A UNIFIED DESIGN FRAMEWORK FOR VECTOR ROTATIONAL CORDIC FAMILY
BASED ON ANGLE QUANTIZATION PROCESS

An-Yeu Wu
Dept. of Elec. Engin., National Taiwan Univ.,
Taipei, 106, Taiwan, R.O.C.

Cheng-Shing Wu
Dept. of Elec. Engin., National Central Univ.,
Chung-Li, 320, Taiwan, R.O.C.

ABSTRACT
Vector rotation is the key operation employed extensively in many
digital signal processing applications. In this paper, we
introduce a new design concept called Angle Quantization (AQ).
It can be used as a design index for vector rotational operation, where
the rotational angle is known in advance. Based on the AQ
process, we establish a unified design framework for cost-effective
low-latency rotational algorithms and architectures. Several existing
works, such as conventional CORDIC, AR-CORDIC, MVRCORDIC, and EAS-based CORDIC, can be fitted into the design
framework, forming a Vector Rotational CORDIC Family. Based
on the new design framework, we can realize high-speed / low-
complexity rotational VLSI circuits, whereas without degrading
the precision performance in fixed-point implementations.

1. INTRODUCTION
Vector rotation plays an important role in many digital signal pro-
cessing (DSP) applications. It is extensively employed as the
processing kernel in discrete orthogonal transformations (DCT, DST,
and DFT), lattice-based (rotation-based) digital filtering, sinewave
generation, and digital modulation/demodulation in communication
systems. Let \([x_{in}, y_{in}]^{T}\) and \([x_{out}, y_{out}]^{T}\) denote the input
and output vectors, respectively. Vector rotation of \([x_{in}, y_{in}]^{T}\) by
a rotational angle \(\theta\) can be formulated as

\[
\begin{bmatrix}
  x_{out} \\
  y_{out}
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
  x_{in} \\
  y_{in}
\end{bmatrix},
\]

Figure 1 shows the direct implementation of Eq. (1). As one can see,
the direct implementation is very area-consuming and low-
speed when rotational operations are heavily utilized in VLSI
circuits.

In this paper, we propose a novel framework to design high-
speed/low-cost vector rotational VLSI circuits. Instead of per-
foming quantization on the coefficient parameters (\(\cos \theta\) and \(\sin \theta\))
in fixed-point implementation, the proposed design framework origi-
nates from the concept of Angle Quantization (AQ). The AQ de-
rites the name from the fact that we perform the quantization pro-
cess on the rotational angle, \(\theta\), directly. That is, we decompose
the original rotational angle \(\theta\) into several sub-angles, \(\theta_i\)’s.
Then, we try to sum up those sub-angles to approximate the original angle
as close as possible; or equivalently, we try to minimize the angle
quantization error

\[
\xi_m = \theta - \sum_{i=0}^{N_A-1} \theta_i,
\]

where \(N_A\) denotes the number of sub-angles. The AQ process is
demonstrated in Fig. 2(a). Based on the AQ process, the vector
rotation operation can be realized as shown in Fig. 2(b). Each
rotation module is dedicated to performing a particular rotation
of sub-angle \(\theta_i\). Then, the rotation of \(\theta\) can be accomplished by
cascading these \(N_A\) rotation modules.

In the AQ process, there are two key design issues:
1. Firstly, we need to determine (or construct) the sub-angles,
and each \(\theta_i\) needs to be easy-to-implement in practical VLSI
circuits.
2. Secondly, we have to find out how to select and combine
these sub-angles such that the angle quantization error \(\xi_m\)
can be suppressed.

