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

APPLICATION NOTE NO. 352

May 2002

PHASE ERRORS, AMPLITUDE ERRORS, AND UNWANTED SIDEBANDS IN FREQUENCY SHIFTERS

INTRODUCTION

The more or less standard frequency shifter design [1] used in audio works is essentially the same structure as the single-sideband modulator used in communications. The device is based on the production of quadrature phases of two signals: a program signal and a shifting signal (Fig. 1). The multipliers are essentially "balanced modulators" which are double sideband devices. By taking the sum and differences of the multiplier outputs, the upper and lower sidebands are separated. This is clearly shown by the trig functions written over Fig. 1. Note however that the exact result shown depends on having quadrature pairs that are exact: exactly 90° out of phase, and which have the exact same amplitude.

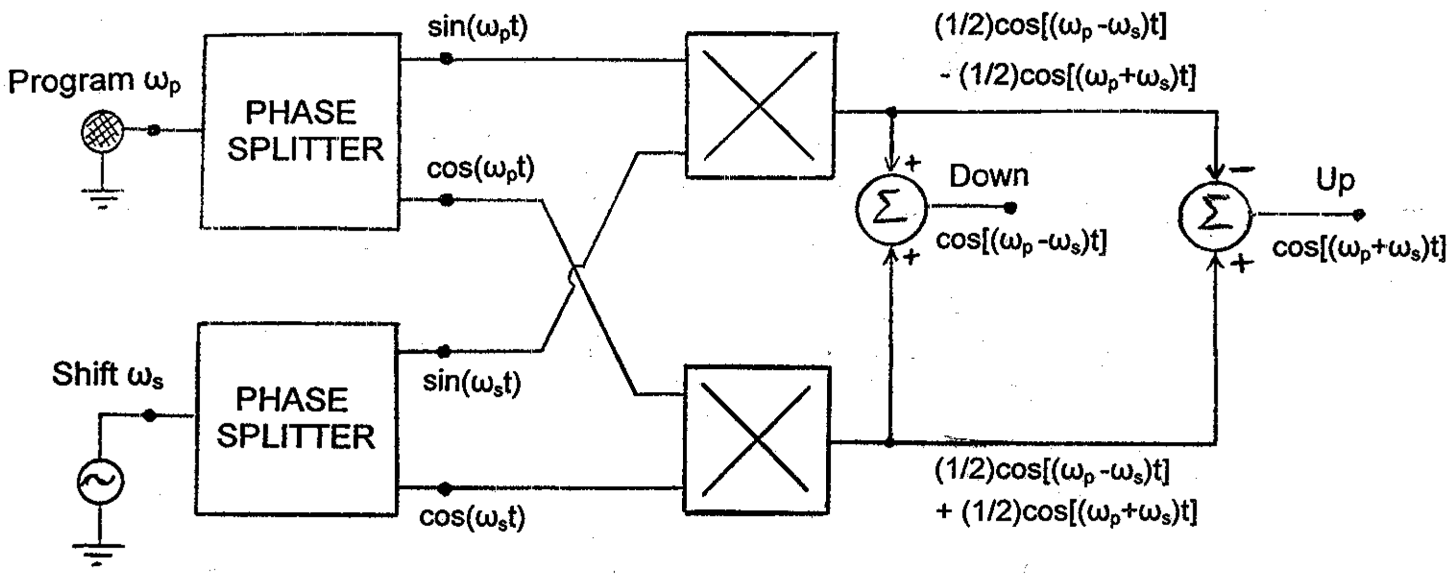

Fig. 1 Standard Frequency Shifter

In practice, it is seldom the case that exact quadrature is obtained. In cases where signals to be processed are of diverse variety, and generally broadbanded, the devices which separates the signals into quadrature components usually have well-specified but limited capabilities. Perhaps surprisingly, it is the usual case that one or the other of phase, or amplitude, is actually very near to being perfect by default. In an analog context, the phase splitters can be formed from all-pass networks with flat amplitude over very wide ranges of frequency. The 90° difference however is approximated by the difference of two all-pass cascades which have phase responses staggered with respect to each other. These are usually called 90° phase differencing networks (PDN), and they approximate a 90° phase difference to within defined tolerances over specified bands.

In the discrete-time case however, the 90° phase difference is easily made to be perfect, (by odd symmetry in an FIR digital filter) and it is the amplitude response that is not flat, being only an approximation to a constant over a range of frequency. These networks are usually called Hilbert transformers (HT) [2]. Figure 2 summarizes the two cases.

[It should perhaps be mentioned before going on that in cases where a signal involved in the frequency shifter is not broadbanded, but rather is perhaps just a single frequency, achieving a perfect quadrature pair may be both trivial and cheap. Single all-pass sections, pure delays, and oscillators with multiple phase outputs are among the options when a broadbanded 90° PND or HT is not required.]

