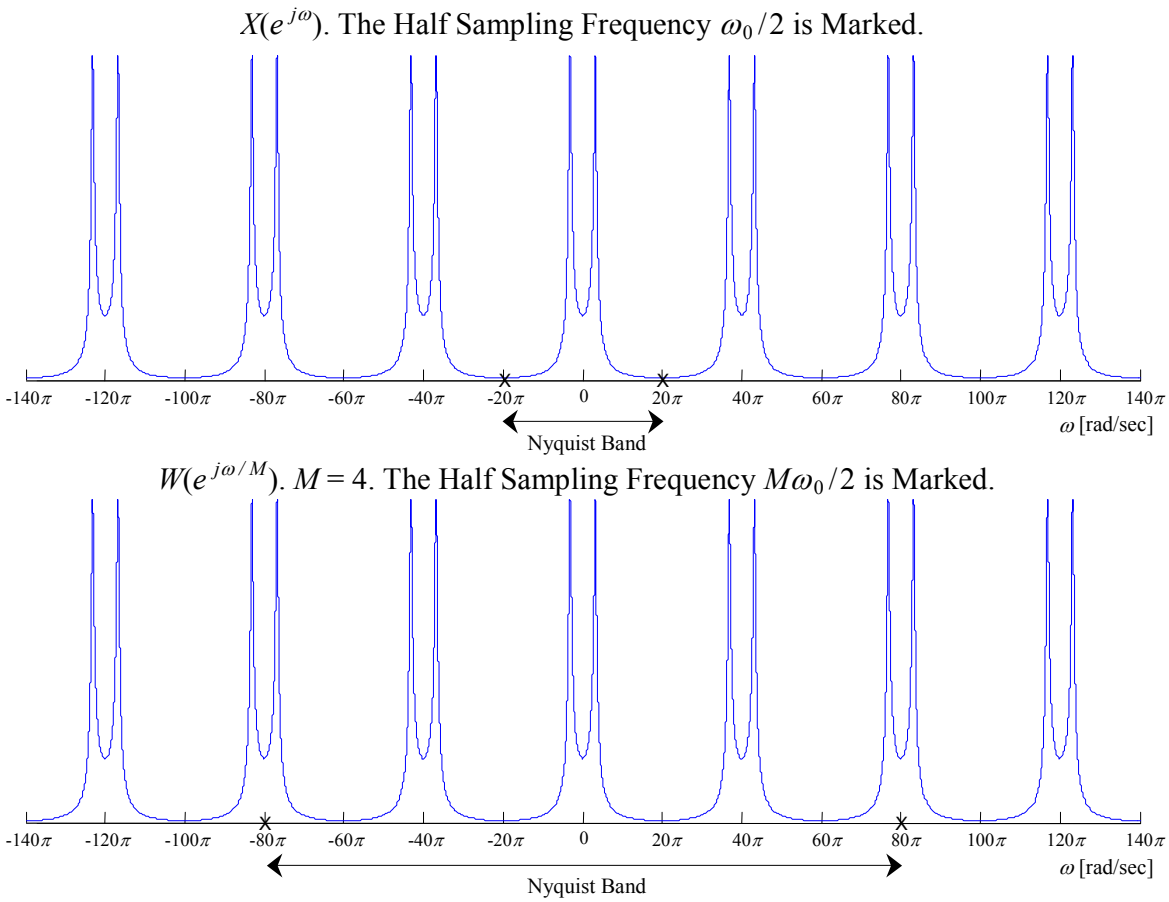


Figure Five Zero-Fill Upsampling Interpreted in the Frequency Domain

Notice that the original signal $X(e^{j\omega})$ has a Nyquist band of 20π (-10π to 10π) and $W(e^{j\omega})$ has a Nyquist band of 4*20π= 80π (-40π to 40π), yet the frequency content is the same in both graphs. Only the Nyquist band has changed as a result of $w[k]$ having a higher sample rate than $x[n]$.

Upsampling with a discrete-time PAM Filter

The second block in the Digital Signal Processing section of Figure Four above is a digital filter, which is used for digital interpolation. This filter is also called a discrete-time PAM filter. The output of this filter is the convolution of the zero-filled sequence $w[k]$ with the pulse response of a PAM (or smoothing) filter $p[k]$:

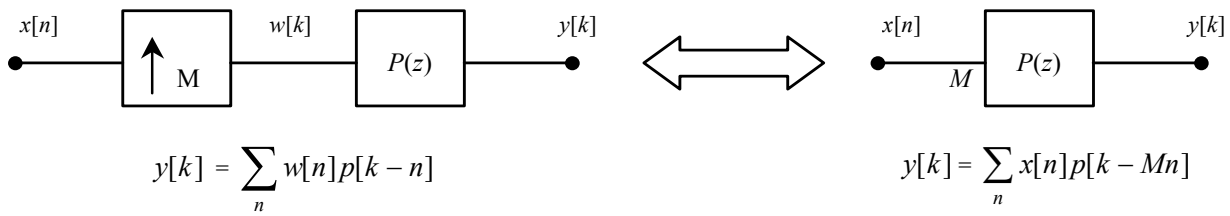
$$(8) \qquad y[k] = \sum_{n=-\infty}^{\infty} w[n]p[k-n] = \sum_{n=-\infty}^{\infty} x[n]p[k-Mn] \quad \xleftrightarrow{DTFT} \quad Y(e^{j\omega t_0/M}) = W(e^{j\omega t_0/M}) \cdot P(e^{j\omega t_0/M}) = X(e^{j\omega t_0}) \cdot P(e^{j\omega t_0/M})$$

where $P(z)$ is the transfer function of the PAM filter:

$$(9) \qquad P(z) = \sum_{k} p[k]z^{-k}$$

Note here that the discrete-time PAM filter, $p[k]$, runs at a rate M times faster than $x[n]$, meaning it outputs values $y[k]$ at rate Mf_0 in response to nonzero inputs at rate f_0. In order for the convolution in equation (8) to be

conventional, the rate of $x[k]$ had to be increased to the rate of $p[k]$, hence $w[k]$ was created to match the rate of $p[k]$. From equation (8) we can see that the following notations are equivalent:



$$y[k] = \sum_n w[n]p[k-n] \qquad y[k] = \sum_n x[n]p[k-Mn]$$

Figure Six Equivalent Implementations of the Same Digital Interpolation Scheme

Figure Six illustrates two equivalent digital interpolation schemes that take a low rate input signal, $x[n]$, and output a higher rate signal $y[k]$. On the left side of Figure Six, a more traditional and mathematically conventional implementation is given. In this case, the signal rate of $x[n]$ is increase to the rate of the interpolation filter by the use of a zero-fill upsampler or rate converter. The output of this upsampler, $w[k]$, is then passed through the interpolation filter $p[k]$ where a straight convolution is used to generate the output $y[k]$. In this case, $p[k]$ is running at the same signal rate as $w[k]$. This is the way that many textbooks explain the digital interpolation process. The right side of Figure Six uses a different representation of digital interpolation. A low rate signal, $x[n]$, is running through a high rate filter. This filter is outputting samples $y[k]$ at a rate M times faster than the inputs $x[n]$ are coming in. Given that both of the systems are *linear*, an input of zero must output a zero. For the case on the left, $w[k]$ and $p[k]$ are running at a rate that is M times faster than the input $x[n]$, but only the low rate inputs from $x[n]$ produce non-zero outputs from the high rate filter $p[k]$. From this, it should be clear that both implementations produce the same exact output, $y[k]$.

