No-overflowing Magnification Scheme in Redundant CORDIC algorithm and its implementation in FPGA

Zhang Jing*

School of electrical information engineering, Wanjiang University of Technology, Ma’anshan City, Anhui Province, 243031, P. R. China
515069431@qq.com

Abstract. In this paper, normalization and magnification methods to maintain the calculating accuracy of CORDIC algorithm for small amplitude inputs are discussed, especially the overflowing boundaries are analyzed through mathematical deduction and the correctness has been proven by simulation. Finally, the FPGA implementations of different methods are conducted, which shows that the proposed magnification factor chosen scheme has an effective reduction on hardware resources consumption compared with the conventional normalization method.

1. Introduction
The coordinate rotation digital computer (CORDIC) algorithm is a powerful numerical approximation method for computing a variety of complicated and nonlinear elementary functions [1][2], and is widely used in digital signal processing applications, such as digital waveform generation [3], high-speed discrete transformations [4][5], and matrix decompositions[6-8]. In actual applications, since the power of a feedback signal is varying, the vectoring CORDIC which is used to derive the amplitude and angle information from the signal may has a variable input over a wide range. To maintain a certain computational accuracy of the implemented CORDIC algorithm, the input should be normalized or magnified when the input amplitude is small.

2. Basic Principles of CORDIC
The basic rotation equation of CORDIC algorithm is

$$\begin{align*}
  x_{i+1} &= x_i - d_i \cdot \frac{y_i}{2^{i-1}} \\
  y_{i+1} &= y_i + d_i \cdot \frac{x_i}{2^{i-1}} \\
  z_{i+1} &= z_i - d_i \cdot a_i
\end{align*}$$

(1)

where $i=1, 2, 3, \ldots, N$, is the index of rotation stage, $N$ is the total number of rotation stages; $d_i$ indicates the rotation direction of the $i$-th rotation, $d_i = -1$ if $y_i \geq 0$ and $d_i = 1$ otherwise in vectoring CORDIC; $a_i$ represents the fixed rotation angle of the $i$-th rotation, which is given by

$$a_i = \arctan\left(\frac{1}{2^{i-1}}\right),$$

(2)
Generally, the values of $a_i$ are pre-calculated, quantized, and stored in the algorithm program. After the total $N$ stages of rotation, the final outputs of Vectoring CORDIC are

$$[x_{N+1}, y_{N+1}, z_{N+1}] = [K \cdot A_m, 0, \phi_m],$$

where $K$ is a scale gain, and

$$K = \prod_{i=1}^{N} \sqrt{1 + 2^{-2i}},$$

$A_m$ and $\phi_m$ is the amplitude and angle of the input vector $(x_m, y_m)$, respectively, as

$$A_m = \sqrt{x_m^2 + y_m^2},$$

$$\phi_m = \arctan \frac{y_m}{x_m}. $$

3. Normalization and Magnification methods

In theory, the computing error of CORDIC algorithm is comprised with two kinds of errors [9], the inherent approximation error because of finite rotation stages and the unavoidable truncation error because of finite representation accuracy. In actual digital implementation, there is a third kind error [10], i.e., rotation error, which is involved when the truncation error is accumulated during the rotations and has a much greater effect on the final computational accuracy especially for small inputs.

3.1 Input normalization

As for small inputs, normalization of the input is an effective method to improve the final computing accuracy. The modified rotation expression of normalization CORDIC is

$$\begin{align*}
\Delta x_{i+1} &= \Delta x_{i} - d_{i} \cdot \frac{\Delta y_{i}}{2^{2i}}, \\
\Delta y_{i+1} &= \Delta y_{i} + d_{i} \cdot \frac{\Delta x_{i}}{2^{2i}}.
\end{align*}$$

where $\Delta x_i = 2^i \cdot x_i$, $\Delta y_i = 2^i \cdot y_i$, $2^i$ is the normalization multiple, which is a power of two for ease of implementation; $d_{i} = -1$ if $\Delta y_i \geq 0$, $d_{i} = 1$ otherwise.

3.2 Key variable magnification

As a matter of fact, the correct selection of rotation direction individually depends on the sign of variable $y$ in vectoring CORDIC. In hence, magnification of variable $y$ with no magnifying the value of variable $x$ is enough. Furthermore, variable $y$ is in downdraught to zero, which allows an increased multiplication factor during rotations rather than a fixed multiple as in normalization CORDIC. The rotation equation of this key variable magnification is [11]

$$\begin{align*}
x_{i+1} &= x_i - d_{i} \cdot \frac{\Delta y_{i}}{2^{2i+1} \cdot 2^{2i}}, \\
\Delta y_{i+1} &= 2 \cdot \Delta y_{i} + d_{i} \cdot x_i \cdot 2^{2i+2},
\end{align*}$$

where $\Delta y_i = 2^{i+1} \cdot y_i$, and $2^{i+1}$ is the multiple factor according to the rotation stage index.