CALCULATING THE UNWANTED SIDEBAND

Since in the general case we expect amplitude errors, phase errors, or possibly both in our broadband phase shifter networks, it is the purpose of this note to discuss how one calculates the degraded performance in terms of the amount of unwanted sideband that corrupts the wanted sideband. This we accomplish easily by assuming specific errors and overwriting our basic diagram with the revised trig calculations. The theoretical equations are then easily verified by calculations of test cases using Fast Fourier Transforms.

The easiest of the errors to analyze is the amplitude error, and this is shown in Fig. 3. Here the frequency shifter for the program channel is assumed to have a different amplitude (A) for the sine output, as opposed to an amplitude of 1 for the cosine. In general, we expect A to be a function of frequency - not a fixed constant. So the worse-case amplitude error might be used to find the worse case of unwanted sideband. It is easy to calculate that the downshifted output is:

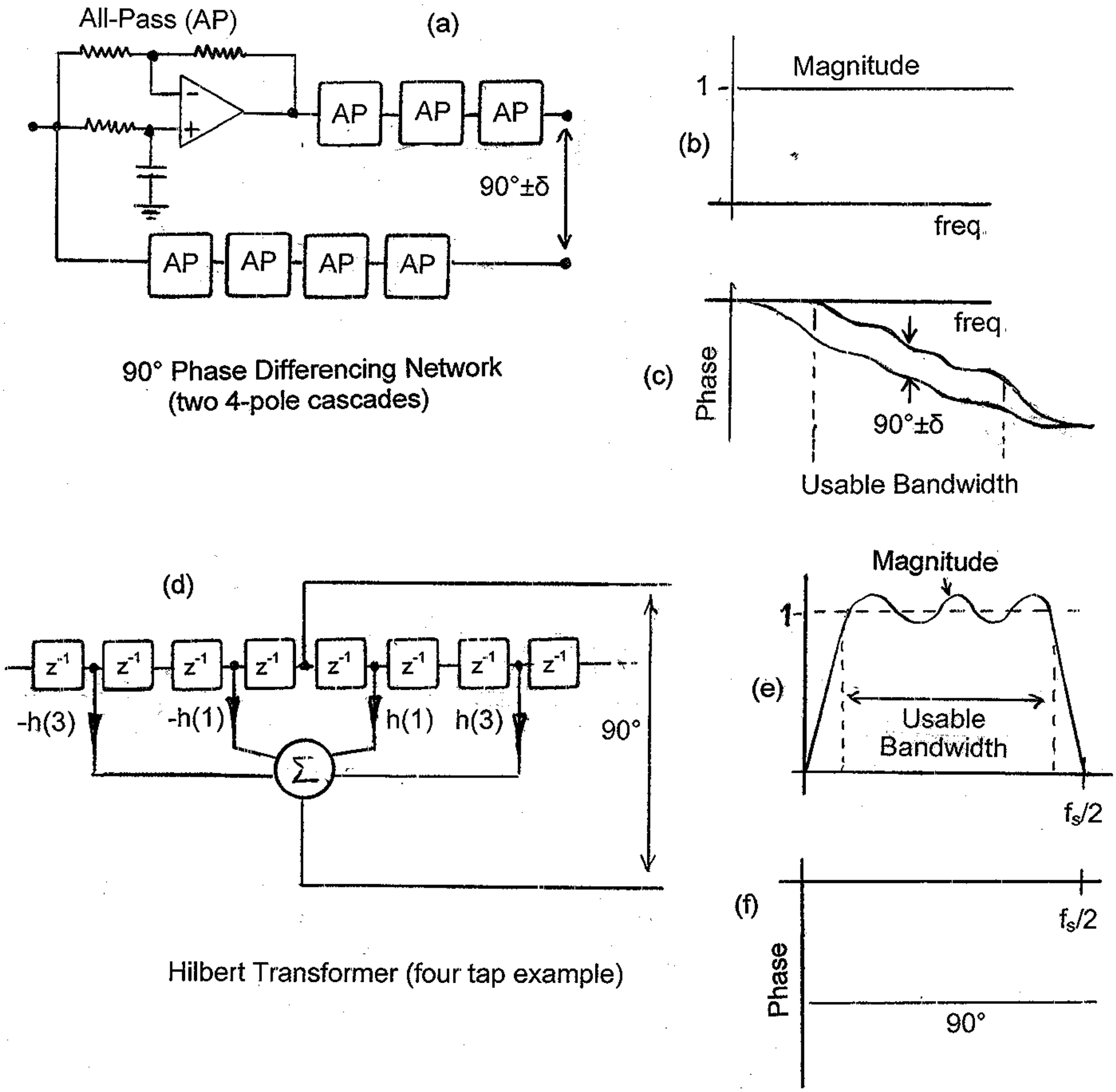

Fig. 2 A 90° PDN (a) is comprised of a series connection of analog all-pass networks (flat frequency response magnitude as in b) staggered so that a phase difference of 90° is maintained to within some tolerance δ over a suitable range (c). The Hilbert transform approach (d) approximates a unity gain (e), but the phase response is exactly 90° (f). Thus the analog approach has a "perfect" magnitude response while the digital approach has a "perfect" phase response.

