A NOVEL TRELLIS-BASED SEARCHING SCHEME FOR EEAS-BASED CORDIC ALGORITHM

Cheng-Shing Wu

Dept. of Electrical Engineering,
National Central University,
Chung-Li, 320, Taiwan, R.O.C.

An-Yeu Wu

Dept. of Electrical Engineering,
National Taiwan University,
Taipei, 106, Taiwan, R.O.C.

ABSTRACT

The CORDIC algorithm is a well-known iterative method for the computation of vector rotation. For applications that require forward rotation (or vector rotation) only, the *Extended Elementary Angle Set (EEAS) Scheme* provides a relaxed approach to speed up the operation of the CORDIC algorithm. When determining the parameters of EEAS-based CORDIC algorithm, two optimization problems are encountered. In the previous work, *greedy algorithm* is suggested to solve these optimization problems. However, for the application that requires high-precision rotation operation, the results generated by greedy algorithm may not be applicable. In this paper, we propose a novel searching algorithm to overcome the aforementioned problem, called *Trellis-based Searching (TBS)* algorithm. Compared with the greedy algorithm used in the conventional EEAS-based CORDIC algorithm, the proposed TBS algorithm yields apparent performance improvement. Moreover, derivation of error boundary as well as computer simulations are provided to support our arguments.

1. INTRODUCTION

The *COordinate Rotational DIgital Computer (CORDIC)* algorithm is a well-known iterative technique to perform various basic arithmetic operations [1]. The algorithm is very attractive for hardware implementation because it uses only elementary shift-and-add operations to perform the vector rotation in 2D plane. The major problem of CORDIC algorithm is the slow computational speed because of the large iteration number. For some of these aforementioned applications, the large iteration number can be significantly reduced by taking the advantage of prior knowledge of the rotation angles. The reduction of iteration number comes from the relaxation of rotation sequence of $\mu(i)$. The set of $\mu(i)$ is extended from $\{-1, 1\}$ to $\{-1, 0, 1\}$, *i.e.,* some rotation of elementary angles is skippable. Such a technique is called *Angle Recoding (AR) technique* [2].

In our previous work, by taking the similar concept of AR technique, we proposed an algorithmic-level improvement scheme, called *Extended Elementary Angle Set (EEAS) scheme* [3]. Based on the EEAS scheme, the iteration number of CORDIC algorithm can be further reduced compared to that of AR technique, implying faster operation speed in iterative CORDIC structure, or reduced hardware complexity in parallel CORDIC structure [1].

In the application of EEAS scheme, two constrained optimization problems are encountered, one in the micro-rotation phase and the other in the scaling correction phase. Unlike conventional CORDIC algorithm, these two optimization problems raise due to the extension of rotation sequence. One intuitive solution is to perform the exhaustive searching algorithm. However, the computational complexity is extremely high to be practically employed. Instead, *greedy search* can be used to solve the optimization problems [2][3]. In general, the greedy algorithm leads to local optimal solution to the optimization problem; it trades the error performance for the computational complexity. For situations that require high-precision (*i.e.,* high-SQNR) rotational operations, the greedy algorithm may not be applicable.

To deal with the aforementioned optimization problems, in this paper, we propose a novel searching algorithm, called *Trellis-based Searching (TBS) algorithm*. The proposed searching algorithm operates in the similar way to the trellis search (also known as Viterbi decoding algorithm [4]), which is frequently used in decoding convolutional codes in communication systems. Moreover, by exploring the relationship between greedy algorithm and the proposed TBS algorithm, we can show that the proposed TBS algorithm always has superior precision performance to the greedy algorithm. In addition, the mathematical derivation is supported by extensive computer simulations. The simulation results show that the averaged residue angle error generated by TBS algorithms is only about 15% of the greedy algorithm.

2. REVIEW OF EEAS-BASED CORDIC ALGORITHM

In the conventional CORDIC algorithm, each elementary angle needs to be performed sequentially so as to complete the micro-rotation phase. However, in the applications where the rotation angles are known in advance, it would be advantageous to relax the sequential constraint on the micro-rotation phase. The *Angle Recoding (AR) technique* is done by extending the set of $\mu(i)$ from $\{1, -1\}$ to $\{1, -1, 0\}$ [2]. With the relaxation on $\mu(i)$, for certain angles, we can obtain better approximation of θ (*i.e.,* smaller residue angle, ϵ) but with reduced iteration number.

