A Discrete Time Oscillator for a DSP based Radio

Clay S. Turner
TGA Systems, Inc.
3100 Medlock Bridge, Suite 150
Norcross, GA 30071
(770) 441-2100

1 Introduction

With the proliferation of digital signal processors, **DSP**s, and the ever increasing need for higher data rates over radio channels, it is no surprise that **DSP**s are becoming integral parts of modern radios. Some of the early uses were for performing spectral and delay equalization tasks and straightening out the non-linearities of the mixers and amplifiers. Now they are also being used for performing the modulation/demodulation tasks. Not only does this enhance the radio's performance, but also it puts more of the radio under software control which simplifies the configuration task of multifunction radios. In the context of the paper, the term radio refers to both transmitters and receivers.

High data rate modulation methods utilize optimal filters, precise phase and frequency shifters, and harmonically pure numerically controlled oscillators. There is a lot of information in the literature about the design of such modulators and demodulators along with most of their components. But there is relatively little about discrete time oscillators. This paper presents an oscillator design that can be used in almost any modulator topology.

The oscillator features a stabilized amplitude with quadrature outputs, and it may be smoothly tuned. The author has implemented the algorithm in a Motorola DSP56002. The subroutine consists of 23 instructions and consumes just 46 processor clock cycles per oscillator output. This includes frequency tuning and both the I and Q outputs. If tuning is not required, then only 28 cycles are needed.

2 Oscillators

The classical approach to oscillator design uses an amplifier whose output is fed back to its input via a gain reducing and phase shifting network. The Barkhausen criterion for an oscillator requires the forward gain times the reverse gain to equal 1 and that the phase shift must be a multiple of 2π. This is easily done with a **DSP** via multiplies and delays.

2.1 Table Look Up

A simple, yet frequency limited, way to generate sinusoids is by table look up. The sinusoid is evaluated and stored in a table. To generate the sinusoid, the table's values are read off in succession with table wrap around. This is easily done with a pointer using modulo arithmetic. Since a phase continuous sinusoid is desired, the table needs to hold an integral number of periods. Despite the simplicity of the table method, it does possess two major problems. The first is the requirement that the frequency must divide evenly into some integral multiple of the sampling rate. This can result in lengthy tables. Also if quadrature outputs are needed, then the table's length must be a multiple of 4, so an exact 90° phase relation is maintained. The second drawback is the question of smooth frequency adjustment.

Some deal with these problems by using shorter tables and accepting the harmonic distortion caused by the phase discontinuities while nonuniformly stepping through the table. However, some applications require harmonically pure waves and find this method unsuitable. Additionally, for quadra-

ture outputs, this method has a phase, between the carriers, noise component.

The table look up method's harmonic distortion and phase noise can be reduced by using interpolation, but then the total computational effort exceeds that of the recursive oscillators. When pure sinusoids are needed, the recursive oscillators become the method of choice.

2.2 Recurrence Relations

Since classical oscillators function by feeding their output back into their input with an appropriate delay, it is desirable to find a similar relationship that is operable with a discrete time system. This discrete time relation is called a "recurrence relation." The recurrence relation states that when given n consecutive values, the next value can be calculated. If the recurrence is such that the next value can be written as a linear combination of the past n values, the recurrence relation represents an nth order all pole filter. An all pole filter has a feedback only topology.