Figure Seven depicts what happens in this *upsampling* process, for a particularly uninspired kind of interpolation. Suppose that we increase the sampling rate by a factor of four by simply repeating each sample four times (known as zero-order hold upsampling by a factor of four). Clearly this won't contribute anything to the DAC stair step output problem, since it will lead to the same eventual analog signal as we would have had without the sample rate increase, but we want to analyze the general case, and this is one example. All five of the signals in Figure Seven have the same time base. The first signal is the original continuous time signal $x(t)$. We will use a simple pulse shape. (This pulse was constructed in MATLAB using the statement `x=pulse(20,9,3)`. The function 'pulse.m' can be found on the webpage.) The sequence of samples is shown in the second panel of Figure Seven. The upsampled sequence $w[k]$, which is four times as fast as $x[n]$, is shown in the third panel. The impulse response of $p[k]$ is shown the fourth panel and the final output $y[k]$ is shown in the last panel.

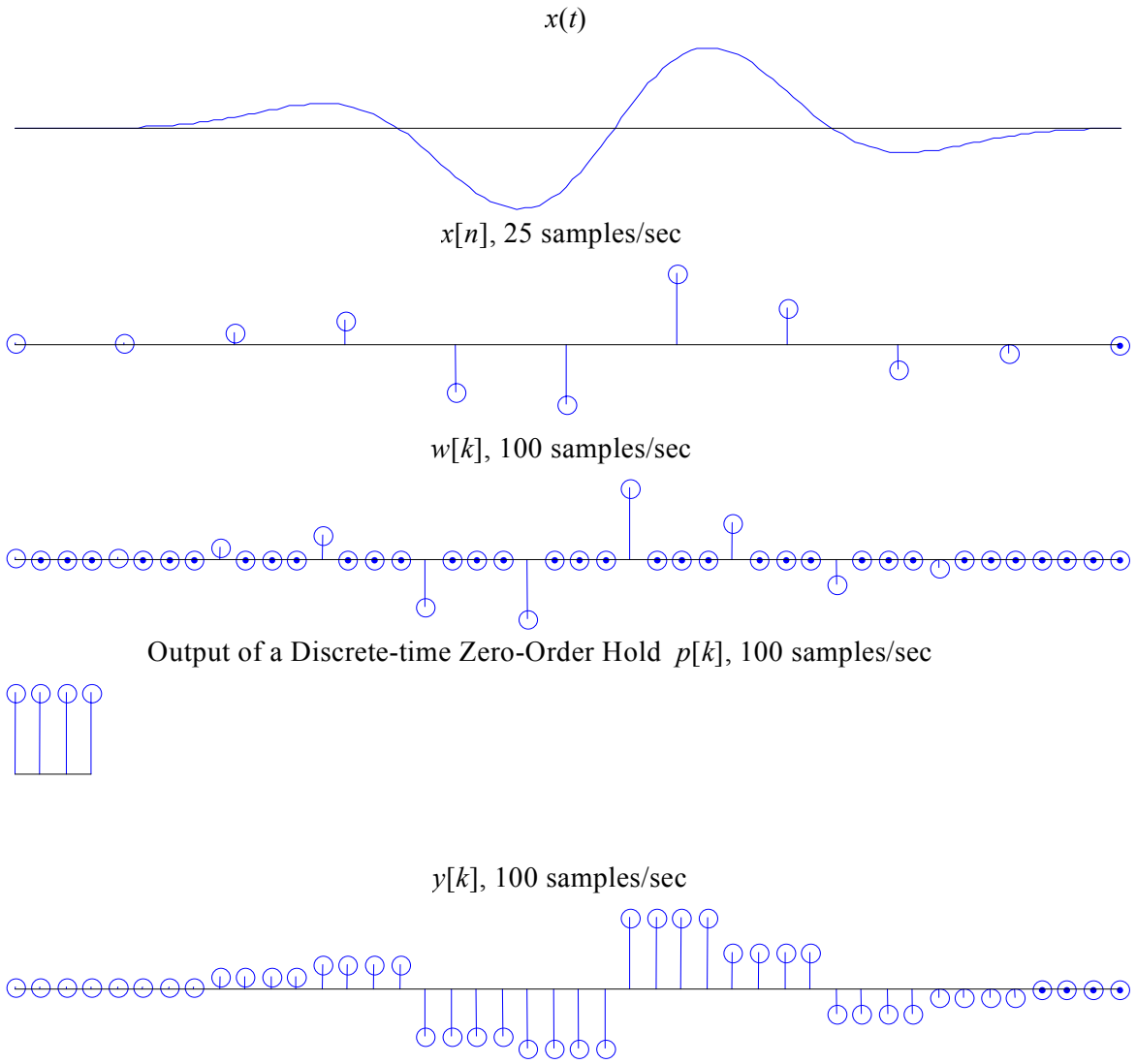


Figure Seven Simple repetition of known samples

Even though Figure Seven does not help with DAC problem, it gives invaluable insight to the PAM idea that was illustrated in Figure Six. Assume that $w[k]$ was never created and that $p[k]$ is running at a rate 4 times faster than the $x[n]$. As each sample of $x[n]$ enters the filter, the full impulse response of $p[k]$ is outputted before the next impulse of $x[n]$ excites the filter. This is a special case where the filter is only outputting a response based on a single input sample. In other words, the filter does not have to perform any non-zero summing operations before it outputs its current sample. Even though this is a trivial case, it helps to demonstrate the idea of how a high rate discrete-time PAM filter responds to a low rate input signal, without the signal rate conversion. See the m-file 'demo_PAM_filter.m' located on the webpage under 'Demos for Lab 10'. Down load this file and run it a few times to get the feel of how discrete-time PAM filters work. For your own amusement, re-write 'demo_PAM_filter.m' to model the left half of Figure Six. Note, only the signal rate of the input will change. The output should look exactly the same.

