ELECTRONOTES
1016 Hanshaw Rd
Ithaca, NY 14850
(607)-257-8010

APPLICATION NOTE NO. 356

October 2003

SPECTRAL RECOVERY FOR A CLASS OF NON-UNIFORM ("BUNCHED") SAMPLING

1. INTRODUCTION:

Previously [1,2] we have remarked on the fact that while "standard" sampling with uniform spacing of the samples in time is common (and usually perfectly convenient), non-uniform sampling is often workable when the situation presents itself. In such a case, as long as the average sampling rate is at least twice the bandwidth of the signal being sampled, we can still recover the original signal completely, without error.

Specifically, here we will look at cases where possible sample points are uniformly spaced, but where not all these points are occupied by non-zero samples. In addition, there is a specific pattern to the non-uniform sampling, which we shall describe in terms of a "sampling cell" or SAMCELL. For example, if we were keeping one sample, skipping one, taking another, and then skipping three, we would have the SAMCELL represented as s=[1 0 1 0 0 0]. It is to be understood that the first number in the brackets of the SAMCELL corresponds to time index 0, and that the cell repeats periodically to form the actual sampling function. In this notation, reference [1] dealt with s=[0 1 1 1] and reference [2] with s=[1 1 0 0].

What we are looking at is the case where the bandwidth of the signal to be sampled is significantly less than half the rate of the uniformly spaced points. We find that because of this defacto smaller bandwidth, we can live with a proportional loss of samples. For example, for s=[1 1 1 0 0], a bandwidth of (0.5)(3/5)=0.3 can be supported. This reminds us of "bandpass sampling" in the uniform spacing case, and indeed, the spectral support on the frequency range 0 to 1/2 need not be continuous starting at zero (i.e., low-pass). For example, for s=[1 1 1 0 0] the non-zero bandwidth can be 0.3, and we might have this as one segment from 0 to 0.2 and another segment from 0.3 to 0.4. In such a case, it is convenient to denote this spectral support which we can call a SPECCELL as W=[1 1 0 1 0], dividing the spectrum into five segments (the same number as the length of the SAMCELL), where a 1 indicates that the segment is occupied and a 0 that it is empty. In fact, W, along with s, will be the input parameters to our procedures and programs.

It will be noted that recovery in the case of "bunched" samples is generally more difficult than in the case of uniform sampling, perhaps requiring resources augmented by something like an order of magnitude. Nonetheless, in the same sense that engineers contend that we recover signals from uniform sampling "completely," we can recover from non-uniform sampling "completely." Both cases may suggest the need for ideal devices (such as ideal filters) which need to be approximated in practice. As such, both the uniform and non-uniform cases may be approximations in some practical sense. But the methods here for non-uniform sampling are not (further) approximated as a consequence of the non-uniformity of sampling times. Mathematically, ideally, they are every bit as good as uniform sampling.

2. AN EXAMPLE WHERE ONE SAMPLE IN THREE IS LOST

2a. Uniform Sampling - Setting Up the Problem

If we suppose that we have a set of samples that have been obtained in compliance with our simplest understanding of the (uniform) sampling theorem, it is fair and conservative to stipulate that the bandwidth of the signal that was sampled was half the sampling rate. We will take the sampling rate to be a normalized value of 1, and thus we expect a bandwidth (denoted w) of 1/2. Even in theory, the mathematics says that we can only approach this value of 1/2, arbitrarily closely. (And, in actual practice, values like 0.4 or 0.45 might be expected.) However, when we make the assumption of an arbitrary closeness to 1/2, any loss of samples (loss of information) will make exact recovery of the signal impossible.

In this note, we will be dealing with signals that are (non-uniformly) sampled, and that have bandwidths that do not approach w=1/2, but rather which approach, arbitrarily closely, some smaller fraction such as w=1/3 or w=3/8. In addition to stating here this idea of approaching w, we will draw spectra (Discrete Time Fourier Transforms - DTFT's) as triangles or other shapes that seem to reach zero at w. Fig. 1 shows the spectrum of a signal that has a bandwidth of w=1/3. This signal will be supported by an average sampling rate of 2/3. Accordingly, we can anticipate a SAMCELL of length 3 for this example, in which case the SPECCELL is W=[1 1 0], in our notation.

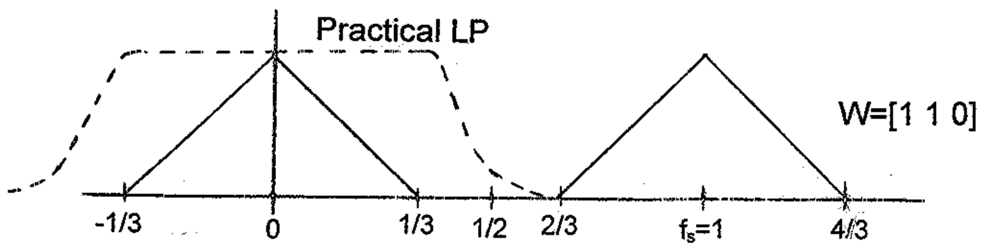


Fig. 1 An example spectrum (DTFT) of a signal that is bandlimited to w=1/3, or W=[1 1 0]. Here we assume that the spectrum is purely real.

We know from Fig. 1 that it is possible to recover the continuous-time signal that corresponds to samples taken at a rate $f_s=1$. In practice, any suitable continuous-time filter that is sufficiently flat for frequencies from 0 to 1/3, and which gets close enough to zero by frequency 2/3, will do (suggested by dashed line of Fig. 1). If we had been a bit more stingy with sampling rate, we might well have noted that the same signal could have been sampled at a rate of 2/3 instead of 1 (Fig. 2) in which case, the triangular repetitions of the spectrum would just touch. This is still just the extreme case of rather ordinary sampling, and we would need an ideal low-pass filter for perfect recovery. This does illustrate that a sampling rate of 2/3 is sufficient for a bandwidth of 1/3, at least for uniform sampling.

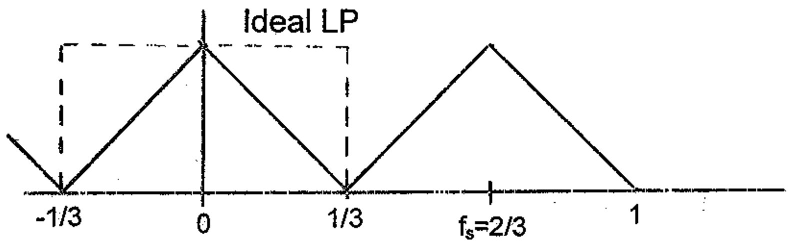


Fig. 2 A bandwidth of 1/3 sampled at 2/3 is recoverable if we had an ideal low-pass filter.

2b. Now Take 2 of 3

The new problem we intend to consider here is different in that it is the average sampling rate that is 2/3. That is, the sampling interval corresponds to a sampling rate of 1, but every third sample is somehow skipped and its position is represented by a zero value (Fig. 3). This, in our SAMCELL notation is s=[0 1 1]. It may also be thought of as a "resampling" problem: some of the samples from the full set corresponding to a rate of 1 are removed, leaving a smaller set with only 2/3 of the original numbers. These ideas relate to situations that are sometimes called "non-uniform," "bunched," or "gated" sampling.

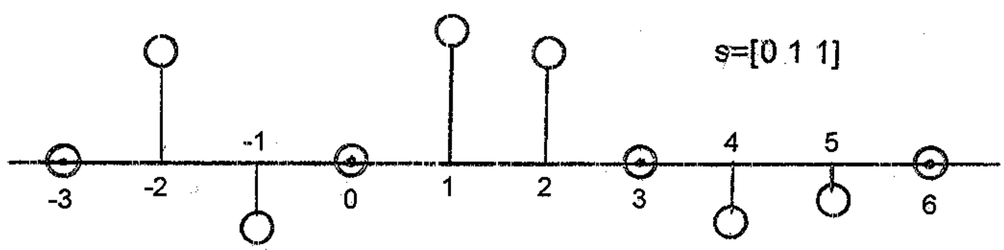
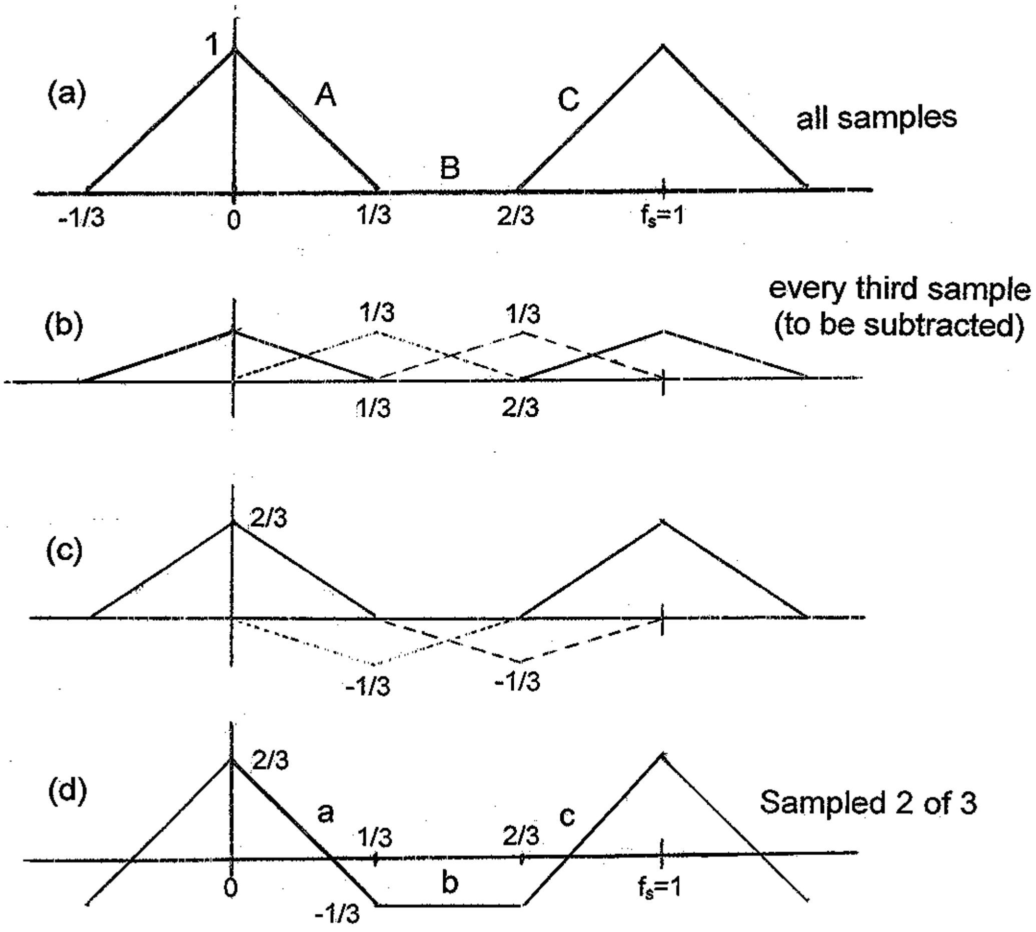


Fig. 3 Sampling 2 of 3. SAMCELL s=[0 1 1]

The problem is most naturally approached by considering it as the superposition of uniform sampling problems which we do understand quite well. For example, Fig. 3 can be considered to be the sum of two sampling situations where samples are taken at a spacing of 3 (a sampling rate of 1/3). Thus we have two sub-sequences: samples ..... ,-2, 1, 4, ..., and samples ...., -1, 2, 5, ... We note immediately that neither of these subsequences, by itself, is sufficient for a bandwidth of w=1/3 (w=1/6 would be the maximum). However, it is clear that both subsequences, taken together, represent twice the information contained in either subsequence taken alone - so perhaps we will have some chance.

In fact, it is convenient in this case to view the sampling of Fig. 3 as a different superposition; not as the sum of two subsequences sampled at 1/3, but rather as the difference between two sequences, one sampled at 1, and the other sampled at 1/3. Thus we look at Fig. 3 as the case where all samples are kept <u>minus</u> the case where samples .... , -3, 0, 3, 6, ... are kept. This will have the advantage that spectra will remain real, and we can more easily sketch the results. But this simplification is not essential as we shall see.



<u>Fig. 4</u> The spectrum of all samples (a) minus the spectrum of every third sample (b) results in (d), where (c) is an intermediate step in the superposition.

Fig. 4 shows the original spectrum (a), which has a bandwidth w=1/3, with shifted and scaled versions of itself as seen in (b) superimposed to give the spectrum of the sampled signal (d). While (a) contains all samples, (b) shows the case where only every third sample is taken: the spectrum is scaled to 1/3 its original height, and spectral replicas are added, centered at 1/3 and at 2/3. (This is ordinary re-sampling.) At this point, it is not clear that the original spectrum, and thus, the original signal (that is, the missing samples at ... , -3, 0, 3, 6, ... , and even the continuous-time signal) can be recovered. Clearly a low-pass filter, ideal or not, will not works.