An oscillator that generates a single sinusoid is called a simple harmonic oscillator, **SHO**. From the theory of difference equations, it can be shown that a discrete time sinusoid has a two term recurrence where the coefficients are real valued. This two term recurrence turns out to be a classical trigonometric theorem. It is:

$$\sin(a+b) = 2\cos(b)\sin(a) - \sin(a-b) \quad (1)$$

To interpret this theorem as an oscillator recursion iteration, let $a = \frac{2\pi f}{f_s}n$ and $b = \frac{2\pi f}{f_s}$, where f is the oscillator frequency, f_s is the sampling rate, and n is the sample number. Then the theorem becomes

$$\begin{aligned}\sin\left(\frac{2\pi f}{f_s}(n+1)\right) &= 2\cos\left(\frac{2\pi f}{f_s}\right)\sin\left(\frac{2\pi f}{f_s}n\right) \\ &\quad - \sin\left(\frac{2\pi f}{f_s}(n-1)\right) \quad (2)\end{aligned}$$

Written in this way, the recurrence is found to be

$$\begin{aligned}\text{NextSin} &= 2\cos\left(\frac{2\pi f}{f_s}\right)\text{PresentSin} \\ &\quad -\text{LastSin} \quad (3)\end{aligned}$$

This relation says that with two starting values and two feedback values, one can recursively generate a sine wave. One of the feedback values is the constant,-1. Since the value of a in the trig theorem is arbitrary, the recursion is phase invariant. It is the starting values that allow for both amplitude and phase determination. The frequency enters into both the starting values and one of the feedback values.

For example, the initial values could be set to 0 and $A\sin\left(\frac{2\pi f}{f_s}\right)$ to make a sinusoid with frequency f and amplitude A. Notice that the filter's first output is $A\sin\left(2\frac{2\pi f}{f_s}\right)$.

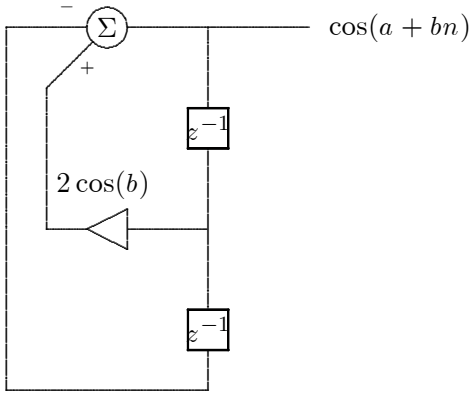

Figure 1: Simple Harmonic Oscillator

2.3 Quadrature Oscillators

Previously, a two term recurrence (2nd order) relation was used to make a **SHO**. Now two coupled single term (1st order) recurrence relations will be used to make an oscillator that has two outputs where one is 90° out of phase with respect to the other. To construct such an oscillator requires two classical trigonometric theorems. They are:

$$\begin{aligned}\sin(a+b) &= \sin(a)\cos(b) + \cos(a)\sin(b) \quad (4)\\ \cos(a+b) &= \cos(a)\cos(b) - \sin(a)\sin(b) \quad (5)\end{aligned}$$

Substituting the same values for a and b as before, the prescription for a quadrature oscillator is:

$$\begin{aligned}\text{NextSin} &= \cos\left(\frac{2\pi f}{f_s}\right)\text{CurrentSin} + \end{aligned}$$

$$\sin\left(\frac{2\pi f}{f_s}\right)\text{CurrentCos} \qquad (6)$$

$$\text{NextCos} = \cos\left(\frac{2\pi f}{f_s}\right)\text{CurrentCos} - \sin\left(\frac{2\pi f}{f_s}\right)\text{CurrentSin} \qquad (7)$$

Figure 2 shows the network form of the quadrature oscillator.

Like before, there are two delay elements; however, the initial values are different. With the **SHO** case, they are two consecutive samples of a sinusoid— In this case, they are samples of a sinusoid 90° apart. This last fact makes the quadrature oscillator easy to control.

The usefulness of the quadrature oscillator has its roots in Fourier transform theory. Most important is the frequency shifting theorem which says for the fourier transform pair:

$$x(t) \Leftrightarrow X(\omega) \qquad (8)$$

that

$$x(t)e^{i\Delta\omega t} \Leftrightarrow X(\omega - \Delta\omega) \qquad (9)$$

The quadrature oscillator generates the components of $e^{i\Delta\omega t}$. Thus the quadrature oscillator is a key component of the frequency shifting process.

