AN OPTIMIZED DIGITAL FREQUENCY SYNTHESIZER

Mohamed F. Mansour
Texas Instruments Inc.
DSP Solutions R&D Center
Dallas, Texas, USA

ABSTRACT
We develop an efficient algorithm for generating the complex vector \( \{\exp(j\omega_n)\}_{n=0}^{N-1} \) on standard fixed-point systems. The proposed algorithm employs optimal factorization of the recursion step to achieve different optimality criteria. It provides a good compromise between the contradicting requirements of memory, computation, and quantization error.

Index Terms—Frequency synthesizer, Fixed-point, Quantization.

1. INTRODUCTION
The generation of sinusoids is of central importance in many signal processing and communication systems. In particular, it is a fundamental part in the calculation of the Fast Fourier Transform (FFT), and the Discrete Cosine Transform (DCT), which are frequently encountered in digital receivers, and modern multimedia coding systems. This problem has been long studied, and many hardware and software solutions were proposed. In this work, we describe a novel algorithm that can be efficiently implemented on current commercial fixed-point DSP’s even with limited memory and resolution. In particular, we are interested in generating vectors of the form

\[ u[n] = \exp(j(\omega_n + \theta)) = \cos(\omega_n + \theta) + j\sin(\omega_n + \theta), \quad \text{for } 0 \leq n < N \]  

(1)

The algorithm proposed in this paper is a variation of the well known CORDIC ([1], [2]) and it is a generalization of classical digital frequency synthesizer [4], where fixed-point arithmetic results in excessive error accumulation. The main objective of the proposed algorithm is to minimize the propagation of the quantization error (which is unavoidable in fixed-point arithmetic) without using standard sigma-delta compensator that usually requires memory and computation resources comparable to the original algorithm itself. This is done using multiple incremental phase steps and factorizing the data vector to finer resolutions so as to minimize the recursion length for the worst case coefficient calculation. Many optimization criteria are proposed within the same framework, and their solutions are shown to be relatively simple.

2. PROPOSED ALGORITHM
The proposed algorithm is based on the two basic rotation equations:

\[
\begin{align*}
(x_{k+1}, y_{k+1}) &= (\cos \omega_o - \sin \omega_o, \sin \omega_o \cos \omega_o) (x_k, y_k) \\
(x_{k-1}, y_{k-1}) &= (\cos \omega_o, \sin \omega_o) (x_k, y_k)
\end{align*}
\]  

(2)

where \( x_k = \cos(k\omega_o + \theta) \) and \( y_k = \sin(k\omega_o + \theta) \). The above equations should be evaluated \( N \) times to generate the complex vector in (1) [4]. Direct calculation of the successive rotations results in excessive quantization error in fixed-point arithmetic (similar to that in recursive filter implementation). The signal to quantization noise ratio is proportional to \( N \) [3]. Moreover, the first order recursion limits the possibility of accelerating the algorithm using parallel structure. Rather than direct recursion, we propose using two phase steps to calculate the complex exponential in (1). These two phases provide two resolutions for phase change, one is a coarse resolution with step \( r\omega_o \) and the other is the fine resolution \( \omega_o \).

The basic idea of the algorithm is to first calculate the complex exponential values with coarse resolution at points that are multiple of \( r\omega_o \). These calculated points are then used as pivots to calculate the points in between using fine resolution. The total number of pivots in this case is \( N/r \). This simple structure reduces the maximum number of terms in the worst case calculation (which is proportional to the worst case signal to quantization noise ratio) to \( N/r + r - 2 \), rather than \( N - 1 \) in the direct recursion. For example, if \( N = 1024 \) and \( r = 32 \), the worst case calculation requires 62 terms rather than 1023 terms in direction recursion.

Instead of starting the recursion at \( k = 0 \), it may be started at \( k = N/2 \). In this case, we need both forward and backward recursion to calculate the pivot points with \( N/2r \) pivots in each direction. This significantly reduces the accumulative quantization noise in the pivots. Also in calculating the fine resolution points between two pivots, both forward and backward recursion are used. Between each two pivots \( x_k \) and \( x_{(k+1)r} \), we have \( r - 1 \) fine resolution points to be calculated. The first \( r/2 \) points are calculated using forward recursion.
starting from $x_{kr}$. The other $r/2 - 1$ points are calculated using backward recursion starting from $(x_{(k+1)r})$. With this forward/backward recursion, the maximum number of recursion is reduced to $N/2r + r/2 - 2$, e.g., for $N = 1024$, and $r = 32$, the worst case calculation requires only 30 terms. Moreover, there is no additional memory or computation cost of using forward/backward recursion. The overall algorithm is illustrated in Fig. 1.

