ELECTRONOTES 201

NEWSLETTER OF THE
MUSICAL ENGINEERING GROUP
1016 Hanshaw Rd., Ithaca, NY 14850

Volume 21, No. 201 February 2002

GROUP ANNOUNCEMENTS

Contents of EN#201

Analog Signal Processing Corner

Second-Order Time Overshoot
A Second-Order Envelope Generator?

Another of our loose-ends left from the series on Analog Signal Processing was the question of time overshoot of a second-order filter [EN#192 (11)]. There we seemed to reference what was, probably, our own application note: "Second-Order Step Response" which was AN-103, Sept 18, 1978. This note in turn referenced the book by J.B. Murdock, Network Theory, McGraw-Hill (1970). At issue is mainly the formula giving the maximum overshoot. From the perspective of 2002, here is what we can consider:

(1) How was the equation derived? It should be straightforward differential calculus.

(2) Is the formula exact, or does it involve an approximation (such as we found in the decrement method of measuring filter Q as discussed in EN#198)?

(3) In 1978, I was working with a TI-59 programmable calculator. Today I have Matlab and a fast computer to both calculate and to answer the more basic questions. Will this help?

(4) And the question: Did we ever, or did anyone ever, consider using the step-response of a second-order low-pass as an envelope generator? Most analog envelope generators were first-order low-pass of course.

There is no reason to attack these issues in order, so let's look at (2) and (3). We must review a bit, and those interested in the full derivation of the overshoot equation can look at AN-103. Keep in mind that by a second-order low-pass we mean a pair of poles in the left half of the s-plane, characterized by their radius and angle. When the angle is smaller than 45 degrees from the jω-axis, we know that the frequency response will have peaking as we approach the cutoff frequency. We are not talking about this peaking in the frequency domain. Rather we are talking about an overshoot seen in the step response. When we apply a step to the filter's input, the output moves from zero toward some other level (which may be just the level of the input step, but this is not required). Eventually, given a long enough time, it will converge on this final level. The question is: in the process or reaching this asymptotic level, did the output ever exceed that level? The answer for all dampings less than 2 is yes.

In AN-103 we started with the damping D, which is the reciprocal of Q (D= 1/Q). (With low-pass filters, it was usually the case that damping was discussed, while of course Q seemed more appropriate for bandpass.) From D, we define parameters α and β:

$$\alpha = -D/2 \qquad (1)$$

$$\beta = \sqrt{(1 - D^2/4)} \qquad (2)$$

from which we found the step response to be:

$$x(t) = 1 - e^{\alpha t}\cos(\beta t) + (\alpha/\beta)\, e^{\alpha t}\sin(\beta t) \qquad (3)$$

Here we have chosen a filter normalized to 1 rad/second, a filter with dc gain of 1, and a step height of 1. The Murdock formula for maximum overshoot is:

$$p = e^{\alpha\pi/\beta} \qquad (4)$$

so the maximum value of equation (3) should be:

$$x_{max} = 1 + p = 1 + e^{\alpha\pi/\beta} \qquad (5)$$

At this point, we can easily check the formulas by calculating a dense set of values for equation (3), search it for a maximum, and compare to equation (5). The result, using Matlab, is that there appears to be no possibility that the equation is other than exact. The results agree to 6 or 8 decimal places for values of D from 2 (critical dampling) down to 0.001. This is important when we try to do the calculus. Never underestimate the value of knowing or suspecting the correct answer!

So now we can try differentiating equation (3) which is not too hard:

$$dx(t)/dt = \beta\, e^{at}\sin(\beta t) - \alpha\cos(\beta t)\, e^{at} + \alpha\, e^{at}\cos(\beta t) + (\alpha^2/\beta)\sin(\beta t)\, e^{at} \qquad (6)$$

The two middle terms on the right side cancel, and we then set dx(t)/dt = 0 to get:

$$\beta \sin(\beta t) + (\alpha^2/\beta) \sin(\beta t) = 0 \qquad (7a)$$

$$(\beta + \alpha^2/\beta) \sin(\beta t) = 0 \qquad (7b)$$

$$(1/\beta) \sin(\beta t) = 0 \qquad (7c)$$

from which we conclude that local min/max occur at $t = m\pi/\beta$ where m is an integer. The case of m=1 is the first maximum and when we plug $t=\pi/\beta$ into equation (3) we indeed get:

$$x_{max} = 1 + e^{\alpha\pi/\beta} \qquad (8)$$

Fig. 1 shows a typical step response for the case where D=0.5 (Q=2) and we note something like a 45% overshoot followed by "ringing" converging toward the level of the step. Some additional examples are shown in Fig. 2

This clears up the loose ends from the ASP material and leaves us to wonder if we ever considered a second-order low-pass as an envelope generator. One thing is clear that the cutoff would have to be fairly low - probably well into the sub-audible range. What we are suggesting here is that instead of a special purpose ADSR circuit, we use the step overshoot for an elaborated "DS" part. The obvious problem is that we have an overshooting step response here, which is fine for the beginning of a gate (the attack part of the envelope), but probably not for the end. Fig. 3 shows an example of D=0.8 in response to a gate.

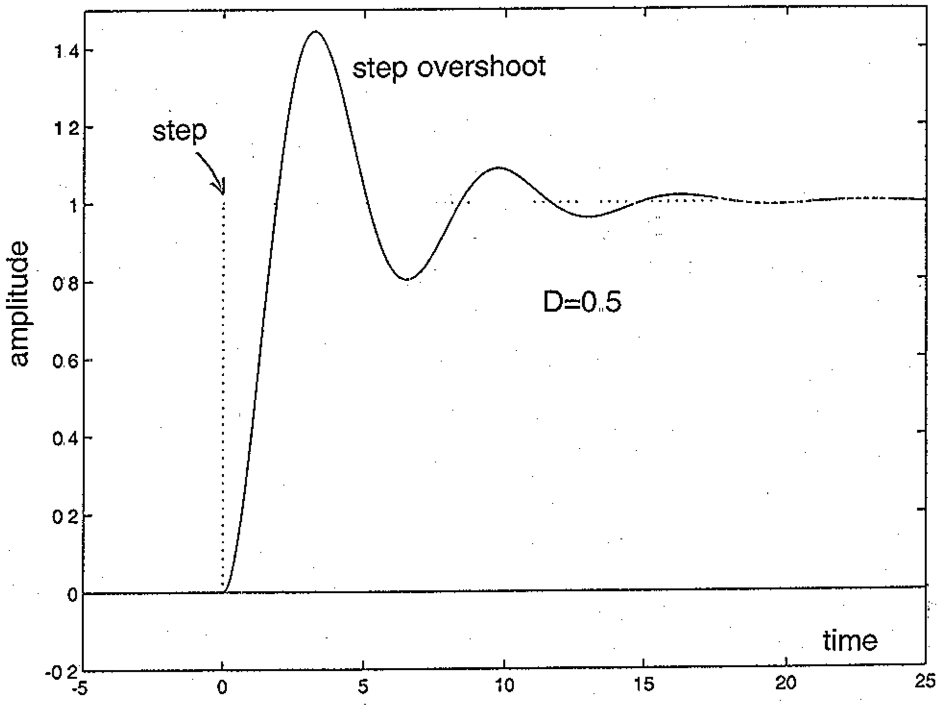
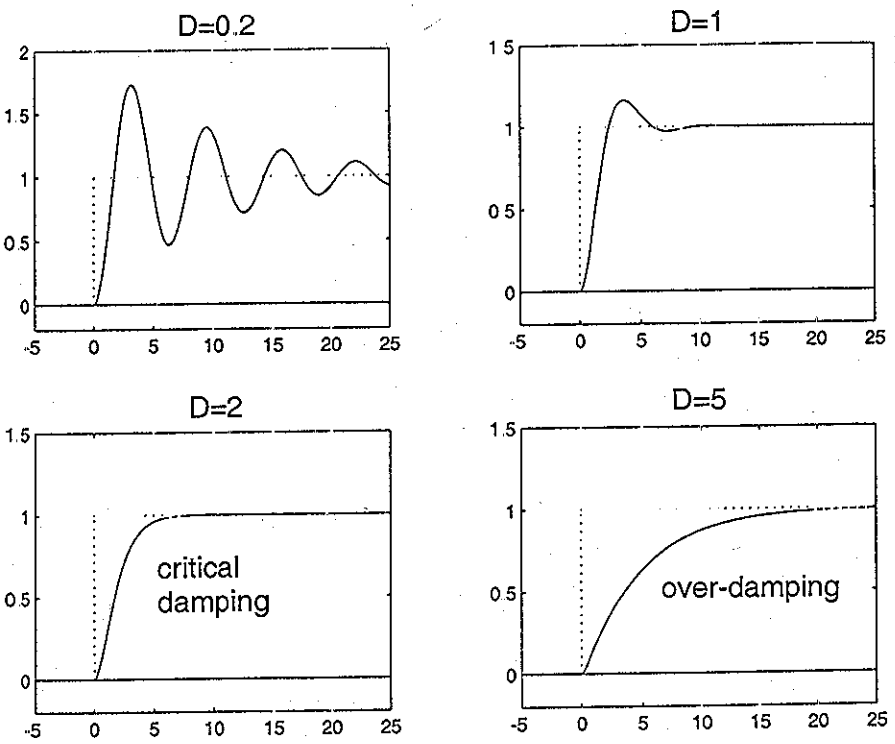
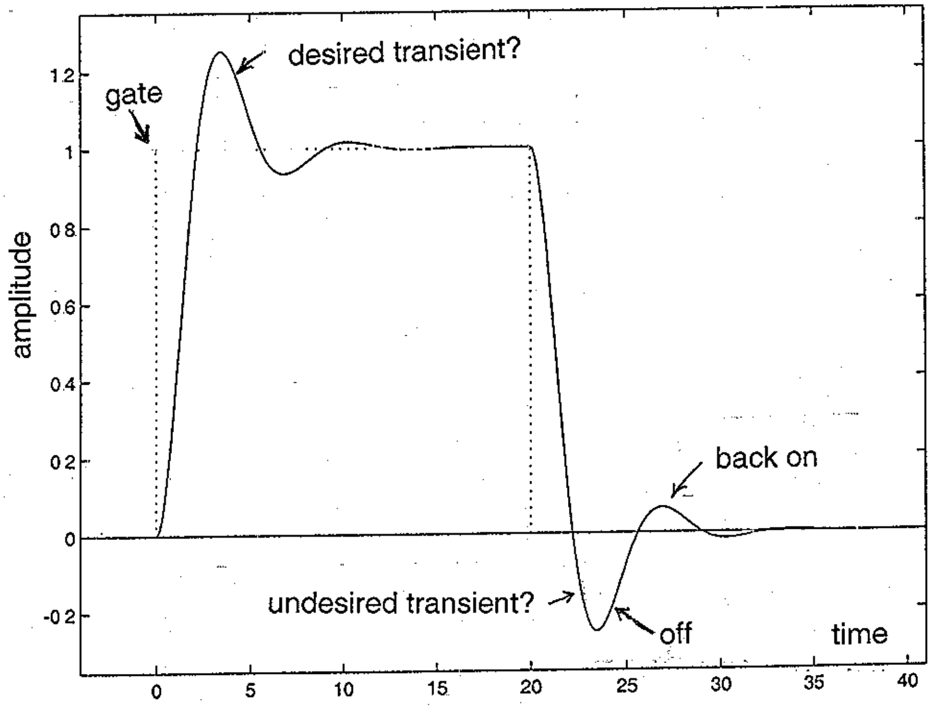


Fig. 1 Second-Order Step Response (D=0.5)



<u>Fig. 2</u> More Examples of 2nd Order Step Response



<u>Fig. 3</u> Response of 2nd Order Filter to Gate (D=0.8)