Motivated by this, we proposed an algorithmic-level improvement scheme, called *Extended Elementary-Angle Set (EEAS) scheme*, in our previous work [3]. In addition to the relaxation on $\mu(i)$, the EEAS scheme further applies the relaxation on *the elementary angles*. We first treat the elementary angles as the set of *arctangent of single signed-power-of-two (SPT) term*. From such a different viewpoint, employing one additional SPT term provides a very straightforward way to relax the constraint of conventional elementary angles, and the precision/range of the elementary angle set can be significantly extended. Specifically, given the wordlength W, the elementary angle set (EAS) $\mathcal{S}_1$ and EEAS $\mathcal{S}_2$ can be represented in the form of

$$\begin{aligned} \mathcal{S}_1 &= \left\{\tan^{-1}\left(\alpha^\star \cdot 2^{-s^\star}\right)\right\} \\ \mathcal{S}_2 &= \left\{\tan^{-1}\left(\alpha_0^\star \cdot 2^{-s_0^\star} + \alpha_1^\star \cdot 2^{-s_1^\star}\right)\right\} \end{aligned} , \qquad (1)$$

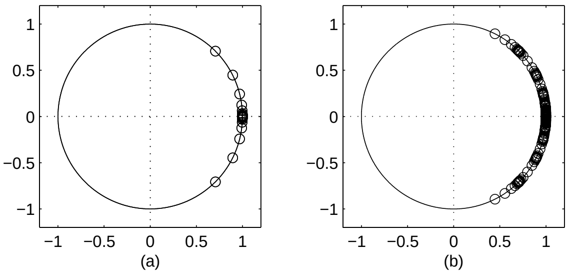

Figure 1: Constellation of (a) EAS $\mathcal{S}_1$ (b) EEAS $\mathcal{S}_2$, with wordlength W=8.

```
% Initilization
Given initial vector of $[x(0), y(0)]^T$
% Micro-rotation phase
For $j$ = 0 to $R_m$-1
    $\begin{bmatrix} x(j+1) \\ y(j+1) \end{bmatrix} = \begin{bmatrix} 1 & -\alpha_0(j)\,2^{-s_0(j)} - \alpha_1(j)\,2^{-s_1(j)} \\ \alpha_0(j)\,2^{-s_0(j)} + \alpha_1(j)\,2^{-s_1(j)} & 1 \end{bmatrix} \begin{bmatrix} x(j) \\ y(j) \end{bmatrix}$
END
% Sclaing phase
For $m$ = 0 to $R_s$-1
    $\begin{bmatrix} \tilde{x}(m+1) \\ \tilde{y}(m+1) \end{bmatrix} = \begin{bmatrix} k_0(m)\,2^{-q_0(m)} + k_1(m)\,2^{-q_1(m)} & 1 \\ 1 & k_0(m)\,2^{-q_0(m)} + k_1(m)\,2^{-q_1(m)} \end{bmatrix} \begin{bmatrix} \tilde{x}(m) \\ \tilde{y}(m) \end{bmatrix}$
END
```

Table 1: Summary of the EEAS-based CORDIC scheme.

where $\alpha^\star$, $\alpha_0^\star$, $\alpha_1^\star \in \{-1, 0, 1\}$, and $s^\star$, $s_0^\star$, $s_1^\star \in \{0, 1, \cdots, W-1\}$. In fact, the use of EEAS scheme has the effect of improving the SQNR performance of the CORDIC algorithm in vector rotation mode. The reason is that now we have more choices of elementary angles in approximating the target rotation angle.

To demonstrate the concept of EEAS scheme, the constellation of the elementary angles from $\mathcal{S}_1$ and $\mathcal{S}_2$ are shown in Fig. 1 (a) and (b), respectively. As we can see, the number of elementary angles in $\mathcal{S}_2$ is much larger than $\mathcal{S}_1$. This implies that EEAS $\mathcal{S}_2$ can yield better error performance than $\mathcal{S}_1$ with a fixed number of micro-rotations of elementary angles (iterations).