The algorithm has another important advantage for parallel structures that are usually included in fixed-point DSPs. The algorithm allows four threads (for forward and backward recursions) to run in parallel during coarse resolution recursion and up to $4N/r$ parallel threads ($N/r$ pivots, with forward and backward recursion for each) during fine resolution recursion. This maximizes the utilization of parallel structures that may exist on the DSP.

Note that, the fine resolution points around the initial point may be calculated first and then the other points are generated recursively from them using coarse phase step. Although the two approaches are similar, we’ll focus on generating the coarse points first.

3. STATE-SPACE REPRESENTATION

In our analysis, we follow the conventional error model of fixed-point multiplication [3], where an error with uniform probability distribution is added to the correct output. The error model of the proposed algorithm is shown in Fig. 2. Note that, the error with forward or backward recursion is symmetric. Therefore we’ll focus only on forward recursion

$$\bar{x}_k = x_k + v_k, \quad \bar{y}_k = y_k + w_k$$  (4)

where $v_k$ and $w_k$ are the accumulative error signals in both branches. Using the above error model, we have

$$\bar{x}_{k+1} = \bar{x}_k \cdot \cos \omega_o - \bar{y}_k \cdot \sin \omega_o$$  \quad (5)

$$x_{k+1} + v_{k+1} = x_k \cdot \cos \omega_o + v_k \cdot \cos \omega_o - y_k \cdot \sin \omega_o - w_k \cdot \sin \omega_o + e_1(k+1) - e_2(k+1)$$  \quad (6)

A similar equation can be obtained for $\bar{y}_k$. Therefore we have

$$v_{k+1} = v_k \cdot \cos \omega_o - w_k \cdot \sin \omega_o + e_1(k+1) - e_2(k+1)$$  \quad (7)

$$w_{k+1} = w_k \cdot \cos \omega_o + v_k \cdot \sin \omega_o + e_3(k+1) + e_4(k+1)$$  \quad (8)

An important characteristic of the above equation is that, $\{e_i(k)\}_{i=1:4}$ are deterministic quantities. They are determined by the quantization error of $\cos \omega_o$ and $\sin \omega_o$. In quantizing the coefficients $\cos \omega_o$ and $\sin \omega_o$, we have the choice of picking the upper or the lower integer (the common practice is to round to the nearest integer). The choice of the quantized value of $\cos \omega_o$ and $\sin \omega_o$ may be different in the two branches of Fig. 2. We have,

$$e_1(k+1) - e_2(k+1) = \bar{x}_k \cdot \Delta_1 \cos \omega_o - \bar{y}_k \cdot \Delta_1 \sin \omega_o$$  \quad (9)

$$e_3(k+1) + e_4(k+1) = \bar{y}_k \cdot \Delta_2 \cos \omega_o + \bar{x}_k \cdot \Delta_2 \sin \omega_o$$  \quad (10)

where $\Delta_{1,2} \cos \omega_o, \Delta_{1,2} \sin \omega_o$ are the quantization error of the coefficients in the upper and lower branches respectively. The recursive error relations in sine and cosine calculations are summarized in (7)-(10). The state-space representation of this equation has the form,

$$v(k+1) = A v(k) + B u(k)$$  \quad (11)

where $v(k) = (v_k \ w_k)^T$ is the states vector (and the output), $u(k) = (x_k \ y_k)^T$ is the input, and $A$ is the state matrix, $B$ is the input matrix. They are defined as

$$B = \begin{pmatrix} \Delta_1 \cos a \omega_o & -\Delta_1 \sin a \omega_o \\ \Delta_2 \sin a \omega_o & \Delta_2 \cos a \omega_o \end{pmatrix}$$  \quad (12)

$$A = \begin{pmatrix} \cos a \omega_o & -\sin a \omega_o \\ \sin a \omega_o & \cos a \omega_o \end{pmatrix} + B$$  \quad (13)

with $a = r$ for coarse recursion, and $a = 1$ for fine recursion.