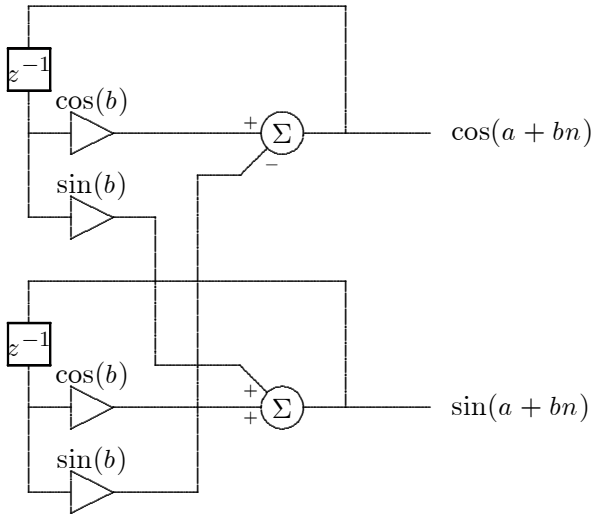

Figure 2: Quadrature Oscillator

2.4 Amplitude Stabilized Oscillators

So far the described oscillators are ballistic in the sense that they are loaded with some preset values and allowed to free run. Since the filter's coefficients and data must be quantized to be practical, a **DSP** implementation may yield an oscillator whose amplitude will change with time. If the word size of the **DSP** is large, i.e., 24 bits, the oscillator can operate for a large number of iterations before the amplitude change becomes significant. Some numerical experiments have demonstrated over a million iterations are needed before the amplitude changes more than 10 percent. So ballistic oscillators function very well for short duration tone bursts. However, if one needs to generate a carrier, which can operate indefinitely, a ballistic oscillator won't do.

To stabilize the amplitude, one measures the oscillator's amplitude and compares it with the setpoint (desired) amplitude and adjusts the feedback accordingly. The quadrature oscillator allows for trivial, non-frequency dependent, amplitude measurement. By denoting the outputs I and Q. The amplitude, A_o is just $\sqrt{I^2 + Q^2}$. If the setpoint amplitude is denoted A_s, then the stabilization gain G is

$$G = \frac{A_s}{\sqrt{I^2 + Q^2}} \qquad (10)$$

The factor G is used in the feedback loops.

If the instantaneous error correction requirement is relaxed, the costly division and square root functions can be avoided. This is done by finding the 1st order Taylor's series expansion of G about A_s. This approximation has the property of becoming more accurate as A_o approaches A_s. This approximation is:

$$G \approx \frac{3}{2} - \frac{1}{2}{A_s}^{-2}(I^2 + Q^2) \qquad (11)$$

Since the amplitude variations, without stabilization, are very small, the first order correction works very well.

With the consideration of a fixed point **DSP** in mind, it becomes convenient to let $A_s = \frac{\sqrt{2}}{2}$. Also, if $G_{\frac{1}{2}}$ is used and the corrected values are scaled up by 2, then all of the values are in $[-1, 1)$.

Thus,

$$G_{\frac{1}{2}} = \frac{3}{4} - \frac{1}{2}(I^2 + Q^2) \qquad (12)$$

The network for the amplitude stabilized oscillator is shown in figure 3. The circles with the Πs in them represent multiplication by a variable, whereas, the triangles represent multiplication by a constants. This form of the oscillator when used in conjunction with a delay line, and a Hilbert transformer, may be used for single side band amplitude modultion, **SSB AM**. Likewise, this oscillator may be used to implement both quadrature amplitude modulation, **QAM**, and quadrature amplitude demodulation, **QAD**.