Moreover, to improve the scaling phase of the EEAS-based CORDIC algorithm, we introduce the technique of *Extended Type-II (ET-II) scaling operation* to perform the scaling operations. It helps to preserve the vector norm after the CORDIC operation [3]. The ET-II scaling operation inherits the feature of EEAS scheme in the micro-rotation phase, which makes the VLSI architecture of the EEAS-based CORDIC algorithm regular and reusable. It can also achieve much smaller quantization error than conventional Type-I scaling operation (*Canonical-Signed-Digit (CSD)-based representation*) in the approximation of scaling factor.

Table 1 summarized the detailed recurrence equations of EEAS-based CORDIC algorithm, including the EEAS scheme in the micro-rotation phase and ET-II scaling operation in the scaling phase. The readers may refer to [3] for detailed operations.

3. OPTIMIZATION PROBLEMS AND GREEDY ALGORITHM

Unlike conventional CORDIC algorithm, two optimization problems are encountered due to the modified basic operations in the EEAS-based CORDIC algorithm, one in the micro-rotation phase and the other one in the scaling phase. In the micro-rotation phase, we try to minimize the residue angle error ξ_m, which represents the difference between target angle θ and the angle that can be composed by elementary angles from EEAS $\mathcal{S}_2$. The residue angle error ξ_m relates to the parameters in Table 1 in the form of

$$\xi_m = \left| \theta - \sum_{j=0}^{R_m-1} \tan^{-1} \left[\alpha_0(j) \cdot 2^{-s_0(j)} + \alpha_1(j) \cdot 2^{-s_1(j)} \right] \right| . \tag{2}$$

In the scaling phase, we try to minimize the quantization error of ξ_s, which represents the difference between the ideal scaling factor and the quantized one that can be represented by ET-II scaling operation. Similarly, the residue angle error ξ_s relates to the parameters in Table 1 in the form of

$$\xi_s = \left| \frac{1}{P} \right| \cdot \left| P - \prod_{m=0}^{R'_s-1} \left[1 + k_0(m) \cdot 2^{-q_0(m)} + k_1(m) \cdot 2^{-q_1(m)} \right] \right| , \tag{3}$$

where P denotes the floating-point represented scaling factor.

These two optimization problems can be solved by the greedy algorithm (GA) [3]. The greedy algorithm tries to approximate the remaining angle using a closest elementary angle at each search step without looking ahead of future steps. By successively applying such an operation, the accumulated angle can continually approach the target angle θ until the searching algorithm is terminated; it terminates when no further improvement can be found, or the $R_m{}^{th}$ micro-rotation angle is determined.

4. THE PROPOSED TRELLIS-BASED SEARCHING (TBS) ALGORITHM

In this paper, we propose a novel searching algorithm, called the *Trellis-based Searching (TBS) algorithm*, to solve the optimization problem described in Eqs. (2) and (3). To facilitate our discussion, we use an simple example to illustrate the proposed TBS algorithm. Suppose that we need to perform the rotation of angle $\theta = \pi/3$, the wordlength $W = 3$, and the maximum iteration number is restricted to $R_m = 4$.

Step 1. Initialization

First of all, let $Z(\mathcal{S}_2)$ denote the number of the elementary angles in the extended set $\mathcal{S}_2$, and each distinct elementary angle in the set is expressed as $r(k)$, for $1 \leq k \leq Z(\mathcal{S}_2)$, *i.e.*, $\mathcal{S}_2 = \{r(1), r(2), \cdots, r(Z(\mathcal{S}_2))\}$. In this example, $Z(\mathcal{S}_2) = 15$.

In the TBS algorithm, there are $Z(\mathcal{S}_2)$ states in each step. For k^{th} state ($1 \leq k \leq Z(\mathcal{S}_2)$) of i^{th} search step, we use the *Cumulative Angle*, $\phi(i, k)$, to denote the *best approximation* of angle θ in the k^{th} state up to the i^{th} step. The TBS algorithm is performed column-wise from left to right. Initially, we start the TBS algorithm by setting all $\phi(1, k)$ as the corresponding elementary angles. That is, $\phi(1, k) = r(k)$ for all k, which is illustrated in the left-most part of Fig. 2.