Thus we proceed to write down the equations that describe how the superposition spectrum (Fig. 4d) comes about. Here we use the notation A, B, and C for the segments of the original spectrum {ABC} and a, b, and c for the segments of the superimposed spectrum {abc}. Recall that in our notation, the SAMCELL here is s=[0 1 1] and the SPECCELL is W=[1 1 0].

Thus, in matrix notation: (M labels the 3×2 matrix)

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \underbrace{\begin{bmatrix} 2/3 & -1/3 \\ -1/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}}_{M} \begin{bmatrix} A \\ C \end{bmatrix} \qquad (1)$$

We propose to invert this to recover the original segments A and C from the superposition {abc}. [We notice that because B≡0, we could have kept this in the rightmost column vector, in which case there would have been a center column in the matrix, which could have had arbitrary values! Clearly inverting a matrix for which one column is arbitrary is not going to give us a useful answer.] Since the matrix is not square, we need to use a pseudo-inverse. Calling the superposition matrix M, we have the simplest calculation of the pseudo-inverse as:

$$\begin{bmatrix} A \\ C \end{bmatrix} = (M^tM)^{-1}M^t \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$= \left[\begin{bmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} \right]^{-1} \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad (2)$$

The same result is obtained using the pseudo-inverse, pinv, in Matlab.

We note, trivially, by inspection, that this works. Section a minus section b clearly gives A. Likewise, c minus b gives C. What about B? There was no equation for B. B≡0 by the bandlimiting assumption, so we did not solve for it. That was the whole point.

Perhaps the reader is thinking: "OK - This works for the triangle in the example, but what about a general spectrum?" First of all, nothing in the mathematics used the actual shape of the spectrum. All that we achieved by using a triangular shape is an easy-to-verify result - and it seems to work perfectly.

2c. Working About the Limits

Fig. 5 shows some additional example spectra (Using Program 1). Here we generate (by choosing DFT values) a random but bandlimited spectrum. We then take the inverse DFT to get the corresponding time sequence, which is then sampled by setting every third sample to zero. The DFT of the sampled result gives us {abc} and we get {ABC} using equation (2).

Fig. 5a shows the case of w=0.1. Sampling 2 of 3 gives us images as expected. Complete recovery could have been obtained more simply with a low-pass filter in this case. Fig. 5b shows a similar case except here w=0.15, and things have gotten more crowded. Again, we do get perfect recovery using equation (2), and a low-pass would still have worked, since w=0.15 and a sampling rate of 1/3 would support a bandwidth up to (1/2)(1/3)=1/6=0.16667. (In this case, this means that the loss of samples at the rate of 1/3 would not have aliased the result.) Note as well the "recoverable" copies of the original spectrum on the interval 1/6 to 1/3 in both Fig. 5a and Fig. 5b.

[We should perhaps mention that here we are working with spectra that are limited to a bandwidth w where we are choosing w over a range to illustrate various results. At the same time, our recovery equations came from an assumption of a SPECCELL W=[1 1 0] which really means that w=1/3. Because, as is obvious, a spectrum bandlimited to some frequency w_1 is also bandlimited to w_2, as long as $w_2 \geq w_1$, W=[1 1 0] works for the examples of w=0.1 and w=0.15. We could have used W=[1 0 0] for these two cases as well. W=[1 1 0] will continue to work for $1/6 < w \leq 1/3$, although W=[1 0 0] would not.]

Fig. 5c shows w=0.25, which is greater than 1/6 of course. We note that the sampled spectrum is now distorted on the interval of 0 to 0.25, and that the region from 0.25 to 0.5 is imaged. No clear copy of the original is now available for filtering. Yet equation (2) returns the original spectrum correctly.

Fig. 5d is a limiting case as w becomes 0.33, just short of 1/3. We see that equation (2) does still work. This compares to the triangle example of Fig. 4. Now, critically, Fig. 5e shows w=0.34, and we see that the original spectrum is not recovered exactly anymore, as the limit was 1/3. [Since the spectrum is random, not all examples using w=0.34 will clearly show a failure when plotted on this scale, but a more sensitive analysis shows that w=0.33 always works and w=0.34 never works, as expected.] As we might expect, as the bandwidth continues upward beyond 1/3, the failure to recover is more easily seen. Fig. 5f shows the case of w=0.4.

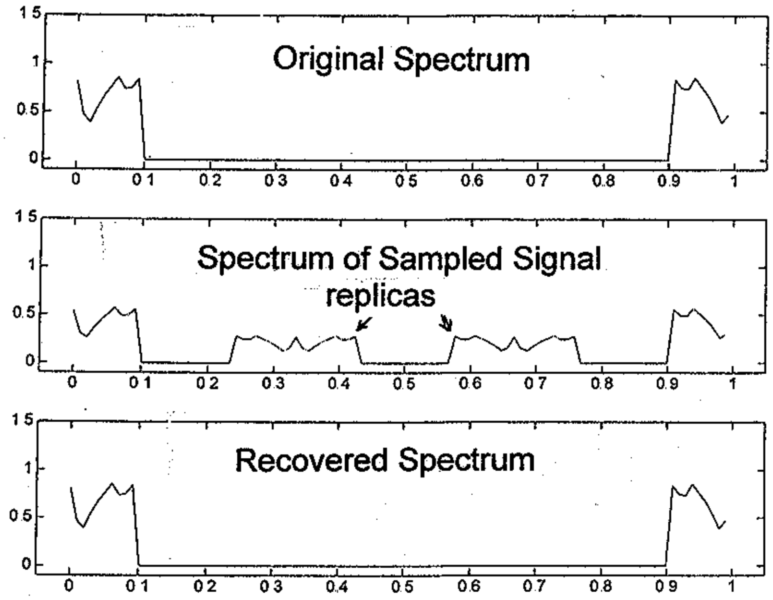


Fig. 5a. w=0.1
Here we find that sampling 2 of 3 gives two additional replicas as in equation (1). Recovery by equation (2) is complete. Recovery by low-pass filter would also be practical here. [All spectra in Fig. 5 are shown as magnitudes.]

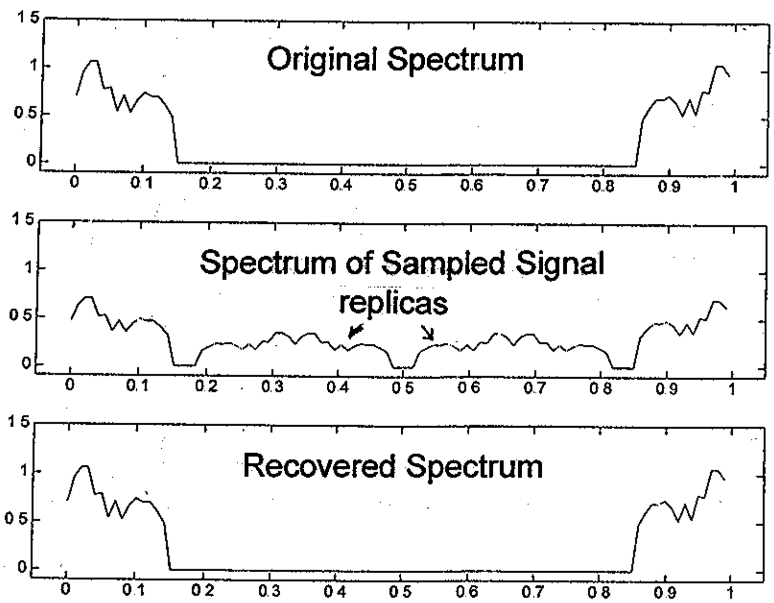


Fig. 5b. w=0.15
Similar to Fig. 5a except here we have a wider bandwidth, but one that is still less than 1/6=0.16666. Recovery using equation (2) is complete, and a low-pass filter with cutoff around 1/6 is also still possiblle, although we would need a fairly sharp cutoff.

In the case of ordinary uniform sampling, when the bandwidth just slightly exceeds the limit, only the actual overlapped portion of the spectrum is non-recoverable. The same is true here, where we see in Fig. 5e and Fig. 5f that a region centered about 1/6 is recovered perfectly. This is also true for a region around 1/2, but the spectrum is still zero there.

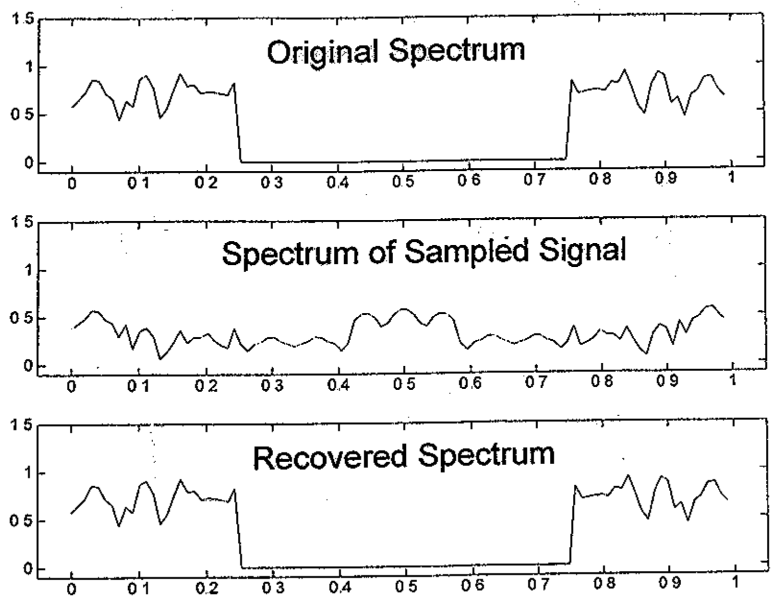


Fig. 5c. w=0.25
Here the bandwidth exceeds 1/6, and we see substantial spectral overlap in the sampled spectrum, making a low-pass recovery impossible. Equation (2) still gives complete recovery.

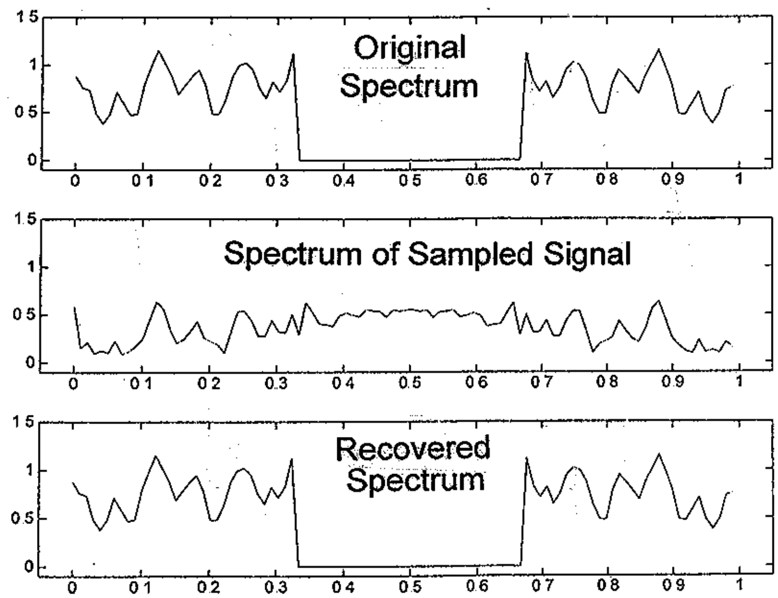


Fig. 5d. w=0.33
Here the bandwidth is only slightly less than 1/3. We have nearly complete spectral overlap in the sampled case. Equation (2) still gives complete recovery.

The fact that some portions of a spectrum are recoverable without error may be of some use at some times. But it is not true that any portion of the corresponding time-domain sequence is recoverable without error in such cases. All samples are now wrong to some degree - easily seen as a consequence of the uncertainty principle.

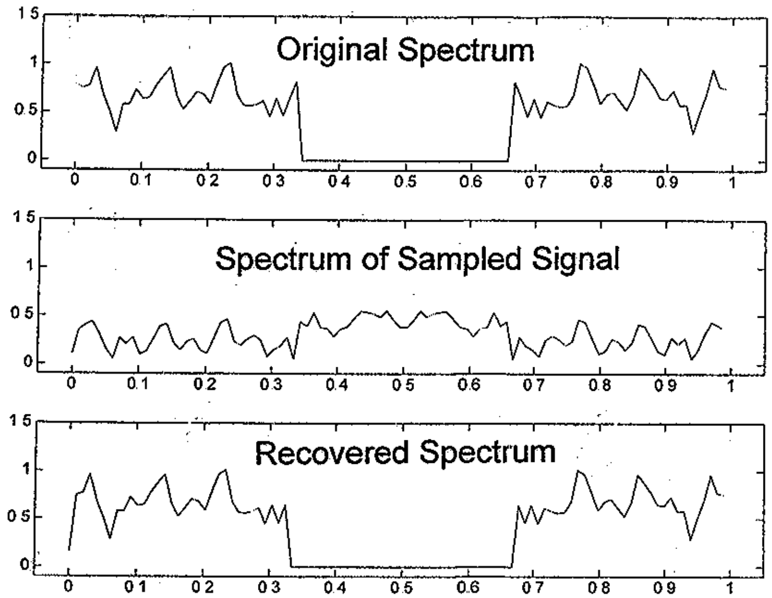


Fig. 5e. w=0.34
Here the bandwidth exceeds 1/3 (ever so slightly) and recovery by equation (2) fails. This is most evident right around 1/3. Note that the spectrum is still completely repaired for a region centered at 1/6. This does not mean that any portion of the time-domain sequence is recovered without error.