The frequency responses of each of the five panels in Figure Seven above are given in Figure Eight below. Another common way to graph a complex frequency response is to use the identity $\omega = 2\pi f$. Now $X(j\omega) = X(2\pi jf)$, and the frequency axis is now in Hz instead of radians/second. This is the technique used in the following complex frequency response graphs. The same five signals are shown with a common frequency axis. The half sampling

frequency for the four discrete-time signals are marked on the horizontal axis. Without this mark, the second and third panels, which show $X(e^{j2\pi f t_0})$ and $W(e^{j2\pi f t_0/M})$ would look exactly the same, because of equation (7). Indeed the only difference in the frequency domain is the sampling rate. The signals in the three lowest panels are related by equation (8), i.e. $Y(e^{j2\pi f t_0/M})$ is the product of the aliased spectrum $X(e^{j2\pi f t_0})$ and the interpolation filter frequency response $P(e^{j2\pi f t_0/M})$. In Figure Eight $P(e^{j2\pi f t_0/M})$ is the DTFT of a square pulse and is a *sinc* pulse. Therefore the high frequency aliases in $X(e^{j2\pi f t_0})$ appear in the final signal $Y(e^{j2\pi f t_0/M})$.

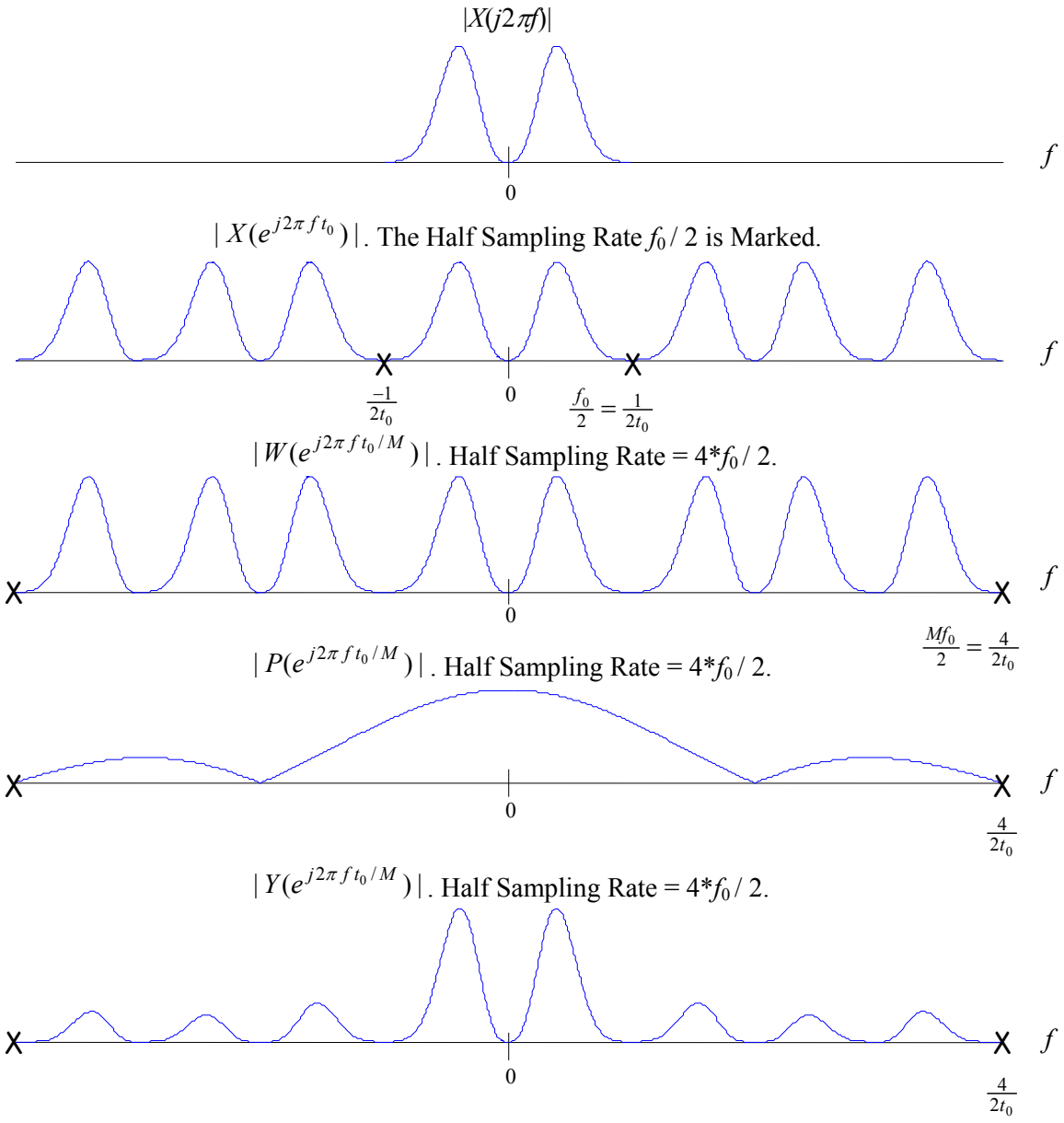


Figure Eight The Frequency domain equivalent of Figure Seven.

You should be able to compute the complex frequency response $Y(e^{j2\pi f t_0/M})$ to verify that panel 4 of Figure Six looks right.

A more inspiring type of digital interpolation can be seen when $p[k]$ is a *sinc* function. The advantage to using this is that a *sinc* function acts as a lowpass filter in the frequency domain, which will do a better job of suppressing the higher frequency artifacts of $Y(e^{j2\pi f t_0/M})$ that can be seen in the fifth panel of Figure Eight. This is demonstrated in Figures Nine and Ten below. Figure Nine contains the time domain graphs and Figure Ten the frequency domain graphs. The same pulse $x(t)$ was used in Figures Seven through Ten.