Indeed we might well like the fancy overshoot transient during the attack. But, we get the exact same (well - upside down) transient when the sound goes away. The ringing about zero at the end would give some "echoes" which we would not always want. [Specifically, an amplitude envelope would be typically fed to a VCA which is a two-quadrant multiplier. This means that when the envelope goes negative (at about time 22 on Fig. 3), the amplitude would turn off rather hard. It would then return for a "blip" for around time 26 to 29, etc.] Of course, we should make no hard rules about how a music synthesizer should be used, but we always do ourselves a service by showing that we can get our schemes to do something relatively

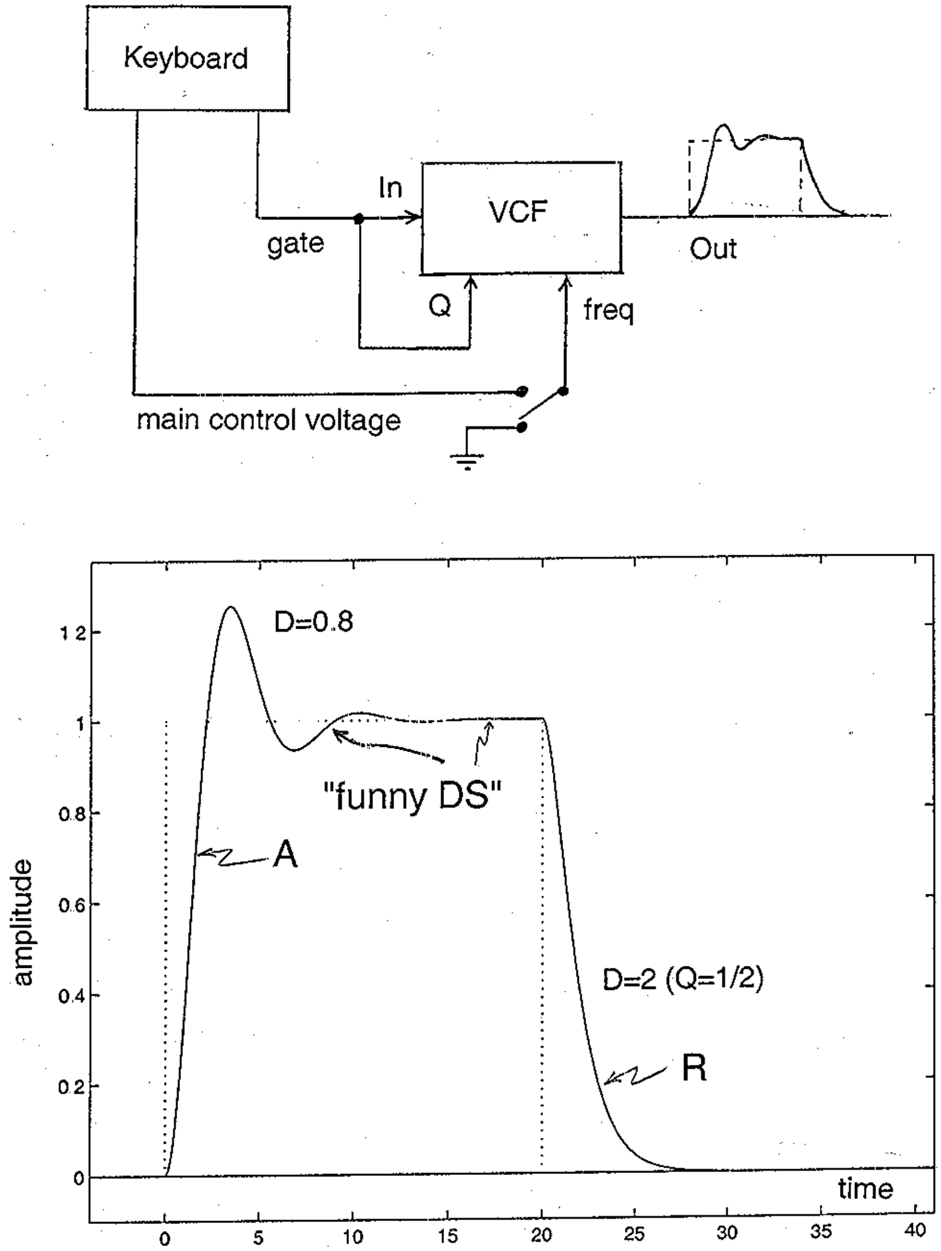


Fig. 4 Scheme for Using VCF as Envelope Generator (top) and a typical envelope (bottom)

conventional. There is time later for quirks! But, we can of course get rid of the decay overshoot by changing the damping factor during decay. This might seem inconvenient, but after all, we are probably going to try this with an existing voltage-controlled filter, and it may well have its own voltage-controlled Q (many of our designs did).

So to attempt this experiment, you would manually set your VCF low-pass to a very low frequency (perhaps 3 Hz). Then you would connect the keyboard gate to the VCF input, and also to the VC Q control, with the Q set initially to 1/2 (critically damped). When the gate arrives, the Q jumps up and the filter responds to the gate with overshoot. Now, when the gate goes low, the Q drops back to 1/2 and the envelope decays down to zero without overshoot. Fig. 4 shows the general idea.

One thing that should always bother us when we suddenly change the parameters of a filter is that we may create, internally, a transient just as severe as we might if we suddenly changed the input. What happens here when the Q suddenly changes from some value greater than 1/2 down to 1/2. If we get a big blip, then the idea of getting a smooth decay to zero, as in Fig. 4, may be fiction. Here it is useful to recall the state-variable filter, that is probably the basis for the VCF we are using (Fig. 5). The Q is controlled by the feedback from the bandpass output (V_B) to the summer by gain path 1/Q. But, after the "attack," the bandpass output, in response to a step (the gate) at the input goes to zero. Another way to look at this is to note that the envelope, at the low-pass output V_L, has become a constant (the "sustain"). Since V_L is the output of an integrator (the input of which is V_B), V_B must be zero, or else V_L is not a constant (it would be ramping). Likewise, V_H must be zero! With V_B zero, the 1/Q path could be any value and it would make no difference until V_B begins to ramp again. In short, it seems that this works the way we suggest, at least under the conditions we suggest.

This is such a simple idea that someone must have tried it. Did we? If not, someone may want to try it and let us know how it works. One additional idea is that we have specifically said here that the VCF frequency would be set low, manually. It is the setting of this frequency that sets the time constants of the entire envelope. [With a conventional ADSR envelope generator, the time constants during the A, D, and R parts were usually set independently, and manually. A few voltage-controlled envelope generators were likely tried.] But here we could set the frequency low, and then still have it respond to the main keyboard control voltage, shortening up the transient with increasing pitch, which would seem to be the sort of thing a lot of acoustic instruments do.

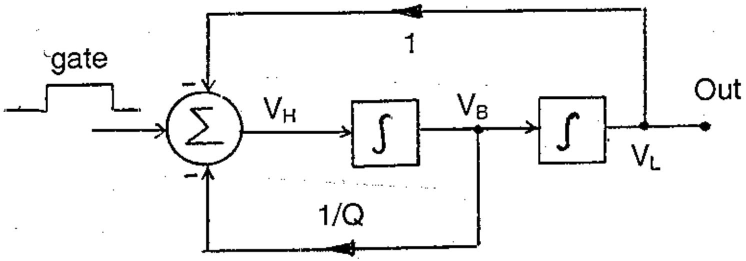


Fig. 5 The State-Variable Structure

SOME ADDITIONAL SAMPLING EXAMPLES

-by Bernie Hutchins

Last issue we presented as part of the Basic Elements of DSP the main material on sampling. At that time, it was said that we had some additional examples of sampling that we wanted to get in at some point. For reasons unrelated to this publication, I found it necessary to review one of these examples, and at that point, it became obvious that I had better get them written up before I forgot how to do them.

Example 1 Scaling Every Other Sample - No Information Lost:

In this example, we suppose that we have a sequence of samples x(n) and that something happens to the sequence (perhaps a serial/parallel converter slips by one bit) such that every other sample is divided by 2:

$$\begin{aligned} y(n) &= x(n) \quad && n \text{ odd} \\ y(n) &= x(n)/2 \quad && n \text{ even} \end{aligned} \qquad (1)$$

If we suppose that the spectrum of x(n) is nicely bandlimited (Fig. 1) then we find that the spectrum of the modified signal y(n) is significantly different (Fig. 2). It is not difficult to understand what has happened. One way we could do this would be to recognize that y(n) is composed of two sequences, both of which are samples of x(n), but at half the sampling rate. One is the odd samples, and the other is the even samples (scaled by 1/2). As such, both sequences have replicas as multiples of 1/2 (not just at multiples of 1) and they are summed. Hence, the mess - the spectra overlap and add. We could approach the analysis in this manner.

Another, equivalent way to look at this would be to say that y(n) is x(n) with the even samples scaled by 1/2 and subtracted. Working with the even samples in this way has the advantage that the replicas are not rotated (see second example). While we might well become fond of the moose-like fancy spectra of Fig. 1 and Fig. 2, it will be simpler to draw triangular shapes for the remainder of the analysis. Keep in mind that when we set up equations to recover the original spectrum, it does not depend in any way on the triangular shape, and would work equally well for the moose.

Well, we did just say we were going to recover the original spectrum. Clearly here, by either interpretation, we have an overlap and the addition of different spectral images. In a "classical" sense we might suppose that we have messed up things to the point where no recovery of x(n) from y(n) is possible. Yet - OBVIOUSLY it is possible. We know this because no information has been lost. Knowing that the even samples are only half what they should be, it is trivial to multiply these by 2. Thus knowing that we can recover x(n) in the time domain, it is clear that we should also be able to recover in the frequency domain. We will work this out, and see (not surprisingly) that both frequency and time domain solutions suggest the same implementation for actual recovery.

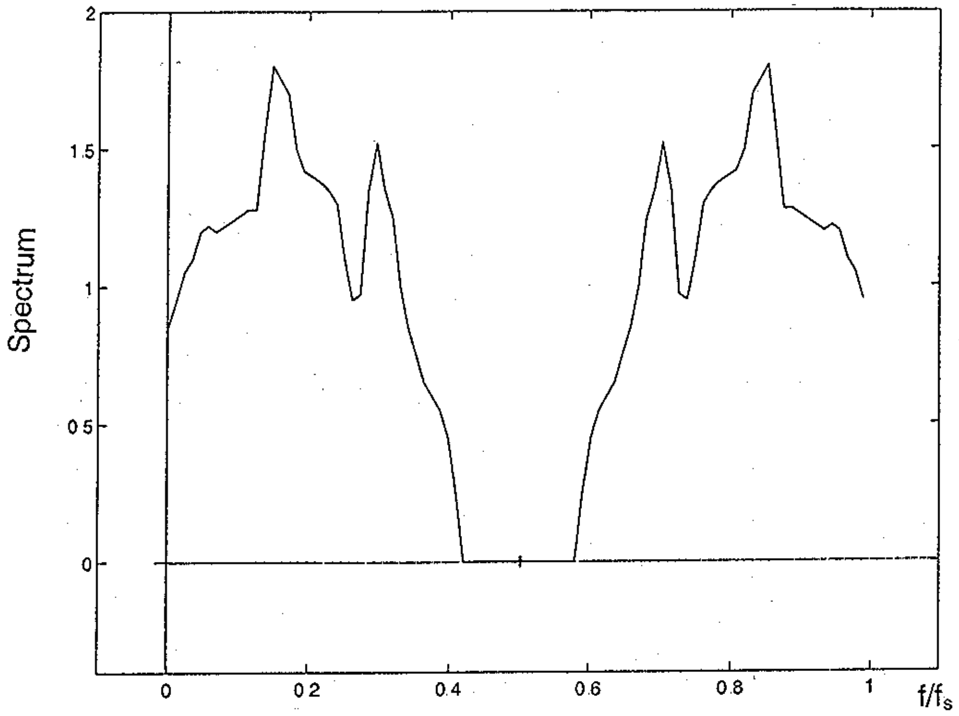


Fig. 1 An original "moose-like" spectrum is bandlimited to just above 0.4. This is perhaps a typical spectrum, but it is difficult to draw and to work with.