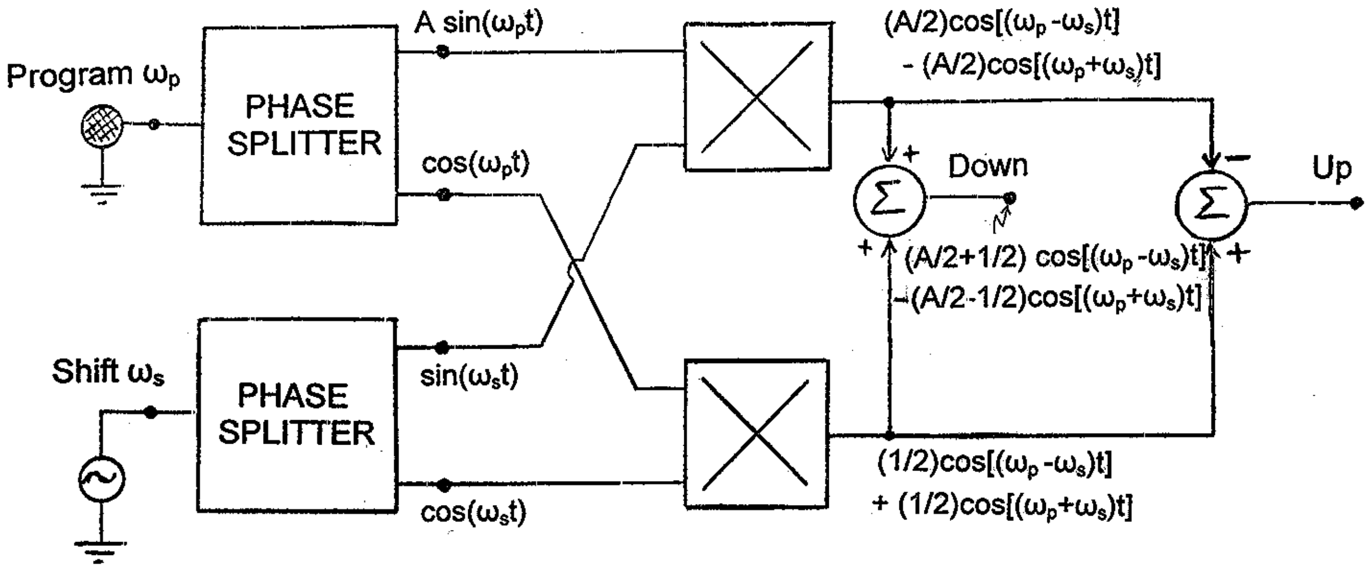

Fig. 3 Frequency Shifter with Amplitude Error

$$d(t) = (A/2 + 1/2)\cos[(\omega_p-\omega_s)t] - (A/2 - 1/2)\cos[(\omega_p+\omega_s)t] \quad (1)$$

from which we note that the ratio of the unwanted sideband (the sum frequency) to the desired sideband (the difference frequency) is:

$$u_1 = (A-1)/(A+1) \quad (2)$$

This is the sort of error we expect with a digital filter approach.

In a second case, we want to look at a phase error (Fig. 4) where we assume the sine output of the program signal is not a pure sine, but rather has an additional phase δ. As was the case with amplitude, δ is likely a function of frequency, and again, we look for worse-case errors. In fact, we have noted that such phase error is likely to be found in an analog PDN, and most PDN design procedures yield a performance specification giving the maximum expected phase error. In Fig. 4, the two components of the downshift d(t) that correspond to the difference frequency are:

$$d_1(t) =(1/2)\cos[(\omega_p-\omega_s)t + \delta] + (1/2)\cos[(\omega_p-\omega_s)t]$$