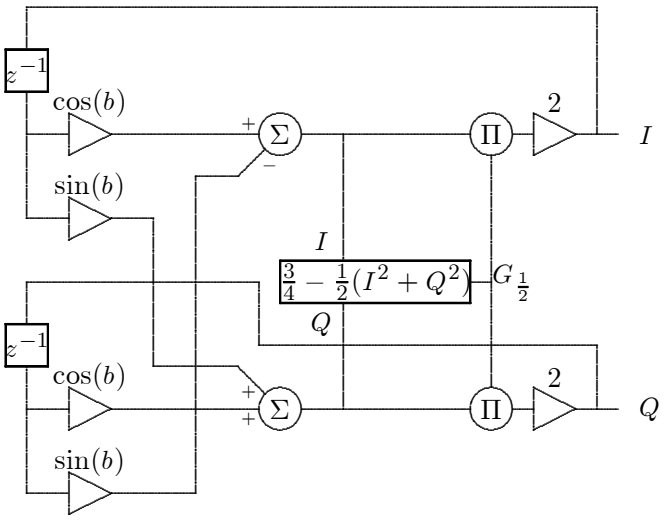

Figure 3: Amplitude Stabilized Quadrature Oscillator

2.5 Frequency Tuning

As mentioned earlier, the quadrature oscillator has many merits. Not only is it easy to stabilize its amplitude, but it is also easy to adjust its frequency while in operation. Since the two term recursive oscillator has only one frequency dependent coefficient, if it is adjusted while the oscillator is running, the oscillator will change amplitude in addition to changing frequency. However, the quadrature oscillator with its two frequency dependent coefficients will not change amplitude when the frequency is changed. The resulting sinusoid is continuous. Thus, the tunable oscillator functions like a voltage controlled oscillator, **VCO**, in a phase locked loop, **PLL**. This discrete time tunable oscillator is called a numerically controlled oscillator, **NCO**.

This makes the **NCO** ideal for use as a frequency shift keying, **FSK**, modulator. Additionally, if the **NCO**'s, input is first differentiated, then one has a phase shift keying, **PSK**, modulator.

The amplitude independence during tuning arises from the fact that the two stored values are always 90° apart. This is not true with the **SHO**. For a two or four level **FSK** system, the $\cos(b)$ and $\sin(b)$ terms for each frequency are precalculated and stored in a table.

For a continuously variable frequency modulation, **FM**, system, $\cos(b)$ and $\sin(b)$ will have to be evaluated in real time. Since the amplitude stabilized form is used for long term stability, slight errors in the $\sin(b)$ and $\cos(b)$ terms are tolerated. This means moderate order series expansions may be used for $\sin(b)$ and $\cos(b)$.

A wide band **FM** system will need a Chebyshev expansion where the maximum error is minimized over an interval that is matched to the working range of the oscillator. A narrow band **FM** system can use a Taylor's approximation. The expansions are about the center frequency. Since the error is zero at the center frequency and small frequency deviations are common in **FM** systems, a low order approximation can be used. Since order of kilohertz deviations are performed on order of 100 megahertz carriers, it is easy to see that the relative frequency shift, h, is a tiny number.

If b is the center frequency, the following 2nd order Taylor's approximations are found:

$$\cos(b+h) \approx \left(1 - \frac{h^2}{2}\right)\cos(b) - h\sin(b) \quad (13)$$

$$\sin(b+h) \approx \left(1 - \frac{h^2}{2}\right)\sin(b) + h\cos(b) \quad (14)$$

where h is a small frequency deviation.

A similar functional form arises for the Chebyshev expansions.

$$\cos(b+h) \approx C(h)\cos(b) - S(h)\sin(b) \quad (15)$$

$$\sin(b+h) \approx C(h)\sin(b) + S(h)\cos(b) \quad (16)$$

Here $C(h)$ and $S(h)$ are low order polynomial expansions of $\cos(h)$ and $\sin(h)$ about 0 repectively.

This formulation for both the Taylor and Chebyshev expansions uses an orthogonal transformation from the center frequency b to the origin. The orthogonal transformation's advantage is that the polynomial approximation errors are not magnified by the transformation process. This allows for lower order polynomial approximations and smaller magnitudes for the polynomial's coefficients.