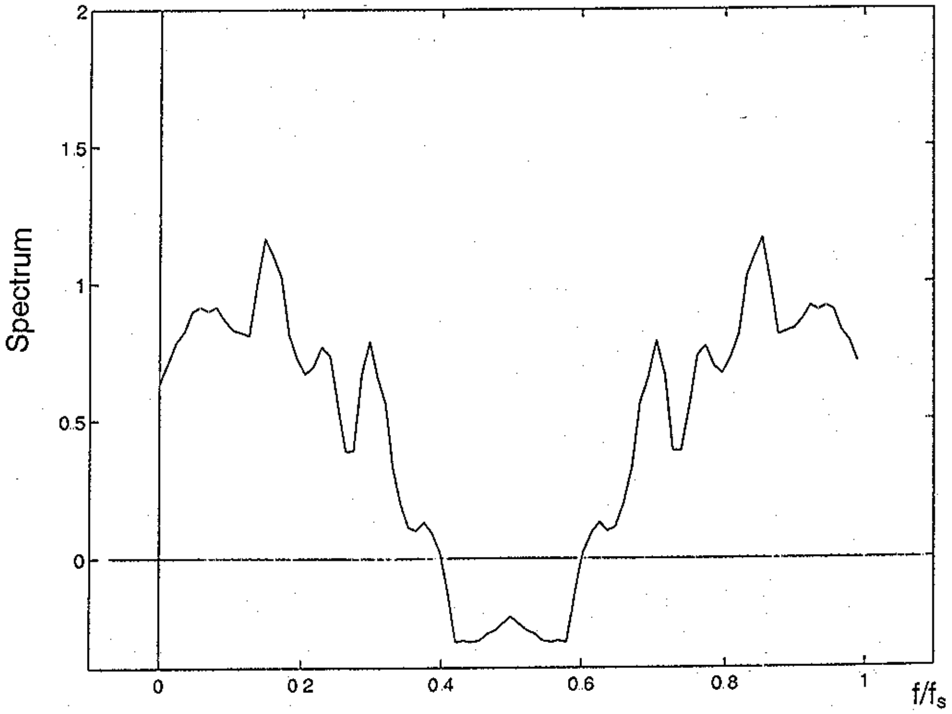


Fig. 2 Above spectrum after dividing even samples by 2. We really do need to try a simpler shape (see Fig. 3).

Fig. 3 shows the original spectrum X(f) as a dashed line. It also shows the negative of the spectrum of the even samples, multiplied by 1/2, as a dotted line, replicated about multiples of 1/2. Note that these replicas are of height 1/4; first they are halved because the samples were multiplied by 1/2, and then again halved because there are twice as many of them (think Parseval). The sum of these two is Y(f) shown as a solid line. It is Y(f) that we have, and we want to get back X(f).

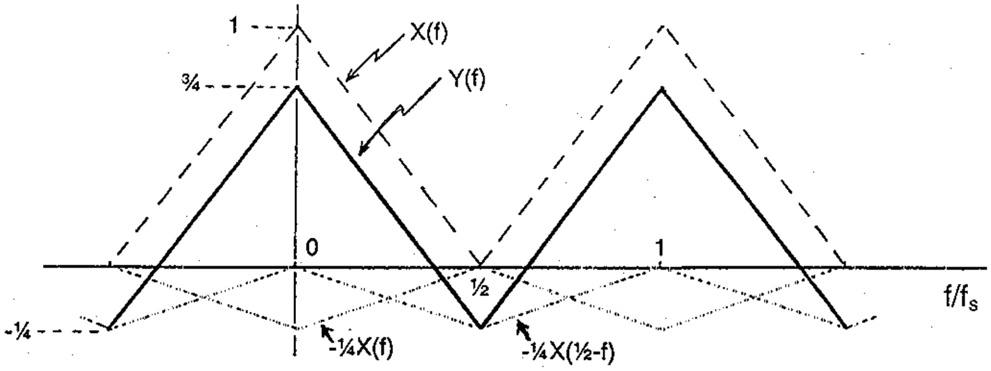


Fig. 3 Triangular X(f) with even samples scaled by 1/2 and subtracted gives Y(f)

Noting that Y(f) is everywhere the sum of three pieces, we can try to set up some equations. Let's concentrate on the frequency region from 0 to 1/2. Here X(f) contributes a full valued triangle, but there is an identical triangle with weight -1/4 exactly aligned with it, for a net 3/4 of X(f). The remaining contribution is just the triangle of weight -1/4 centered about and coming back from 1/2. Thus we can write:

$$Y(f) = (3/4)\ X(f) - (1/4)\ X(1/2 - f) \qquad (2a)$$

We can also write the sum at Y(1/2 - f) to give:

$$Y(1/2 - f) = (3/4)\ X(1/2 - f) - (1/4)\ X(f) \qquad (2b)$$

We know Y(f) and have two equations, so we can solve for X(f) or X(1/2 - f). Solving for X(f) we have:

$$X(f) = (3/2)\ Y(f) + (1/2)\ Y(1/2 - f) \qquad (3)$$

This is the recovery we wanted. Keep in mind that this also works for the moose - not just for the triangles.

This suggests the scheme of recovery by which we take y(n), multiply its terms by $(-1)^n$ (thus shifting the spectrum by 1/2), and sum half of this result with 3/2 of y(n). This is seen in Fig. 4. What have we arrived at? Well, exactly what we would have done if we wanted to recover the samples in the time domain!

For additional insight, consider the scheme of Fig. 5 which is the time-domain "maker" of the bad sequence: messing up x(n) to get y(n). In the frequency domain, this describes exactly equation (2a).

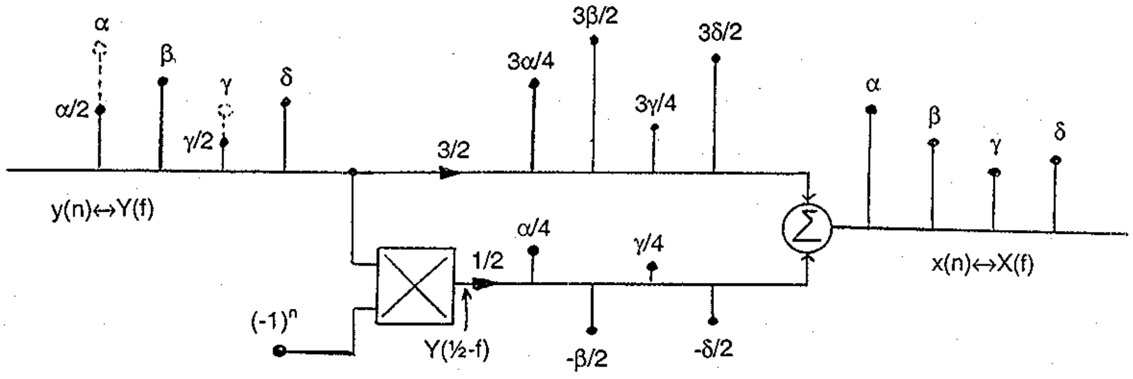


Fig. 4 Recovery scheme is seen to work in the time domain, and relates to equation (3) in the frequency domain

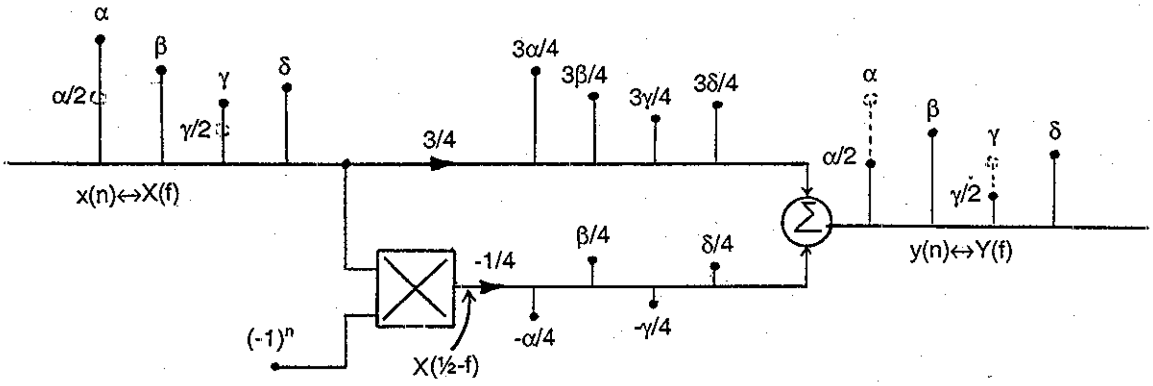


Fig. 5 The time-domain "Maker" of the sequence y(n) relates to equation (2a)

The lesson here is that spectral components may be overlapped and added, and we do not have to loose anything. In this case, we could see in the time domain that no information had been lost.

Example 2 Half the Samples Lost - Bunched Samples

Here we will once again have overlapping spectra which add but we are actually going to throw out information. We will see that this restricts our available bandwidth. But, perhaps not as much as we at first might suppose. Here is the situation. We have a sequence x(n) and we keep two samples and then throw away two, and so on.

$$\begin{aligned} y(0) &= x(0) \\ y(1) &= x(1) \qquad (4) \\ y(2) &= 0 \\ y(3) &= 0 \\ &\text{etc.} \end{aligned}$$

Because we are throwing out information (multiplying by 0 this time) we do not expect to support a bandwidth of 1/2 any more. Think of a simpler problem where we zero every other sample. In this case, we know that if the bandwidth is only 1/4, we have no spectral overlap. We pay the expected price for throwing out half the information, and only get half the bandwidth.

So let's suppose we start with the idea of having only a bandwidth of 1/4 and we want to see if and how the information is preserved if we discard half the samples in various manners. Fig. 6a shows a spectrum (taken to be a triangle - again just for ease of sketching) where all samples are kept. There is all that empty space between 1/4 and 3/4. We know that if we throw out every other sample (as suggested above), that the situation of Fig 6b must obtain (an additional replica centered at 1/2). In order to understand the new case proposed here, it will be easiest to first consider the case where only one in four samples is kept.

Fig. 6c shows the case where x(0) is kept along with x(4), etc., but three of four of the samples of x(n) are set to zero. There are now three additional images (centered at 1/4, 1/2, and 3/4) added to the originals, and we have a hopeless overlap. In fact, these add to a constant of 1/4 for all frequencies. The same sort of situation is true for any initial position of kept samples. (These would work, of course, if the bandwidth were 1/8 instead of 1/4.) If we keep samples x(1), x(5), etc., or x(2), x(6), etc., or x(3), x(7), etc., we again have hopeless overlaps. But these are cases where we really do keep only one of four samples, not two of four on average. What happens if we get some help? Well, we know that if we consider the particular instance of keeping two samples of four, i.e., x(0), x(2), x(4), x(6), etc., that we have the happy result of Fig. 6b. So somehow, two unhappy situations: x(0), x(4), etc. along with x(2), x(6), etc. must combine to cancel some overlapping images.