$$= \cos(\delta/2)\cos\{(\omega_p-\omega_s)t + \delta/2] \quad (3a)$$

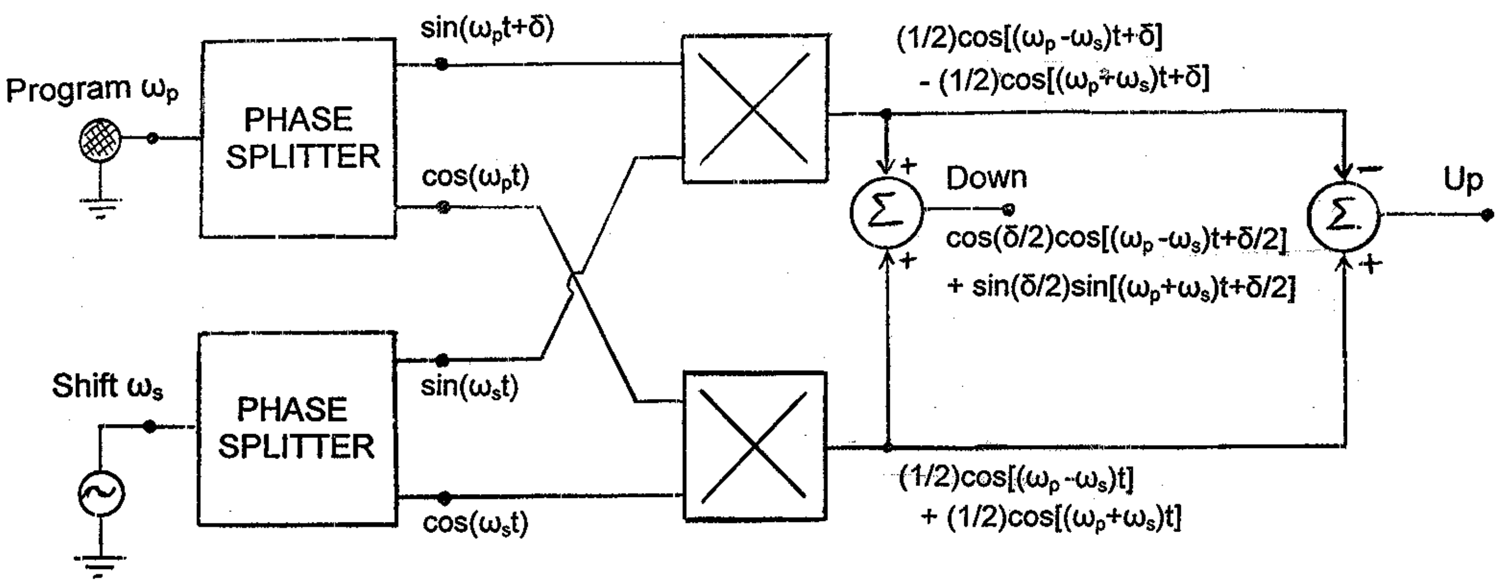

Fig. 4 Frequency Shifter with Phase Error

where we have use the trig identity for the sum of cosines. A very similar calculation gives the sum frequency (unwanted sideband) at d(t) as:

$$d_2(t) = (1/2)\cos[(\omega_p+\omega_s)t] - (1/2)\cos[(\omega_p+\omega_s)t + \delta]$$

$$= \sin(\delta/2)\sin[(\omega_p+\omega_s)t + \delta/2] \qquad (3b)$$

and then

$$d(t) = d_1(t) + d_2(t) \qquad (4)$$

Note that the angle $\delta/2$ appears in four places in d(t). The phase shifts $\delta/2$ in the cosine and sine that are time varying are not of much interest. But the $\cos(\delta/2)$ and $\sin(\delta/2)$ that determine the amplitudes are exactly what we need to use. We see that the ratio of undesired sideband (upshift) to desired (downshift) is:

$$u_2 = \sin(\delta/2)/\cos(\delta/2) = \tan(\delta/2) \qquad (5)$$

which is the measure used for analog shifters [1].

