# A Digital Class-D Single-Sideband Transmitter

*__Graham Naylor__ has discovered a novel method of SSB synthesis that makes use of digital techniques, for which he has received the 2006 CREG award*. Together with __David Gibson,__ he describes how it works.*

This article describes a novel method of generating a single-sideband modulated carrier for a cave radio transmitter. The method is suitable for implementation in a embedded digital system, with a digital switched-mode (class D) output. The use of Class D allows the construction of a simple power amplifier, which is efficient, cheap and easily extendable to high power.

## Generating the SSB Signals

There are several analogue methods of generating a single-sideband (SSB) modulated carrier. The HeyPhone uses the so-called 'phasing' method, illustrated in **Figure 3**.

The method proposed in this article is digital and is not merely the copy of an analogue method in a digital domain. Instead of re-modulating the *I* and *Q* signals with quadrature carriers, values of instantaneous amplitude and phase are calculated from I and *Q* using a cordic (a software algorithm that converts rectangular co-ordinates to polar) and are then used to separately modulate the amplitude and phase of a digital carrier, which would not be possible if the amplitude and phase were not separated in this way. See **Figure 4** and box: **How the Cordic Method Works** on the next page.

Both the 'phasing' filter and the cordic would operate at audio sampling rates (between 6 and 12kHz) and could therefore be implemented on a processor such as a dsPIC. Similarly the phase-modulated and pulse-width-modulated signal for the class D output could, in principle, be generated by the PWM stage of the dsPIC by modifying the counter registers, but the preference may be for an implementation in an FPGA, which, at under £10, is now very affordable.

## A Class D Amplifier

**Figure 1** shows a class-D amplifier built around a MOSFET H-Bridge. This particular example is a trimmed-down version of Beat Heeb's PA stage for his underground text messaging system. A filter network between the outputs of the H-bridge (not shown) would act as a low-pass filter so as to present only the first harmonic of the rectangular pulses to the output. The phase of the output sine wave is determined by the phase (i.e. delay) of the pulses provided to each half of the bridge and the amplitude of the output is determined by the width of the pulses. The two drivers operate in anti-phase.

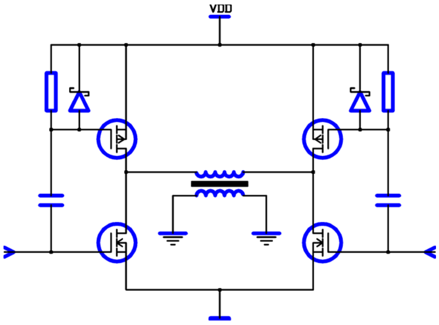

***Figure 1 – H-Bridge for Class D Output***
*This diagram shows the basic components of a typical MOSFET H-bridge, that might be used to drive an earth-current transmitter with a class D signal. It is based on the design by Beat Heeb.*

## FPGA Simulation

Field-programmable Gate Array design software was used to simulate the modulation of a 1kHz sine wave on to the upper sideband of an 87kHz carrier using this method.

**Figure 2** shows a spectral plot of the carrier (ch. 1, solid line) and the generated SSB signal (ch. 2, dotted line) displaced by 1kHz. The modulated output sine wave has a purity of about 40dBc measured with an equivalent resolution bandwidth of 1kHz, which is nothing to write home about, but certainly sufficient for readable voice communications. Some spurious lines are present at around –30dB that represent a negligible fraction of the output power and which would be undetected by the receiver. This non-ideal performance is due to small glitches in the class D signal calculation as the calculated phase wraps from –π radians to +π radians but it is certainly adequate. Note that there is no spectral line in the unwanted sideband.

From the results of the above simulation, it can be concluded that, by driving a MOSFET H-Bridge as an r.f. output stage for an SSB transmitter, a high power output is possible with optimal efficiency and at low cost. The signals to drive the class D stage can easily be generated digitally.

A detailed explanation of the modulation method is given on the next two pages.

## Further Reading

Naylor, Graham (2005a), *Using FPGAs for Digital Filtering*, CREGJ **59**, pp17-18.

Naylor, Graham (2005b), *Development of an FPGA-Based Cave Radio*, CREGJ **62**, pp4-5.

Heeb, Beat (2004) *Underground Text Messaging,* CREGJ **57**, pp4-7.