To see what is going on, and to eventually attack the problem at hand, we need to look at additional instances of keeping one sample in four, and these are shown in Fig. 6d, Fig. 6e, and Fig. 6f. Here we see the important fact that, being sampled at instances displaced relative to the original set (Fig. 6c) causes the additional images to "rotate" as shown. (See Section 4c-1 of Sampling Element, EN#200, pg 42. Here we have four different sampling functions, [1 0 0 0], [0 1 0 0], [0 0 1 0], and [0 0 0 1] and each of these sequences has a different DFT.) Note that some images are accordingly imaginary, and we have tried to plot this rotation (by showing a tipped imaginary axis). All that remains to do is sum the results.

Thus we see that summing Fig. 6c with Fig. 6e does in fact give the happy result of Fig. 6b. The images centered at 1/4 and at 3/4 cancel each other. We get a clean original copy between -1/4 and +1/4, of height 1/2 (easy enough to scale by 2).

Finally, the new case. We note that in keeping samples x(0), x(1), x(4), x(5), etc. that we are keeping half the information. We should be able to recover the original spectrum. It may not be quite as easy as in Fig. 6b. We have only to sum Fig. 6c and Fig. 6d. This is real hard to draw (attempted in Fig. 6g - see also Fig. 7b, etc.). Note that the real images about f=0 sum to an amplitude of 1/2. The image at f=1/4 is real in the case of Fig. 6c and imaginary in the case of Fig. 6d, so we get something that is now complex (tipped back 45 degrees) and at amplitude $\sqrt{2}/4$. At f=1/2, two real images cancel. This is good news. At f=3/4, we again get a complex result, tipped forward by 45 degrees. The recovery is possible in the un-overlapped region between 1/4 and 3/4. True they are complex, tipped, and reversed, but they are clean. Now what?

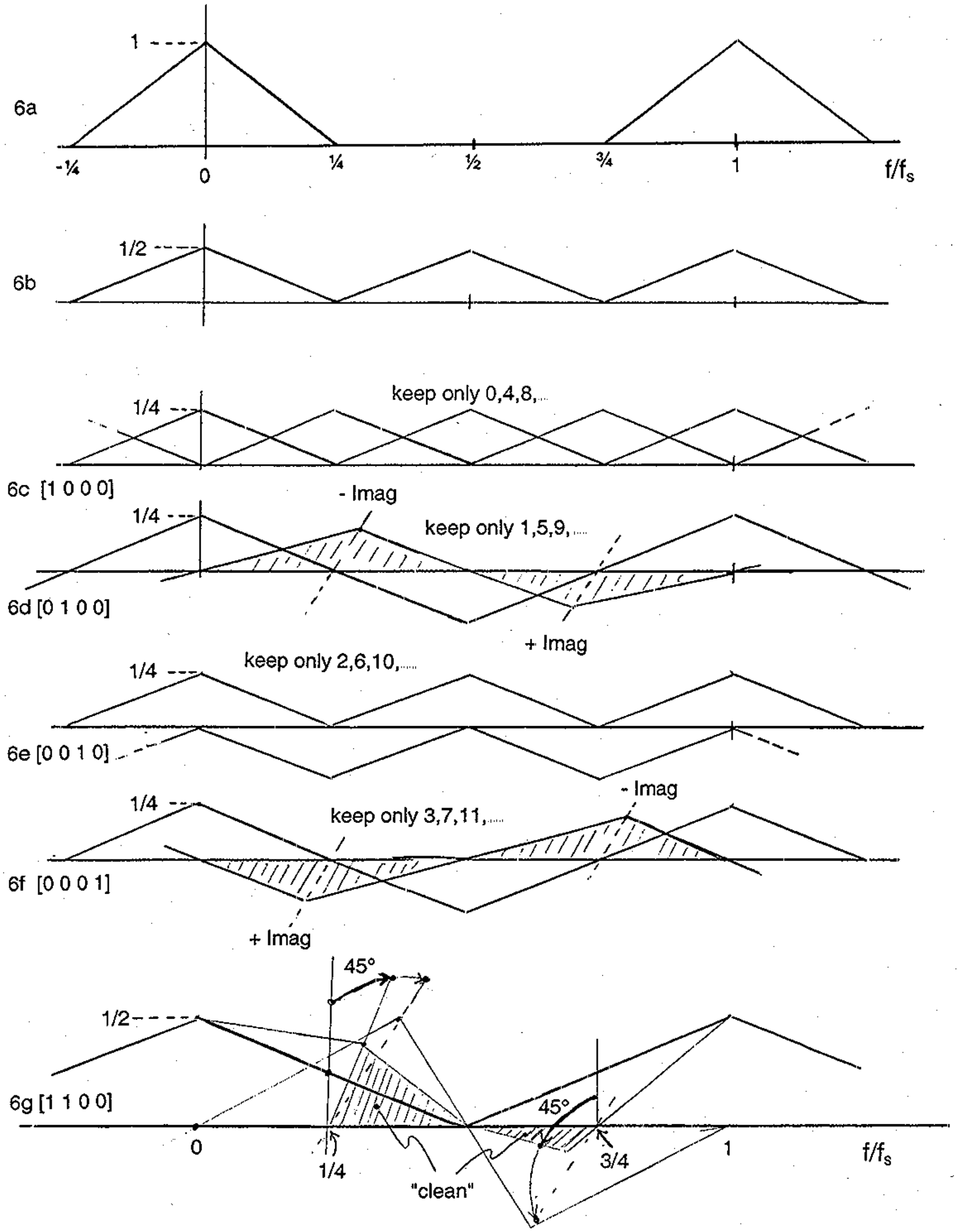
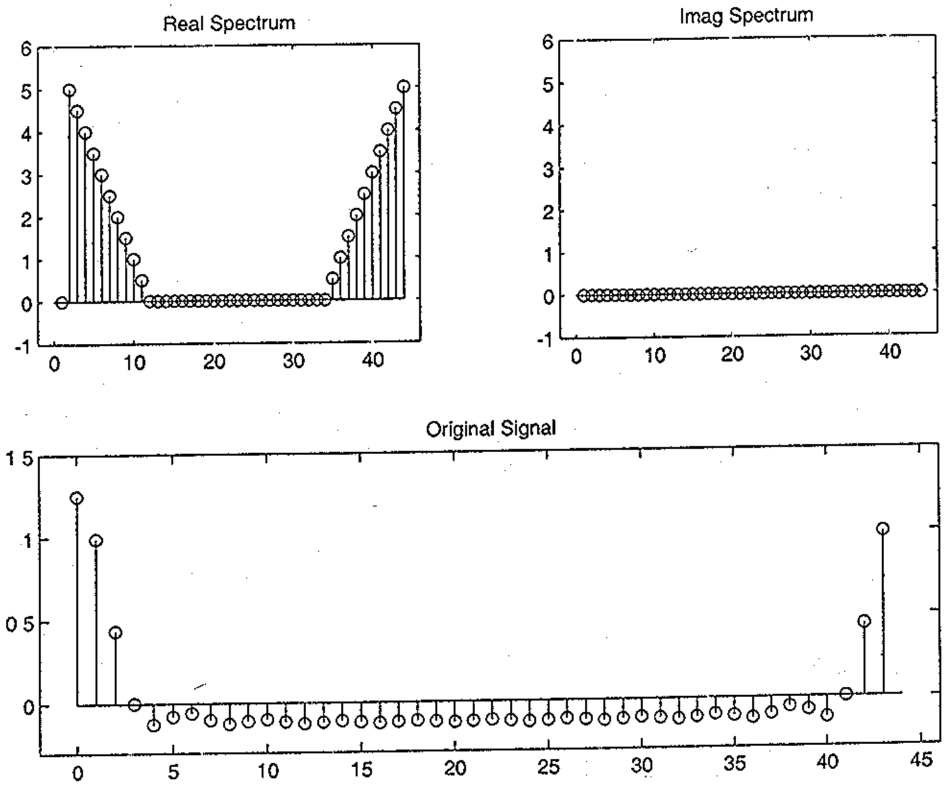
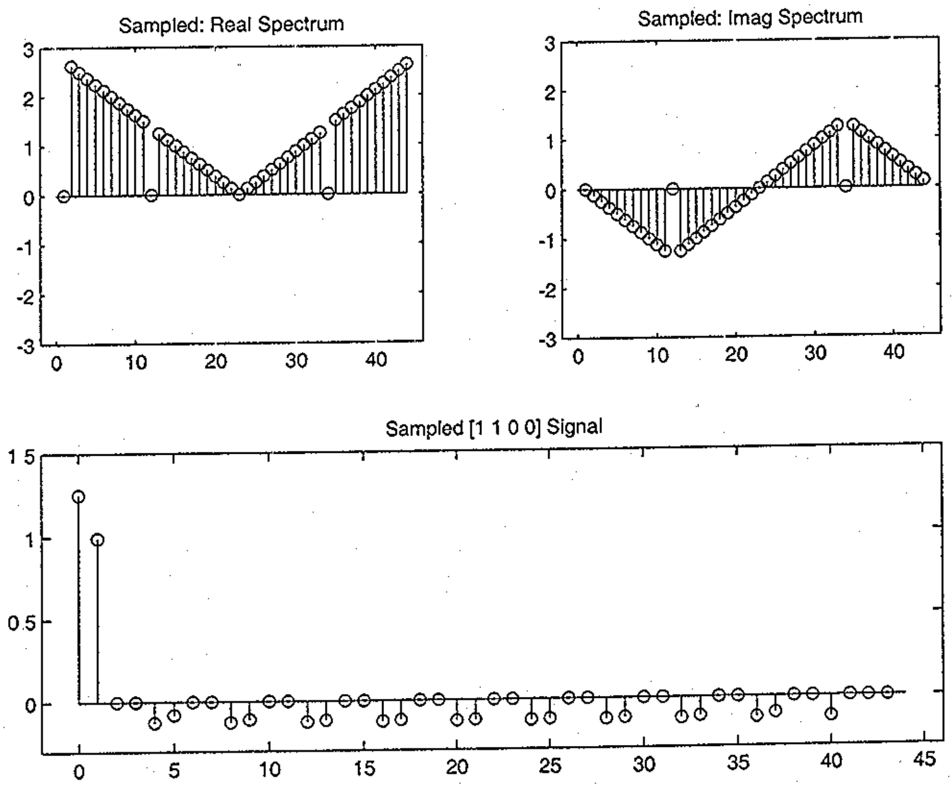


Fig. 6 Sampled in pairs and zeroed in pairs.

Well, we have to try to recover the signal. The first step would seem to be to high-pass the spectrum of y(n), so that we have only the clean copies. Then we would need to modulate the high-passed output back down to its original position. Finally, we low-pass out the desired images (and scale for any amplitude losses). This whole procedure we illustrate in a series of plots in Fig. 7, first creating a triangular test spectrum, and manipulating with the FFT for convenience of illustration. In each of the nine sets of three plots in Fig. 7, the top two plots show the real and the imaginary parts of the spectrum, while the lower plot shows the time-domain samples. The signals are length 44.



<u>Fig. 7a</u> An original triangular spectrum (Compare to Fig. 6a)



<u>Fig. 7b</u> Sampling in pairs and discarding in pairs results in this complex (and complicated) spectrum.