3 DSP Code

This section shows three useful oscillator macros, written in Motorola DSP56002, demonstrating various forms of the algorithm's implementation. The first is the ballistic form of the oscillator. The second is an amplitude stabilized quadrature oscillator, and the third is a smoothly tunable stabilized oscillator. Each uses two long memory locations. On entry and exit `r\reg` points to the first of the memory locations. The first long address contains $\cos\left(\frac{2\pi f}{f_s}\right)$ in x space and has the cosine output in the y space. The cosine output is initialized to $\frac{\sqrt{2}}{2}$. The second long address contains $\sin\left(\frac{2\pi f}{f_s}\right)$ in x space and has the sine output in the y space. The sine output is initialized to 0.0. Since the macros make extensive use of the processor's parallelism, the long memory data should be placed in internal **DSP** ram to avoid wait states.

The macros may easily be modified to provide stackable oscillators by changing the autodecrement part of the each macro's last instruction to an autoincrement. The pointer would also need to be configured to be modulo 2 times the number of oscillators. Likewise, the memory locations will need to be at the proper type of modulo address.

```
;ballistic quadrature
; oscillator iteration (CST)
;6 instructions -- 12 cycles
;182 nSec @ 66MHz

Q_OSC MACRO reg
move L:(r\reg)+,x
mpy x0,x1,a L:(r\reg)-,y
mac -y0,y1,a
mpy x1,y0,b
mac x0,y1,b a,y:(r\reg)+
move b,y:(r\reg)-
ENDM

;amplitude stabilized quadrature
;  oscillator iteration (CST)
;14 instructions -- 28 cycles
;424nSec @ 66MHz

AQ_OSC MACRO reg
move L:(r\reg)+,x
mpy x0,x1,a L:(r\reg)-,y
mac -y0,y1,a
mpy x1,y0,b
mac x0,y1,b a,x0
mpy -x0,x0,a b,y0
mac -y0,y0,a #0.75,b
addr b,a
move a,x1
mpy x1,x0,a
asl a
mpy x1,y0,b a,y:(r\reg)+
asl b
move b,y:(r\reg)-
ENDM

;Continuously tunable,
; stabilized quadrature oscillator (CST)
;Uses 2nd order Taylor's series.
;23 instructions -- 46 cycles
;697nSec @ 66MHz

AFQ_OSC MACRO reg
move a,y0 #0.5,b ;a has delta freq.
asl b L:(r\reg)+,x
asl b
mac -y0,y0,b L:(r\reg),y
asr b a,y0
mpy -y1,y0,a b,x0
mpy x1,y0,b
mac y1,x0,b L:(r\reg)-,y
mac x1,x0,a L:(r\reg),x
move a,x1
mpy x0,x1,a b,y1
mac -y0,y1,a
mpy x1,y0,b
mac x0,y1,b a,x0
mpy -x0,x0,a b,y0
mac -y0,y0,a #0.75,b
```

```
addr b,a
move a,x1
mpy x1,x0,a
asl a
mpy x1,y0,b a,y:(r\reg)+
asl b
move b,y:(r\reg)-
ENDM
```

4 Conclusion

Several oscillator forms have been described along with sample code and a continuously variable frequency, amplitude stabilized quadrature oscillator is developed. While this last form seems to be a bit convoluted, its execution is straight forward. It needs only four memory locations— two for the outputs and two for the feedback values. These four locations may be combined into two long memory locations as is done in the sample macros.

The quadrature oscillator is not only extremely flexible in that it finds numerous uses in modulators and demodulators, but this implementation also allows for easy control. Also, The efficiency of the oscillators allows one to design a radio with a general purpose **DSP** with an IF in the 100 kHz range. An ASIC could execute the algorithm at a much higher rate.