Mixed Signal Frequency Mixers with Intermodulation Product Cancellation

William A. Ling
Johns Hopkins University & Sotekco Electronics LLC
Baltimore, MD, USA
Email: wling [at] jhu [dot] edu

Paul P. Sotiriadis
Sotekco Electronics LLC
Baltimore, MD, USA
Email: pps [at] ieee [dot] org

Robert Adams
Analog Devices Inc.
Wilmington, MA
USA

Abstract—We propose a nearly all-digital, broadband frequency mixer which synthesizes a sinewave at the frequency difference of two periodic input square waves. The mixer has high output spurious free dynamic range by cancelling the dominant intermodulation products in the sinewave generation process. Due to its use of almost exclusively digital circuitry, it can easily be integrated with CMOS digital circuits. A programmable logic implementation and spectral measurements demonstrate its feasibility and performance.

I. INTRODUCTION

Frequency mixers are an important class of circuits used in many wireless and microwave applications [1]. The ideal frequency mixer takes two input frequencies \( \omega_1 \) and \( \omega_2 \) to produce an output frequency \( \omega_1 \pm \omega_2 \). In reality, however, the output spectrum of a mixer contains many additional, undesired intermodulation products \( k\omega_1 \pm m\omega_2 \) resulting from the nonlinear behavior of the mixer.

Frequency mixers are typically implemented using either an active-device multiplier [2], or, nonlinear elements, such as a diode or diode-based structures [3].

Diversion from the ideal multiplication introduces undesired intermodulation products that in many cases are difficult or even impossible to reject using filters since they may appear inside the working band. Furthermore, the use of analog circuitry complicates, and in some cases limits, integrability of a frequency mixer with a digital circuit.

In this work, we address these disadvantages of analog frequency mixers by proposing a mixed-signal frequency mixer architecture which utilizes minimal analog circuitry (in fact, only resistors) but maintains high SFDR through intermodulation product cancellation, achieved in part through multiphasing, a concept that was introduced for up-and-down-conversion in radio tranceivers in [4], and was exploited in [5] for almost-all digital frequency synthesis.

II. SPECTRAL PROPERTIES OF SAMPLED SINUSOIDS

In a digital representation of sinusoidal signals, one can only represent the ideal sinusoid using an approximation signal that is quantized in time. A synchronous digital system further restricts that the quantization in time be uniform:

\[
a(t) = \frac{p_n \mod N}{T} \quad \text{for } t \in \left[\frac{nT}{N}, \frac{(n+1)T}{N}\right),
\]

where \( T \) is the period of \( a(t) \) and \( N \) is the number of intervals per period on which \( a(t) \) is constant.

One would desire to choose the \( p_i \) and \( N \) such that this approximation signal is as spectrally pure as possible under implementation constraints. In this work we consider the class of signals generated by the following choice of the \( p_i \):

\[
p_i = \cos \left( \frac{2\pi i}{N} + \theta \right),
\]

where \( \theta \in [0, 2\pi / N) \). We state the following result regarding such a function:

**Fact 1:** A signal of the form (1) with coefficients of the form (2) has non-zero discrete spectral components only at frequencies \( (rN + 1)\omega \) and \( (rN - 1)\omega \), where \( r \) is any integer. For example, if \( N = 8 \), then \( a(t) \) has only harmonics 1, 7, 9, 15, ….

III. MULTIPHASE MIXING FOR INTERMODULATION PRODUCT CANCELLATION

Consider a function \( a(t) \) with frequency \( \omega_1 \) and period \( T_1 \). Consider also a function \( b(t) \) with frequency \( \omega_2 \) and period \( T_2 \). We assume that these functions have no DC value. These functions can be represented by the following Fourier series:

\[
a(t) = \sum_{k=1}^{\infty} A_k \cos(k\omega_1 t + \theta_k)
\]

\[
b(t) = \sum_{m=1}^{\infty} B_m \cos(m\omega_2 t + \phi_m)
\]