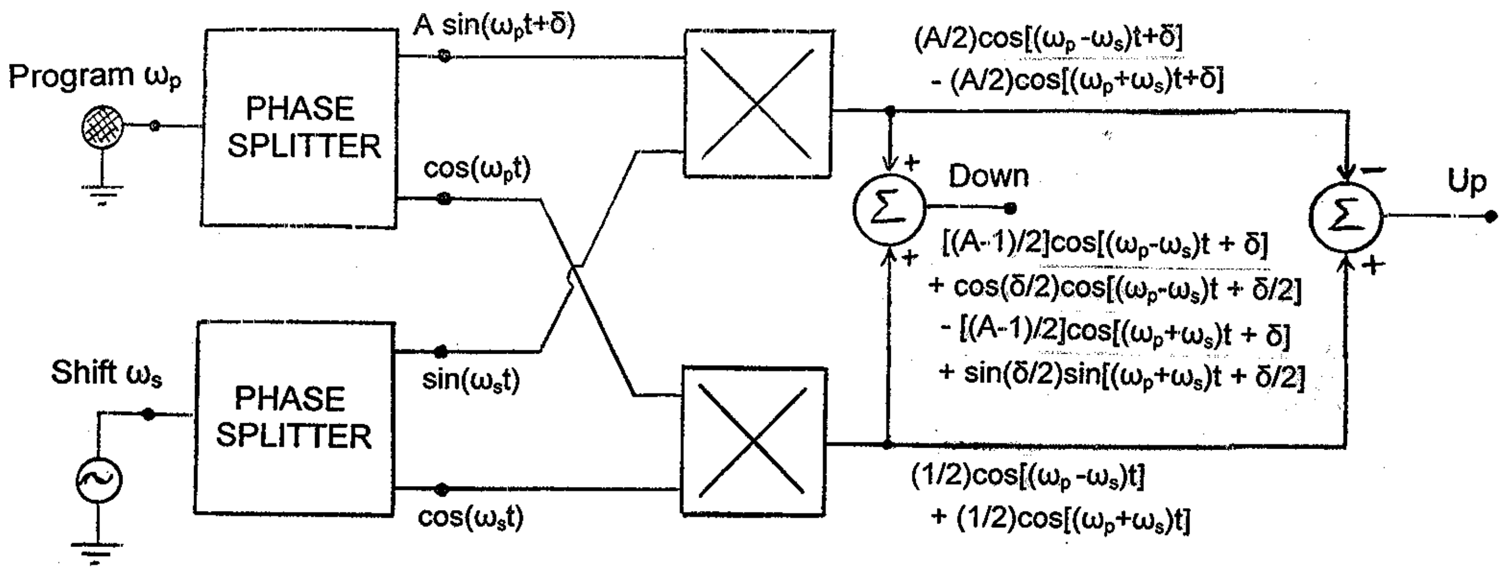

Fig. 5 Frequency Shifter with Amplitude and Phase Errors

Although we have noted that many applications will have only one of the two errors (amplitude or phase), it is not overly difficult to calculate the unwanted sideband ratio for the case where both errors occur. This is seen in Fig. 5. At the downshift output, the downshifted component is represented by:

$$d_1(t) = (A/2)\cos[(\omega_p-\omega_s)t + \delta] + (1/2)\cos[(\omega_p-\omega_s)t] \qquad (6)$$

which is not something we can sum with the sum of cosine trig identity. But we can rearrange it and use equation (3a) to get:

$$d_1(t) = [(A-1)/2]\cos[(\omega_p-\omega_s)t + \delta] + (1/2)\cos[(\omega_p-\omega_s)t+\delta] + (1/2)\cos[(\omega_p-\omega_s)t]$$

$$= [(A-1)/2]\cos[(\omega_p-\omega_s)t + \delta] + \cos(\delta/2)\cos[(\omega_p-\omega_s)t + \delta/2] \qquad (7)$$

We note that $d_1(t)$ consists of two cosines at the difference frequency, with amplitudes $a=(A-1)/2$ and $b=\cos(\delta/2)$ which have a phase difference of $\delta/2$. The amplitude of the sum is thus obtained by the "law of cosines" as:

$$r_1 = [\, a^2 + b^2 - 2ab\cos(\pi-\delta/2)\,]^{1/2} \qquad (8)$$

At the downshifted output, the upshifted component is given by (also using equation 3b):

$$d_2(t) = (1/2)\cos[(\omega_p+\omega_s)t] - (A/2)\cos[(\omega_p+\omega_s)t+\delta]$$

$$= (1/2)[(\omega_p+\omega_s)t] - (1/2)\cos[(\omega_p+\omega_s)t+\delta] - [(A-1)/2]\cos[(\omega_p+\omega_s)t + \delta]$$

$$= \sin(\delta/2)\sin[(\omega_p+\omega_s)t + \delta/2] - [(A-1)/2]\cos[(\omega_p+\omega_s)t + \delta] \quad (9)$$

Here we have a sine and a cosine with added phase δ. Now taking $c=\sin(\delta/2)$ as the amplitude of the sine component and $b=(A-1)/2$ as before, again using the law of cosines:

$$r_2 = [\, c^2 + b^2 - 2cb\cos(\pi/2 + \delta/2)]^{1/2} \quad (10)$$

Finally, this yields the undesired sideband ratio for both errors as:

$$u_3 = r_2/r_1 \quad (11)$$

These same equations derived here for the downsampled output also apply to the upsampled output.

A TEST PROGRAM

We find it useful to write a Matlab program for several purposes: to evaluate the various equations above, to simulate the frequency shifter, and to plot some example figures. The program below does these things. We can think of the simulation as an "experimental" verification of our theoretical calculations above. The program takes as input an amplitude error A and a phase error del. If A=1 and del=0 we have the no error case. A value of A of 1.1 would be an example of an amplitude error and a value of del of 0.05 (meaning radians) would be a phase error.

The program first generates portions of the sequences representing the outputs of the phase splitters, and these are multiplied and added (downshift). The FFT is then taken of the output sequence, and the magnitude for the upshift (unwanted) sideband frequency is divided by the magnitude of the downshift (desired) sideband frequency. For this test, the program frequency is taken to be 50 while the shift frequency is taken to be 20, so the downshift is 30 while the upshift is 70. The experimental value for the unwanted sideband is u. This we compare with values of u1, u2, and u3 which are calculated from A and del. The value u3 always agrees with u, while u1 agrees with u when there is only an amplitude error, and u2 agrees with u when there is only a phase error. Thus the derived equations are verified.