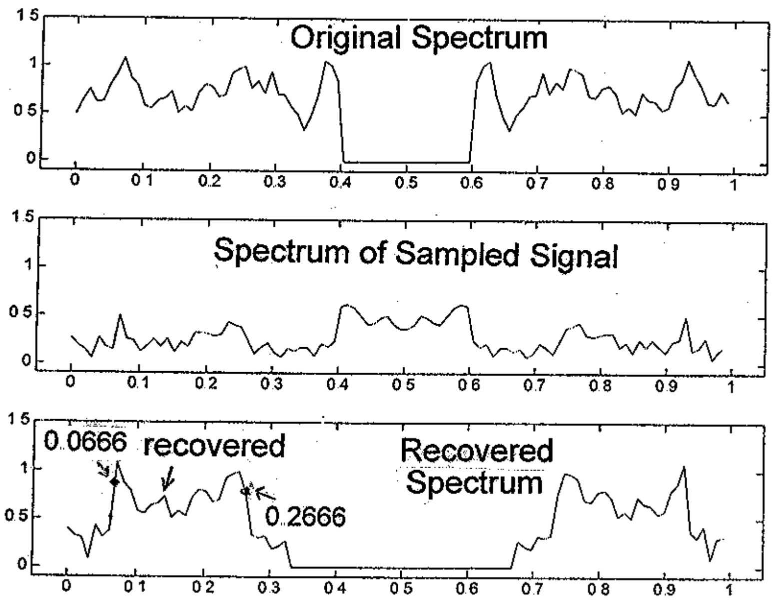


Fig. 5f. w=0.4
At this point, the bandwidth exceeds 1/3 by an amount of 1/15. Equation (2) does not recover the entire spectrum. There are regions of width 2/15= 0.13333 centered about 1/3 and about 0 that are corrupted. The region of width 0.2 centered about 1/6 is recovered without error. Again, each time domain value will have some error.

3. A SECOND EXAMPLE - PLAYING WITH LENGTH 4

3a. SAMCELL = [1 1 0 0], Toward a General Approach

It will be useful to look at a second example which we have previously studied by using a different approach that was possible in this case [2]. This is the case of keeping two samples and discarding two. The average sampling rate is thus 1/2 and we should be able to support a bandwidth of 1/4. This example also illustrates the summation approach rather than the subtraction approach (which works best when only one sample is discarded in any SAMCELL). The SAMCELL for this case is s=[1 1 0 0].

Here, even though we start with a real spectrum, the sampled spectrum becomes complex. We will form the sum of samples ...., -4, 0, 4, 8, . . and ..... ,-3, 1, 5, 9, ... Fig. 6 shows the situation for a triangular shape (again - just used for easy sketching). Here the original spectrum, Fig. 6a, has four segments {ABCD}, and we know that B=0 and C=0. Fig. 6b and Fig. 6c show the spectra of the two subsequences. Fig. 6b for the samples .... , -4, 0, 4, 8, ... corresponds to the SAMCELL s=[1 0 0 0] component of the superposition and is just Fig. 6a multiplied by 1/4 and replicated about frequencies 1/4, 1/2, and 3/4. Fig. 6c shows the spectrum of samples .... , -3, 1, 5, 9, ... corresponding to the SAMCELL s=[0 1 0 0] component, and is complex (due to the shift of one sample). (The magnitudes are the same for both subsequences.) Fig. 6d is intended to be the sum of Fig. 6b and Fig. 6c, which is difficult to draw [2], so we just show it as a squiggly line segment {abcd}. Note that it was the cancellation of the replicas centered at 1/2 that allowed us to recover the spectrum with filtering and shifts [2].

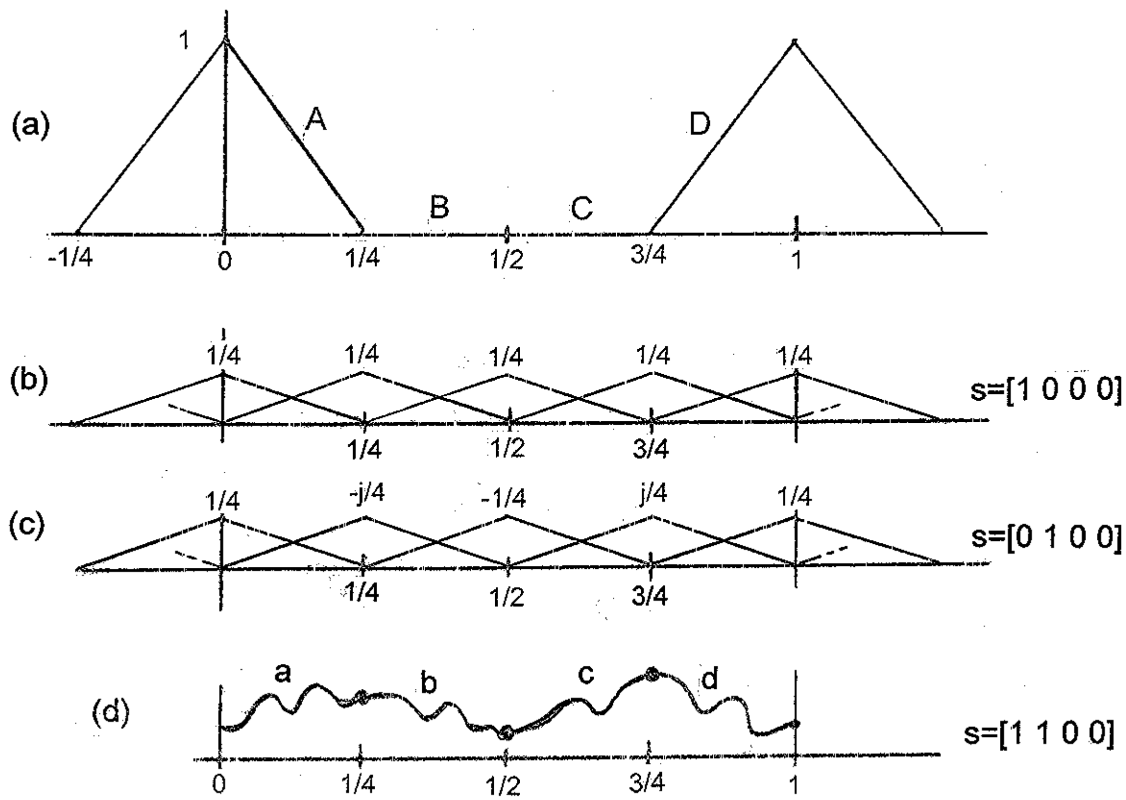


Fig. 6 Sampling 2 of 4: SAMCELL s=[1 1 0 0]

Using the current method we again start with the equations that give the superposition leading to Fig. 6d.

$$a = A/4 + D/4 + A/4 - jD/4 = A/2 + [\,(1-j)/4\,]D \tag{3a}$$

$$b = A/4 + D/4 - jA/4 - D/4 = [\,(1-j)/4\,]A \tag{3b}$$

$$c = A/4 + D/4 - A/4 + jD/4 = [\,(1+j)/4\,]D \tag{3c}$$

$$d = D/4 + A/4 + D/4 + jA/4 = D/2 + [\,(1+j)/4\,]A \tag{3d}$$

or in matrix form

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1/2 & (1-j)/4 \\ (1-j)/4 & 0 \\ 0 & (1+j)/4 \\ (1+j)/4 & 1/2 \end{bmatrix} \begin{bmatrix} A \\ D \end{bmatrix} \tag{4}$$

This has the pseudo-inverse solution:

$$\begin{bmatrix} A \\ D \end{bmatrix} = \begin{bmatrix} 1 & 1+j & j & 0 \\ 0 & -j & 1-j & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{5}$$

Since we started out with a real spectrum {ABCD} we need to see how this matrix multiply gets rid of imaginary parts. Thus in verifying the recovery, using equation (5) and Fig. 6, we calculate:

$$\begin{aligned} A = {} & A/4 + D/4 + A/4 - jD/4 \\ & + D/4 + A/4 - jA/4 - D/4 + jD/4 + jA/4 + A/4 - jD/4 \\ & + jA/4 + jD/4 - jA/4 - D/4 = A \end{aligned} \tag{6a}$$

$$\begin{aligned} D = {} & -jA/4 - jD/4 - A/4 + jD/4 \\ & + A/4 + D/4 - A/4 + jD/4 - jA/4 - jD/4 + jA/4 + D/4 \\ & + A/4 + D/4 + jA/4 + D/4 = D \end{aligned} \tag{6b}$$

3b. A Simple Case Going Back to Uniform

Clearly, if the SAMCELL is s=[1 0 1 0] we have the case of sampling at a rate of 1/2 and a bandwidth of 1/4 can be supported: SPECCELL W=[1 1 0 0]. Even the easiest cases should work!

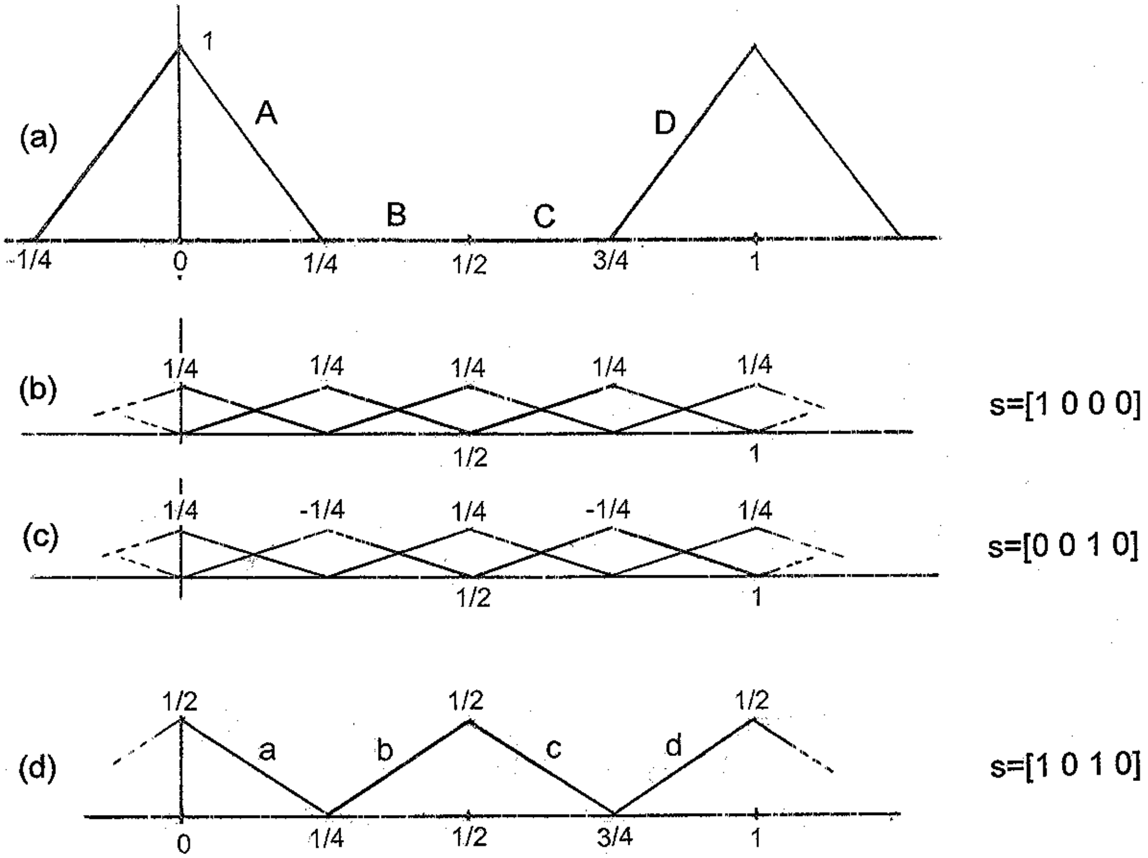


Fig. 7 SAMCELL s=[1 0 1 0] reverts to uniform downsampling

Fig. 7 shows the sampling situation much as we have done it before. We easily write down the equations in matrix form:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \\ 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} A \\ D \end{bmatrix} \tag{7}$$

which has the simple pseudo-inverse solution

$$\begin{bmatrix} A \\ D \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{8}$$

This is just:

$$A = a + c \tag{9a}$$

$$D = b + d \tag{9b}$$

For this example, we see that again the pseudo-inverse gives us a perfectly correct answer. What does not come out here is a slightly simpler answer, evident from Fig. 7, that A=2a, since a=c. This is really equivalent to A=A, which would come out in the case of SAMCELL s=[1 1 1 1] in which case the superposition matrix and the recovery matrix are both 4x4 identity matrices. While we would need an ideal low-pass filter for this simple solution, the s=[1 0 1 0] case is still ordinary sampling (or re-sampling if you prefer). So perhaps the <u>simplest</u> answer does not come out. Or does it......?