Consider also phase-shifted versions of \( a(t) \) and \( b(t) \):

\[
a\left(t + \frac{nT_1}{P}\right) = \sum_{k=1}^{\infty} A_k \cos \left( k\omega_1 t + \theta_k + \frac{2\pi kn}{P} \right)
\]

\[
b\left(t + \frac{nT_2}{P}\right) = \sum_{m=1}^{\infty} B_m \cos \left( m\omega_2 t + \phi_m + \frac{2\pi mn}{P} \right)
\]
where \(n\) and \(P\) are integers. Now consider the sum of products of phase-shifted pairs:

\[
\sum_{n=0}^{P-1} a \left( t + \frac{nT_1}{P} \right) b \left( t + \frac{nT_2}{P} \right) = \sum_{n=0}^{P-1} \sum_{k=1}^{\infty} \sum_{m=1}^{\infty} A_k \cos \left( k\omega_1 t + \theta_k + \frac{2\pi kn}{P} \right) B_m \cos \left( m\omega_2 t + \phi_m + \frac{2\pi mn}{P} \right)
\]

\[
= \sum_{k=1}^{\infty} \sum_{m=1}^{\infty} \frac{P A_k B_m}{2} \left[ \cos \left( (k\omega_1 + m\omega_2)t + \theta_k + \phi_m + \frac{2\pi (k + m)n}{P} \right) + \cos \left( (k\omega_1 - m\omega_2)t + \theta_k - \phi_m + \frac{2\pi (k - m)n}{P} \right) \right]
\]

This expression can be rewritten using the following fact:

\[
\sum_{n=0}^{P-1} \cos \left( x + \frac{2\pi r n}{P} \right) = P \cos(x) \delta[r \mod P],
\]

where \(r\) is any integer. Using this relation, (5) can be written as

\[
= \sum_{k=1}^{\infty} \sum_{m=1}^{\infty} \frac{P A_k B_m}{2} \left[ \cos \left( (k\omega_1 + m\omega_2)t + \theta_k + \phi_m \right)[(k + m) \mod P] + \cos \left( (k\omega_1 - m\omega_2)t + \theta_k - \phi_m \right)[(k - m) \mod P] \right]
\]

This important result shows that most intermodulation products formed from multiplying \(a(t)\) with \(b(t)\) are cancelled by the phase shifting and addition. For our purposes, we desire the difference frequency \(\omega_1 - \omega_2\) as the output of the system. The sum frequencies \(k\omega_1 + m\omega_2\) are all much larger than this difference frequency, so they can all be safely ignored.

**IV. MIXED-SIGNAL CIRCUIT IMPLEMENTATION**

**A. Representing Piecewise-Constant Functions with Digital Circuits**

As an example, consider a function \(a(t)\) which shall approximate the sine wave using \(N = 4\) divisions per period \(T\). Let the frequency of \(a(t)\) be \(\omega = 2\pi/T\). Let the four possible amplitudes of \(a(t)\) be \(p_0 = a(0), p_1 = a(T/4), p_2 = a(2T/4)\), and \(p_3 = a(T)\). It can be shown that \(p_2 = -p_0\) and \(p_3 = -p_1\); in fact as long as \(N\) is even, the \(p_i\) will assume values which occur in equal-magnitude but oppositely-signed pairs.

First consider fig. 1. There are two shift registers, each four bits long and with the same clock input of frequency \(4\omega\) (rad/s). Bits shift from left to right, except for the right-most bit, which shifts its output into the left-most bit. The contents of the shift registers is initialized to hold a particular pattern of bits. Consider the right-most bits of the shift registers. Currently, this pair of bits holds the binary number 00, with the top shift register holding the most significant bit and the bottom shift register holding the least significant bit.
which rotates at the rising edge of a clock of frequency $4\omega_1$ (rad/s). The ports labeled $Q_0$ through $Q_3$ are the outputs of the shift register.

Register $A_2$ is constructed similarly, but is initialized with a different 4-bit sequence. Together, the $Q_3$ outputs of registers $A_1$ and $A_2$ form a 2-bit representation of the amplitude of $a(t)$. Since $a(t)$ has 4 possible amplitudes, these two bits uniquely represent the value of $a(t)$ at any instant in time.

Shift registers $B_1$ and $B_2$ are used in an analogous manner to represent $b(t)$. The 4 bits entering the address inputs of the ROM thus uniquely represent the current amplitudes of $a(t)$ and $b(t)$. The ROM uses this to determine the corresponding product. Thus, assuming the ROM has negligible propagation delay, at any instant in time, the output of the ROM contains all the information necessary to determine the product $a(t)b(t)$.

The only missing component is a circuit to convert this digital code from the ROM into an analog value which closely approximates $a(t)b(t)$.

Digital to analog conversion can be accomplished using a resistor network driven by digital gates, in this case a ROM’s output logic gates. Consider fig. 3, where we show a 3-bit example of a nonlinear digital-to-analog converter (DAC). The output is the following:

$$V_{out} = c(V_2/R_2 + V_1/R_1 + V_0/R_0),$$

where $c$ is a constant.