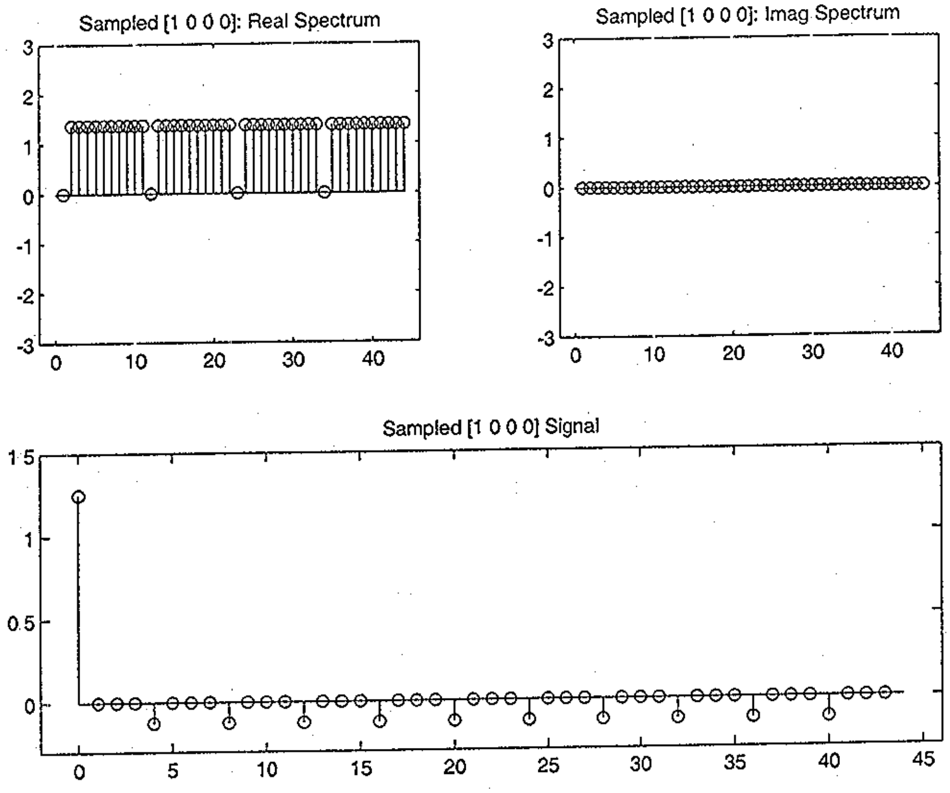


Fig. 7c Sampling one in four, starting with 0

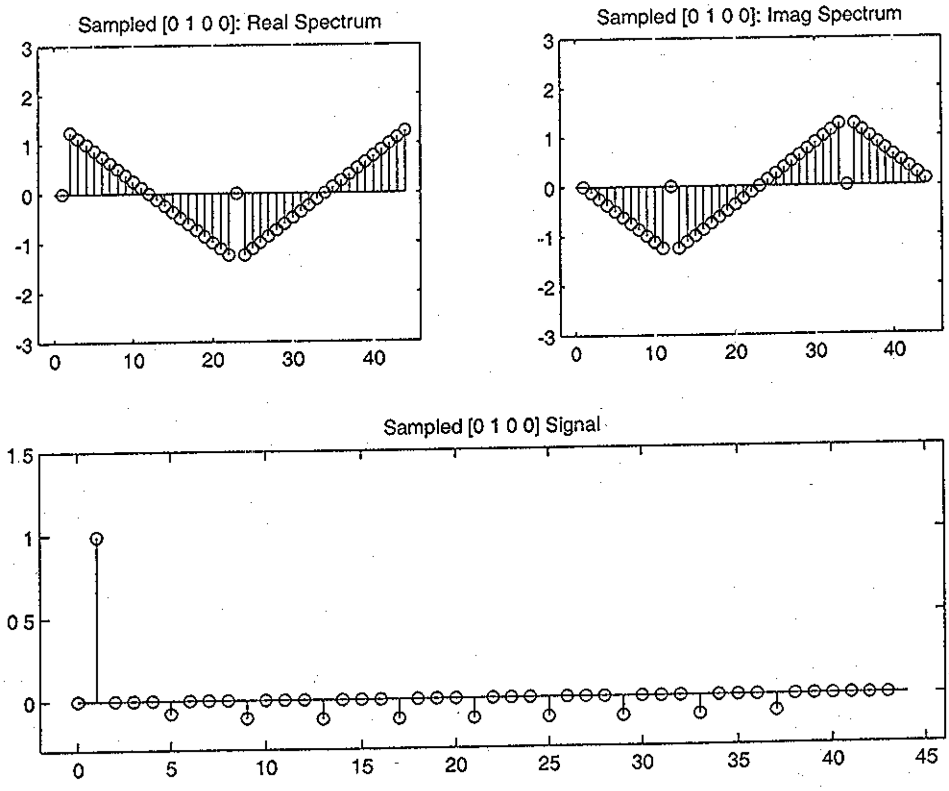


Fig. 7d Sampling one in four, starting with 1

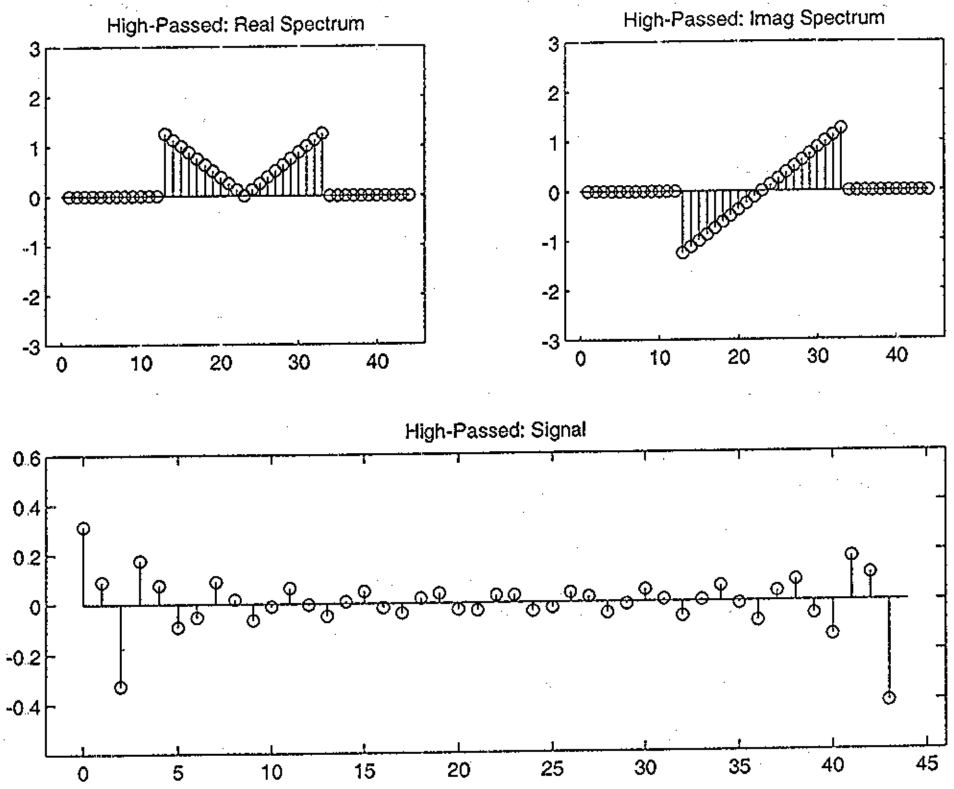


Fig. 7e High-pass of sampled spectrum is "clean" portion (compare to Fig. 6g)

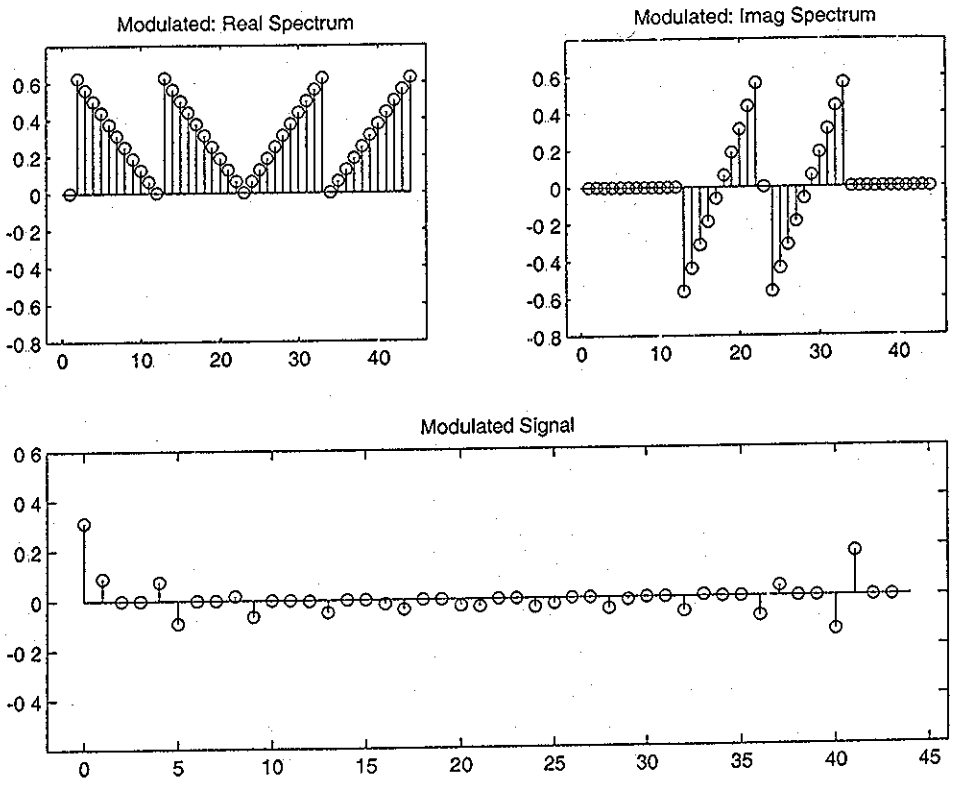


Fig. 7f Modulating the high-passed signal by [1 1 0 0]

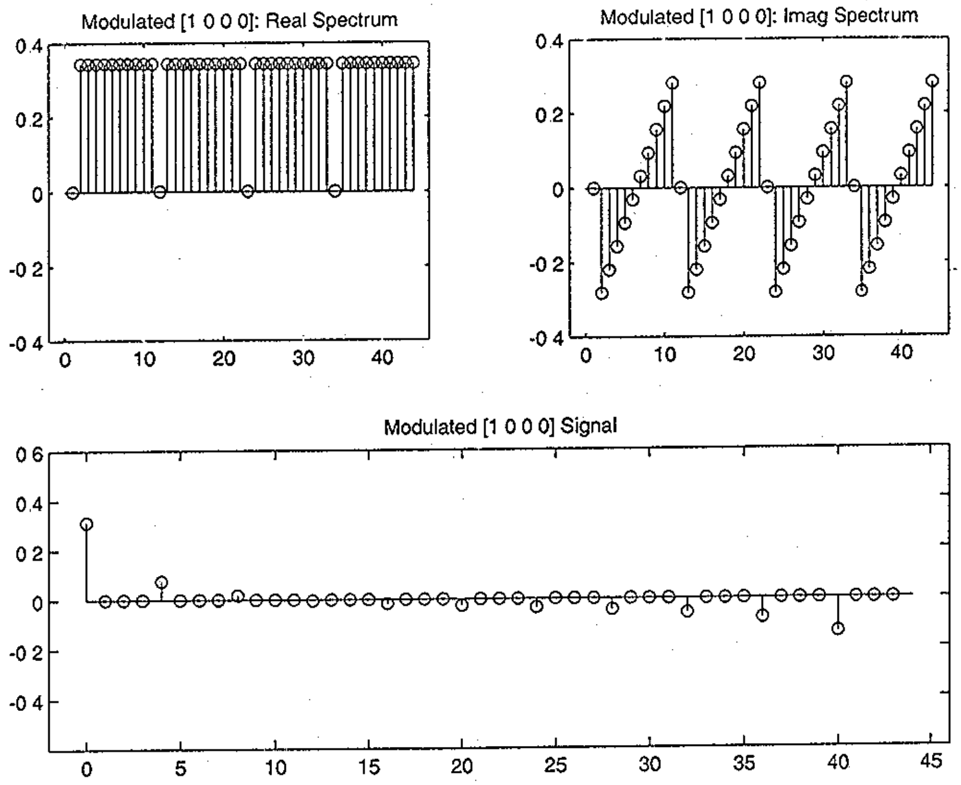


Fig. 7g High-passed spectrum modulated by [1 0 0 0] (portion of Fig. 7f)