Fig. 8 is offered to address this question, and to remind us of what an actual recovery would involve in terms of ideal filters and spectral shifts. In Fig. 8a, the superposition spectrum is separated into its four segments a, b, c, and d by ideal filters. In getting segment A back we need to add segment a to a shifted version of segment c - hence the multiplication of c by a sequence $(-1)^n$. Likewise D is obtained by adding d to a shifted version of b. Adding A to D recovers the original spectrum. Fig. 8b and Fig. 8c show simplifications of Fig. 8a down to a single filter with gain 2. (Of course the filter from 3/4 to 1 is the same as a filter from -1/4 to 0.) It is thus possible to see that the actual recovery reduces to the traditional approach for this case.

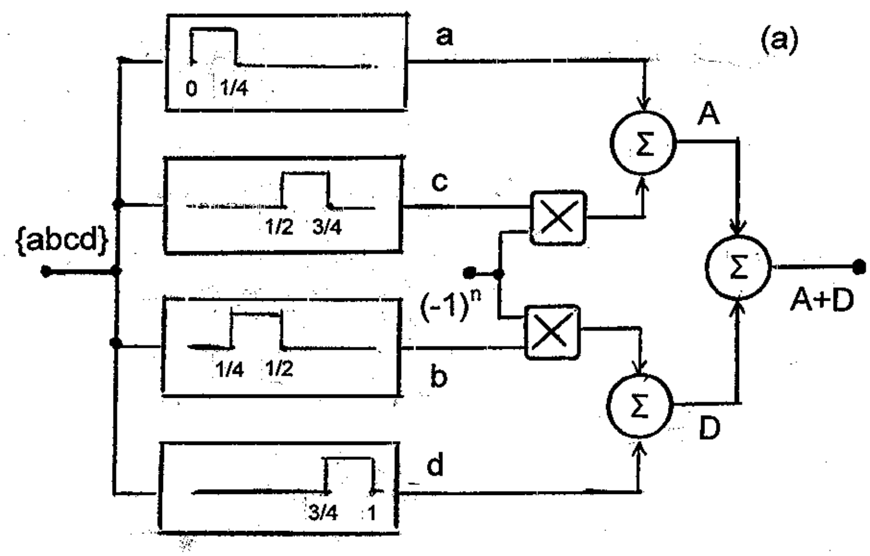
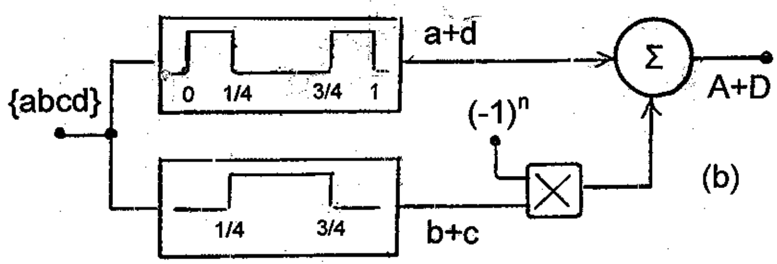
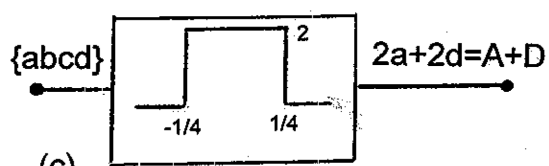


Fig. 8. Fig. 8a corresponds to equation (8). This reduces to Fig. 8b and Fig. 8c, which suggests the traditional low-pass route to recovery.





4. KEEPING THREE OF FOUR AND THE USE OF A SPECTRAL SHIFT

Our previous example [1] kept three of four samples, and we divided the frequency range into eight segments to do it. Such a division seems convenient for the general case, but here we can redo the problem using only four segments. Further, we need to keeps such alternatives in mind as they may be useful in an actual implementation.

Keeping three of four samples should support a bandwidth of (1/2)(3/4)=3/8. Since we are discarding only one sample, we can use our subtraction method (as we did with the s=[0 1 1] SAMCELL) by using the s=[0 1 1 1] SAMCELL. Fig. 9a shows the generation of the superimposed spectrum {abcd} from the original {ABCD}. Note that here we have centered the segments in a different way, so that A is centered about 0. Also, note that we have chosen C=0, centered about 1/2, so that w=3/8.

Here it is easy to obtain the superposition equations much as we have done before:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3/4 & -1/4 & -1/4 \\ -1/4 & 3/4 & -1/4 \\ -1/4 & -1/4 & -1/4 \\ -1/4 & -1/4 & 3/4 \end{bmatrix} \begin{bmatrix} A \\ B \\ D \end{bmatrix} \qquad (10)$$

and the recovery equations using pseudo-inverse are:

$$\begin{bmatrix} A \\ B \\ D \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad (11)$$

Fig. 9b and Fig. 9c show exactly how these segments are separated from the sampled spectrum and recombined according to equation (11).

If this had been done with eight segments, the matrices would have been twice the size [1]. Each matrix element will appear in diagonal positions, and half the matrix will be zeros. We will need to recognize this pattern as implying that the solution is actually smaller than the matrix size appears. Keeping extra segments may well permit us to use one general program for a wider variety of cases.

5. THE GENERAL CASE OF A LENGTH FOUR SAMCELL

We have seen above that we could solve length four SAMCELL recovery problems by using four segments in frequency. For the case of Section 4 however, this involved a shifting of the segment boundaries. That is, there is a difference

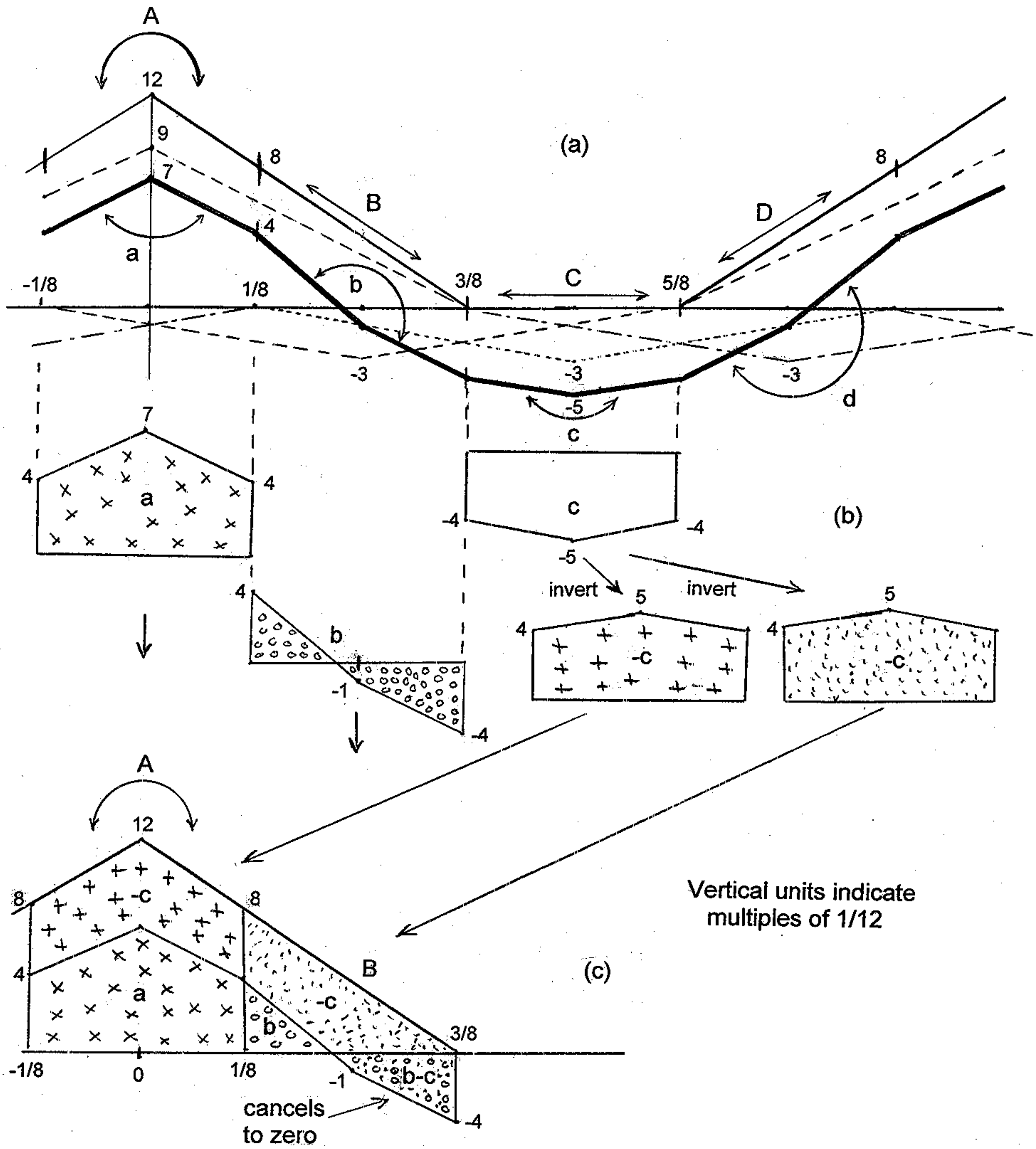


Fig. 9 Here a bandwidth of w=3/8, W=[1 1 1 0] is sampled by s=[0 1 1 1] so that segments {ABCD} are replaced by segments {abcd}, as shown in Part (a) of the figure. In Part (a), the replicas about 0, 1/4, 1/2, and 3/4 are shown as various dashed and dotted lines. Equation (11) indicates a reconstruction of the original spectrum from the pieces indicated in Part (b) of the figure. Thus we have A = a - c and B = b - c, forming Part (c) of the figure. Segment C is zero by assumption, and segment D is d-c (not shown).

between problems with 2 of 4 kept and problems with 3 of 4 kept. In general, it will make a difference whether we keep an even number or an odd number of samples in our SAMCELL. To avoid this, and to simplify the writing of a general program, we can choose twice as many frequency segments. For a length 4 SAMCELL, this means we will have eight frequency segments of width 1/8, and so on.

Fig. 10 shows such a spectrum with eight segments {ABCDEFGH} of which pairs that are symmetric about 1/2 may be zero. For example, for the problem of Section 4 we choose segments D and E to be zero, and this is exactly what we mean by a SPECCELL of W=[1 1 1 0]. While Fig. 10 shows all of {ABCDEFGH} drawn as non-zero, our interesting (solvable) problems will always have segments that are zero (reducing the spectral size and thus allowing samples to be skipped). Note that the zero segments need not be D and E, but might be C and F, B and G, or A and H: W=[1 1 0 1], W=[1 0 1 1], and W=[0 1 1 1] respectively.

Fig. 11a, Fig. 11b, Fig. 11c, and Fig. 11d show the spectral components corresponding to the four possible sets of samples, s=[1 0 0 0], s=[0 1 0 0], s=[0 0 1 0], s=[0 0 0 1]. The spectrum of the sampled signal is the sum of these four, weighted according to the particular SAMCELL. For example, if s=[1 1 0 0] then we add Fig. 11a and Fig. 11b. The weights on the spectral replicas as in Fig. 11, are the values of a length 4 DFT matrix:

$$D_4 = (1/4)e^{-j(2\pi/4)\,nk} = (1/4)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \qquad (12)$$

where n and k are indices running from 0 to 3, right to left, and top to bottom, and deployed as seen in Fig. 11a through 11d. This weighting is just a fundamental observation from the sampling of a discrete time signal.