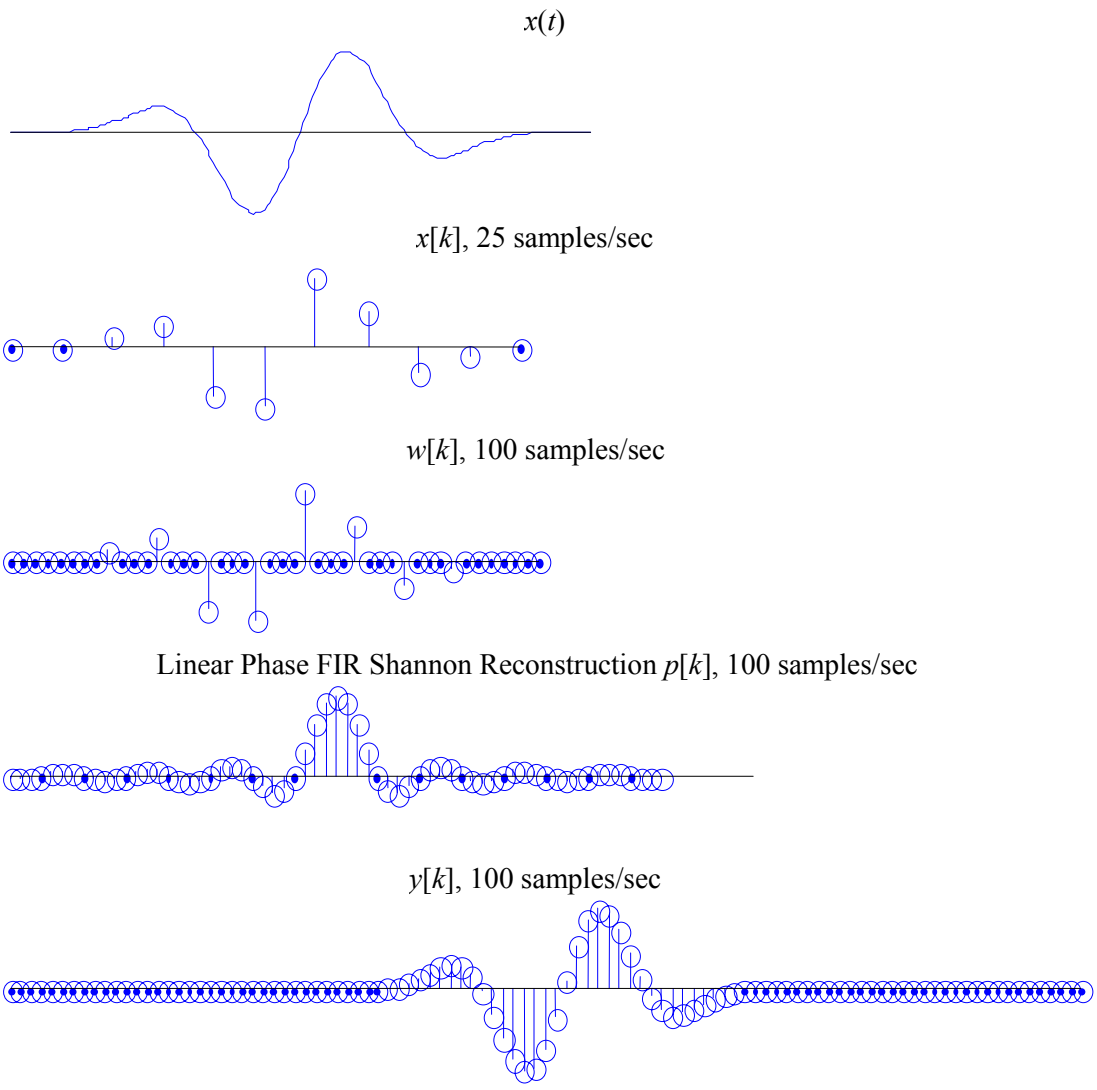


Figure Nine Smoothing with a digital lowpass filter

Notice that $y[k]$ now looks more like the original analog signal $x(t)$, excepted that it is delayed in time. This is the price that one must pay for better digital interpolation.

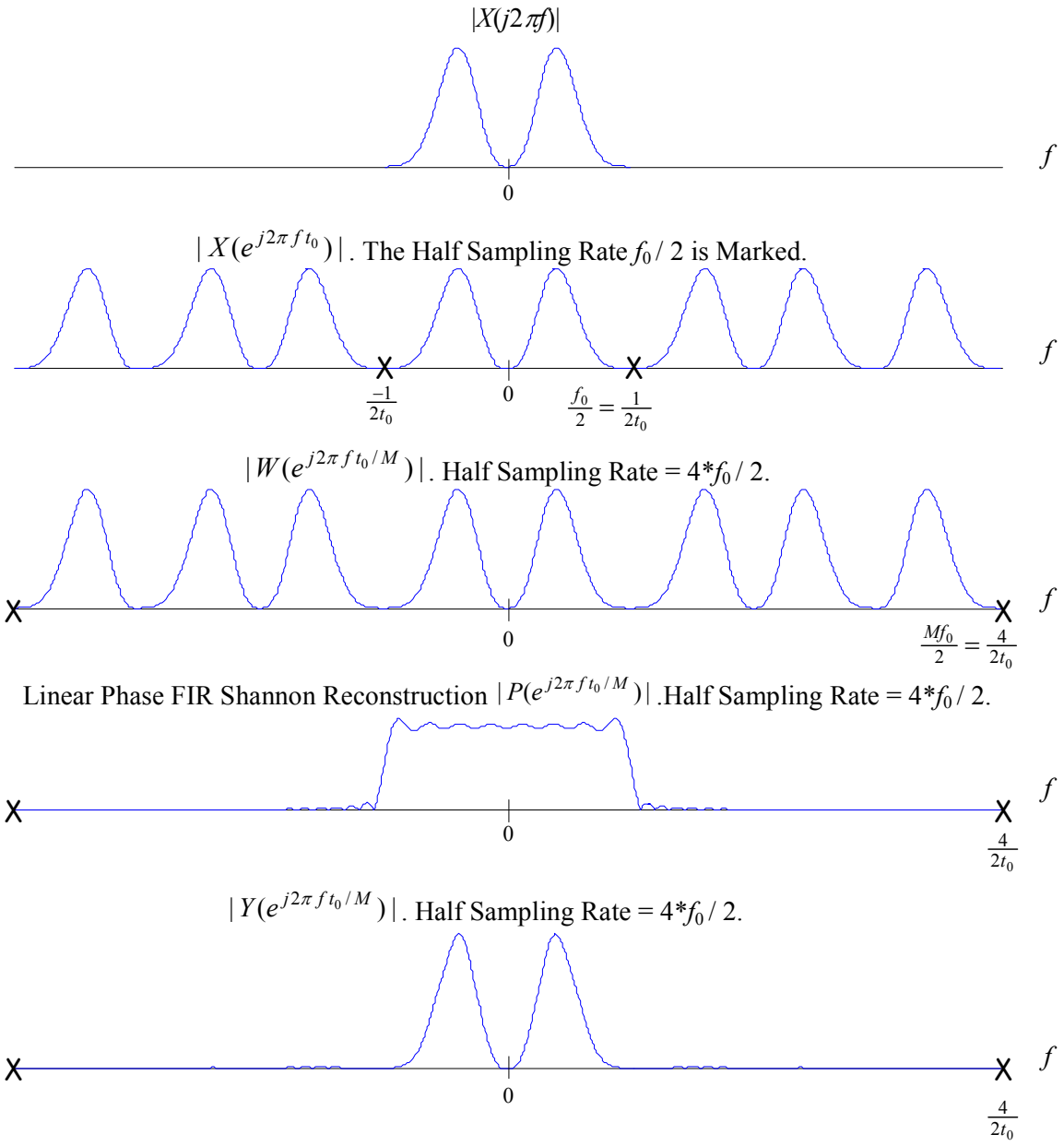


Figure Ten The Frequency domain equivalent of Figure Nine.