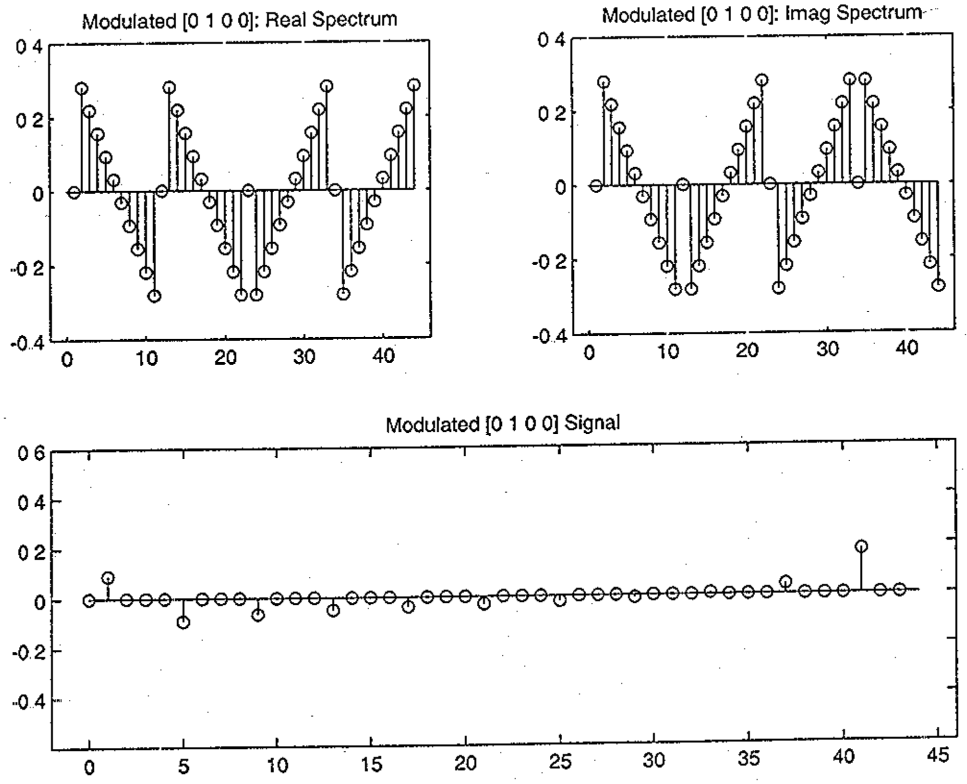
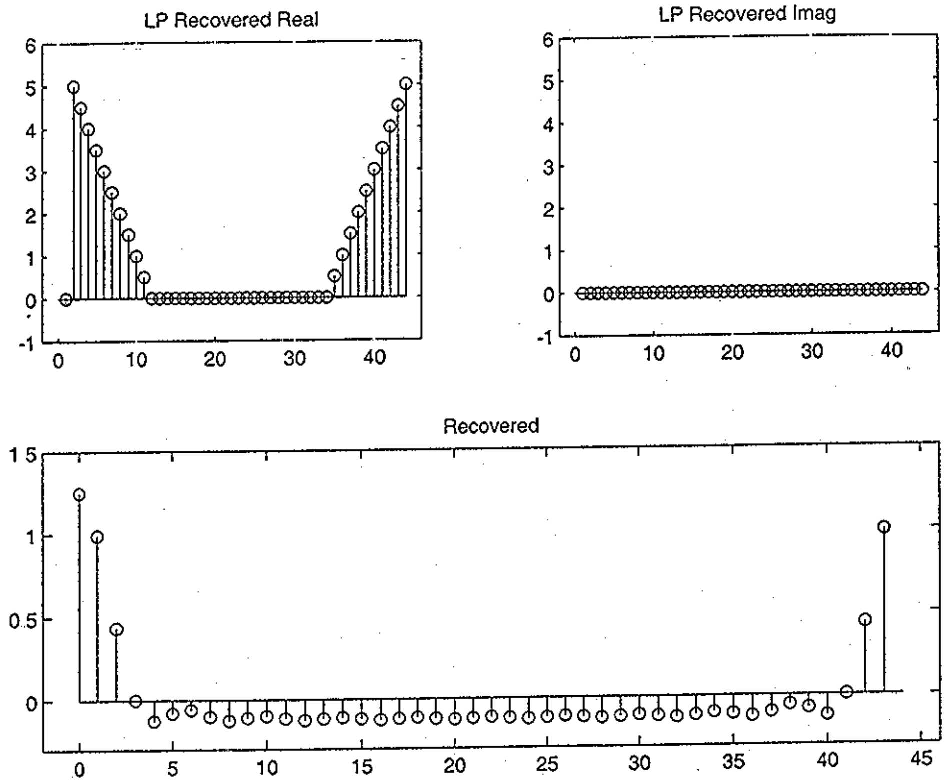


Fig. 7h High-passed spectrum modulated by [0 1 0 0] (portion of Fig. 7f)


LP Recovered Real

LP Recovered Imag

Recovered

Fig. 7i Low-passing recovers original spectrum from modulated high-pass

Fig. 7a shows the original spectrum. This was created simply by specifying a real, triangular FFT as shown. The spectrum chosen uses most of the region between 0 and 1/4, but we have left a bit of empty space on the ends so that we can see the limits. The time signal is obtained by inverting the FFT. While the processing in Fig. 7a is down, that of Fig. 7b is upward. We take the time sequence of Fig. 7a and sample it by keeping two samples, zeroing two, etc. to get the time sequence at the bottom of Fig. 7b. Hence working upward, we obtain the FFT, which now has an imaginary part as well as a real part. This is the result we were trying to draw in Fig. 6g (except here we have left a couple of zeros in the spectrum as markers). Fig. 7c and Fig. 7d dissect the results of Fig. 7b into the results of taking samples 0, 4, 8, ... (7c) and samples 1, 5, 9,..... (7d). These we can compare with Fig. 6c and Fig. 6d. Note that in Fig. 7c the spectrum is purely real and constant (except for the intentional marker holes). In Fig. 6c we would have had a constant if we had added the triangular pieces. Fig. 7d corresponds to the "rotating" triangular images.

Going back to Fig. 7b, we note the "clean" spectrum between spectral samples 12 and 32. Because we are working with the FFT here, it is trivial to make a high-pass filter here by simply masking out the FFT values outside this range. The simple result is seen in Fig. 7e, where we have then also taken the inverse FFT of the high-passed spectrum to see the corresponding time waveform, at the bottom.

So far, all this was easy and logical. But at this point, what do we do other than resort to the general idea of modulating the signal back down? We are reminded of the saying about the "proof of the pudding" and we must admit that a certain amount of guessing and trying is not

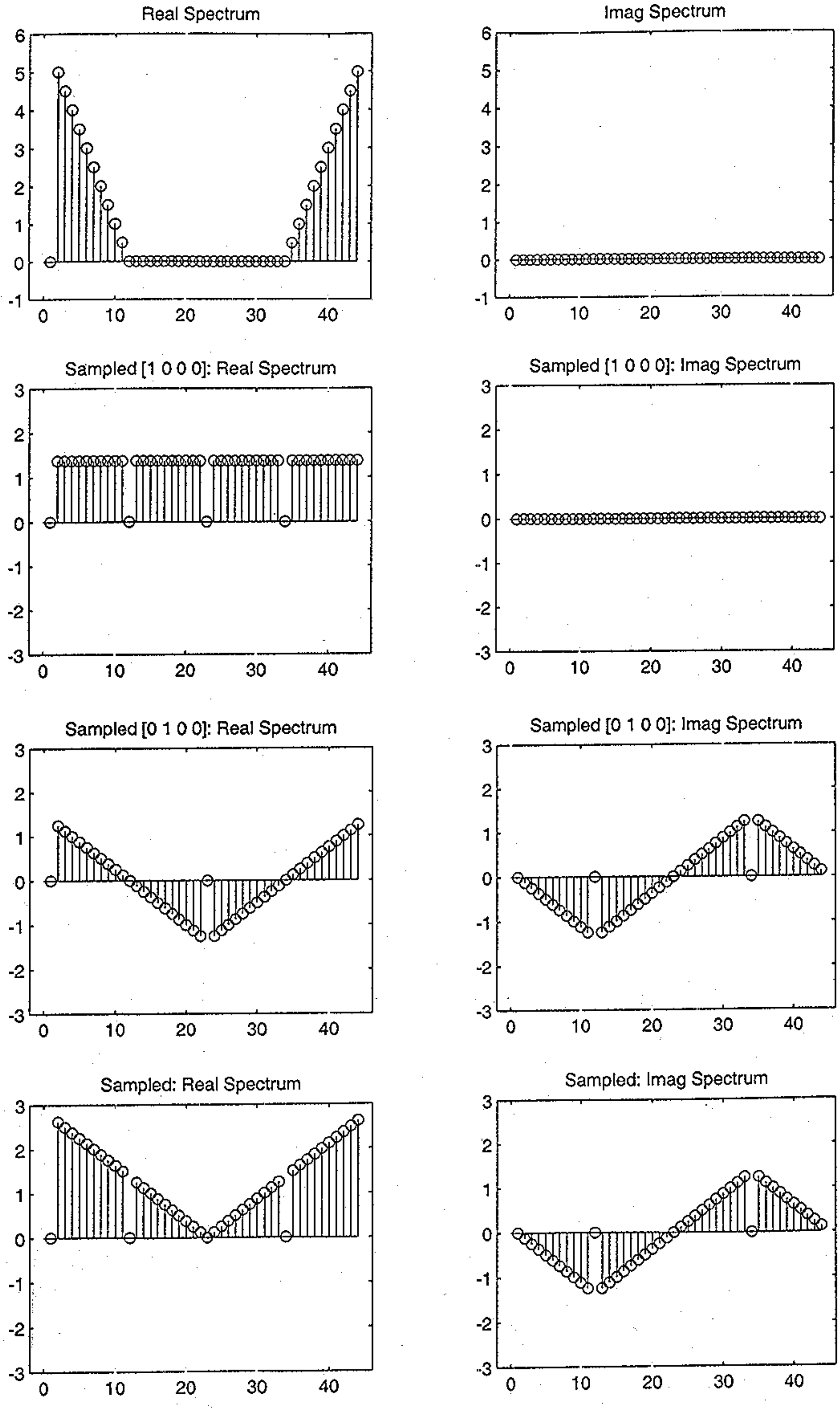
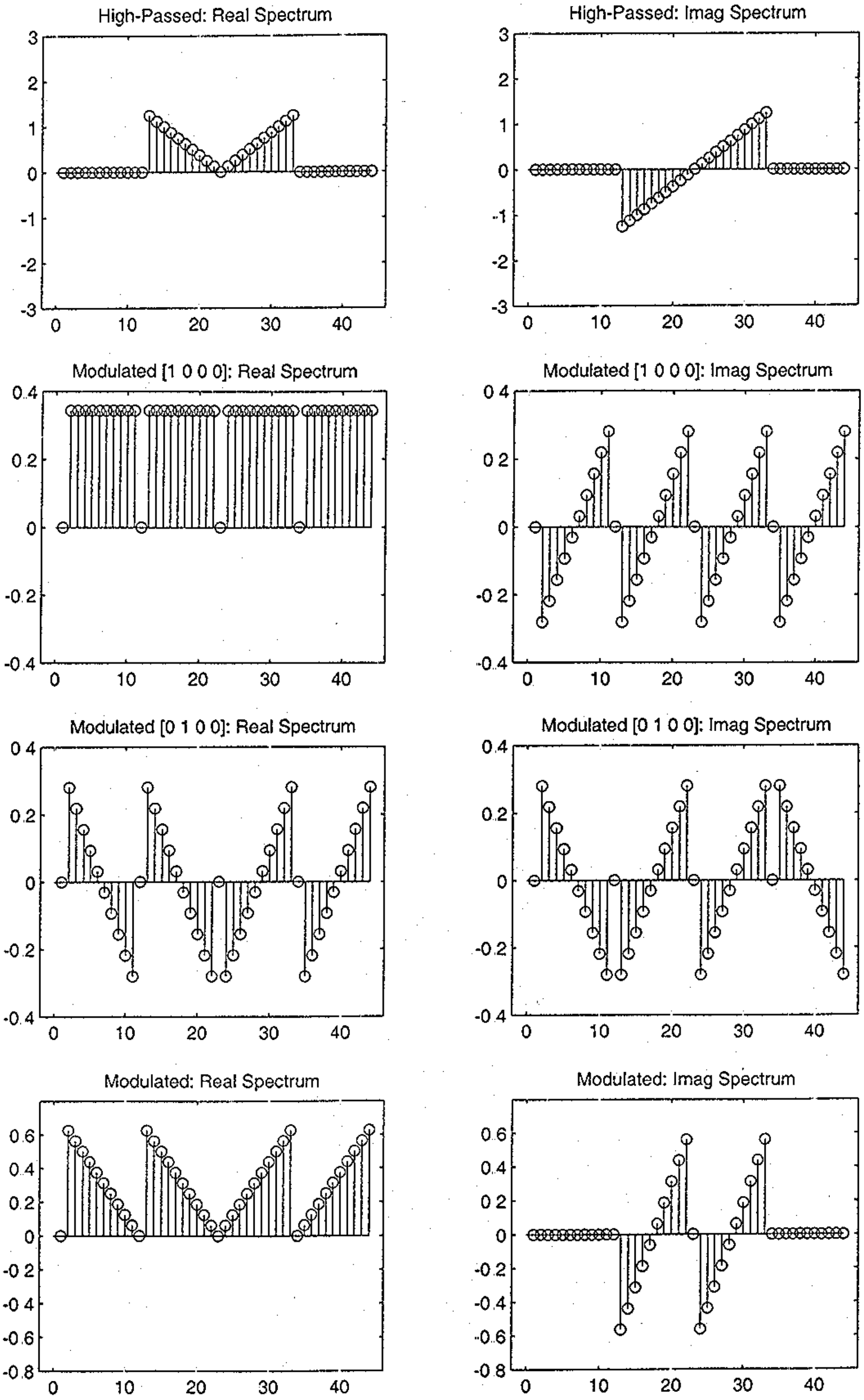