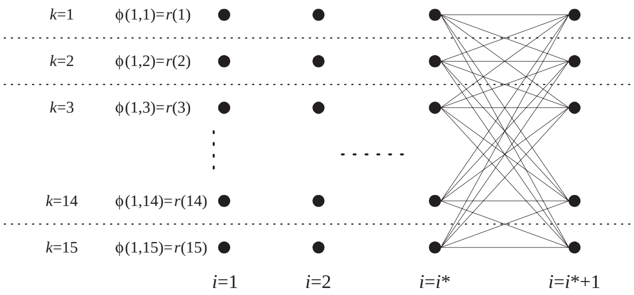

Figure 2: Illustration of initialization process and transition paths of the proposed TBS algorithm.

Step 2. Accumulation

A path in the trellis, which leaves the k^{th} state at i^{th} step and enters the k'^{th} state at $(i+1)^{th}$ step, corresponds to an operation of adding $\phi(i,k)$ by $r(k')$. Then, the appended angle of $\phi(i,k)+r(k')$ becomes the candidate for $\phi(i+1,k')$. Moreover, as shown in the right part of Fig. 2, from a given state at step i, the paths can diverge to all the states at the next search step (i^*+1). Namely, there are $Z(\mathcal{S}_2)$ paths, carrying the corresponding appended angles of $\phi(i^*,k)+r(k')$ for all k, enters the k'^{th} state at $(i^*+1)^{th}$ step. Then, those appended angles form the candidate set for the cumulative angle of $\phi(i^*+1,k')$.

Step 3. Comparison and Selection

Conceptually, the whole process is similar to the trellis decoding of convolutional code [4]: The TBS algorithm involves calculating and minimizing the difference between the target angle θ and $\phi(i,k)$ for all k at each search step i. To be specific, $\phi(i+1,k)$ is determined such that

$$|\phi(i+1,k)-\theta| = \min_{1\le k^\star \le Z(\mathcal{S}_2)} |\phi(i,k^\star)+r(k)-\theta|\,. \qquad (4)$$

Then, the selected path is denoted as the *surviving path*. Note that we have to calculate all the cumulative angles $\phi(i,k)$ for all k (thus their corresponding surviving paths) before moving to the $(i+1)^{th}$ step. Continuing in this manner, we can successively advance deeper into the trellis (set $i=i+1$), until the maximum iteration number is reached ($i=R_m$).

Consider our design example, in which $i=2$ and $k=12$. The process of Eq. (4) is illustrated in Fig. 3. In this case, the third path ($k^\star$=3), which is marked by the solid line, is selected. Then, the resultant angle is assigned to $\phi(3,12)$ for the subsequent search process.

Step 4. Determination of the Global Result and Trace Back

After calculating all the cumulative angles at the last search step, the next procedure for the TBS algorithm is to determine the *global result*, θ_{TBS}. Similar to the determination of surviving path, we decide θ_{TBS} as follows

$$|\theta_{TBS}-\theta| = \min_{1\le k^\star \le Z(\mathcal{S}_2)} |\phi(R_m,k^\star)-\theta|\,. \qquad (5)$$

Next, we can determine all the micro-rotations by tracing from the state, whose corresponding $\phi(R_m,k)$ is best approximation of θ, along its surviving path backward.

In our example, the procedure for trace back is illustrated in Fig. 4. All the surviving paths for each state at each step are represented by the dash line. First, $\phi(4,13)$ is selected as the *global result*. Then, trace along the surviving path that connecting the 13^{rd} state at final step backward. Next, find the state from which the surviving path leaves in the previous step. By doing the process repeatedly, we can thus determine the *global surviving path* of the TBS algorithm, as marked by the solid line in Fig. 4. By travelling along the global surviving path, we can find the visited states and read all the micro-rotation angles that form the global result θ_{TBS}. In this case, $\phi(4,13)=r(9)+r(13)+r(2)+r(13)$, which is the best approximation of angle θ generated by the proposed TBS algorithm.