The program also plots example waveforms. Fig. 6 shows plots for a no-error case, while Fig. 7 shows a case where A=1.1 and del=0.1 (about 6 degrees).

```
function [u,u1,u2,u3]=usb(A,del)
% function [u,u1,u2,u3]=usb(A,del)
% unwanted upper sideband of frequency shifter
%
% A = amplitude relative to 1
% del = phase error (radians)
%
% u  = unwanted "experimental"
% u1 = calculated based on amplitude error only
% u2 = calculated based on phase error only
% u3 = calculated total error
% B. Hutchins                    Fall 2001

n=0:599;
% program signals
xps=A*sin(2*pi*50*n/600 +  del);
xpc=cos(2*pi*50*n/600);
% shifting signals
xss=sin(2*pi*20*n/600);
xsc=cos(2*pi*20*n/600);
% multipliers
ms=xps.*xss;
mc=xpc.*xsc;
% downshift is sum
down=ms+mc;
up=-ms+mc;

n=0:50
figure(1)
subplot(221)
plot(n,xps(1:51))
axis([-5 55 -1.2 1.2]);
title('program sine')
subplot(222)
plot(n,xpc(1:51))
axis([-5 55 -1.2 1.2]);
title('program cosine')
subplot(223)
plot(n,xss(1:51))
axis([-5 55 -1.2 1.2]);
title('shift sine')
subplot(224)
plot(n,xsc(1:51))
axis([-5 55 -1.2 1.2]);
title('shift cosine')
figure(2)
subplot(221)
plot(n,ms(1:51))
axis([-5 55 -1.2 1.2]);
title('multiply sines')
subplot(222)
plot(n,mc(1:51))
axis([-5 55 -1.2 1.2]);
title('multiply cosines')
subplot(223)
plot(n,down(1:51))
axis([-5 55 -1.2 1.2]);
title('downshift')
subplot(224)
plot(n,up(1:51))
axis([-5 55 -1.2 1.2]);
title('upshift')

D=abs(fft(down))/300;
figure(3)
stem([0:1:100],D(1:101))
axis([-5 105 -.2 1.2]);
% "experimental"
u=D(71)/D(31)
% calculated due to amplitude error
u1=abs((A-1)/(A+1))
% calculated due to phase error
u2=tan(del/2)

a=cos(del/2);
b=(A-1)/2;
r1s=a^2+b^2-2*a*b*cos(pi-del/2);
r1=sqrt(r1s);
c=sin(del/2);
r2s=c^2+b^2-2*c*b*cos(pi/2+del/2);
r2=sqrt(r2s);
% calculated due to total error
u3=r2/r1


figure(3)
```