In Figure Ten, the high frequency aliases are suppressed. If the filter $P(z)$ were an ideal lowpass filter with sharp cutoff, then the whole baseband $0 \le f < f_0/2$ would be usable. But this is not practical, and therefore a compromise must be made. In high fidelity digital audio, the analog signals are bandlimited to 20 kHz in the recording studio with a high quality lowpass filter. Then the sampling is done at 44.1 kHz, which is of course 10 percent above the Nyquist rate of twice 20 kHz. The extra band above 20 kHz and below 22 kHz is used for a *transition region* in the design of the digital PAM interpolation filter. Such a transition can be seen in the fourth panel of Figure Ten. In the aliasing measurements that will be made, we will see a degradation in the reconstruction of sinusoids about 10 percent below the half sampling frequency.

THE FINAL STAGE OF CONVERSION: THE ANALOG PIECE

The upsampled and interpolated sequence $y[k]$ is now passed through a DAC. The high frequency artifacts due to this operation will be multiples of Mf_0 or the upsampling ratio times the original sampling frequency. Here they can easily be removed with a cheap analog lowpass filter. The process is shown in Figure Eleven. Note that this is the last section of Figure Four.

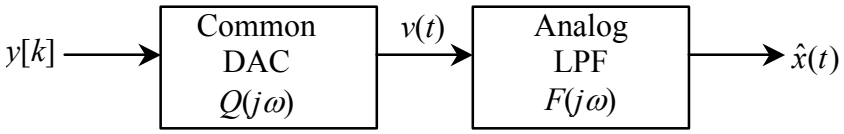


Figure Eleven Analog output stage

The final continuous time output signal is

$$(10) \qquad \hat{x}(t) = \int f(t-\tau)\sum_k y[k]\,q(\tau - k\frac{t_0}{M})d\tau \quad \xleftrightarrow{DTFT} \quad \hat{X}(j\omega) = Y(e^{j\omega t_0/M})Q(j\omega)F(j\omega)\,.$$

Here, $Q(j\omega)$ represents the DAC. In time the DAC filter will have a unit step response, which is a square pulse of width t_0/M. The Fourier transform will have magnitude

$$(11) \qquad \left|Q(j\omega)\right| = (t_0/M)\left|\operatorname{sinc}(\pi\omega/M\omega_0)\right|.$$

Let the output of the DAC be

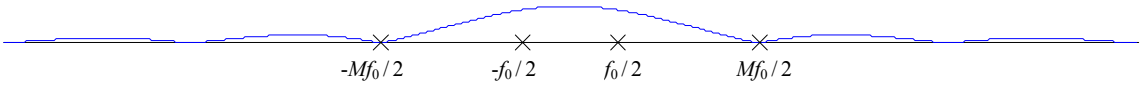
$$(12) \qquad v(t) = \sum_k y[k]q(t - k\tfrac{t_0}{M}), \text{ where } q(t) = \begin{cases} 1, & 0 \le t \le \frac{t_0}{M} \\ 0, & otherwise \end{cases},$$

with the following spectrum

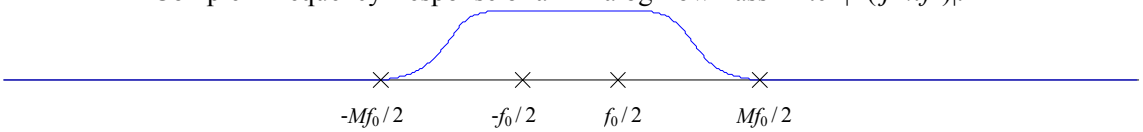
$$(13) \qquad \begin{aligned} v(t) \xleftrightarrow{DTFT} V(j\omega) &= \sum_k y[k]\int q(\tau - k\tfrac{t_0}{M})e^{-j\omega\tau}d\tau \\ &= \sum_k y[k]e^{-j\omega k\frac{t_0}{M}}Q(j\omega) \\ &= Y(e^{j\omega t_0/M})Q(j\omega) \end{aligned}$$

Notice that the analog reconstruction in equation (12) is a PAM equation just as it was in equation (5). The difference is that equation (12) uses a causal square pulse, and equation (5) uses a non-causal *sinc* function. Let us examine the frequency responses of each of the boxes in Figure Eleven.

Complex Frequency Response of a Common DAC $|Q(j2\pi f)|$.



Complex Frequency Response of an Analog Low Pass Filter $|F(j2\pi f)|$.



Complex Frequency Response $|Q(j2\pi f)F(j2\pi f)|$.

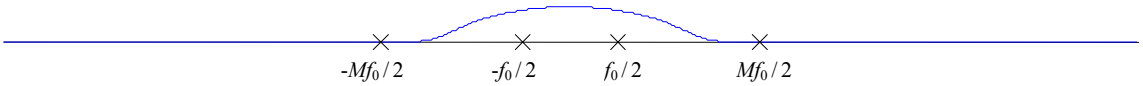
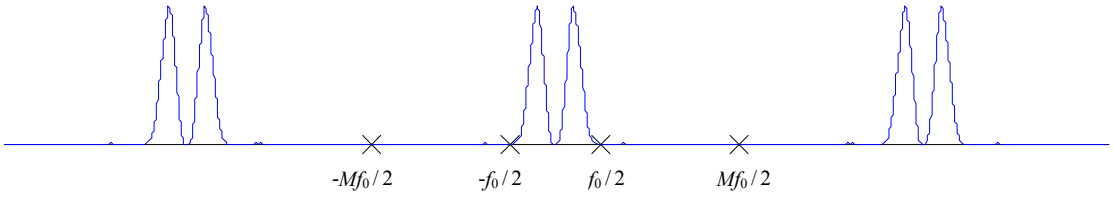


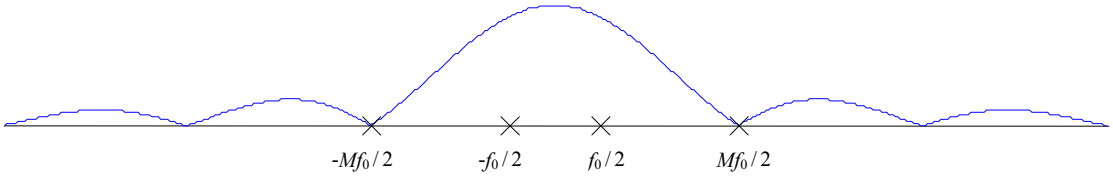
Figure Twelve Complex Frequency Response of Figure Eleven

In Figure Twelve, we can see that the DAC filter $Q(j2\pi f)$is a very poor lowpass filter whose first zero crossing is at $Mf_0/2$. Notice also that $Q(j2\pi f)$ does not fully suppress high frequency components of $y[k]$, which are *images* or suppressed aliases of the original signal $X(j2\pi f)$. To compensate for this, the signal $v(t)$ in Figure Eleven is passed through an analog lowpass filter $F(j2\pi f)$. This filter could be considered either a smoothing filter (for the effects that are seen in the time domain) or an *anti-imaging filter* (for the effects seen in the frequency domain). Either way, the product $Q(j2\pi f)F(j2\pi f)$ is then a better lowpass filter which is essentially constant on the frequency band from $-f_0/2$ to $f_0/2$, but effectively attenuates all frequency components above $Mf_0/2$. This leaves a long gap for a *transition band* from $f_0/2$ to $Mf_0/2$. This is crucial because our information lies in the frequency band $-f_0/2$ to $f_0/2$, so this is the only part of the spectrum that we want. Since the alised spectrum of $Y(e^{j2\pi f t_0/M})$ from $f_0/2$ to $Mf_0/2$ was taken out with digital signal processing, the final stage of analog signal reconstruction is not critical. Any reasonable filter will work because the difficult part has been done by the digital filter $P(z)$. We will not simulate the final output stage in Figure Nine, because (if M is reasonably large, say 4 or more) then the analog output will be indistinguishable from a graph of the discrete time signal $y[k]$ with the points connected by straight line segments. This is what the MATLAB 'plot' function does. We will however demonstrate this last step in the frequency domain. Figure Thirteen below shows the reconstruction of $Y(e^{j2\pi f t_0/M})$ from Figure Ten.