4. Overflow Boundary and the chosen of Magnification Factor

Compared to the basic CORDIC, normalization and key variable magnification CORDICs involve an overflowing issue to maintain the convergence of CORDIC algorithm.

4.1 Overflow boundary of basic CORDIC

Assume the variable $x$ and $y$ are signed and b-bit width wide. In the rotation of equation (1), the variable $x$ is a non-decreasing positive value, in hence, the no-overflow $x$ should satisfy

$$x_{\text{max,bs}} = x_{N+1} \leq 2^{b-1} - 1.$$ 

Thus, according to equation (3) and (4), the no-overflow input amplitude is required as
\[ A_{\text{in,BS}} \leq \frac{(2^{b-1} - 1)}{K}. \]  

If a vector satisfied equation (9) inputs, the variable \( y \) is forwarding 0 during rotations, and would not overflow in equation (1).

We consider this boundary \( A_{\text{in}} \in (0, \frac{(2^{b-1} - 1)}{K}] \) in equation (9) as a basic full input range, restricted only by the constant scale factor \( K \), should be fulfilled by every advanced CORDIC.

4.2 Overflow boundary of normalization CORDIC (NM CORDIC)

As in equation (6), the variable \( x \) and \( y \) are magnified by a fixed multiple \( 2^s \), the overflow situation is similar to equation (1) except

\[ x_{\text{max,NM}} = 2^s \cdot x_{i+1} \leq 2^{b-1} - 1, \]  

where the maximum \( s \) is \((b-2)\) for avoiding left shift overflow.

Therefore, the no-overflow input amplitude in NM CORDIC is required as

\[ A_{\text{in,NM}} \leq \frac{(2^{b-1} - 1)}{2^s} \cdot \frac{1}{K}. \]  

4.3 Overflow boundary of key variable magnification CORDIC (KVM CORDIC)

A relationship between the residual angle and the fixed rotation angle of the last rotation stage is concluded as

\[ |\delta_{y_i}| \leq a_{y_i}. \]  

A further promotion of equation (12) is that the residual angle and the fixed rotation angle of each rotation stage has a relationship as

\[ |\delta_i| \leq a_i. \]  

Substituting \( |\delta_i| = \arctan\left(\frac{y_i}{x_i}\right) \) and equation (2) into equation (13), in hence,

\[ \arctan\left(\frac{y_{i+1}}{x_{i+1}}\right) \leq \arctan\left(\frac{1}{2^{s-i}}\right), \]  

\[ 2^{s-i} \cdot |y_{i+1}| \leq x_{i+1}. \]  

A particularly different situation is involved in equation (7). Taking the multiple \( 2^{s+2} \) into consideration, the magnified variable \( \Delta y \) may overflow even the variable \( x \) does not. The no-overflow has a necessary condition that

\[ 2^{s+2} \cdot |\Delta y_{N+1}| \leq 2^{s+2} \cdot x_{N+1} \leq 2^{b-1} - 1. \]  

where the maximum \( s \) is \((b-4)\) for avoiding left shift overflow.

Thus, the no-overflow input amplitude in KVM CORDIC is required as

\[ A_{\text{in,KVM}} \leq \frac{(2^{b-1} - 1)}{2^{s+2}} \cdot \frac{1}{K}. \]  

Since \( s = -2 \) and \( \Delta y_i = 2^{s-i} \cdot y_i \), then,

\[ \Delta y_{i+1} = 2^{s-i} \cdot y_{i+1}. \]  

Hence,

\[ |\Delta y_{N+1}| \leq x_{i+1}. \]  

As a result, the magnified variable \( \Delta y \) in equation (7) will not overflow before the variable \( x \) does. Compared the \( x \) computing path in equation (7) with in equation (1), we can find that the two \( x \) computing paths have an equal expression essentially. In other words, the two \( x \) variables have a same overflow boundary.

A specific KVM CORDIC with a constant magnification factor, \( s = -1 \), was proposed in [11], and was inherited in papers [6], [7], [12]–[16]. In all these papers, with the same magnification multiple and without proof of convergence, a complicated selection function and several correcting iterations are usually included for the implementation of scale factor redundant CORDIC.
In fact, the no-overflowing of variables indicates the convergence of the algorithm. As shown in equation (11) and equation (17), for no overflow during rotation processes, the input range of NM and MV CORDIC will be narrowed by the multiple $2^s$ and $2^{s+2}$, respectively. As a result, hardware consumption will be substantially declined when realize a NM or KVM CORDIC in digital processing chip.