Looking at Fig. 11, top to bottom, segment by segment, we see that each segment of {abcdefgh} is the sum of 16 possible pieces - four overlaps of each of the four possible phases of sample spacing 4. For example, with s = [s(1) s(2) s(3) s(4)] :

$$\begin{aligned} a = {} & s(1)[A/4 + G/4 + C/4 + E/4] \\ & + s(2)[A/4 - jG/4 + jC/4 - E/4] \\ & + s(3)[A/4 - G/4 - C/4 + E/4] \\ & + s(4)[A/4 + jG/4 - jC/4 - E/4] \end{aligned} \qquad (13a)$$

$$\begin{aligned} = {} & (1/4)[\, s(1) + s(2) + s(3) + s(4) \,]\, A \\ & + (1/4)[\, s(1) - js(2) - s(3) + js(4) \,]\, G \\ & + (1/4)[\, s(1) + js(2) - s(3) - js(4) \,]\, C \\ & + (1/4)[\, s(1) - s(2) + s(3) - s(4) \,]\, E \end{aligned} \qquad (13b)$$

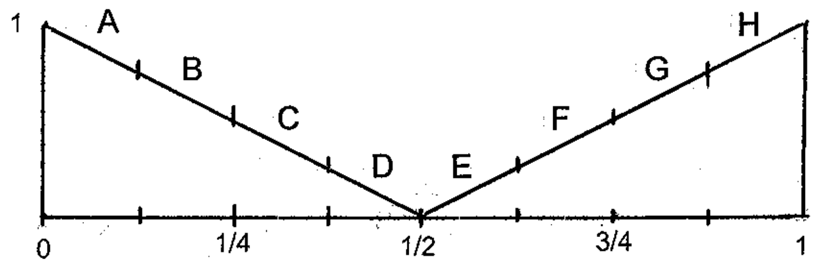


Fig. 10 An original triangular spectrum for w as much as 1/2

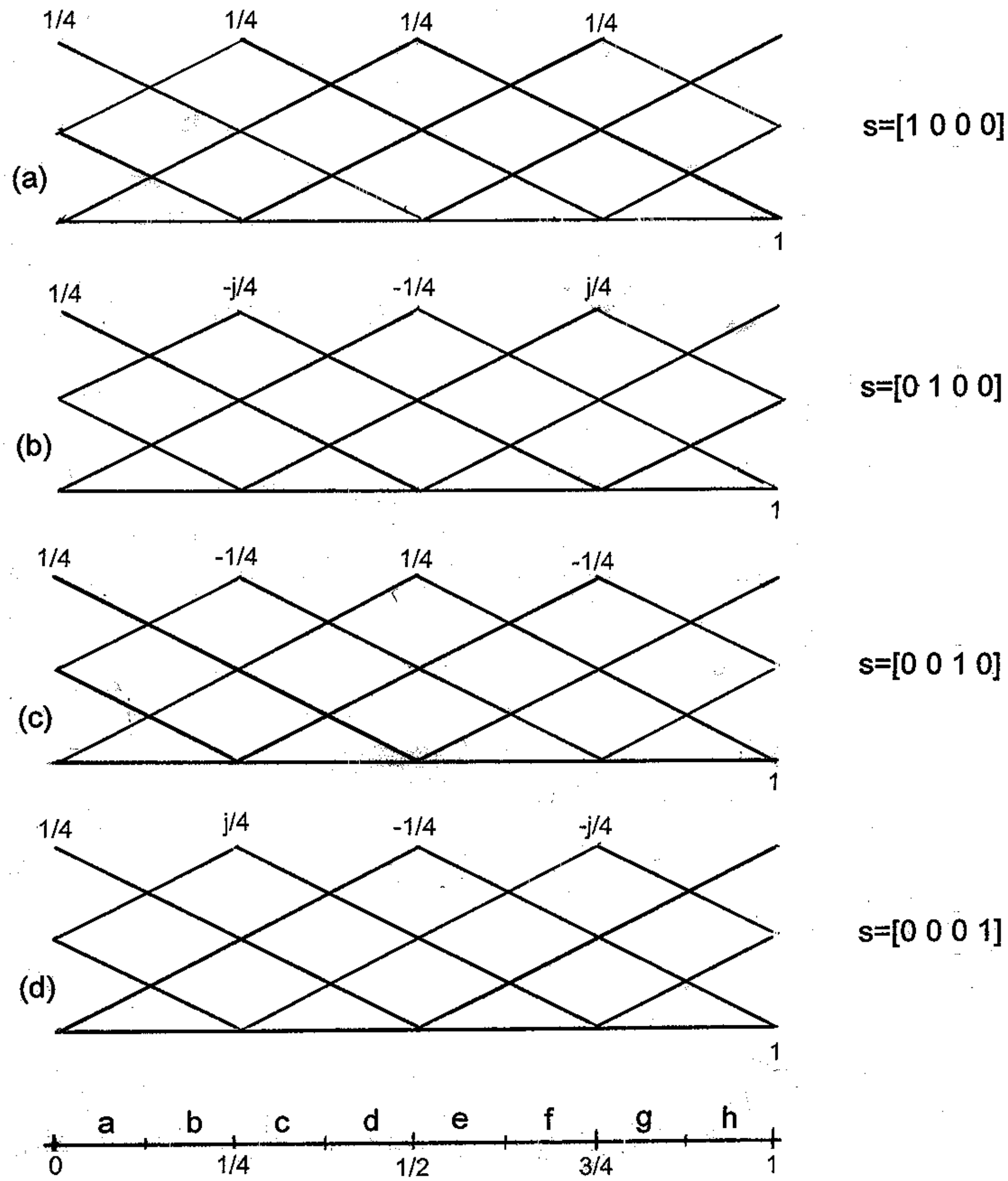


Fig. 11 Four different phases of sampling by 4

This describes the segment a in terms of segments A, G, C, and E. Note that we need to consider in general all cases when s(1), s(2), s(3) and s(4), the elements of SAMCELL s, may be 1 or 0. If all of s(1), s(2), s(3) and s(4) are 1, then a=A as it must.

In setting up a general 8x8 superposition matrix M that relates the segments {abcdefgh} to the segments {ABCDEFGH} we would have eight equations similar to equation (13b). There are only four different values for the matrix elements, obtained from D_4 (the DFT matrix) and from s as:

$$\begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ r(4) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} s(1) \\ s(2) \\ s(3) \\ s(4) \end{bmatrix} \tag{14}$$

These end up populating the matrix as:

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \underbrace{\begin{bmatrix} r(1) & 0 & r(4) & 0 & r(3) & 0 & r(2) & 0 \\ 0 & r(1) & 0 & r(4) & 0 & r(3) & 0 & r(2) \\ r(2) & 0 & r(1) & 0 & r(4) & 0 & r(3) & 0 \\ 0 & r(2) & 0 & r(1) & 0 & r(4) & 0 & r(3) \\ r(3) & 0 & r(2) & 0 & r(1) & 0 & r(4) & 0 \\ 0 & r(3) & 0 & r(2) & 0 & r(1) & 0 & r(4) \\ r(4) & 0 & r(3) & 0 & r(2) & 0 & r(1) & 0 \\ 0 & r(4) & 0 & r(3) & 0 & r(2) & 0 & r(1) \end{bmatrix}}_{M} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{bmatrix} \tag{15}$$

At this point, it is useful to reflect that if s=[1 1 1 1] then r(1)=1 while r(2), r(3), and r(4) are zero, and the matrix is just the identity matrix as it should be.

Of course in general s is not [1 1 1 1] and all four of the r values are non zero. In the cases where s has one or two values that are zero, two or four of original spectra segments must be zero to reflect the reduction of available bandwidth. This is the final key to finishing the problem. In the case where D and E are zero, we remove the center two rows of M, the ones on the same level as D and E. If other pairs are zero, we remove those rows from M. If SPECCELL is W=[1 0 1 0] for example, we would remove rows 2, 4, 5, and 7, corresponding to segments B, D, E, and G. This would means that we could expect recovery if the SAMCELL has one or two zeros.

The final step is simply to obtain the recovery matrix as the pseudo-inverse of the reduced M.

6. PROGRAMS USED

While it is often not necessary to present program listings that are used to generate calculations and figures in a report, it is also true that sometimes the code offers a useful, unambiguous, and definitive statement of what is actually being done. Such seems to be indicated here where the programs are more definitive than the equations.

6a. A Note on Bandlimiting

A condition on our procedures here is that the signal to be sampled is bandlimited in a well-defined way. It is also the case that our transformation between time and frequency domains pretty much has to be through the use of the DFT (and its inverse). When we think about obtaining bandlimited signals, we may tend first to think in terms of sinusoidal waveforms and sinc functions in the time domain. In Program 1 below, however, we are not really concerned with the time-domain waveform, except as we want to sample it in a particular way. Accordingly we find it convenient to just select values for the DFT and set the DFT to zero for regions outside the bandwidth we have selected for our test. In this case, the time-domain waveform is actually a finite sum of sinusoidal waveforms all of which are integer multiples of some fundamental ("DFT Harmonics").

In Program 3 we do want to look at the actual time-domain waveform - to see it before and after sampling, and after recovery. For example, if we want a waveform bandlimited to 1/4, it is natural to choose something like a cosine at a frequency of 0.2. Clearly this is bandlimited, but does the DFT, our only available spectrum analyzer, think it is bandlimited? Of course, this all depends on the length of the DFT. If we do not choose the length correctly, even a sequence that "is" bandlimited can and does, properly, come back without bandlimiting. In our example, if we do choose the frequency 0.2, then there are 5 samples per cycle in our sequence. Any DFT length that is not a multiple of 5 will give us spectral "leakage." So why not use length 5 if we know it is right?

Well, we also need to consider the remainder of our study. We are going to sample by a SAMCELL that is length 4. Thus since we are looking at DFT's after this sampling, we will need to choose a DFT that is a integer multiple of 4. To meet both conditions, we choose length 20. If this seems too contrived, we can just take a very long FFT of many many cycles and accept some minor recovery errors.

6b. Program 1 - twoof3.m

This program, twoof3.m, was used to generate Fig. 5a through Fig. 5f, the case where two or three samples are kept (one of three is replaced by 0), for a variety of bandwidths.

```
% twoof3.m
function twoof3(B)
%
%  Sampling Two of Three Samples
%     for Different Bandwidths
%
%  B. Hutchins       Sept. 2003
%  (For Fig. 5)

B=round(100*B);

% Random Complex, but Bandlimited Spectrum X1(k)
X1(1)=rand+j*rand;
for k=1:B
  X1(k+1)=rand + j*rand;
end
% Smoothen for better display example X(k)
for k=2:B-1
  X(k)=(X1(k-1)+X1(k)+X1(k+1))/3;
end
X(1)=X1(1);
X(B)=X1(B);
% Symmetric Upper Side
for k=1:B
  X(99-k+1)=X(k+1);
end
% FFT is Length 99

figure(1)
subplot(311)
plot([0:1/99:.99],abs(X))
axis([-.05 1.05 -.1 1.5]);


% Sampling Sequence
s=[0 1 1 0 1 1 0 1 1];
s=[s s s s s s s s s s s];
% Sequence
x=ifft(X);
% Sampled Sequence
xs=x.*s;
% Spectrum of Sampled Sequence
XS=fft(xs);

subplot(312)
plot([0:1/99:.99],abs(XS))
axis([-.05 1.05 -.1 1.5]);
```

```
% Repair the FFT
for k=1:33              % Segment A from a and b
XR(k)=XS(k)-XS(k+33);
end
for k=34:66             % Segment B
XR(k)=0;
end
for k=67:99             % Segment C from b and c
XR(k)=-XS(k-33)+XS(k);
end

subplot(313)
plot([0:1/99:.99],abs(XR))
axis([-.05 1.05 -.1 1.5]);

figure(1)
```

6c. Program 2 - pinvs.m

This program calculates the matrix M that describes the spectral superpositions and the pseudo-inverse matrix p that describes the repair procedure. This follows a general case of which the length 4 steps are given in Section 5. The input parameters are the SAMCELL and the SPECCELL as described. Because we need to be able to handle even and odd lengths for the non-zero elements of a SAMCELL, in many cases the matrices M and p are twice as large (in each dimension) as seems necessary (see comments in Section 4). In such cases, the actual matrix elements are repeated twice, along the diagonal, for each 2x2 submatrix of the results.

```
function [M,p]=pinvs(samcell,speccell)
% Find the matrix M that determines the sampled spectrum
%      in terms of the original spectrum. Find the matrix p that
%      inverts this, based on the sampling cell and the bandwidth.
%  samcell is the basic sampling cell.  For example, if we
%    are keeping three samples of four, samcell is [1 1 1 0]
%    where the samples kept are at 0, 1, and 2.  L = length of
%    the samcell.
%  speccell is another vector the same length as samcell.
%    It divides the frequency range of 0 to 1/2 into L segments
%    For example, keeping three of four samples allows us to
%    support a bandwidth of (1/2)*(3/4) = 3/8 so the speccell
%    could be [1 1 1 0].
%  B. Hutchins                                               Oct. 2003
```

```
s=samcell;
W=speccell;
L=length(s);

% DFT Matrix
n=0:L-1;
k=0:L-1;
nk=n'*k;
nk=(-j*2*pi/L)*nk;
m1=exp(nk);

for k=1:L
   r(k)=(1/L)*sum(s*m1(:,k));
end
r;
r=[r(1) r(L:-1:2)];
LL=2*L;
mm=zeros(LL,LL);

% for convenience, make three copies, populate matrix, and then
% keep the center one.
mm=[mm,mm,mm];
ri=0;
for k=0:2:2*LL
      ri=ri+1;
      for n=1:LL
        mm(n,k+n)=r(ri);
      end
      if ri==L;ri=0;end
end
for k=1:LL
      for n=1:LL
       m(n,k)=mm(n,k+LL);
      end
end
m;

% now remove rows according to bandwidth
W=[W fliplr(W)];

mm=[];
for n=1:LL
  if W(n)==1;
     mm=[mm; m(:,n).'];
  end
end
M=mm;
```

```
% simplify the output display
sz=size(m);
for k=1:sz(1)
   for n=1:sz(2)
      if abs(real(m(k,n)))<0.0001;m(k,n)=j*imag(m(k,n));end
      if abs(imag(m(k,n)))<0.0001;m(k,n)=real(m(k,n));end
   end
end

% Use Matlab psseudo-inverse to avoid possible numerical
%                  problens using matricies
p=pinv(m).';

% simplify the output display
sz=size(p);
for k=1:sz(1)
   for n=1:sz(2)
      if abs(real(p(k,n)))<0.0001;p(k,n)=j*imag(p(k,n));end
      if abs(imag(p(k,n)))<0.0001;p(k,n)=real(p(k,n));end
   end
end

m=m.';
```