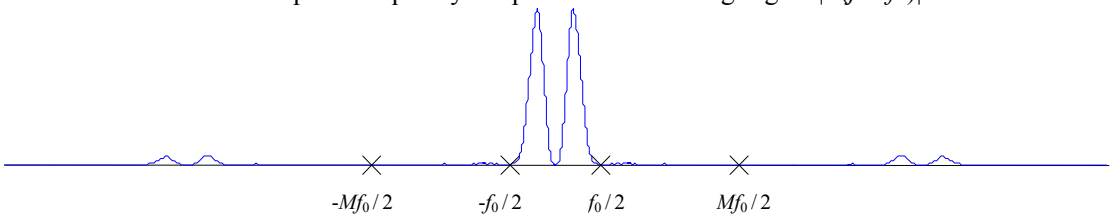
$|\ Y(e^{j2\pi f t_0/M})|$. Half Sampling Rate = $4*f_0/2$.



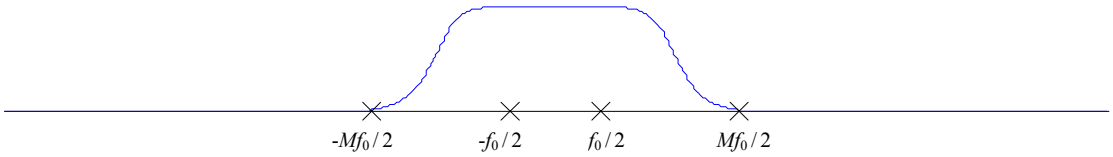
Complex Frequency Response of a Common DAC $|Q(j2\pi f)|$.



Complex Frequency Response of the Analog Signal $|V(j2\pi f)|$.



Complex Frequency Response of the Analog Low Pass Filter $|F(j2\pi f)|$.



Complex Frequency Response of the output $|\ \hat{X}(j2\pi f)\ |$.

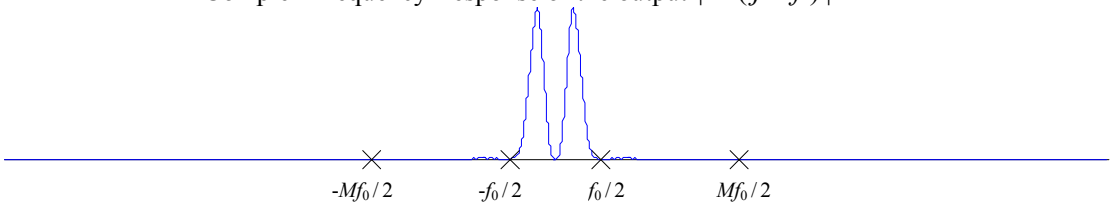


Figure Thirteen Analog Reconstruction of Figure Ten

Figure Ten showed only the Nyquist band of $Y(e^{j2\pi f t_0/M})$, but the Nyquist band aliases every Mf_0. This is shown in the first panel of Figure Thirteen. The shortcoming of just using a common DAC is that the higher frequency aliases of $Y(e^{j2\pi f t_0/M})$ will not be suppressed. The combination of a DAC and a reasonable analog LPF will then be enough to give reasonably accurate analog reconstruction of a digital signal. You may notice some frequency components in $\hat{X}(j2\pi f)$ that were not in the original signal $X(j2\pi f)$. This is due to the fact that a mid-grade digital filter was used in Figure Ten. In practice, digital filters are used that have a better complex frequency response, but it helps to show that the quality of the final analog signal depends almost entirely on the quality of the digital signal processing performed and not the quality of the analog components used.

MATLAB TOOLS FOR RATE CONVERSION

You will find two MATLAB functions for up and down sampling, and two demonstration m-files, which simulate experiments, involving rate conversion. The PAM filter upsampling process described in the text above can be done with the function 'd_up.m' found on the web page under 'Functions For Lab 10'. Type

```
»help d_up

y=d_up(x,ratio,type);
upsample x, at an integer (ratio)=f_out/f_in
using PAM. Time delay is returned in second output.
pulse types:
0 zero fill preserved
1 square or zero-order hold (ordinary DAC)
2 triangular (linear interpolation)
3 FIR linear phase hamming of order 8(ratio)+1
```

There are four PAM filters available, as one can see from the above 'help' listing. These are zero-fill upsampling without any filtering at all, the square pulse used in Figures Seven and Eight, a linear interpolation scheme which uses a triangular pulse response, and the Shannon approximation used in Figures Nine and Ten. Any integer upsampling ratio, from 0 to 16 can be used. The original samples will be preserved by any of these upsampling schemes. However the Shannon approximation will have a latency (delay) of d. To get the delay as well as the output signal, use the form [y,d]=d_up(x,ratio,type).

The down sampling process (subsampling) uses the tool 'd_down.m', which is also on the web page under 'Functions For Lab 10'. Type

```
»help d_down

x=d_down(u,decimate);
downsample x, at an integer (decimate)=f_in/f_out
```

The demonstration m-file 'demo10a.m' produces three panels of output like those shown in Figure Fourteen. In the first panel, the original signal $x(t)$ and the (time delay adjusted) final continuous time output $\hat{x}(t)$ are overlayed. Also the samples of $x(t)$ which were used to produce $\hat{x}(t)$ are shown with small circles. The spectra of these two signals are shown in the second and third panel. This simulation necessarily uses short signals with a small number of samples. One can choose three kinds of signals, shown in the note below Figure Fourteen. There is a frequency parameter alpha that is normalized. The sampling frequency is alpha = 2. The half sampling frequency is alpha = 1. The PAM filter type and upsampling ratio are arbitrary. Since this is an m-file, one must specify all parameters in command mode.