Fig. 8a Summary of sampling by [1 1 0 0]



High-Passed: Real Spectrum

High-Passed: Imag Spectrum

Modulated [1 0 0 0]: Real Spectrum

Modulated [1 0 0 0]: Imag Spectrum

Modulated [0 1 0 0]: Real Spectrum

Modulated [0 1 0 0]: Imag Spectrum

Modulated: Real Spectrum

Modulated: Imag Spectrum

Fig. 8b Summary of high-pass modulateion by [1 1 0 0]

uncommonly encountered in our bag of tricks. (Indeed, there are several ways to go from here.) The most logical modulation to try, and the one which works with the least additional fudging, is the modulation that got us here. We multiply the sequence at the bottom of Fig. 7e by the sampling sequence 1 1 0 0 1 1 0 0 ...... and this gives us the sequence at the bottom of Fig. 7f, from which the FFT gives the spectrum at the top of Fig. 7f. Much to our delight we see the original spectrum in the low-pass region. Fig. 7i shows the spectrum of Fig. 7f low-passed (again by just masking the FFT) and from this low-passed spectrum, the inverse FFT gives us the recovered original signal (we did have to scale by 4 due to the two modulations and filterings). It worked perfectly.

Because this looks a bit like magic (perhaps it is!) we might want to look at some of the details. Just as we dissected the original sampling into two phases, Fig. 7g and Fig. 7h show the two phases of the modulation. It is interesting to see how the real and imaginary parts of the two phases combine. In the case of the real part, a "bipolar" triangle is offset by a constant to give us the original spectrum. In the case of the imaginary part, the low-pass portions cancel.

For ease of viewing, Fig. 8a shows the dissection of the spectra for the original sampling, while Fig. 8b shows the dissection of the modulation following the high-pass.

Example 3 - A Toy Hold

Instructors in digital signal processing (and probably most academic subjects) need to make up situations for use as examples and as problems for homework and exams. (In fact, Examples 1 and 2 above are of this nature, as is obviously, the one coming up.) Often, by objective standards these are mere toys, yet they very often yield considerable insight. By this statement one might suppose that we are suggesting that instructors design these to provide insight to the students. Sometimes this is the case. But at least as often, the design of the example or problem provides unexpected insight first to the instructor, as the example is invented and analyzed, and not infrequently, the way students actually solve the problem adds additional perspective, often in the form of a better approach than the instructor had in mind. For the most part the students do not begin their answers with "Look stupid! All you really have to do is..........!" But it sometimes comes down to that.

In this third example (Fig. 9), we suppose that a signal is first sampled so that only even samples are kept, with odd samples set to zero. Then the zeroed samples are set to the value of the sample before them - that is, the even samples are held. Thus we have a simple length-two discrete time hold - a toy sample-and-hold if you wish. At the same time, the length-two hold is a simple low-pass filter with a zero at z=-1.

Fig. 10a shows an original triangular spectrum or width 0.2. (Here we are using the FFT for our studies, as previously, but are plotting the spectra with straight line segments so as to represent the DTFT.) Fig. 10b shows the (expected) result of the sampling procedure of setting every other sample to zero. We see a spectral replica centered at 0.5, and an amplitude scaling of 1/2. Fig. 10c shows the spectrum that follows the hold. Here a quick glance indicates that the amplitude goes back up, and the replica centered at 0.5 is greatly reduced. This is an indication of the modest low-pass reconstruction effort of the hold. Indeed, in Fig. 10c, the spectral shaping of the hold is indicated. This comes from the length 2 filter:

$$H_{hold}(z) = 1 + z^{-1} \tag{5a}$$

$$H_{hold}(f) = 1 + e^{-j2\pi f} = 2\, e^{-j\pi f} \cos(\pi f) \tag{5b}$$

Here the 2 is responsible for bringing the dc value of the spectrum back up to 2, and the null at f=0.5 is responsible for reducing the full-sized replica at 0.5 to two modest lobes centered about 0.4 and 0.6. The phase term in equation (5b) suggests that the spectrum at the hold output is likely complex, and this we see is true in Fig. 10d (note different scale on real and imaginary plots).

In our academic exercise, we would invite our audience to now reverse the spectral shaping of the hold device. This is similar to the "sin(x)/x correction" often employed as a result of sample-and-hold [Section 2d, EN#200 (20)]. The answer (one answer) would be to use the filter that is inverse to equation (5a):

$$H_{undo\text{-}hold}(z) = 1/(1 + z^{-1}) \tag{6a}$$

$$H_{undo\text{-}hold}(f) = e^{j\pi f} / [2 \cos(\pi f)] \tag{6b}$$

This is doable, as shown in Fig. 9. [Note however that because of the sampling by 2, the whole device is not time invariant. Further, the inverse filter has a pole at z=-1, on the unit circle, so we would need to take care to see that it was initialized properly. But clearly, from the time domain, it does work.] We obtain our sampled sequence back from the held version. The cascade of the hold with its inverse is just a path of 1.

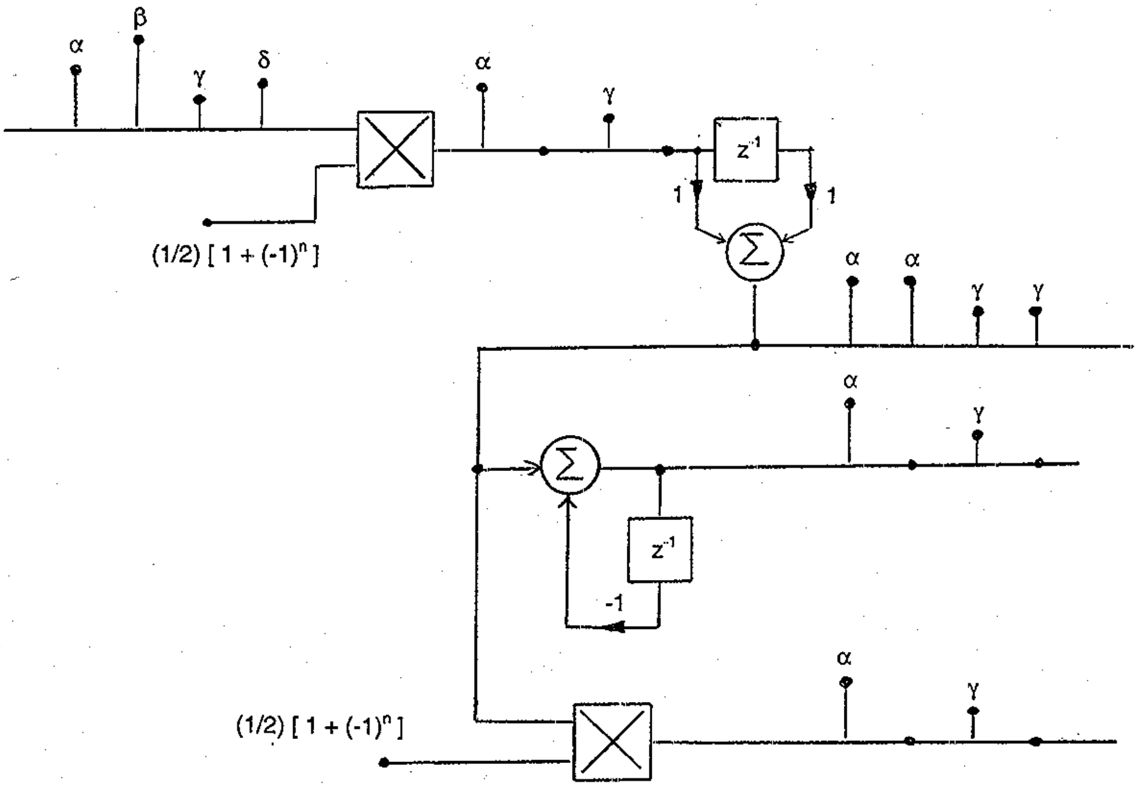


Fig. 9 Discrete length-two hold, and two ways of undoing hold.