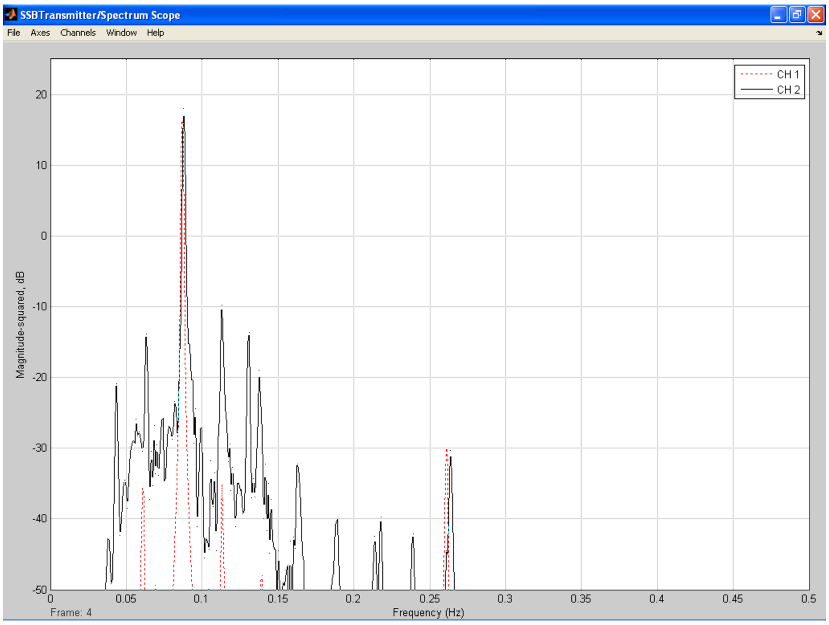

***Figure 2 – FPGA Simulation***

* See page 9 of this issue.

## How the Cordic Method Works

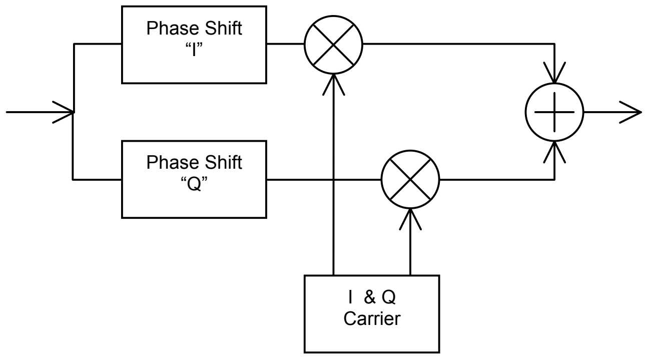

***Figure 3 – The 'phasing' method of SSB generation***

*The audio signal is modified by two filters such that the outputs are 90° out of phase. (This is easier to implement that a single filter with a 90° phase shift). The resulting in-phase and quadrature components are multiplied by an in-phase and quadrature carrier, and the resulting DSB signals are summed. One of the sidebands will cancel, leaving an SSB signal at the output.*

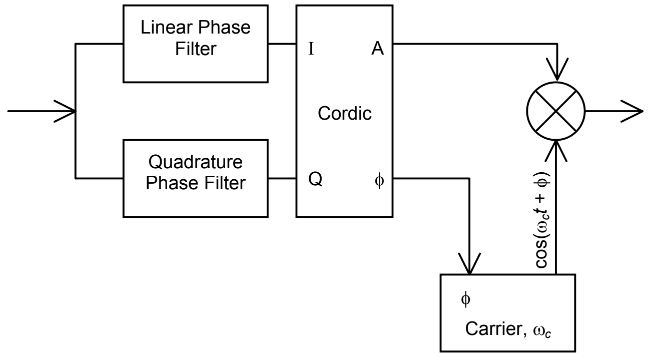

***Figure 4 – Graham Naylor's 'cordic' method of SSB generation***

*The audio signal is modified by two filters, exactly as in the 'phaser' method. Since the implementation may well be digital, the in-phase channel may use a 'linear phase filter' (i.e. simply a delay) and the quadrature channel may use a 'quadrature phase filter', which is simpler to implement than a duplication of the conventional analogue 'phaser' function in a digital domain. The resulting I and Q signals are converted to amplitude and phase using a cordic (a device that converts rectangular co-ordinates to polar). The phase output of the cordic is used to phase-modulate a carrier, which is then amplitude modulated by the cordic. The result is an SSB signal. The advantage of this method is that the carrier may be digital in nature, as explained in the text.*

Graham Naylor's cordic method, as described in **Figure 4** makes use of a **cordic** – a device that converts rectangular to polar co-ordinates. The relationship between the inputs and outputs is

$$A(t) = \sqrt{I^2 + Q^2}\ ,\ \phi(t) = \arctan\frac{Q}{I} \tag{1}$$

This would be difficult to implement in analogue circuitry, but it is easier than one might imagine in software. Several algorithms exist that are efficient and easily implemented. One method uses the concept of a synthesised 'rotating vector', and the output 'servos' to track the input. In this way, the difficult and/or time-consuming calculations of an inverse tangent and square-root do not have to be performed. The phase and amplitude modulation of the carrier $\omega_c$ is simply

$$S(t) = A \cdot \cos(\omega_c t + \phi) \tag{2}$$

The phase modulation is trivial to implement digitally as it involves only the adding or subtracting of a value from a counter that is being incremented by the system clock. The amplitude modulation can utilise a digital PWM module. The figures to the right show the results of a MatLab simulation of the technique. The program is given in the box on the next page.