In fact, the well-known Coordinate Rotational Digital Computer
(CORDIC) algorithm \([1]\) can be considered as an approach to per-
fom the AQ process. Recall that in the CORDIC algorithm, the
rotation of angle \(\theta\) is performed by sequentially rotating element-
ary angle of \(a(i) = \tan^{-1}(2^{-i}),\) for \(0 \leq i \leq W - 1\), where
\(W\) denotes the wordlength. The advantageous feature of the ele-
mental angle is that rotation of \(a(i)\) requires only two shift-and-
add operators. The easy-to-implement feature of \( a(i) \) conforms to the requirements of aforementioned AQ process. In addition, the sequential rotating operation of \( a(i) \)'s is the way to select and combine those sub-angles in conventional CORDIC.

Next, we can link the AQ process with several existing vector rotation schemes such as Angle Recoding (AR) technique [2], Modified Vector Rotational CORDIC (MVRCORDIC) algorithm [3] and Extended Elementary Angle Set (EEAS) scheme [4]. We explore their relationship with the proposed AQ process. Then we will derive a unified framework for all these vector rotational operations. That is, all previous schemes can be considered as subsets of the proposed framework. The unified operations and AQ process of these algorithm suggest a family of rotation algorithms. We call it Vector Rotational CORDIC Family.

2. DESIGN FRAMEWORK FOR VECTOR ROTATIONAL OPERATIONS

2.1. Conventional CORDIC Algorithm

In conventional CORDIC algorithm, the elementary angles, \( a(i) \), is defined as \( a(i) \triangleq \tan^{-1}(2^{-i}) \) [1]. Based on the elementary angles, the conventional CORDIC algorithm can be rewritten as

\[
\theta = \sum_{i=0}^{N-1} \mu(i)a(i) + \xi_m,
\]

where \( N \) denotes the number of elementary angles, \( \mu(i) \in \{1, -1\} \) is the rotation sequence which determines the \( i^{th} \) rotational angle \( a(i) \). In general, for data of \( W \) bit wordlength, the iteration number is less than \( W \), i.e., \( N \leq W \). Basically, the CORDIC tries to decompose the rotation angle, \( \theta \), into the combination of \( a(i) \), for \( i = 0, 1, \ldots, N-1 \). The angle quantization error of the CORDIC algorithm

\[
\xi_{m,CORDIC} \triangleq \theta - \left[ \sum_{i=0}^{N-1} \mu(i)a(i) \right],
\]

represents the residue angle beyond the resolution of CORDIC algorithm.

2.1.1. Link AQ process with conventional CORDIC algorithm

Next, we would like to define Elementary Angle Set (EAS) for the derivation of the proposed vector rotational framework. Basically, EAS consists of all \( a(i) \), for \( 0 \leq i \leq N-1 \), and can be defined as

\[
S = \{ a(i) : 0 \leq i \leq N-1 \}.
\]

With the help of EAS, we can say that the CORDIC algorithm essentially performs the angle quantization. This can be observed from Eq. (3). Given a target rotation angle \( \theta \), CORDIC algorithm determines the first rotation angle \( \mu(0) \) for the most significant elementary angle \( a(0) \), followed by the determination of \( \mu(1) \) for \( a(1) \). The process is repeated until the last elementary angle is applied. That is, the CORDIC algorithm tries to perform the rotation through sequentially applying micro-rotations of all elementary angles.

Referring to Fig. 2, now we can relate AQ to CORDIC algorithm as follows: 1) The sub-angle \( \theta_i \) in AQ now becomes \( \theta_i = \mu(i)a(i) \) in CORDIC algorithm, 2) The number of sub-angles of \( N_A \) in AQ is set to be \( N \) in CORDIC algorithm, 3) CORDIC algorithm sequentially apply all \( \theta_i \), for \( i = 0, 1, \ldots, N-1 \), to approximate the target angle \( \theta \).