4. ALGORITHM OPTIMIZATION

There are many optimization criteria for the algorithm described in the previous section. The objective function could
be either minimizing the quantization error or maximizing the speed given a particular parallel hardware structure. In this section, we focus on minimizing the quantization error on fixed-point systems. The optimization problem considered here is a minimax optimization, where the objective of is to get the optimal coarse/fine factorization and quantized coefficients values so as to minimize the worst case quantization error. From (11), we may express the quantization error at any time as:

\[ \mathbf{v}(k + 1) = \mathbf{A}^{k+1}\mathbf{v}(0) + \sum_{n=0}^{k} \mathbf{A}^{k-n}\mathbf{B}\mathbf{u}(n) \quad (14) \]

Denote the coarse and fine state matrices as \( \mathbf{A}_c \) and \( \mathbf{A}_f \) respectively. Similar notations are used for all other parameters. If we assume that the coarse step is \( r\omega_o \), then we have in the worst case \( N/r - 1 \) coarse steps calculations, and \( r - 1 \) fine steps between each two coarse steps. If we adopt the forward/backward recursion described in section 2, then we have \( N/2r - 1 \) coarse steps calculations, and \( r/2 \) fine steps calculation. Then we have

\[ \mathbf{v}_c(0) = \mathbf{0}, \]

\[ \mathbf{v}_f(0) = \mathbf{v}_c(N/2r - 1) = \sum_{n=0}^{N/2r-1} \mathbf{A}_c^{N/2r-n}\mathbf{B}_c\mathbf{u}_c(n) \quad (15) \]

Therefore the worst case quantization error vector is:

\[ \mathbf{v}_f(r/2 - 1) = \mathbf{A}_f^{r/2} \sum_{n=0}^{N/2r-1} \mathbf{A}_c^{N/2r-n}\mathbf{B}_c\mathbf{u}_c(n) + \]

\[ \mathbf{A}_f^{r/2} \sum_{n=0}^{r/2} \mathbf{A}_f^{r/2-n}\mathbf{B}_f\mathbf{u}_f(n) \quad (16) \]

In this relation, we have

\[ \mathbf{u}_c(n) = (\cos((N/2 + n\omega_o + \theta) \sin((N/2 + n\omega_o + \theta)^T) \]

\[ \mathbf{u}_f(n) = (\cos((N - r + n\omega_o + \theta) + \sin((N - r + n\omega_o + \theta)^T) \]

The objective of the optimization problem is to minimize \( \|\mathbf{v}_f(r/2 - 1)\|^2 \). The variables in (16) are the coarse factor \( r \) and the quantized values of coarse and fine steps of the cosine and sine functions (which are the entries of \( \mathbf{B}_c \) and \( \mathbf{B}_f \)). The matrices in (16) are time-varying, i.e., \( \mathbf{B}_c(n) \) and \( \mathbf{B}_f(n) \). The quantized value of each coefficient is either the floor or the ceiling integer approximation. Therefore for a certain value of \( r \) the optimization problem is an unconstrained integer programming problem. At each iteration, the total number of unknowns is the number of entries in either \( \mathbf{B}_c(n) \) or \( \mathbf{B}_f(n) \) which equals four. Therefore the total number of unknowns (16) (for a certain choice of \( r \)) equals \( (2r + 2N/r) \). Note that, we neglect the error term in \( \mathbf{A}_c \) and \( \mathbf{A}_f \) because it results in a second order error after multiplication with \( \mathbf{B} \).

Analytic solution of (16) is in general expensive to calculate online on a fixed-point DSP. Moreover, if it is solved offline the solution (which is the quantization values at each time) may be expensive to store. Therefore we propose a suboptimal greedy solution and a direct implementation to it.

Instead of minimizing \( \|\mathbf{v}_f(r/2 - 1)\|^2 \), we minimize \( e_1(k) - e_2(k) \) and \( e_3(k) + e_4(k) \) in (9) and (10) at every iteration. Referring to Fig. 2, the objective is to choose the pairs that minimize \( e_1(k) - e_2(k) \) in the upper branch and \( e_3(k) + e_4(k) \) in the lower branch. This is done by testing all four combinations for both the upper and lower recursions in (9) and (10). This is repeated at every iteration of coarse and fine recursions. This in general increases the required computations by a factor of eight. To reduce this overload, the minimization may take place only in coarse recursion, and the conventional coefficients rounding may be used in fine recursion. In this case the computational increase by the ratio \( 8r/N \). This simplification is very effective at low resolutions.