6d. Program 3 - tdr.m

This program begins with twenty samples, four cycles of five samples each, of a cosine of frequency 1/5. (See notes on bandlimiting is Section 6a above). The results will be taken up in Section 7c.

```
% Time-Domain Recovery of 2 of 4 Samples
%  tdr.m
%
% B. Hutchins                         Oct. 2003


% Generate Original Signal and Look at its FFT
n=0:19;
x=cos(2*pi*0.2*n);
figure(1)
subplot(511)
stem([0:19],x)
title('original time-domain signal')
axis([-2 22 -1.2 1.3]);
X=fft(x);
subplot(513)
```

```
stem([0:.05:.95],real(X))
axis([-.1 1.1 -14 14]);
title('real part of spectrum')
subplot(515)
stem([0:.05:.95],imag(X))
axis([-.1 1.1 -12 12]);
title('imag part of spectrum')

% Now Sample With s=[1 1 0 0] and Look at FFT
s=[1 1 0 0];
s=[s s s s s];
xs=x.*s;
figure(2)
subplot(511)
stem([0:19],xs)
title('sampled time-domain signal')
axis([-2 22 -1.2 1.3]);
XS=fft(xs);
subplot(513)
stem([0:.05:.95],real(XS))
axis([-.1 1.1 -8 8]);
title('real part of spectrum')
subplot(515)
stem([0:.05:.95],imag(XS))
axis([-.1 1.1 -8 8]);
title('imag part of spectrum')

% Recover Spectrum using p=[1  1+j j 0; 0 -j 1-j 1]
XR=zeros(1,20);
for k=1:5
   XR(k)=XS(k)+(1+j)*XS(k+5)+j*XS(k+10);
   XR(k+15)=-j*XS(k+5)+(1-j)*XS(k+10)+XS(k+15);
end
xr=ifft(XR);
xr=real(xr);
figure(3)
subplot(511)
stem([0:19],x)
axis([-2 22 -1.2 1.3]);
title('original time-domain signal')
subplot(513)
stem([0:19],xr)
axis([-2 22 -1.2 1.3]);
title('recovered time-domain signal')
subplot(515)
stem([0:19],(xr-x));
axis([-2 22 -1.2 1.3]);
title('error in time-domain')
```

7. A COUPLE MORE NOTES AND EXAMPLES

Here we catalog the sampling cases we have looked at above.

s	W	Where
s=[0 1 1]	W=[1 1 0]	Section 2b
s=[0 1 1]	W=[1 0 0], W=[1 1 0]	Section 2c
s=[1 1 0 0]	W=[1 1 0 0]	Section 3a, Reference [2]
s=[1 0 1 0]	W=[1 1 0 0]	Section 3b
s=[0 1 1 1]	W=[1 1 1 0]	Section 4, Reference [1]

Some extensions and additional examples are offered in this section.

7a. Relation to Downsampling

We saw in Section 2b that we could support a bandwidth of w=1/3, W=[1 1 0] with s=[0 1 1], in which case the spectral recovery equations were:

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{16}$$

Or we could have chosen s=[1 1 0] in which case the recovery equations would become:

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} 1 & 0.5+0.866j & 0 \\ 0 & 0.5-0.866j & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{17}$$

and so on. This bandwidth of w=1/3 is shown in Fig. 12a, and we note that there are three natural segments A, B, and C. Fig. 12b shows the case of w=1/6, W=[1 0 0], in which case we have six natural segments. It is possible of course to represent w=1/3, W=[1 1 0] as six segments as seen in Fig. 12c. Of course a bandwidth of 1/6 can be regarded as a bandwidth of 1/3 for which the portion from 1/6 to 1/3 happens to be zero throughout its length. If we solve the problem with w=1/3, W=[1 1 0], with six segments, the recovery equations with s=[0 1 1] become:

$$\begin{bmatrix} A \\ B \\ E \\ F \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \tag{18}$$

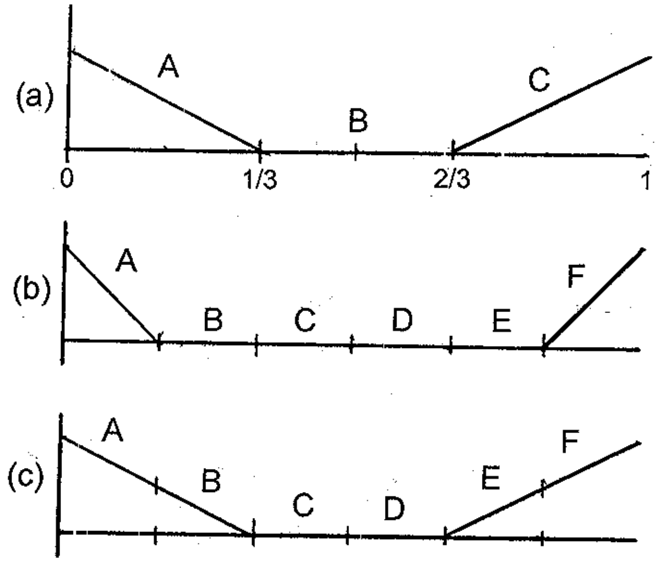


Fig. 12 Choosing different numbers of segments for different bandwidths.

which easily reverts to equation (16) if we recombine the segments from six back to just three. In fact, it is this expanded version that it's the default output in Program 2.

When the bandwidth is truly 1/6, we could of course still use equation (18). But if we do put in W=[1 0 0] instead of W=[1 1 0], we get the recovery equations:

$$\begin{bmatrix} A \\ F \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5 & 0 & -0.5 & 0 \\ 0 & -0.5 & 0 & -0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \quad (19)$$

Can both of these be right? Fig. 13 shows the case of s=[0 1 1] for W=[1 0 0].

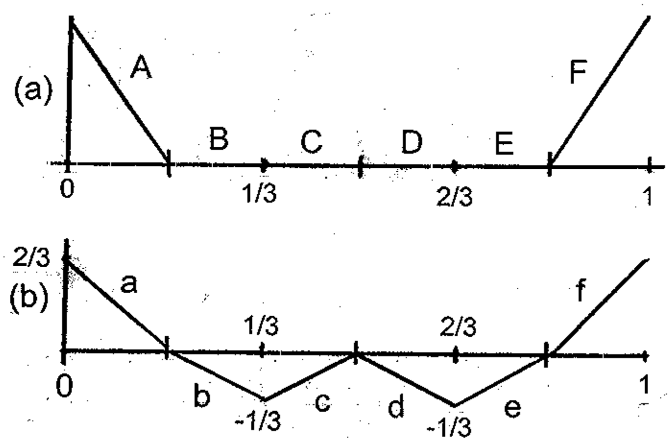


Fig. 13 Sampling a bandwidth of 1/6 (W=[1 0 0]) for case s=[0 1 1].

We see that equations (18) and (19) work. For example, using equation (19) we have:

$$A = a - (1/2)c - (1/2)e \tag{20a}$$

While equation (18) gives us:

$$A = a - c \tag{20b}$$

and both these are right as can be seen by inspection. Note that with equation (19), segment B is zero by assumption, while using equation (18) we have:

$$B = b - d \tag{20c}$$

which is clearly zero, by inspection. And so on.

In this case, where the bandwidth is 1/6, we do not need two of three elements of the SAMCELL to be ones – we can use s=[1 0 0] for example. Fig. 14 shows this sampling – note that we really do need to use six segments here.

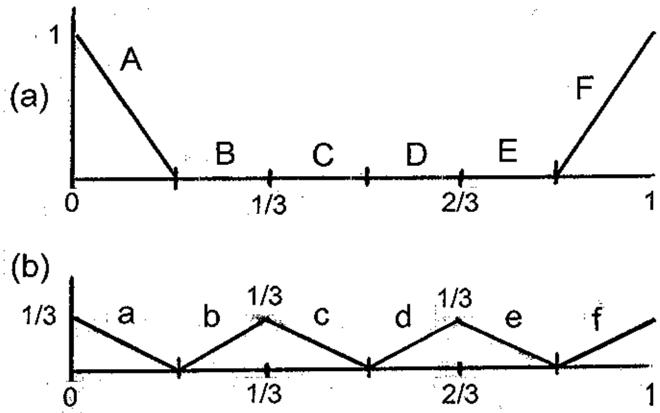


Fig. 14 Sampling a bandwidth of 1/6 (W=[1 0 0]) for case s=[1 0 0] reverts to downsampling by 3.

For s=[1 0 0] and W=[1 0 0] the recovery equations become:

$$\begin{bmatrix} A \\ F \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \tag{21}$$

which is correct by inspection.

To complete this line of study, it is useful to be sure we know what happens when all the samples are kept. That is, for s=[1 1 1] we could start with the assumption that we have a full bandwidth w=1/2, W=[1 1 1], in which case, the recovery matrix is just:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (22a)$$

which is just an identity matrix as it should be. Of course, this should also work if the bandwidth is w=1/3, W=[1 1 0], in which case the recovery matrix is:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (22b)$$

Finally, we can also support the even smaller bandwidth w=1/6, W=[1 0 0]; which gives the simplest recovery matrix:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (22c)$$

7b. Bandpass Sampling

We suggested above that the bandwidth that is supported in our procedures need not be a continuous segment, and this we need to show by an example. Fig. 15 shows the case of W=[1 0 1] with s=[0 1 1]. The recovery equations for such a case are:

$$\begin{bmatrix} A \\ C \\ D \\ F \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \qquad (23)$$

That this works is seen from Fig. 15. We note from equation (23) that:

$$A = a - e \qquad (24a)$$

and

$$C = c - e \qquad (24b)$$

which are correct by inspection.

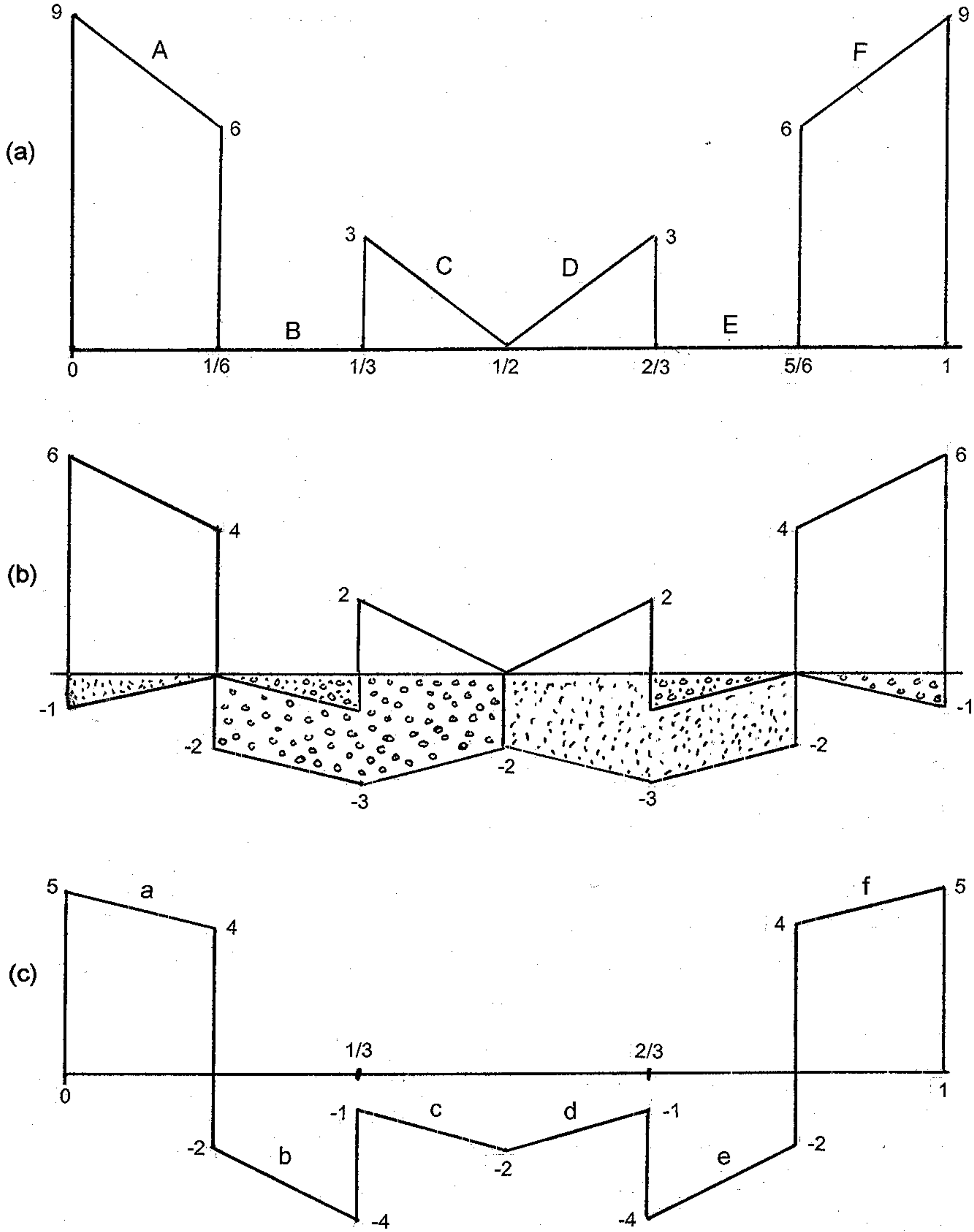


Fig. 15 Bandpass sampling example. Here in (a) the bandwidth from 0 to 1/2 is comprised of two non-continuous segments of total width 1/3. Sampling with s=[0 1 1] gives three overlapping images (b) which sum to (c). Vertical units are multiples of 1/9.