2.2. AR Technique [2]

In conventional CORDIC algorithm, the micro-rotations of all elementary angles are performed in a sequential way. On the contrary, in the Angle Recoding (AR) technique proposed by Hu and Nagathethan [2], certain micro-rotations can be skipped depending on the target rotational angle. Specifically, the modification is done by extending the set of \( \mu(i) \) from \( \{1, -1\} \) to \( \{1, -1, 0\} \). One can skip the micro-rotation of the elementary angle \( a(i) = \tan^{-1}(2^{-i}) \) by setting \( \mu(i) = 0 \). Now, the angle quantization error of the AR technique, \( \xi_{m,AR} \), can be represented as

\[
\xi_{m,AR} = \theta - \left[ \sum_{i=0}^{N-1} \mu(i)a(i) \right].
\]

Basically, Eq. (6) is identical to Eq. (4), except for the extended \( \mu(i) \in \{1, -1, 0\} \).

2.2.1. Link AQ process with AR technique

To make AR technique fit into our design framework, we reformulate Eq. (8) in a compact form as

\[
\xi_{m,AR} = \theta - \left[ \sum_{j=0}^{N'-1} \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \right] = \theta - \left[ \sum_{j=0}^{N'-1} \tilde{\theta}(j) \right],
\]

where \( N' \triangleq \sum_{i=0}^{N-1} |\mu(i)| \) denotes the effective iteration number, \( s(j) \in \{0, 1, \ldots, N-1\} \) is the rotational sequence that determines the micro-rotation angle in the \( j^{th} \) iteration, \( \alpha(j) \in \{-1, 0, 1\} \) is the directional sequence that controls the direction of the \( j^{th} \) micro-rotation of \( a(s(j)) \), and \( \tilde{\theta}(j) \) is the \( j^{th} \) micro-rotation angle, defined as \( \tilde{\theta}(j) = \tan^{-1}(\alpha(j) \cdot 2^{-s(j)}) \). As we can see from Eq. (9), the AR technique essentially tries to approximate \( \theta \) with the combination of selected angle elements from a pre-defined elementary angle set (EAS). The EAS consists of all possible values of \( \alpha(j) \)'s, and the EAS \( S_1 \) used in AR technique can be represented as

\[
S_1 = \left\{ \tan^{-1}(\alpha \cdot 2^{-s}) : \alpha \in \{-1, 0, 1\}, \ s \in \{0, 1, \ldots, N-1\} \right\}.
\]

(8)

The use of the subscript 1 will become apparent later in this section. With the EAS \( S_1 \) in hand, now we can easily link AR technique to the AQ process. By comparing Eq. (7) with the AQ approximation equation of Eq. (2), we find that AR technique indeed performs the angle quantization of target angle \( \theta \). The sub-angle \( \theta_i \) now corresponds to \( \tilde{\theta}(i) = \tan^{-1}(\alpha(i) \cdot 2^{-s(i)}) \) and \( N_A \) is set to be \( N' \).

2.2.2. Optimization Problem

We can consider the optimization problem of AR technique from EAS \( S_1 \) point of view. It can be re-stated as: Given \( \theta \), find the combination of elementary angles from EAS \( S_1 \), such that the angle quantization error \( |\xi_{m,AR}| \leq \alpha(N-1) \) and \( N' \) is minimized.
In [2], the *Greedy algorithm* is proposed to solve the optimization problem.

### 2.3. MVR-CORDIC Algorithm [3]

Based on the AR technique, in the *Modified Vector Rotational CORDIC (MVR-CORDIC) algorithm* [3], two more modifications are proposed.

1. Repeat of elementary angles:
   Referring to Eq. (6), in the AR technique, each micro-rotation angle of \(a(i) = \tan^{-1}(2^{-i})\) is allowed to be used only once. However, in the MVR-CORDIC algorithm, each micro-rotation of elementary angle can be performed repeatedly. The relaxed operation can result in more possible combinations of elementary angles, hence, smaller \(\xi_m\) can be expected.