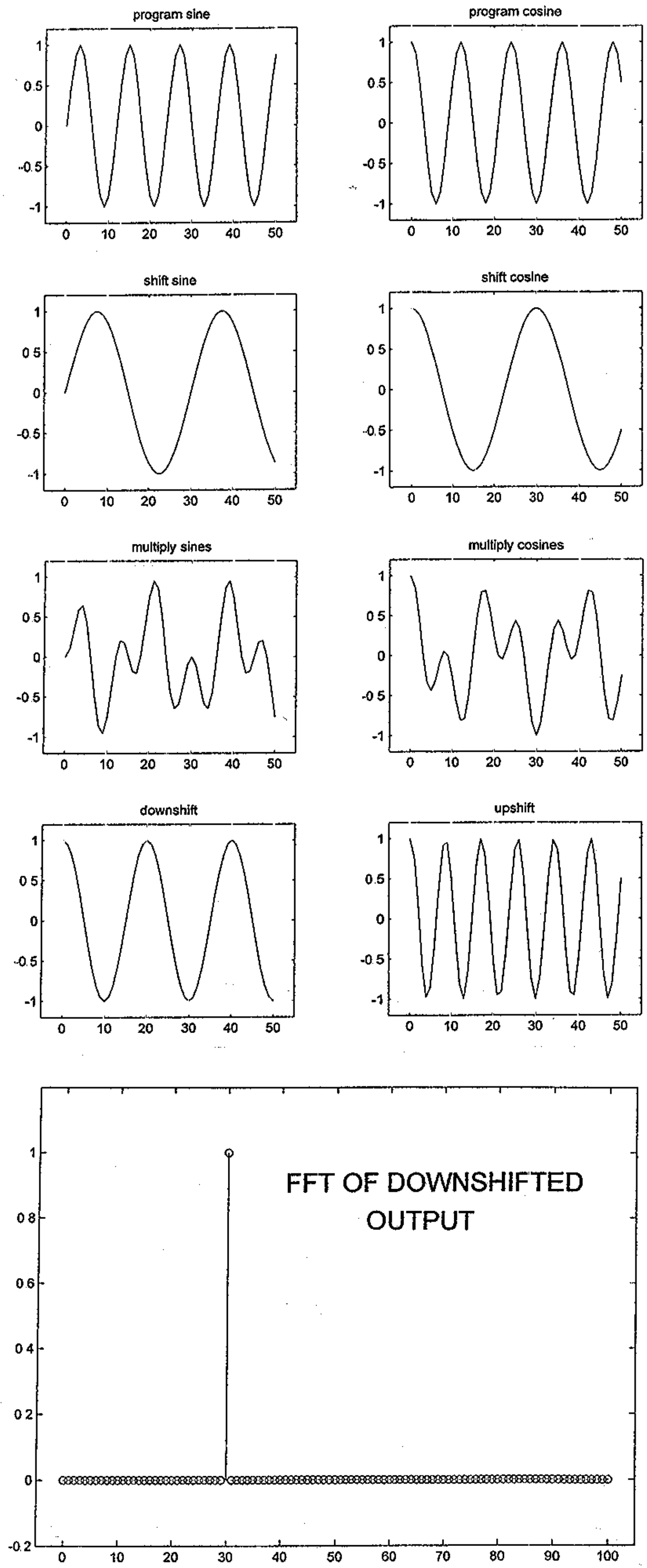

Fig. 6 Here we see the results of running the Matlab program for no errors (A=1, del=0). The program frequency is 50 while the shift frequency is 20. The eight relevant waveforms are shown. At the bottom, the magnitude FFT of downshift sequence shows the perfect downshift at a frequency of 30.

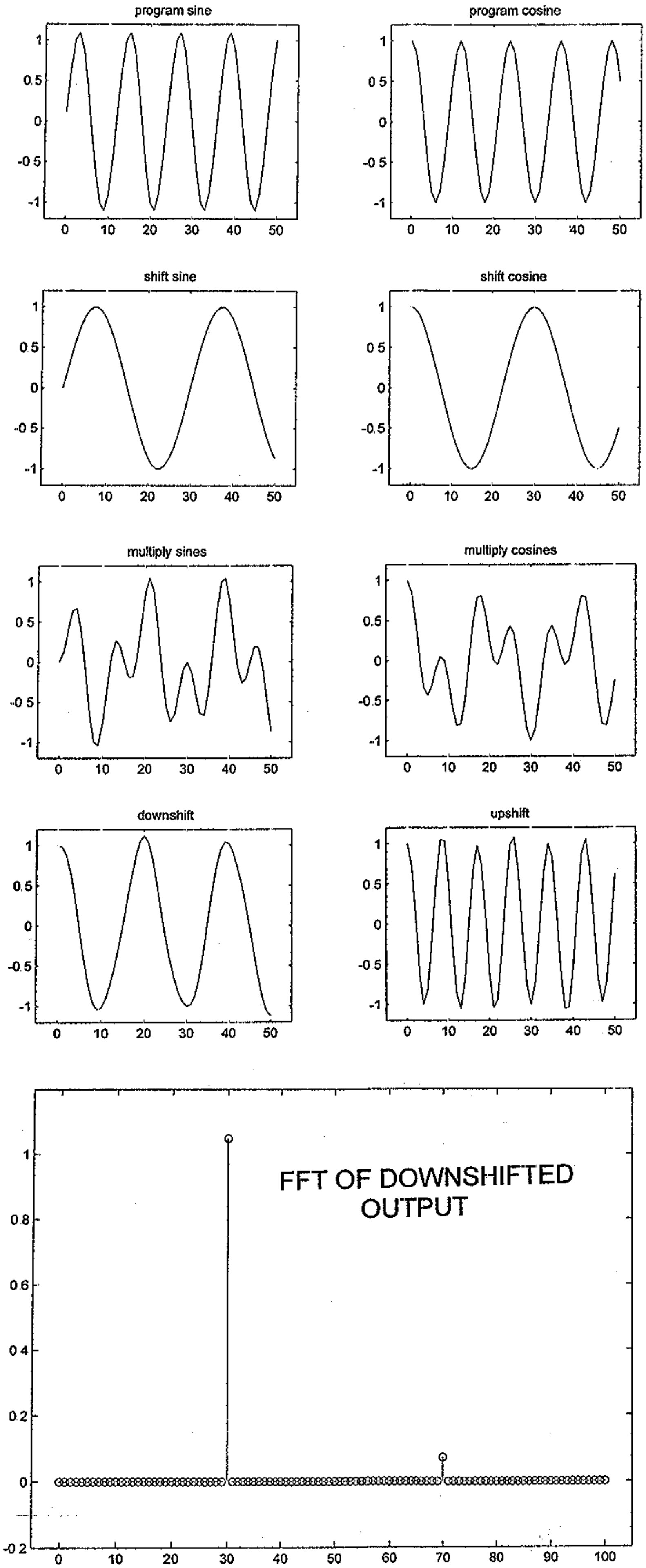

Fig. 7 Here A=1.1 and del = 0.1 (about 6°). We see waveforms that are fairly similar to those in Fig. 6, although there are slight differences that are apparent even in the time domain. In particular, there are some unusual "bending" in the upshifted and downshifted waveforms. The FFT shows the down-shift as expected (at 30) but also a small upshifted component at 70. The unwanted sideband ratio is 0.0691. With only the amplitude error, or only the phase error, the ratios are 0.0476 and 0.0500 respectively

TRYING IT WITH ±45 DEGREES

Something that is sometimes tried, which looks like an astoundingly good idea at first, is to try to get a 90° degree phase difference by using filters giving +45° degrees and -45° degrees instead of 0° degrees and 90° degrees. The thought is that the shifters that give +45° and -45° degrees of shift have identical amplitude responses, which is true. This means that there is no error due to amplitude error.