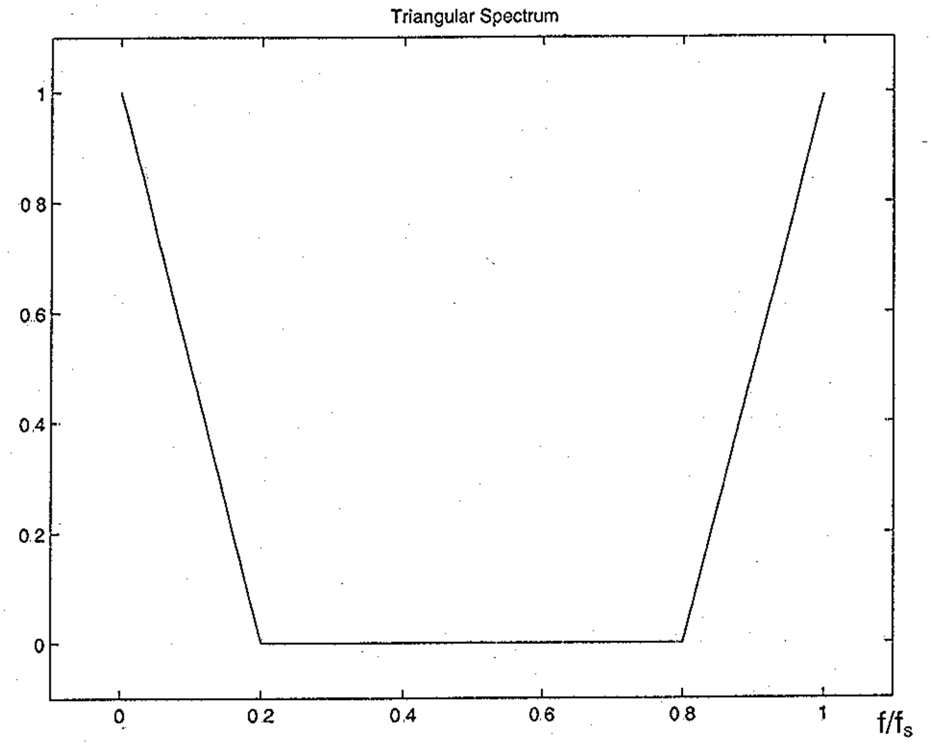


Fig. 10a Original triangular spectrum

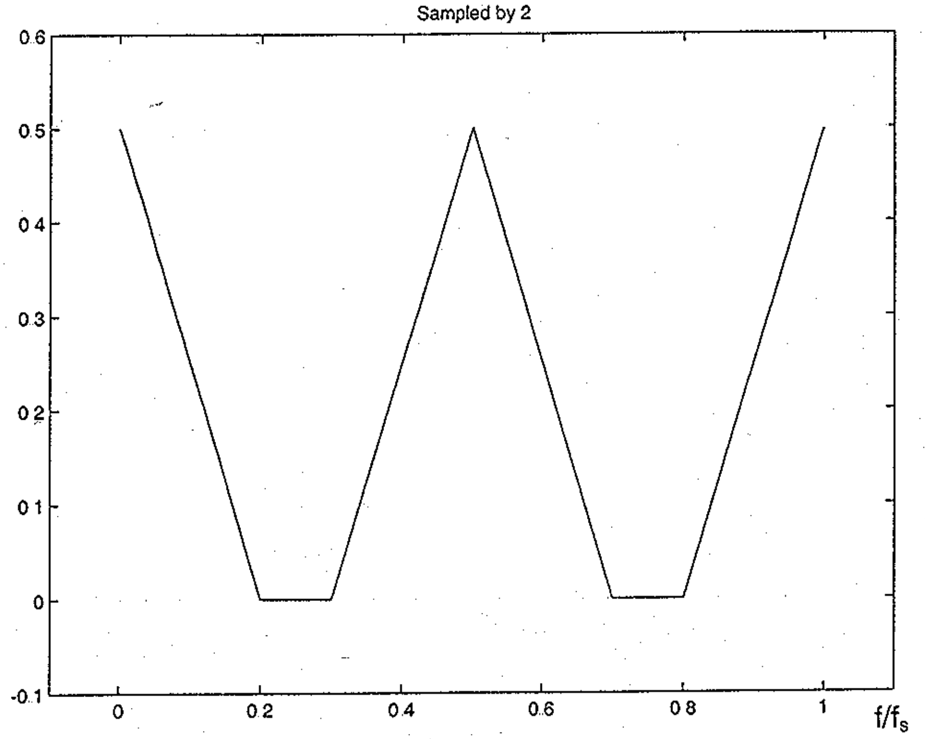


Fig. 10b Sampled triangular spectrum

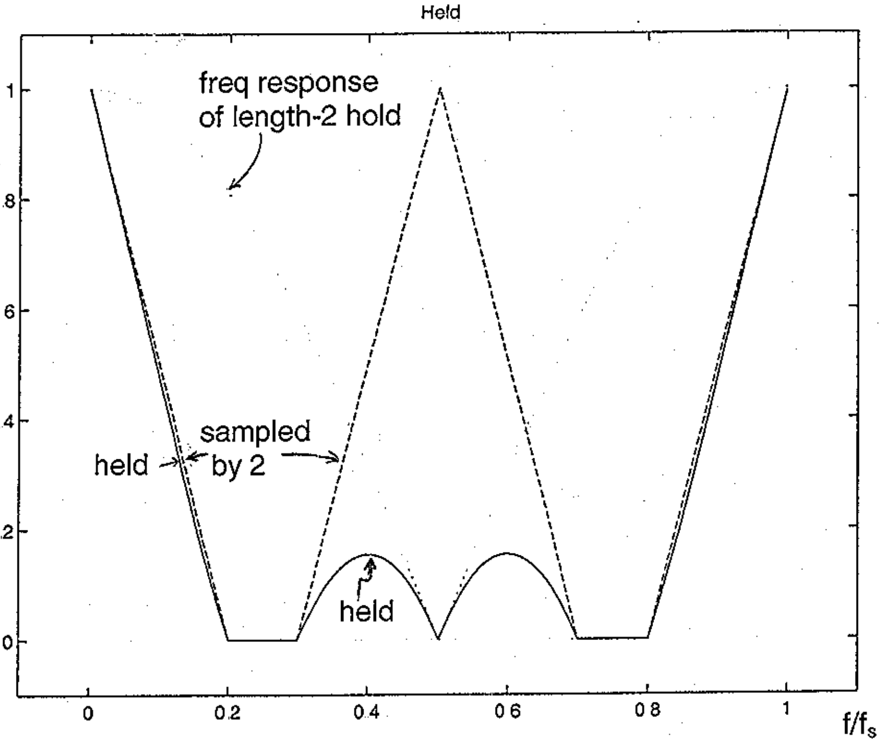


Fig. 10c Held version of sampled triangular spectrum

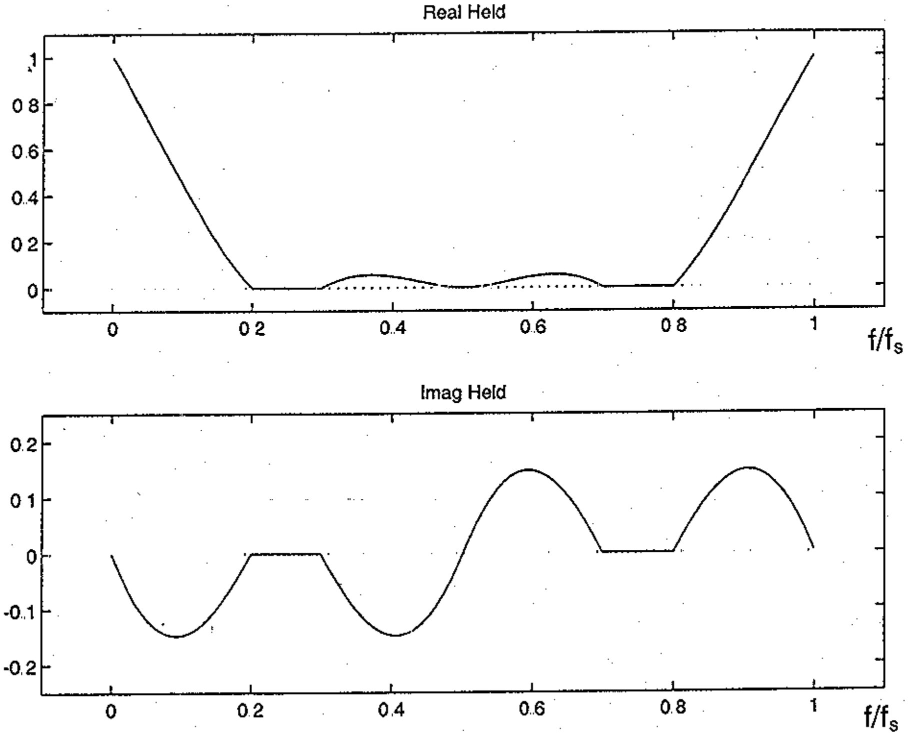


Fig. 10d Real and imaginary parts of held spectrum

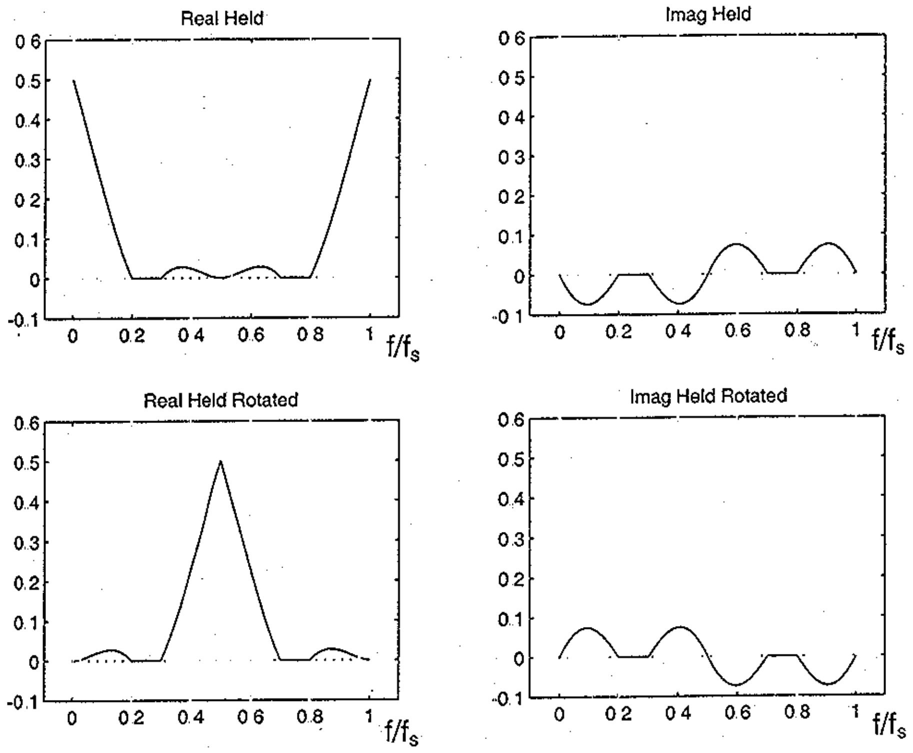


Fig. 11 Real and imaginary parts of held and rotated held spectra.

Now, someone might well point out that a simpler way of getting the sampled sequence back from the held sequence would be to re-sample the held sequence (bottom portion of Fig. 9). Obviously this works too, and we need to show that the held spectrum returns. It is not obvious that a spectrum as twisted and rounded (and complex in the mathematical sense) as the one in Fig. 10c and Fig. 10d can be added to a rotated version of itself to give a spectrum of straight line (and purely real!) pieces (Fig. 10b back). Of course, from the time domain, we see that this must be true, so our interest is only in how it does happen.

Just as the sampling by 2 (multiplication by (1/2)[1 + $(-1)^n$]) gave us a rotated, added replica in the original sampling (Fig. 10a becoming Fig. 10b), the multiplication by (1/2)[1 + $(-1)^n$] at the bottom should rotate and add the output of the hold. Fig. 11 shows what is going on. The top two plots repeat Fig. 10d, showing the spectrum at the output of the hold. The bottom two plots in Fig. 11 show rotated versions. It is clear that in summing the bottoms with the tops, that the imaginary parts cancel. Further, the small "bumps" in the real images correspond to the deviation of the larger "near-triangles" from actual triangles. Summing the real parts of Fig. 11 return Fig. 10b.

****

ELECTRONOTES, Vol. 21, No. 201, February 2002
Published by B. Hutchins, 1016 Hanshaw Rd., Ithaca, NY 14850 (607)-257-8010