2. Confinement of total micro-rotation number:
   From Eq. (7), we can see that the effective iteration number \(N^*\) in the AR technique is not fixed. For certain cases, \(N^*\) is large and very close to the upper bound of \(N/2\) [2]. In the MVR-CORDIC algorithm, we confine the iteration number in the micro-rotation phase to \(R_m\) \((R_m \ll W)\). The role of \(R_m\) is quite similar to the number of non-zero digits, \(N_D\), used in CSD recoding scheme; it will dominate the precision performance and the complexity.

#### 2.3.1. Link AQ Process with MVR-CORDIC algorithm

Putting all the aforementioned modifications together and ignoring the null operations, we can represent the angle quantization error of the MVR-CORDIC algorithm as

\[
\xi_{m,\text{MVR}} \triangleq \theta - \sum_{i=0}^{R_m-1} \alpha(i) a(s(i)) ,
\]

where \(s(i) \in \{0, 1, \ldots, W - 1\}\) is the rotational sequence that determines the micro-rotation angle in the \(i^{th}\) iteration, \(\alpha(i) \in \{-1, 0, 1\}\) is the directional sequence that controls the direction of the \(i^{th}\) micro-rotation of \(a(s(i))\). As one can find that the sub-angle of \((\alpha(i) a(s(i)))\) in Eq. (9) is exactly the same as the definition of \(\theta(j)\) in Eq. (7). Hence, the EAS formed by MVR-CORDIC algorithm is the same as AR technique.

Based on the Eq. (9), it is obvious that MVR-CORDIC algorithm also performs the AQ process as well. The major difference is:

1) The total number of sub-angles \(N_A\) in Fig. 2 (i.e., the total iteration number in the micro-rotation phase) is now kept fixed to a pre-defined value of \(R_m\) \((N_A = R_m)\).

2) The sub-angle \(\theta_i\) corresponds to \((\alpha(i) a(s(i)))\) in MVR-CORDIC algorithm, i.e., \(\theta_i = \alpha(i) a(s(i)) = \theta(i)\).

#### 2.3.2. Optimization Problem

In the application of MVR-CORDIC algorithm, the optimization problem can be stated from EAS point of view as: Given \(\theta\) and \(R_m\), find the combination of \(R_m\) elementary angles from EAS \(S_1\), such that the angle quantization error \(\xi_{m,\text{MVR}}\) is minimized.

In [3], *Semi-greedy algorithm*, which can provide tradeoffs between computational complexity and performance, is proposed to solve the optimization problem.

### 2.4. Extended EAS-based CORDIC Algorithm [4]

In *Extended Elementary Angle Set (EEAS)*-based CORDIC algorithm [4], in addition to applying the relaxation on \(\mu(i)\), we also relax the constraint of elementary angles by extending EAS \(S_1\). Then, we can have more choices (elementary angles) in approximating the target angle \(\theta\). It is expectable that the angle quantization error \(\xi_m\) can be reduced correspondingly.

#### 2.4.1. Extended EAS

By observing Eq. (8), we can see that the EAS \(S_1\) is comprised of *arctangent of single signed-power-of-two (SPT) term*. In the problem of SPT-based digital filter design, one effective way to increase the coefficient resolution (hence the filter performance) is to employ more SPT terms to represent the filter coefficients [5]. Motivated by this, we can easily extend the set by representing the elementary angles as the arctangent of the sum of two SPT terms [4]. That is,

\[
S_2 = \{\tan^{-1}(\alpha_0 \cdot 2^{-s_0} + \alpha_1 \cdot 2^{-s_1}) : \alpha_0, \alpha_1 \in \{-1, 0, 1\}, s_0, s_1 \in \{0, 1, \ldots, W - 1\}\}
\]

We call it *Extended Elementary-Angle Set (EEAS) \(S_2\)*. The subscript is used to denote the number of SPT terms.

Based on the EEAS \(S_2\) developed in Eq. (10), the sub-angle \(\theta_i\) in Fig. 2 now can be represented as

\[
\theta_i = \tan^{-1}(\alpha_0(j) \cdot 2^{-s_0(j)} + \alpha_1(j) \cdot 2^{-s_1(j)}) ,
\]

and the number of sub-angles \(N_A\) is set to be \(R_m\).