7c. Time-Domain Recovery

Most of what we have done in this note involves considering how the spectra of sampled signals are composed of the superposition of component spectra of different sampling sub-sequences. We then take segments of the superimposed spectrum and reassemble the pieces in an appropriate way. This is easy to do on paper where extracting a particular segment is a matter of redrawing the segment with vertical sides - the equivalent of using ideal filters. It is also easy to just shift these segments by redrawing them in different positions, adding them, etc. For example, see Fig. 9.

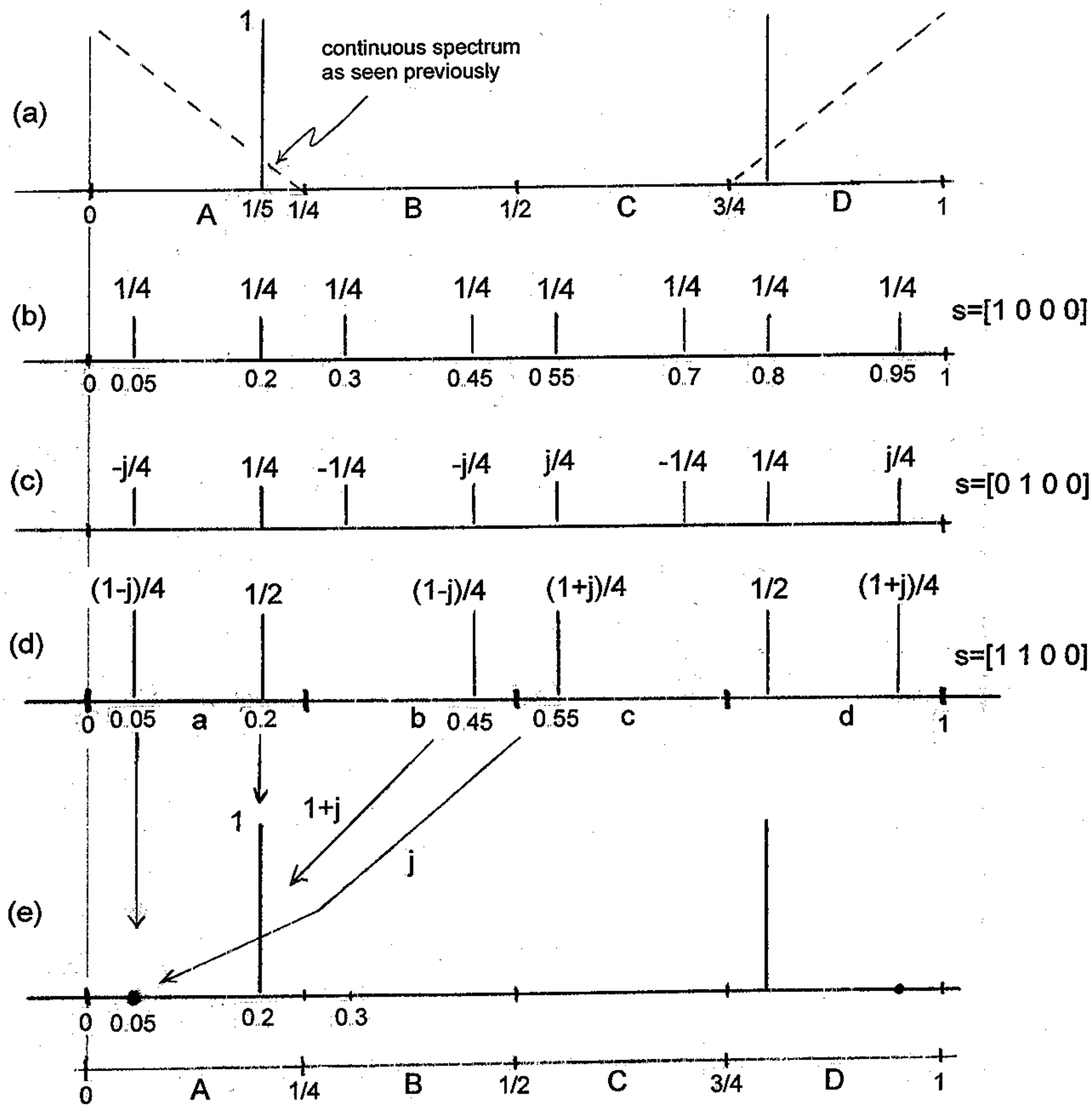


Fig. 16 A bandlimited signal (a cosine of frequency 0.2) is sampled with s=[1 1 0 0]. Here we have chosen W=[1 1 0 0]. The arrows between (d) and (e) indicate the recovery of segment A as A = a + (1+j)b + jc according to equation (5).

In order that we don't totally lose track of the fact that we are talking about time-domain sequences, in this example we will work with an actual bandlimited sequence, a cosine sequence in fact. This will permit two interesting things to happen. First, instead of drawing spectra that are triangles (or similar continuous shapes), the spectrum will become discrete. In addition, when we did choose triangular spectra, the

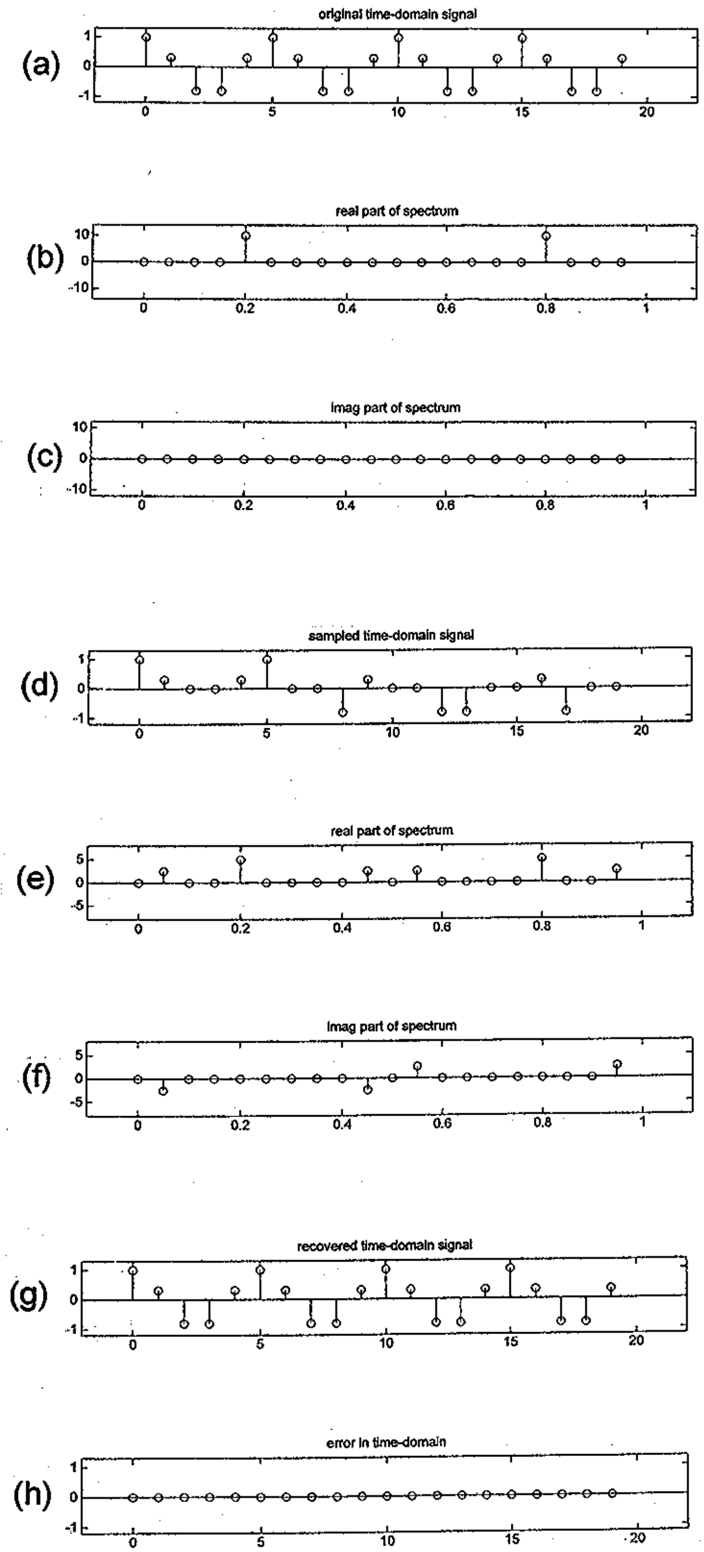


Fig. 17. A cosine of frequency 0.2 (a),(b), and (c) is sampled by keeping two samples and then discarding two samples, and so on (d), (e) and (f). Compare Fig. 17 (e) and (f) to Fig. 16 (d). By reconstructing the spectrum using equation (5) and inverting the resulting DFT, we obtain the time domain cosine (g) with no error (h).

corresponding time-domain sequences were very small except in a very localized position (they were squared sinc functions) and not too easy to plot and study. We never plotted these, and they would not have been convincing had we tried to argue reconstruction based on time-domain plots. Here we can show that we get cosine samples back - exactly.

Fig. 16 shows the sampling situation where s=[1 1 0 0] and the spectrum is a single cosine at the frequency 0.2. Thus we might think of W=[1 1 0 0] as being the appropriate general (continuous spectrum) case here since the frequency 0.2=1/5 is below 1/4.

This situation compares to the continuous spectrum assumed in Section 3a, and is shown dashed in Fig. 16a. The spectrum of the two sampling components, corresponding to s=[1 0 0 0] and s=[0 1 0 0] are shown in Fig. 16b and Fig. 16c, with the sum, the superposition spectrum {abcd} shown in Fig. 16d. Here the recovery equations are represented in equation (5) and Fig. 16e shows the recovered spectrum. We note the cancellation of the replicas at frequency 0.05, which otherwise was inside segment A. This is very similar to Section 3a, except here we have a discrete spectrum.

Fig. 17, generated by Program 3, shows the recovery in the time domain. Fig. 17e and Fig. 17f correspond to Fig. 16d. In Fig. 17g we see the recovered time-domain cosine. Note that this was recovered from Fig. 17d, without error (Fig. 17h).

We saw in Section 7b that a bandpass sampling example for our procedures given here was possible. In the case of a single frequency cosine, we can reduce the sampling rate virtually without limit because the spectrum has zero actual width (one discrete frequency). In fact the procedures in this note are useful for some general bandpass sampling problems (which are often set up and solved individually), even for the uniform sampling case as will be illustrated in Fig. 18 and Fig. 19.