The problem comes up in thinking that the phases of exactly +45° or -45° degrees can come about by mixing exactly 0° degrees (center tap) and exactly 90° degrees (Hilbert transformer) each weighted by $1/2^{1/2}$ (Fig. 8). This would occur only if A = 1 of course, in which case we would already have a perfect Hilbert transformer. But is there perhaps some improvement at least? It seems not.

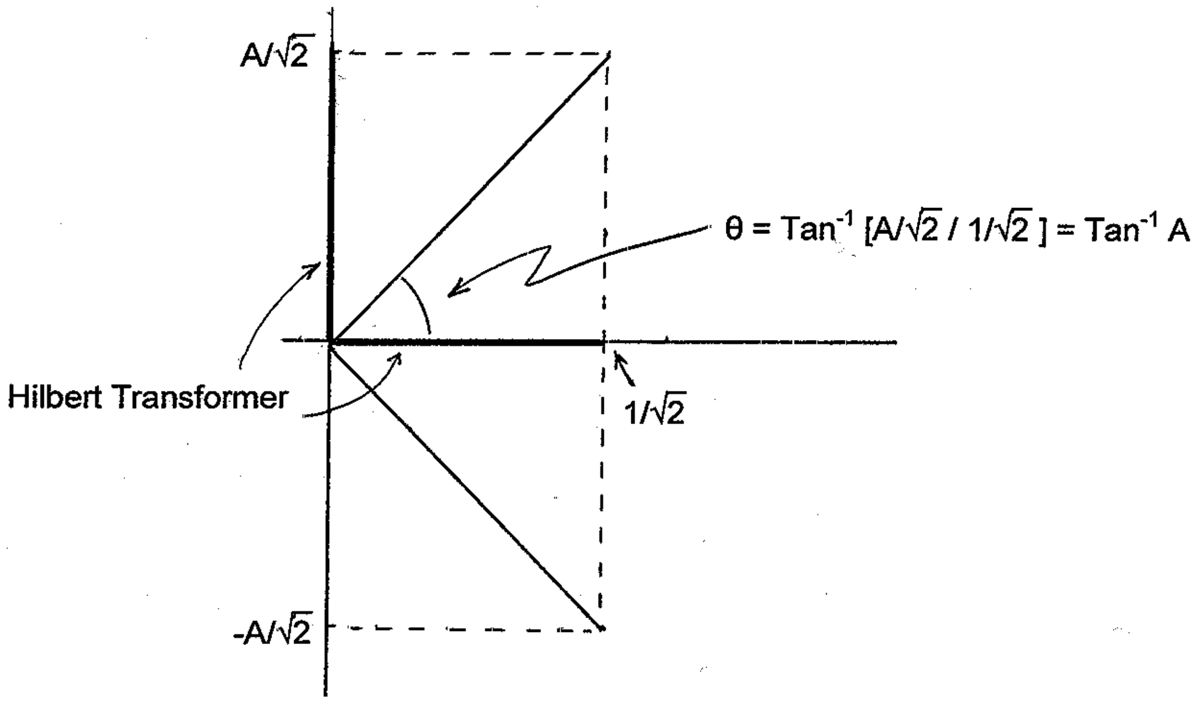

Fig. 8 Attempting to get Good Quadrature with ±45 Degrees!

From Fig. 8 we note that the angle Θ is $\tan^{-1}A$, not exactly 45 degrees. That is, the non-unity amplitude A (which is also a function of frequency) now causes a phase error. It's not going to let us win. In fact, the phase error (difference from 90 degrees) is:

$$\delta = 2(\pi/4 - \tan^{-1}A) \qquad (12)$$

and we know from equation (5) that this phase error results in an unwanted sideband of:

$$u_4 = \tan(\delta/2) = \tan(\pi/4 - \tan^{-1}A) \quad (13)$$

Using the trig identity for the tangent of a difference we get:

$$u_4 = [\tan(\pi/4) - \tan(\tan^{-1}A)] / [1 + \tan(\pi/4)\tan(\tan^{-1}A)] = (1-A)/(1+A) \quad (14)$$

This is just equation (2) coming back to us. So we see that the amplitude error A, through creating a phase error, gives us exactly the same unwanted sideband that we would have had with the original amplitude error.

REFERENCES

[1] B. Hutchins, Musical Engineer's Handbook, Chapter 6a, "Frequency Shifter Design," Electronotes (1975)

[2] B. Hutchins, "An Introduction to Hilbert Transform Theory and Applications," Electronotes, Vol. 14, No. 134, Feb. 1982, pp 7-17