#### 2.4.2. Optimization Problem

With the derived EEAS \(S_2\), now the optimization problem of the EEAS-based CORDIC algorithm can be stated as: Given \(\theta\) and \(R_m\), find the parameters of \(\alpha_0(j)\), \(\alpha_1(j)\), \(s_0(j)\) and \(s_1(j)\) (i.e., the combination of elementary angles from EEAS \(S_2\)), such that the angle quantization error

\[
|\xi_{m, \text{EEAS}}| \triangleq \left| \theta - \sum_{j=0}^{R_m-1} \tan^{-1}(\alpha_0(j) \cdot 2^{-s_0(j)} + \alpha_1(j) \cdot 2^{-s_1(j)}) \right|
\]

can be minimized.

In [6], a novel searching algorithm, called *Trellis-based Searching (TRS) algorithm*, is proposed to solve the optimization problem.

### 2.5. Generalized EEAS Scheme

By following the similar idea of EEAS scheme, it is straightforward to insert more SPT terms in the representation of elementary angles. Hence, the size of EEAS can be increased. With more than two SPT terms, we call such an extension scheme *Generalized EEAS Scheme*. Specifically, the generalized EEAS with \(d\) SPT terms can be represented as

\[
S_d = \left\{\tan^{-1}(\alpha_0 \cdot 2^{-s_0} + \cdots + \alpha_{d-1} \cdot 2^{-s_{d-1}}) \right\},
\]

where \(\alpha_0, \cdots, \alpha_{d-1} \in \{-1, 0, 1\}, s_0, \cdots, s_{d-1} \in \{0, \ldots, W - 1\}\). As one can expect that the size of the EEAS increases exponentially as \(d\) increases. Consequently, with properly chosen design parameters, we can achieve higher precision performance in the AQ process.
### Table 1: Comparisons of members in the Vector Rotational CORDIC family.

<table>
<thead>
<tr>
<th>Vector Rotation Algorithm</th>
<th>Selection of Rotation Sequence</th>
<th>Elementary Angle Set</th>
<th>Micro-rotations</th>
<th>Angle Quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional CORDIC</td>
<td>$\mu \in {-1}$</td>
<td>EAS $S$</td>
<td>Complete</td>
<td>$\theta = a(i)$</td>
</tr>
<tr>
<td>Angle Recording Technique</td>
<td>$\mu \in {-1,0,1}$</td>
<td>EAS $S_i$</td>
<td>Schematic</td>
<td>$\theta = \tan^{-1}(\mu(i))$</td>
</tr>
<tr>
<td>MVR-CORDIC Algorithm</td>
<td>$\alpha \in {-1,0,1}$</td>
<td>EAS $S_i$</td>
<td>Schematic</td>
<td>$\theta = \tan^{-1}(\mu(i))$</td>
</tr>
<tr>
<td>EEAS Scheme</td>
<td>$\alpha, \beta, \gamma \in {-1,0,1}$</td>
<td>EAS $S_i$</td>
<td>Schematic</td>
<td>$\theta = \tan^{-1}(\mu(i))$</td>
</tr>
<tr>
<td>Generalized EEAS Scheme</td>
<td>$\alpha, \beta, \gamma, \delta \in {-1,0,1}$</td>
<td>EAS $S_i$</td>
<td>Schematic</td>
<td>$\theta = \tan^{-1}(\mu(i))$</td>
</tr>
</tbody>
</table>