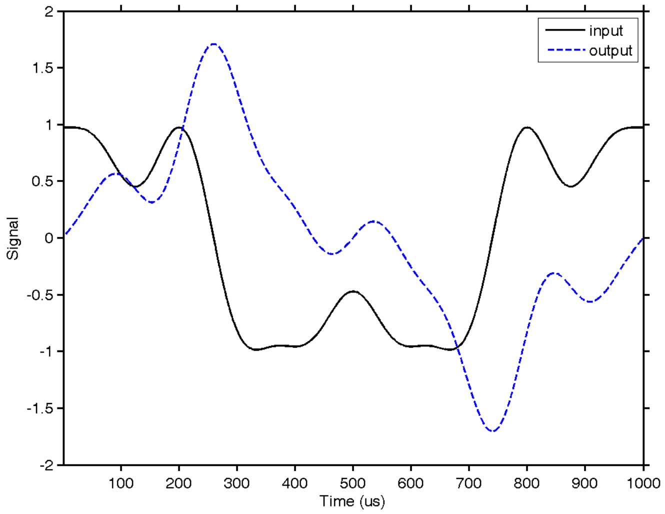

***Figure 5 – Hilbert transformer (Phaser)***

*The solid line represents an example input to the phase-shifter, comprising a 1kHz cosine wave, with 3rd, 4th, 5th and 7th harmonics. The dotted line represents the 90°phase-shifted output.*

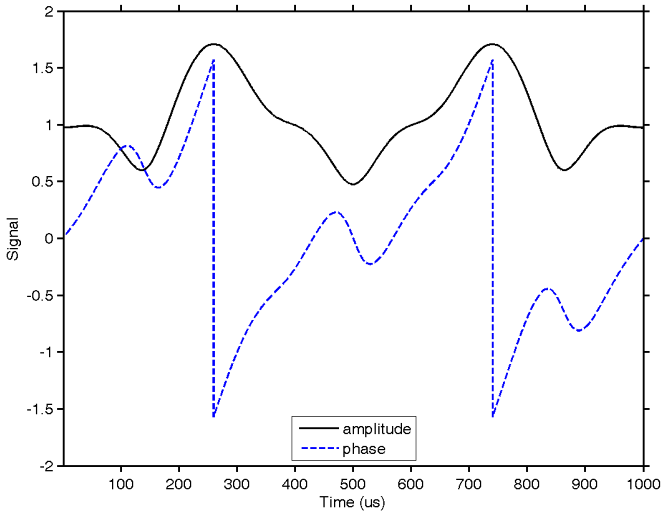

***Figure 6 – Cordic output***

*The quadrature outputs of the phaser (Figure 5) are processed by a cordic (a device that converts rectangular to polar co-ordinates), which produces amplitude (solid line) and phase (dotted line) signals. Note that the amplitude signal bears no obvious resemblance to the original signal in Figure 5 and that the phase has been constrained to ±π/2 rad.*

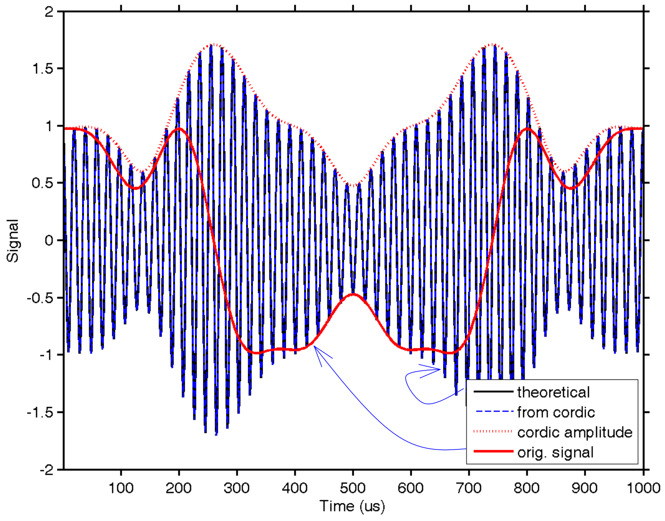

***Figure 7 – SSB generation]***

*A solid line represents a simulation of the signal of Figure 5 up-shifted to 50kHz. A synthesised SSB signal, derived from the cordic outputs of Figure 6 is shown by a dotted line that overlies the solid line exactly (and is therefore not visible), demonstrating that the cordic produces the signal we desire. The envelope of the SSB signal follows the cordic amplitude signal. The original signal is also plotted, in order to demonstrate that it bears no obvious relationship to the SSB signal.*