The demonstration m-file 'demo10b.m' produces two panels of output like those shown in Figure Fifteen. These are simply the time domain and frequency domain representations of the final continuous time output $\hat{x}(t)$. This simulation involves longer signals with many samples. If sound is available, the signal will be played through the audio output. This is especially revealing of the high frequency artifacts of the reconstructions. In both these simulations, the sampling frequency is 4 kHz. The total time duration is about 0.128 seconds. The type of reconstruction filter will be divulged in the titles of the time domain plots. The upsampling ratio can be detected by dividing the maximum frequency of the reconstruction spectrum by 2000, which is the original half sampling frequency.

For sinusoidal signals, the spectra from 'demo10b.m' will be much sharper than those from 'demo10a.m' because the data lengths are greater.

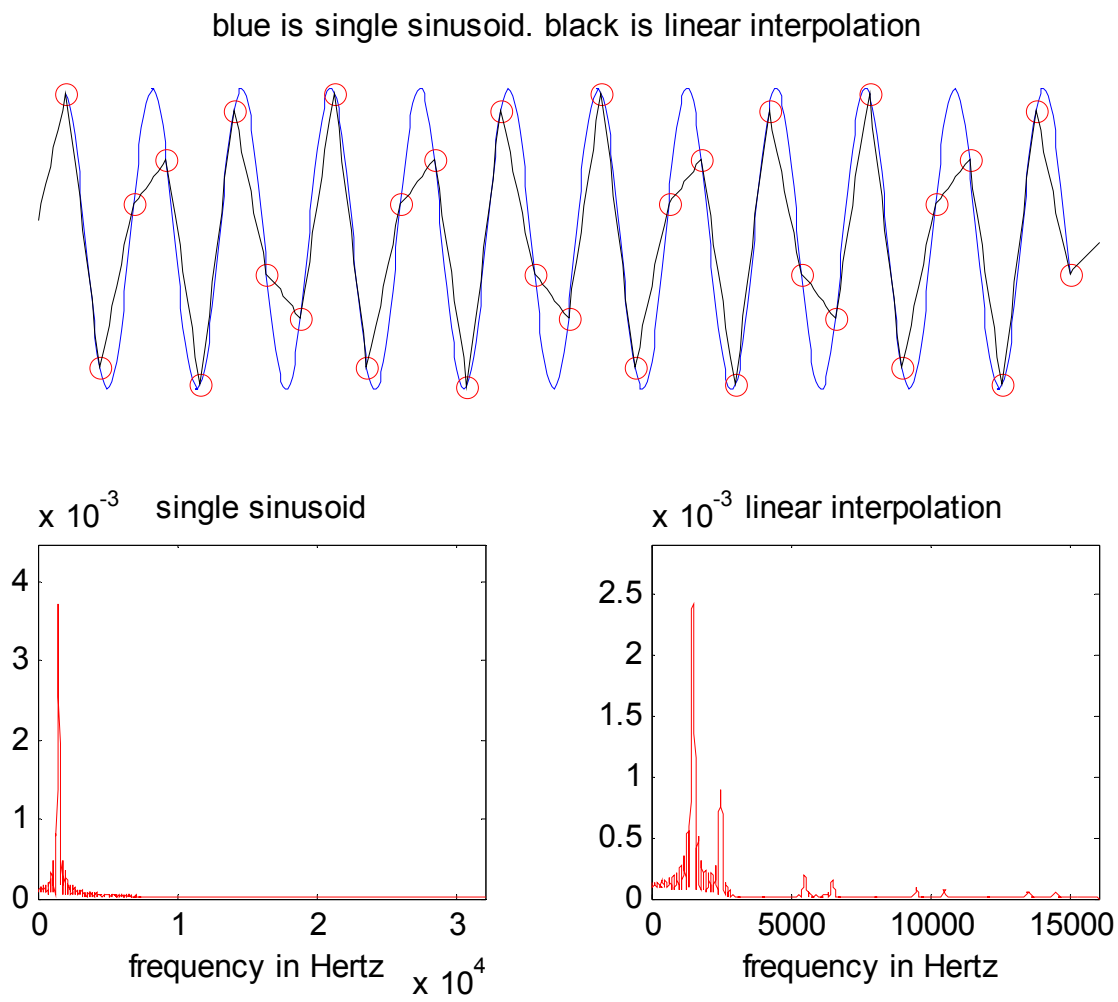


Figure Fourteen Output of 'demo10a.m'.

This example uses type=2, or linear interpolation, and s='s' for a sinusoidal signal. The upsampling ratio is 8. The frequency of the sinusoid is determined by the parameter alpha. In this case, alpha = .75. Observe the errors in the time domain, and the high frequency artifacts in the frequency domain.
To recreate this in MATLAB, type »s='s'; type=2;ratio=8; alpha=.75;demo10a

```
»help demo10a

demonstration of aliasing
undefined vbls: s,type,ratio,alpha (must defined in workspace)
s='s' sinusoid with normalized frequency alpha
s='c' two sinusoids with normalized freq. alpha, alpha/3
s='b' narrowband process with normalized center frequency alpha
(for type and ratio see d_up.m)
```