Fig. 3. Nonlinear DAC using a resistor network

C. Digital-to-Analog Conversion of ROM Output Codes

The design of the DAC is simplified by noting that if $N$ is even, then $p_i = -p_{i+N/2}$ for $i = 0, 1, \ldots, N/2 - 1$. This means that the products also occur in oppositely-signed pairs.

Consider the circuit of fig. 4. For this DAC, the most negative product is represented as 0 V. This corresponds to a ROM output of $V_2V_1V_0W_2W_1W_0 = 000000$.

Similarly, the maximum product will be represented by having the ROM output $V_2V_1V_0W_2W_1W_0 = 111111$. A product of 0 will be represented with the output $V_2V_1V_0W_2W_1W_0 = 000111$, which results in an analog output of

$$c\left(\frac{0}{R_2} + \frac{0}{R_1} + \frac{0}{R_0} + \frac{1}{R_2} + \frac{1}{R_1} + \frac{1}{R_0}\right)$$

where $c$ is a constant. Thus, (8) is the DC offset of the DAC. This DAC can be used to produce any output of the form

$$c\sum_{i=0}^{2} \frac{s_i}{R_i} + c\sum_{i=0}^{2} \frac{1}{R_i}$$

where $s_i \in \{-1, 0, 1\}$. Clearly, we see that if all $W_i = 1$, by adjusting the value of $V_i$ we can make $s_i$ equal 0 or 1. However, we can choose $s_i = -1$ by choosing $V_i = 0$ and changing $W_i$ to 0.

In general, a DAC with $2B$ resistors produces $3^B$ possible values of the form

$$c\sum_{i=0}^{B-1} \frac{s_i}{R_i} + c\sum_{i=0}^{B-1} \frac{1}{R_i}$$

An important property of this DAC is that it produces pairs of values which are exactly opposite in sign if the resistors are pairwise matched. It can be shown that this results in removal of all intermodulation products of the form

$$k\omega_1 \pm m\omega_2, \ k \text{ or } m \text{ even}$$

regardless of the choice of the $R_i$.

V. EXPERIMENTAL VERIFICATION

The proposed frequency mixer was tested using commercial programmable logic. We chose $N$, the number of partitions of the period, to be 10, and we chose $P$, the number of phases, to be 5. The phase shift $\theta$ was chosen as $\pi/10$. We chose the number of resistors per mixer, $2B$ to be 8, with the following values:

$$R_0 = 6R, \ R_1 = 7R, \ R_2 = 12R, \ R_3 = 23R$$
The corresponding non-negative output values of the DAC are given in table I (without DC offset); the negative values are the exact negative of the positive values. Our DAC has an equivalent resolution of approximately 18 bits for producing the desired values. However, it achieves this resolution using only eight small-ratio resistors.

<table>
<thead>
<tr>
<th>exact product</th>
<th>resistor network approximation of product</th>
<th>encoding of ROM s1, s2, s3, s4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6180340</td>
<td>0.6180308</td>
<td>1, 1, 1, 1</td>
</tr>
<tr>
<td>0.3819660</td>
<td>0.3819692</td>
<td>1, 1, 1, 1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0, 0, 1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0, 0, 0, 0</td>
</tr>
</tbody>
</table>

**TABLE I**

ALL POSSIBLE NON-NEGATIVE PRODUCTS OF THE APPROXIMATION FUNCTION FOR TEST CIRCUIT

The spectrum of a single phase mixer is shown in fig. 5 for a discrete-component implementation. The MATLAB simulation of its spectrum is shown in fig. 6.

In the complete multiphase mixer, we combine $P = 5$ phases at the output. Note that $P = 5$ and $P = 10$ will yield approximately the same spectrum. This is because $a(t)$ and $b(t)$ ideally only possess odd harmonics. Since the sum of two odd numbers is even, $k \pm m$ can only equal an even multiple of $P$ if $P$ is odd. The measurement of the spectrum is shown in fig. 7 and the MATLAB simulation is shown in fig. 8.

**VI. CONCLUSIONS**

Our test circuit demonstrates the feasibility of our proposed frequency mixer. A monolithic design would offer improved performance due to its faster ROM and closely matched delays. The imperfect multiphase cancellation was due to unequal delays through the routing of the programmable interconnect matrix and parasitics on the PCB. In finding the optimal resistors, we constrained resistances to be small integer multiples of a base value, thus allowing for accurate monolithic implementation. The same methods used to find these optimal resistances can be used to find optimal transistor sizes for a monolithic current-mode design.

**REFERENCES**