## MatLab Demonstration of Cordic Method

```
% SSB4 investigate Graham Naylor's SSB modulator
% (David Gibson, 30 Nov 2006. MatLab 7.2)

% create a time line. 0 to 10ms, interval 0.1us
time = (0 : 1E-7 : 0.01);
time_axis = time * 1E6;          % time in microseconds

% create a 1kHz waveform with some harmonics
f = 1000;                        % frequency in Hz
fN =    [1  2  3  4  5  6  7];   % frequencies of harmonics
aN = 1./[1 Inf -3 4 5 Inf -7];   % amplitude of harmonics
wave = aN * cos(2*pi*f*fN'*time);

% now shift the waveform by 90 degrees
waveQ = aN * cos(2*pi*f*fN'*time - pi/2);

% Run it though a cordic to generate phase and amplitude
ampl = sqrt(wave.^2 + waveQ.^2);
phase = atan(waveQ./wave);

% Up-shift the original signal onto a carrier
carrier = 50000;                 % frequency in Hz
fc = carrier * ones(size(fN));
waveC = aN * cos(2*pi*(f*fN' + fc')*time);

% Now synthesise the same carrier via the cordic
waveS = ampl .* cos (2*pi*carrier*time + unwrap(2*phase)/2);

%% ========== PLOT THE GRAPHS ==========
% Open a log file in the RESULTS folder
results = 'results\';            % Name of results folder
program = 'ssb4';
logfile = [program '-log-' datestr(now,'HHMM')];
[success message] = mkdir(results);

X = f;                           % length of X-axis
fig = 0;

fig = fig+1; figure(fig); cla;
plot(time_axis, wave, 'k', 'linewidth', 1.5)
hold on
plot(time_axis, waveQ,'b--', 'linewidth', 1.5)
hold off
parms = ['Reference: ' logfile '-' num2str(fig)];
title({'Hilbert Transformer (Phaser)'; parms},'fontsize',10)
legend('input', 'output', 'location', 'NorthEast')

fig = fig+1; figure(fig); cla;
plot(time_axis, ampl, 'k', 'linewidth', 1.5)
hold on
plot(time_axis, phase, 'b--', 'linewidth', 1.5)
hold off
parms = ['Reference: ' logfile '-' num2str(fig)];
title({'Cordic outputs'; parms},'fontsize',10)
legend('amplitude', 'phase', 'location', 'South')

fig = fig+1; figure(fig); cla;
plot(time_axis, waveC, 'k', 'linewidth', 1.5)
hold on
plot(time_axis, waveS, 'b--', 'linewidth', 1)
plot(time_axis, ampl, 'r:', 'linewidth', 2)
plot(time_axis, wave, 'r', 'linewidth', 2)
hold off
parms = ['Reference: ' logfile '-' num2str(fig)];
title({['Up-shift to ' num2str(carrier/1000) 'kHz']; ...
  parms},'fontsize',10)
legend('theoretical', 'from cordic', 'cordic ampl', ...
  'orig. signal', 'location', 'SouthEast')

% embellish all the figures and store them to disc
plot_filetype = '-dtiff';        % Type of graphical file
plot_resolution = '-r150';       % Resolution of output file
for fig=(1:fig),
    figure(fig)
    axis([1 X -2 2])
    set(gca,'LineWidth',1.5,'FontSize',12,'TickDir','out')
    xlabel('Time (us)','fontsize',12)
    ylabel('Signal','fontsize',12)
    print(plot_filetype, plot_resolution, ...
        [results logfile '-' num2str(fig)] )
end
```

## A Proof of the Cordic Method

You may consider that the demonstration in the **How the Cordic Method Works** box is sufficient as a proof. But it could be dangerous to argue from the specific to the general, so an algebraic proof is desirable.

But first, consider what happens if the signal being processed is a simple cosine wave, say ***a*·cos ωt**. In this situation, the phase-shifted version is ***a*·sin ωt** so the output of the cordic has a constant amplitude of ***a***.