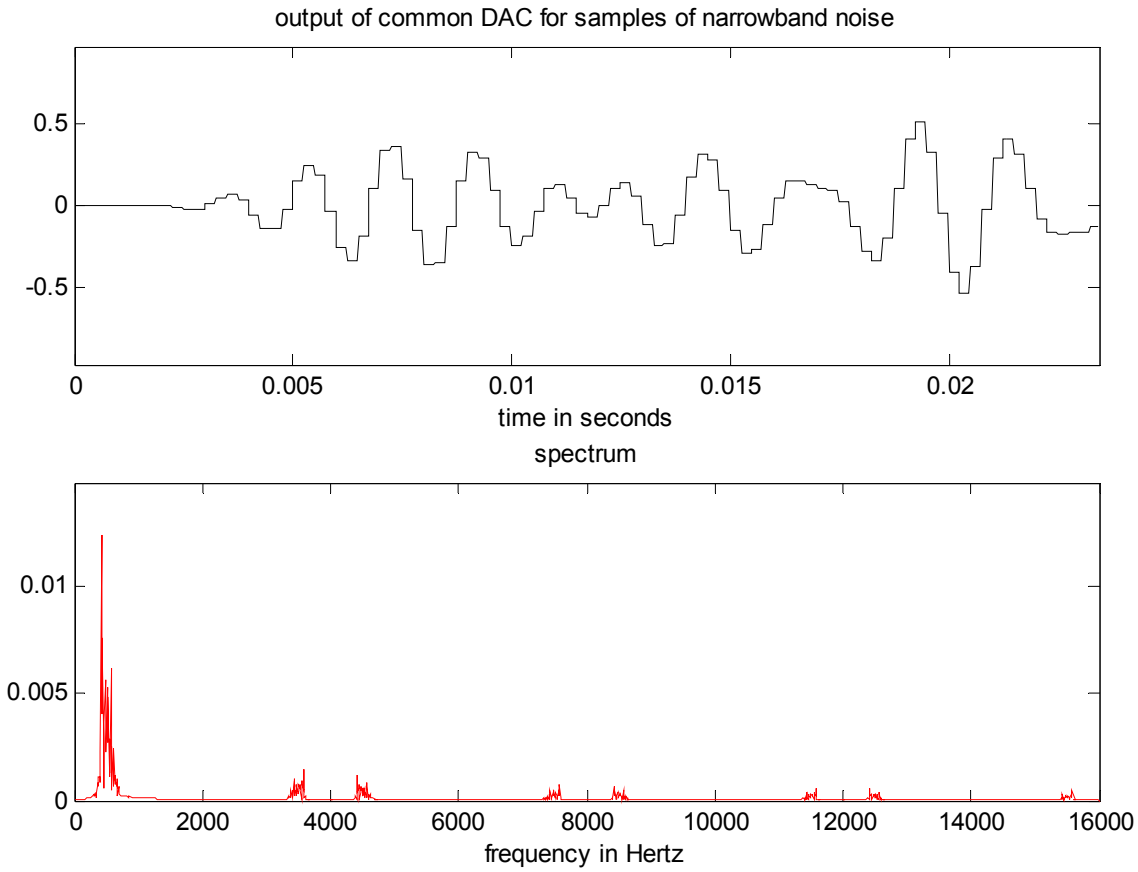


Figure Fifteen Output of 'demo10b.m'.

The signal type is s='b', for a narrowband signal. The center frequency is determined by the parameter alpha. In this case, alpha = .5. The upsampling ratio is 8, and the PAM reconstruction type is 1, for a square pulse response. Observe the steps in the time domain plot, and the high frequency artifacts in the frequency domain.
To recreate this in MATLAB, type »s='b'; type=1;ratio=8; alpha=.5;demo10b

```
»help demo10b

demonstrate interpolation/reconstruction
of analog signals from samples
undefined vbls: ratio, type, s (string), alpha  (must defined in workspace)
signals:
s='s' single tone at frequency alpha*fs/2
s='c' chord
s='u' unit pulse
s='n' noise
s='b' band noise
```

THE DIGITAL CRACKS HYPOTHESIS

Several years ago the following Exchange was made within a certain users group on the net. It involves a theory about the quality of digital audio that says that certain frequencies are favored, because after all, the medium is digital. I have removed all information that would identify the authors, and I hope that I am not breaking any laws by reproducing this material.

(Fan number one.) *Now, I know that CDs have some major advantages over vinyl in terms of wear -n- tear, and background noise due to dust, etc. My friend and I came up with a theory why digital music will never reproduce sound as well as analog devices. Since music is made up of sound waves which range over a 'continuous' spectrum of frequencies, it seems that digital music will suffer in any frequencies 'in between' the ones recognized by the digital equipment.*

(Fan number two.) *This is quite true. The analog-to-digital converters are calibrated to pick up the frequencies of _correctly tuned_ guitar strings, piano keys, etc. These are reproduced unerringly. However, it is seldom the case that instruments are in perfect tune. If, for example, an A is 441 Hz rather than the standard 440, a digital recorder will not reproduce it at full volume, and it may be slightly "muffled". This is the key to the oft-heard complaint that "grungy" music (e.g., Sonic Youth) sounds too thin and not very powerful on CD relative to LP. In the case of SY, they often use non-standard guitar tunings, which, since they do not go to the expense of using specially-calibrated A/D converters, are poorly converted to digital waveforms. A group like Floyd would certainly be in perfect _relative_ tune in the studio, but possibly not in perfect _absolute_ tune. The effect would be the same -- dull sound from the "falling between the binary cracks" effect. If, during the re-mixing from two-track analog masters to 24-track digital tape, the mastering engineer adjusts the numeric representation used in the digital domain, a fairly accurate conversion can be accomplished. In practice, a straight transfer is done with little care, and the result sounds "thin," as essential information was lost. This is probably the effect that you have noticed. It may seem odd that Mobile Fidelity Labs didn't get this right with Dark Side of the Moon, but it is rumored that this was due to their use of fixed-point arithmetic in the D/A converters due to an engineering oversight at the time (DSOTM was one of MFSL's earlier efforts, alas). The Steely Dan album I don't know about. Analogous to the problem of representing certain numbers in binary, such as repeating decimals. The new Sony MD recordable discs avoid this problem entirely by using a continued-fraction representation of the signal. This is possible because the MD format is not purely optical (as with CDs), but rather magneto-optical, thereby incorporating more of the continuous-domain information from the analog signal, much as an analog magnetic tape gives a perfectly continuous playback of an analog signal. Thus, we can expect the new format to have much less of a "binary" problem than current CDs. I hope this cleared things up.*

Assignment:

Before doing the parts below, familiarize yourself with the demonstration m-files 'demo10a.m' and 'demo10b.m'. Exercise them with all input choices, for types 1 and 3 of PAM filters. You do not need to make any prints.

1. *PAM filter types*

   Use 'demo10b.m' with a unit pulse input to observe the four PAM filters in the time and frequency domains. The following command line can be used.

   ```
   »s='u';type=1;ratio=8;demo10b
   ```

   Repeat with type = 0, 2, and 3. Make plots, and write comments on each. Mark the position of the half sampling frequency (2000 Hz). Identify "high frequency artifacts" in the spectrum.

   Now repeat these except using a sinusoidal input signal. Use the following command.

   ```
   »s='s';alpha=.7928;type=1;ratio=8;demo10b
   ```

Assignment:

2. *Aliasing*

   Use 'demo10b.m', with s='s', type=3, and ratio=8. Vary the parameter alpha between 0 and 2 to change the frequency, and observe the output in both time and frequency. It is not necessary to make any prints of plots. We want only to characterize the output. It will be either a single sinusoid (alpha=.3 for instance), or a mixture of two (alpha=.97). Ignore the little transient at the beginning. The parameter alpha is a fraction of $f_s/2 = 2$ Khz. Over the range $0 \le \alpha \le 2$ determine the output frequencies (either 1 or 2) and sketch the spectra in the range $0 \le f \le 2.5$ kHz. Use closely spaced alpha near the value alpha=1.

   For the frequencies alpha= .851,.973, 1.027,1.149, run 'demo10a.m' and observe aliasing.

   Do you detect any experimental evidence for the "digital cracks" hypothesis? If so, find a theoretical explanation. If not, find a theoretical justification.

3. Write a MATLAB m-file to duplicate Figure Nine. Use the tool 'd_up.m' to do the upsampling. Make sure that the same time axis is used for all four plots. In order to do this, it is helpful to compose a time base for each signal, like

   ```
   tw=tsw*[1:length(w)].
   ```

4. Write a function (m-file) to do sample rate conversion by a rational number M / N. This function should call both 'd_up.m' and 'd_down.m' to do the rate conversion. Create an interesting signal, upsample by M and downsample by N. (Use M=5 and N=7.) Plot the samples, using 'stem' on two separate panels, against the same time axis for comparison. Then repeat the process except downsample before upsampling. Which is preferable?