Another simplification of the exhaustive search can be done by online updating of the quantization values based on the signal amplitude. Without loss of generality, we’ll assume that all the angles are in the first quadrant, i.e., \( \tilde{x}_k \) and \( \tilde{y}_k \) are positive for all \( k \). The criterion for minimizing the overall error in (9) and (10) is as follows:

\[
\begin{align*}
  &\text{if } \tilde{y}_k > \tilde{x}_k \\
  &\text{if } \sin \omega_o - |\sin \omega_o| < |\sin \omega_o| - \sin \omega_o \\
  &\text{Upper: } Q(\sin \omega_o) = [\sin \omega_o], \quad Q(\cos \omega_o) = [\cos \omega_o] \\
  &\text{Lower: } Q(\sin \omega_o) = [\sin \omega_o], \quad Q(\cos \omega_o) = [\cos \omega_o] \\
  &\text{else} \\
  &\text{Upper: } Q(\sin \omega_o) = [\sin \omega_o], \quad Q(\cos \omega_o) = [\cos \omega_o] \\
  &\text{Lower: } Q(\sin \omega_o) = [\sin \omega_o], \quad Q(\cos \omega_o) = [\cos \omega_o] \\
\end{align*}
\]

\[
\begin{align*}
  &\text{if } \cos \omega_o - |\cos \omega_o| < |\cos \omega_o| - \cos \omega_o \\
  &\text{Upper: } Q(\cos \omega_o) = [\cos \omega_o], \quad Q(\sin \omega_o) = [\sin \omega_o] \\
  &\text{Lower: } Q(\cos \omega_o) = [\cos \omega_o], \quad Q(\sin \omega_o) = [\sin \omega_o] \\
  &\text{else} \\
  &\text{Upper: } Q(\cos \omega_o) = [\cos \omega_o], \quad Q(\sin \omega_o) = [\sin \omega_o] \\
  &\text{Lower: } Q(\cos \omega_o) = [\cos \omega_o], \quad Q(\sin \omega_o) = [\sin \omega_o] \\
\end{align*}
\]

In this configuration, we choose the quantization with minimum error for the component that is multiplied by \( \max(\tilde{y}_k, \tilde{x}_k) \). Then we choose the quantization error of the other component with the appropriate sign such that the resultant quantization error in either branch is reduced. Note that, we do not need to compare \( \tilde{y}_k \) and \( \tilde{x}_k \) at each iteration as they are known a priori. For example, if we are calculating the first quadrant samples, then \( \tilde{y}_k > \tilde{x}_k \) if the accumulative angle is greater than \( \pi/4 \), and vice versa.

The optimization with respect to the coarse factor \( r \) is done using enumeration. The above algorithms may be performed for each possible value of \( r \) and the minimum is picked. In general (especially at high resolutions), the worst
case quantization error is proportional to the total number of coarse and fine recursions which equals \( N/2r + r/2 - 1 \). This is minimized when \( r = \sqrt{N} \).

5. RESULTS

The proposed algorithms are extensively tested to verify the improvement over previous approaches. The first test is to verify the signal-to-quantization noise improvement of the basic algorithm over the conventional recursive rotation algorithm. In this test, we used simple rounding of the coefficients. The comparison basis is the signal-to-quantization error of the whole complex vector (1). We used sequences of length 4096 samples at different resolutions. We used 50 sequences with different frequencies. Also, we used 64 coarse scales (i.e., \( r = 64 \)) for the proposed algorithm. From the figure, we note that, there is a gain of 35-40 dB over the conventional rotation algorithm at all resolutions.

![Fig. 3. Algorithm performance improvement over conventional recursive rotation](image)

Next, we assess the greedy optimization techniques described in section 4. The first algorithm uses exhaustive search over all possible values of quantized coefficients. The second algorithm performs simplified search based on the sinusoid values as described in the previous section. The performance of both optimizations is compared to the simple rounding in Fig. 4. From the figure we notice that, the exhaustive search algorithm improves the performance by 2-6 dB and the improvement is higher at high resolutions. The simplified algorithms has around 2 dB degradation from the exhaustive search algorithm. However, it performs poorly at low resolutions.

![Fig. 4. Performance of the optimized algorithms](image)

6. DISCUSSION

We described a new algorithm for sinusoid generation on fixed-point systems. The algorithm uses two frequency resolutions in addition to forward and backward recursions. The algorithm performance is more than 35 dB higher than conventional rotation at almost all resolutions.

The problem is formulated as a minimax optimization problem and we provide an approximate greedy solution to it. The proposed solution is around 5 dB higher than the algorithm with coefficients rounding. The proposed algorithm allows parallel computation because of the relatively independent computations. This is desirable for parallel coprocessor structures that are usually added to general-purpose DSPs.

The ideas presented in this paper are general to all stable first order recursive systems that follow the state-space representation in (11). Extending these algorithms to higher order recursive systems is a subject of future research.

7. REFERENCES