For a moment, you might think that this must be true of each frequency component of the signal, and that we can simply add the components together. But, of course, this is not so! As a simple example, consider the case of a composite signal with two frequency components,

$$I = a\cos\omega t + b\cos 3\omega t \qquad (3)$$

The quadrature signal will be

$$Q = a\sin\omega t + b\sin 3\omega t \qquad (4)$$

so, from equation (1), the cordic's amplitude output is

$$A = \sqrt{a^2 + b^2 + 2ab\cos 2\omega t} \qquad (5)$$

which demonstrates that there is a frequency component to the cordic signal that was not present in the original signal. For a moment, you might now start to believe that the method might not work with multiple tones, (or, at least, that a proof would be difficult).

However, we can argue the validity of the technique from a consideration of 'rotating vectors'. But a proof based on the need to draw diagrams is not satisfying because it makes too many 'anthropomorphic' assumptions. (E.g. are the diagrams merely a human artefact and how can we be certain that our interpretation of them is correct?) And, if they *do* stand up to scrutiny, ... well, in that case we should be able to solve the problem algebraically without diagrams!

### The Proof

In fact, there is a simple algebraic proof, which you may think is 'cheating' as it seems merely to undo what the cordic does. Nevertheless, the proof is valid. The cordic produces ***A*** and **φ,** which modulate a carrier as

$$S(t) = A\cdot\cos(\omega_c t + \phi) \qquad (6)$$

which, we are 'claiming', is an SSB signal. We expand the right side to

$$A\cdot(\cos\omega_c t\cos\phi + \sin\omega_c t\sin\phi) \qquad (7)$$

and we note that, from the definition of ***A*** and **φ** in equation (1), that

$$\tan\phi = Q/I \text{ , so} \qquad (8)$$

$$\cos\phi = I/A \text{ and } \sin\phi = Q/A \qquad (9)$$

so our conjectured SSB signal is

$$S = I\cdot\cos\omega_c t + Q\cdot\sin\omega_c t \qquad (10)$$

which is the same form as for the conventional phaser. E.g. ***I* = cos ωt ⇒ *Q* = sin ωt** and ***S* = cos (ω$_c$ - ω) *t***.

This demonstrates that ***S*** is a frequency-shifted signal and, moreover, that the transfer function is linear, and so we can say that what is true for a monochromatic signal will be true for a composite waveform. *QED*.

## Advantages

The advantage of the cordic method is that it allows us to easily synthesise a digital (class-D) SSB signal with the switching frequency equal to the carrier frequency. Were it not for this highly important feature, the cordic method would not have much merit in the digital domain, because other digital methods (e.g. Gibson, 1992, 2002) would be easier even if they did not directly produce a class-D output.

To see why use of a cordic is such an important technique, it is helpful to consider how we might digitally synthesise other modulated carriers.

### Amplitude Modulation

The zero-crossings of an AM carrier occur at fixed intervals of half the carrier period and the information is conveyed by the different amplitudes of the half cycles. It is therefore straightforward to replace the analogue half-cycles by a digital PWM signal. There are issues to do with the positioning (i.e. the phase) of the pulse, which were addressed in Beat Heeb's design (Heeb, 2004) and in (Gibson, 1992) where a design for a double-sideband suppressed carrier modulator was described. However, a hardware design of that type would be obsolete now.

### Frequency Modulation

The zero-crossings of an FM carrier occur at differing intervals and convey the information in the signal. The amplitude of the signal is irrelevant and, in a digital system, we can square it off without loss of information. It is therefore straightforward to replace the analogue half-cycles by pulses of a varying length, which are easily generated digitally.

### Phase Modulation

Analogue PM and FM are related. Phase modulation of the integral of a modulating signal is equivalent to the frequency modulation of the original signal. In a digital system PM may be conceptually easier than FM.

### SSB Modulation

The difficulty with single-sideband modulation is that information is carried in both the amplitude *and* the phase of the signal. Moreover, neither the amplitude nor the phase is related to the modulating signal in the simple way that it is with AM, FM or PM.

Graham Naylor's cordic method derives the signals that are required to separately modulate the phase and amplitude of the carrier. Both of these modulations can then be undertaken using techniques that are simple to implement in the digital domain.

### References

Gibson, David (1992), *Cave Radio PA Design and a Digital PWM DSB Modulator*, CREGJ **10**, p16-24, December 1992.

Gibson, David (2002), *Cave Radio Notebook 48: A DSP implementation of the Turner/ Weaver SSB modulator*. CREGJ **47**, pp 18 & 24. March 2002.

Heeb, Beat (2004) *Underground Text Messaging*, CREGJ **57**, pp4-7, Sept 2004.