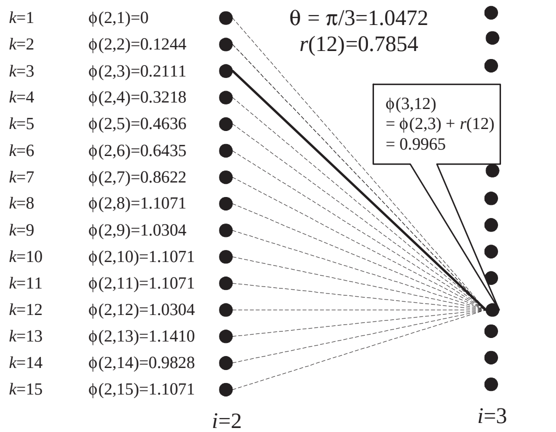

Figure 3: Illustration of the determination of surviving path.

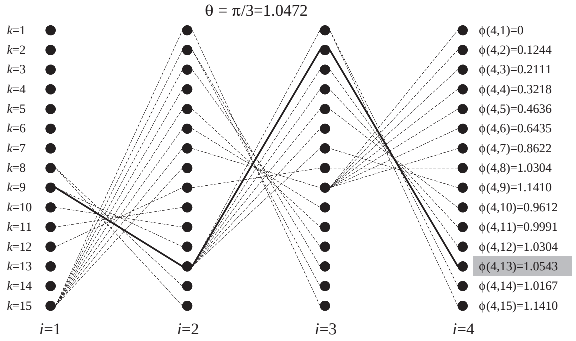

Figure 4: Illustration of trace back procedure.

5. COMPARISON OF TWO SEARCHING ALGORITHMS

5.1. Error Bound of TBS Algorithm

Since the greedy algorithm performs a progressive searching procedure in nature, for comparison purpose, we also illustrate the greedy algorithm in the form of the trellis diagram. Consider the example of $\theta=3\pi/8$, $R_m=4$, and $W=16$. In Fig. 5, the trellis-like representation of the GA process (marked by the bold dash line) is demonstrated. In our discussion, we call such a line as the *greedy path*.

¿From Fig. 5, we can easily explore the difference between the greedy and TBS algorithm. As we can see, for the greedy algorithm, only one single surviving path is allowed from one step to next step; while in the TBS algorithm, there are $Z(\mathcal{S}_2)$ simulta-

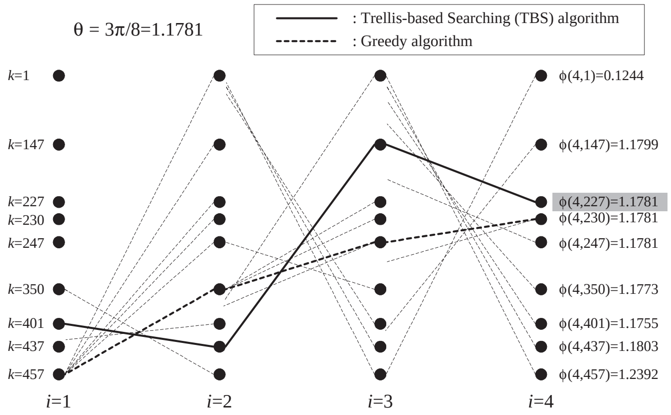

Figure 5: Comparison between greedy algorithm and the proposed TBS algorithm.

neous surviving paths run in parallel. That is, at each search step of greedy algorithm, we are forced to select the best path among $Z(\mathcal{S}_2)$ candidates and eliminate all the other paths before moving to next step.

Lemma: Let ξ_{GA} and ξ_{TBS} denote the residue angle error provided by the greedy algorithm and the TBS algorithm, respectively. *It is guaranteed that the TBS algorithm outperforms the greedy algorithm in terms of residue angle error, i.e.,* $\xi_{TBS} \leq \xi_{GA}$.

Proof: First, assume that the first segment of greedy path starts at l^{th} state at 1^{st} step and enters the k^{th} state at 2^{nd} step (in this case, $l = 457$ and $k = 338$). It is noteworthy that the segment of the greedy path must be one of the candidates of the surviving path of k^{th} state at 2^{nd} step in the TBS algorithm. Recall that $\phi(2,k)$, the best approximation of θ along its surviving path up to the 2^{nd} step in k^{th} state, is determined such that $\phi(1,l') + r(k)$ for $1 \leq l' \leq Z(\mathcal{S}_2)$ is closest to θ. This implies that the difference between the cumulative angle $\phi(2,k)$ and θ must be equal to or smaller than the difference between the greedy results up to 2^{nd} step and θ.