### Table 2: Design example of rotation angle $\theta = 13\pi/32$, where the wordlength $W = 16$.

<table>
<thead>
<tr>
<th>Rotation Angle $d = 132/64$</th>
<th>Searching Algorithm</th>
<th>Rotational Sequence $\alpha$ and $\alpha$-Subangle $\phi$</th>
<th>Angle Approx. Error $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional CORDIC</td>
<td>$\theta = 0.001$</td>
<td>$\phi = 0$</td>
<td>0.0108*10^-6</td>
</tr>
<tr>
<td>Angle Recording Technique</td>
<td>Greedy Algorithm</td>
<td>$\theta = 0.001$</td>
<td>0.0108*10^-6</td>
</tr>
<tr>
<td>MVR-CORDIC Algorithm</td>
<td>$\alpha = [1 -1 -1]$</td>
<td>$\phi = [0 3 5]$</td>
<td>5.2809*10^-6</td>
</tr>
<tr>
<td>GS-greedy Algorithm (2)</td>
<td>$\alpha = [2 -1 -1]$</td>
<td>$\phi = [0 3 5]$</td>
<td>5.2053*10^-6</td>
</tr>
<tr>
<td>GB-Technique</td>
<td>$\theta = 0.001$</td>
<td>$\phi = [0 3 5]$</td>
<td>5.2053*10^-6</td>
</tr>
<tr>
<td>EEAS Scheme with $N_s = 2$</td>
<td>Greedy Algorithm</td>
<td>$\theta = 0.001$</td>
<td>2.9991*10^-6</td>
</tr>
<tr>
<td>GB-Technique</td>
<td>$\theta = 0.001$</td>
<td>$\phi = [0 3 5]$</td>
<td>4.8223*10^-6</td>
</tr>
<tr>
<td>TBS Algorithm</td>
<td>$\theta = 0.001$</td>
<td>$\phi = [0 3 5]$</td>
<td>7.136*10^-6</td>
</tr>
<tr>
<td>EEAS Scheme with $N_s = 3$</td>
<td>Greedy Algorithm</td>
<td>$\theta = 0.001$</td>
<td>3.2503*10^-6</td>
</tr>
</tbody>
</table>

Figure 3: Set diagram of vector rotational CORDIC family.

2.6. Family of Vector Rotational CORDIC Algorithm

So far, we have linked the AQ process with several existing vector rotation approaches, including CORDIC algorithm, Angle Recording technique, MVR-CORDIC algorithm, EEAS scheme, and Generalized EEAS scheme. All algorithms intend to realize the AQ process with various EAS and suitable combinations of sub-angles. That is, they try to decompose the target rotational angle into several easy-to-implement sub-angles, while minimizing the angle quantization error $\epsilon_m$ to obtain the best precision performance.

Based on our discussion, we can link all these rotation algorithms together under a unified design framework, from the AQ point of view. They form a family of vector rotational CORDIC algorithm, called Vector Rotational CORDIC Family. They all conform to the AQ process, but each rotational algorithm uses different AQ setting as summarized in Table 1.

Note that EEAS scheme covers MVR-CORDIC algorithm and the AR technique due to the fact that MVR-CORDIC and AR employ EAS $S_1$ as a searching space that is a subset of EEAS $S_2$. Moreover, MVR-CORDIC algorithm can also be treated as a subset of AR technique due to the fact that we impose one constraint on the total iteration number. Fig. 3 illustrates the relationships among members of vector rotational CORDIC family.

3. DESIGN EXAMPLE

In the design example, we consider the rotation angle of $\theta = 13\pi/32$ All algorithms in vector rotational CORDIC family derived in Section 2 are applied to perform the rotation of $\theta$. Meanwhile, aforementioned searching algorithms are conducted to solve the optimization problems. The results are summarized in Table 2.

4. CONCLUSION

In this paper, we introduce a new design index, called Angle Quantization. Following the new index, designers can explore a bigger design space in deriving low-cost/high-performance rotational circuits. As illustrated in [7], most popular DSP algorithms can be realized via rotational circuits. The new framework proposed in this paper can be employed to design the processing kernel of the DSP engine in [7].

References