Fig. 18 shows the case where we have divided the spectrum of the cosine into eight segments {ABCDEFGH} rather than just the four segments {ABCD} in Fig. 16. Now the cosine appears only in segments B and G (W = [0 1 0 0]), and we have less bandwidth to recover. Further, we can now use a SAMCELL s=[1 0 0 0], a downsampling by 4. Program 2 gives us the recovery equations:

$$\begin{bmatrix} B \\ G \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} \tag{25}$$

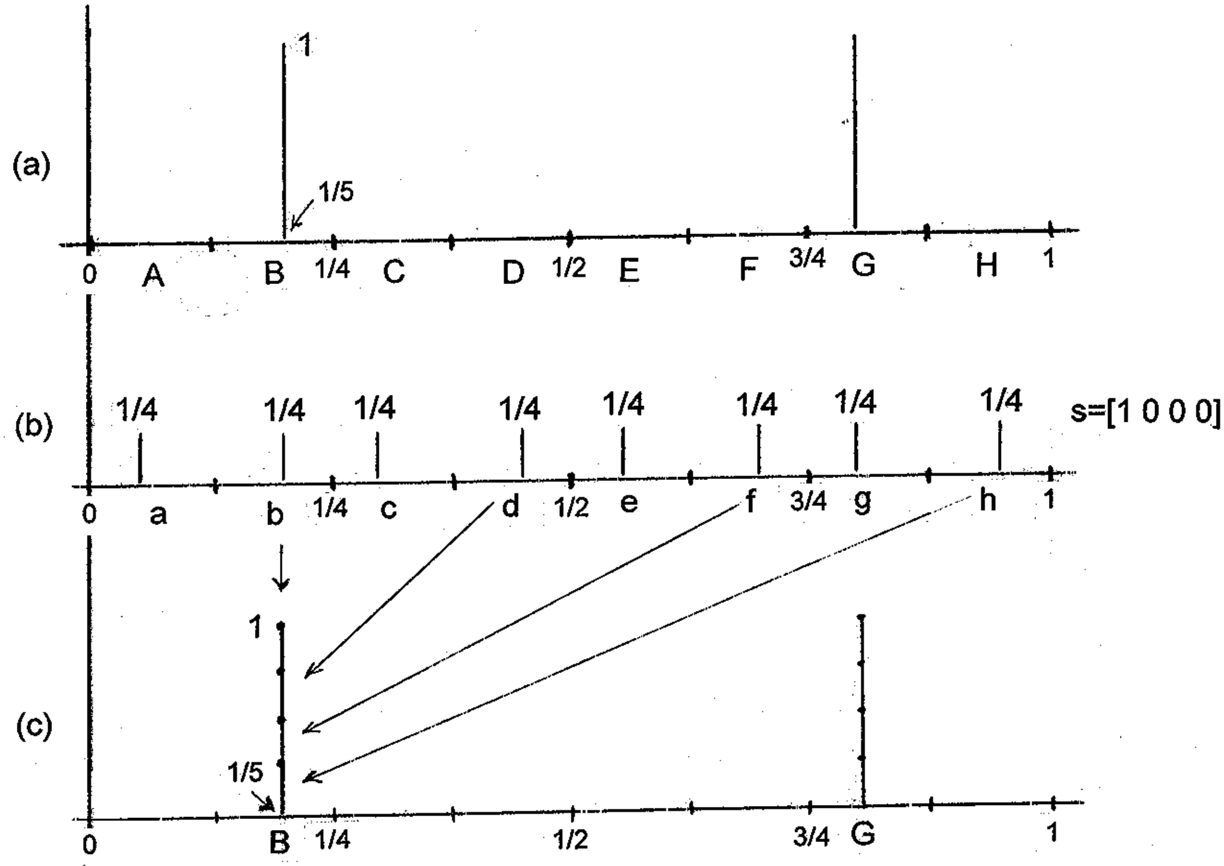


Fig. 18. Here we have a cosine at frequency 0.2 which resides only in segments B and G of (a). Sampled with s=[1 0 0 0] we get four replicas total with amplitudes of 1/4, being purely real, as seen in (b). Equation (22) indicates that segment B is obtained as B = b + d + f + h as seen in (c)

Fig. 18c shows the recovery of the segments B and G from the superposition spectrum of Fig. 18b. The arrows show the reconstruction of segment B from segments b, d, f, and h.

Fig. 19 shows results which correspond to Fig. 18 as obtained with a program similar to Program 3. This we can compare to Fig. 17 as well. Here we have eight cycles of a cosine of frequency 0.2 (Fig. 19a, 19b, and 19c). The time domain sequence in Fig. 19a is sampled with s=[1 0 0 0] to form Fig. 19d. (Here the spectrum is purely real as seen in Fig. 19e and Fig. 19f.) Fig. 19g does show complete recovery of the time domain cosine, from the sampled cosine of Fig. 19 d, without error (Fig. 19h).

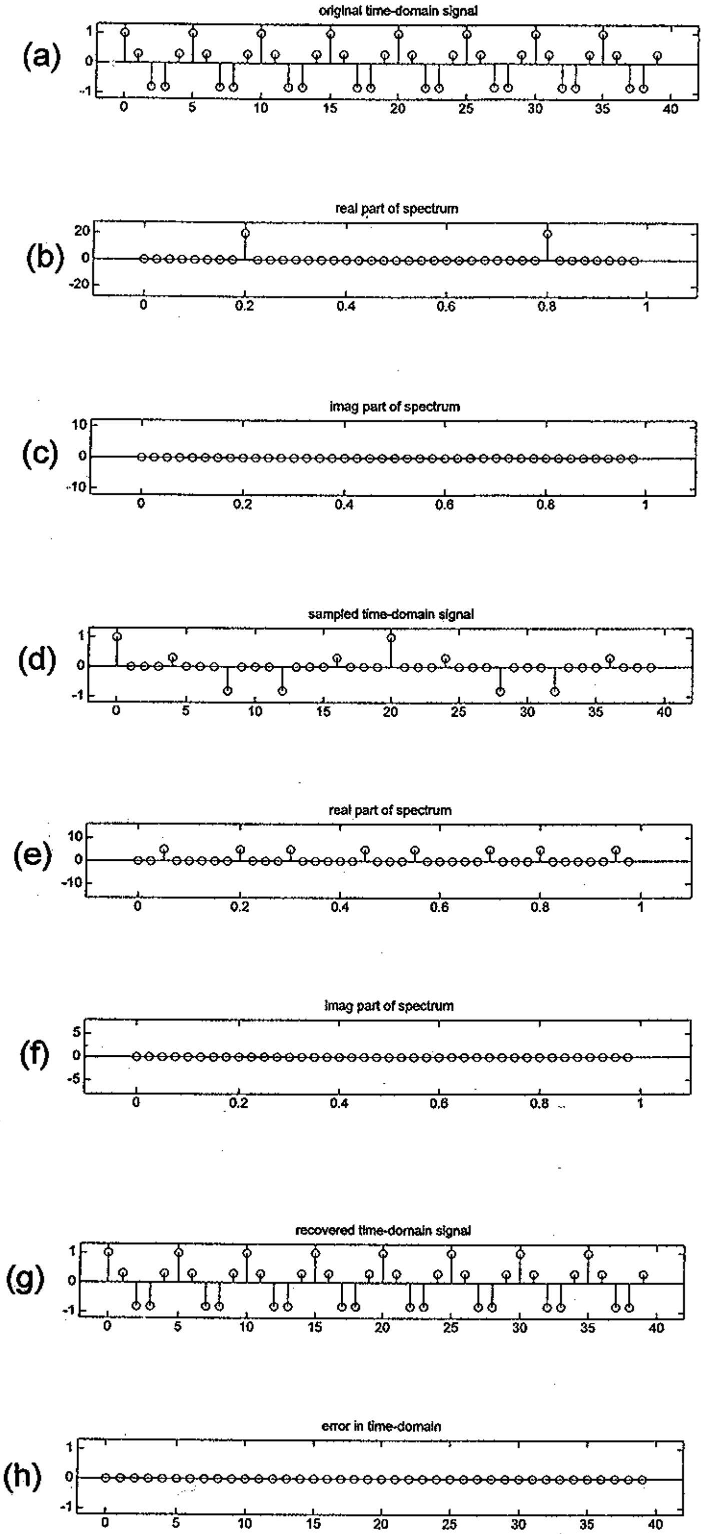


Fig. 19. A cosine of frequency 0.2 (a), (b) and (c) is downsampled by 4 as in (d), (e), and (f). The case in (e) corresponds to Fig. 18b. Using equation (25) we recover the cosine (g) from (d) without error (h)

7d. A SAMCELL with Non-Unity Values

As our final example we can consider a case previously studied [3] where samples are modified, but where no actual information is lost. This example can be expressed in terms of a SAMCELL that has elements that need not be unity. Specifically for this example, s=[1/2 1] with W=[1 1] represents a case where all even indexed samples are divided by two (Fig. 20).

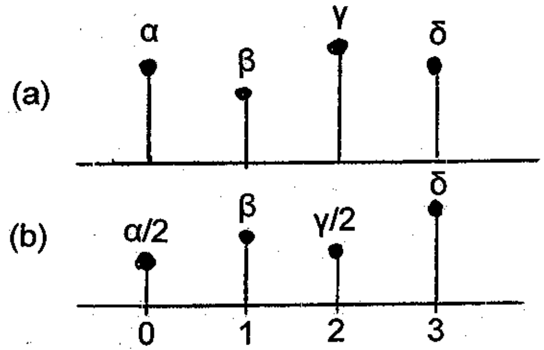


Fig. 20 Every even sample in (a) is set to half its value (b).

Fig. 21 shows the overlap and recovery scheme as a special case of what we have been doing in this note. In (a) we have a triangular spectrum with bandwidth w=1/2 (full size). Since no samples will actually be lost (we can recover them by multiplying by 2), we can expect to support this full bandwidth even though half the samples are modified (divided by 2). Only two segments (A and B) are needed here. It will be convenient, as before, to subtract the case where only the even samples (divided by 2) are kept (Fig. 21b) from the case where all samples are kept (Fig. 21a) to get the case where even samples are divided by 2 (Fig. 21c). Here we have (by inspection, or using Program 2):

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 3/4 & -1/4 \\ -1/4 & 3/4 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \qquad (26a)$$

and the inverse of this (ordinary or pseudo-inverse are the same since the matrix in invertable) is:

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \qquad (26b)$$

We see from Fig. 21d that the recovery is full and correct.

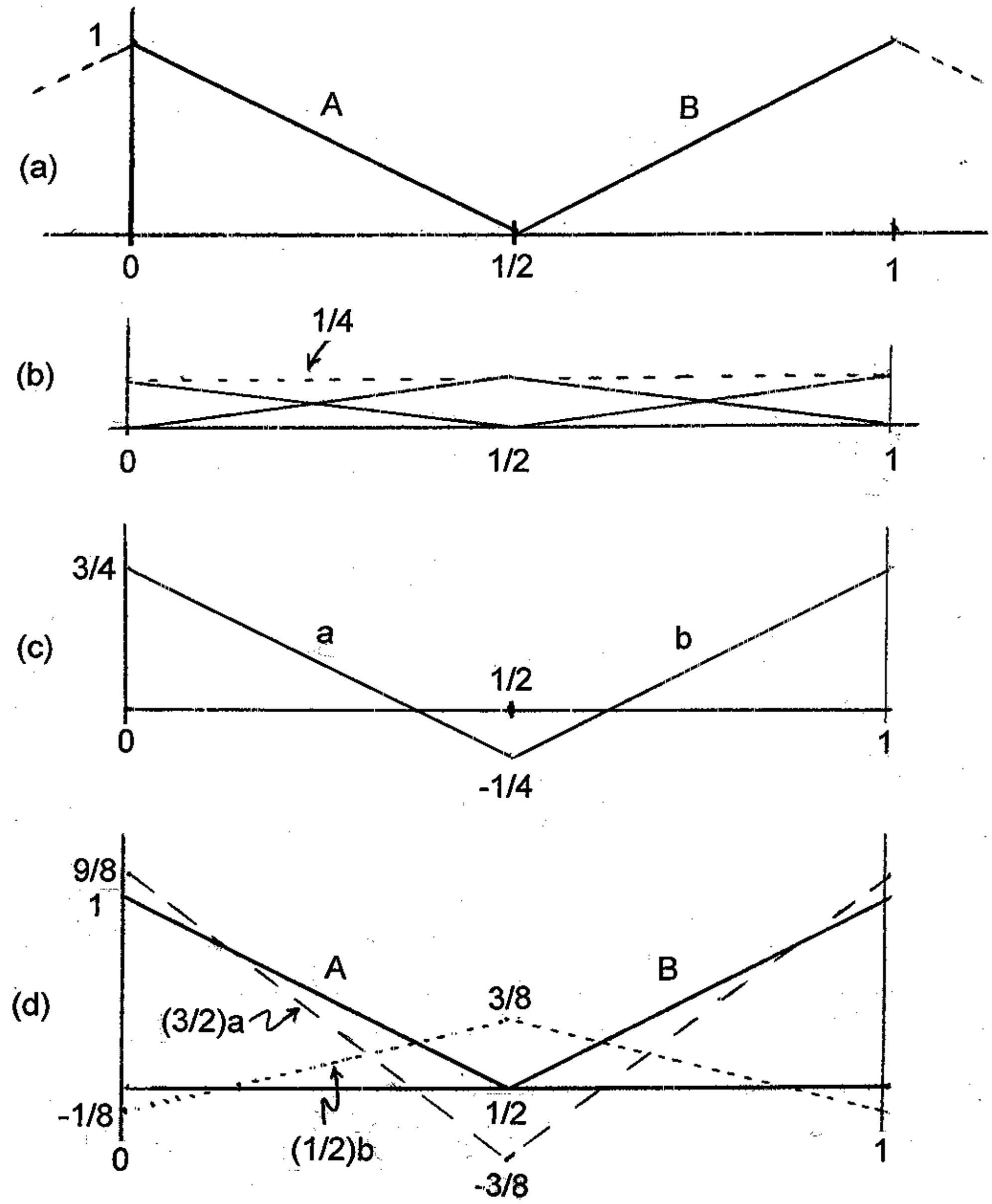


Fig. 21 An original spectrum is seen in (a). The spectrum of 1/2 times the even samples is seen in (b) to sum to 1/4. Subtracting (b) from (a) gives the sampled spectrum (c). In (d) we have the full and complete recovery using equation (26b).

REFERENCES

[1] B. Hutchins, "Sampling and Recovery Based on an Average Sampling Rate," Electronotes, Vol. 19, No. 187, August 1996, pp 3-34

[2] B. Hutchins, "Some Additional Sampling Examples," Electronotes, Vol. 21, No. 201, Feb. 2002, pp 10-20

[3] B. Hutchins, "Some Additional Sampling Examples," Electronotes, Vol. 21, No. 201, Feb. 2002, pp 7-10