Advancing deeper into the trellis by tracing along the greedy path, we find that the statement described above holds for all search step i. That is, for any state at any search step visited by the greedy path, say k^{*th} state at i^{*th} step, the corresponding cumulative angle $\phi(i^*, k^*)$ is always a better (or equally well) approximation of θ than the result of greedy algorithm up to step i^*. Of course, the statement is also true for $i = R_m$. In the last step, moreover, the determination of the global surviving path further pushes the result of TBS algorithm toward the target angle θ. Let θ_{TBS} and θ_{GA} be the resultant angles of the TBS and greedy algorithm, respectively. Then, we have

$$|\theta_{TBS} - \theta| \leq |\phi(R_m, k^\star) - \theta| \leq |\theta_{GA} - \theta| \,, \qquad (6)$$

where $k^\star$ denotes the state that the greedy path terminated at the last search step. It follows that

$$\xi_{TBS} \leq \xi_{GA}. \quad \square \qquad (7)$$

5.2. Performance Simulation

In the simulation, 4097 uniformly spaced rotation angles in the region from 0 to $\pi/2$, *i.e.,* $\theta = 0, \frac{1\cdot\pi}{8192}, \frac{2\cdot\pi}{8192}, \cdots, \frac{4096\cdot\pi}{8192}$, are performed. Both GA and TBS algorithms are applied to solve the optimization problem of EEAS-based CORDIC algorithm. They are run for different values of wordlength, W. The simulation results are shown in Fig. 6.

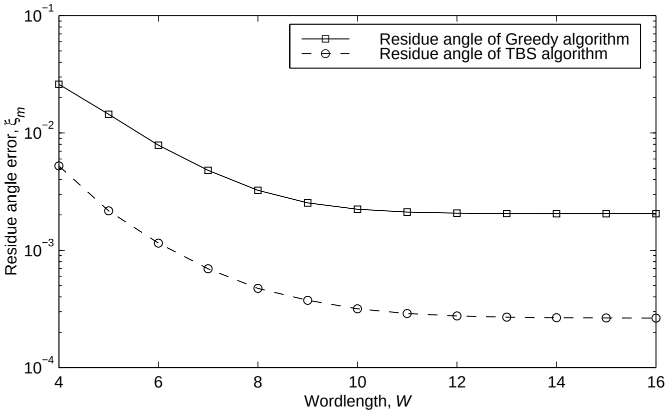

Figure 6: Performance comparison between Greedy algorithm and TBS algorithm with $R_m = 2$.

The results shown in Fig. 6 show that we can obtain superior error performance with the proposed TBS algorithm over the greedy algorithm for all W. This confirms our arguments in Section 5.1. The residue error of the proposed TBS algorithm is only about 15%, in an averaged sense, compared to the greedy algorithm. Combined with the EEAS-based CORDIC algorithm, the improved precision performance also implies that given a precision target, vector rotation can be accomplished with fewer shift-and-add operations. That is, we can reduce the hardware complexity while maintain the precision performance.

6. CONCLUSIONS

In this paper, we presented a novel searching algorithm based on trellis-based operation to deal with the optimization problems encountered in the application of EEAS-based CORDIC algorithm. With the proposed TBS algorithm, the averaged residue angle error is significantly reduced to only about 15% compared to that of greedy algorithm. The improvement makes applications, which call for high complexity, feasible, such as high-point/high-speed discrete transformations (FFT, DCT) and high-order digital lattice filters.

References

[1] Y. H. Hu, “CORDIC-based VLSI architectures for digital signal processing,” *IEEE Signal Processing Magazine*, pp. 16–35, July 1992.

[2] Y. H. Hu and S. Naganathan, “An angle recoding method for CORDIC algorithm implementation,” *IEEE Trans. on Computers*, vol. 42, pp. 99–102, Jan. 1993.

[3] C. S. Wu and A. Y. Wu, “A novel rotational VLSI architecture based on Extended Elementary-angle Set CORDIC algorithm,” in *Proc. IEEE 2^{nd} IEEE Asia Pacific Conference on ASICs*, (Cheju, Korea), pp. 111–114, 2000.

[4] G. David Fornet, Jr., “The Viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, pp. 268–277, March 1973.