5. Implementation in FPGA
Implementation of different CORDIC methods is carried out on the platform of Xilinx’s ISE-12.4 targeted at a Virtex-5 LX50T device, the structures are shown in figure 1, and the detailed rotation structure of each stage is shown in figure 2. To avoid computing overflowing after normalization, an extra evaluation stage is added in the normalization stage in NM CORDIC as shown in figure 1(b). The evaluation of the amplitude of input vector $(x_{in}, y_{in})$ is

$$A_{in} \leq \max\{||x_{in}||, ||y_{in}||\} + \min\{||x_{in}||, ||y_{in}||\}/2$$

(20)

Here, the normalization factor $s$ is depending on $A_{in}$. As shown in figure 1(c), the magnification stage of the KVM CORDIC is placed after rotation-1, where

$$x_1 \geq |y_1|$$

(21)

Hence, the magnification factor $s$ can be totally chosen depending only on $x_1$.

![Figure 1](image-url)

Figure 1. Implementation structures of different CORDICs, (a) is basic CORDIC; (b) is NM CORDIC; (c) is KVM CORDIC
In normalization CORDIC, if a veritable amplitude result is preferred, an extra compensation stage should be added after the last stage to compensate the \( 2^i \) multiplication. Hence, it is obviously that the proposed KVM CORDIC is simpler to implement than NM CORDIC.

Table 1 illustrates the hardware resources consumed for implementations of basic, KVM, and NM CORDIC. It shows that KVM CORDIC has a least resource consumption while NM CORDIC has a maximum one.

|                  | Basic CORDIC | KVM CORDIC | NM CORDIC |
|------------------|--------------|------------|-----------|
| Slice Registers  | 661          | 658        | 740       |
| Slice LUTs       | 1253         | 852        | 1504      |

6. Simulation and Verification

The derived data are analyzed in MATLAB R2013b. During simulations, the major parameters are kept identical, e.g., \( b=16, \ N=14, \ \text{ANG WIDTH} \), the width of variable \( z \), is assigned to 16. According to the above simulation parameters and equation (16), the basic full input range of amplitude can be carried out as (0, 19897].

As presented in figure 3(a), with the increasing of the input amplitude, an impressive improvement in angle computational accuracy achieved by using KVM and NM CORDIC compared with using basic CORDIC. The maximum inadequate improvement compared KVM CORDIC to NM CORDIC is 9.85 degree, while the maximum improvement compared KVM CORDIC to basic CORDIC is 46.72 degree. Figure 3(b) shows that the angle error of KVM CORDIC approaches to the same level as NM CORDIC as soon as the input amplitude increases to about 852, while figure 3(c) shows that the angle error of basic CORDIC approaches to the level of NM CORDIC when the input amplitude increases to about 2971. Figure 3(d) displays that overflowing happens when the input amplitude beyond the basic input range. In this graph, lines of NM and basic CORDIC have been overlapped by each other.

Taking figure 3 as well as table 1 into consideration, KVM CORDIC is a suitable choice of magnification method with an acceptable improvement in computational performance and an effective reduction in hardware resources consumption.
Figure 3. Simulation result of different CORDICs

Table 2 lists the no-overflow boundaries according to the different choice of in KVM and NM CORDIC. The theoretical values are the computing results of equation (11) or equation (17), while the simulated values are picked out from the simulated data where amplitude error or angle error is varied suddenly with an obvious value change. As we can find, the theoretical values are close to the relative simulated values in both NM and KVM CORDIC. Therefore, table 2 verifies the correctness of equation (11) and equation (17).

| $p$ | Theoretical Value | KVM CORDIC ($s = -2$) | NM CORDIC |
|-----|-------------------|------------------------|-----------|
| 0   | 19897.9           | 19900.8                | 19897.3   |
| 1   | 9948.9            | 9952.5                 | 9949.0    |
| 2   | 4974.5            | 4976.4                 | 4974.5    |
| 3   | 2487.2            | 2492.6                 | 2487.2    |
| 4   | 1243.6            | 1246.5                 | 1243.6    |
| 5   | 621.8             | 625.3                  | 621.7     |
| 6   | 310.9             | 312.6                  | 310.8     |
| 7   | 155.5             | 157.7                  | 155.4     |
| 8   | 77.7              | 80.0                   | 77.6      |
| 9   | 38.9              | 39.0                   | 38.8      |

$^a p = s + 2$ in KVM CORDIC, and $p = s$ in NM CORDIC.

7. Conclusion
For the implementation of CORDIC algorithm in circular vectoring mode, the proposed key variable magnification method, which maintains the full basic no-overflow input range, provides a useful improvement not only in computational accuracy, but also in reducing hardware resources consumption, as long as the multiple is selected properly.

References
[1] J. E. Volder. (1959) The CORDIC Trigonometric Computing Technique. IRE Trans. Electron. Comput., vol.EC-8, no.3, :330-334.
[2] J. S. Walther. (1971) A unified algorithm for elementary functions. Proc. Spring Joint Computing Conf., :379-385.
[3] S. Aggarwal, P. K. Meher, K. Khare. (2013) Scale-Free Hyperbolic CORDIC Processor and Its Application to Waveform Generation. IEEE Trans. Circuits Syst. I, Reg. Papers, vol.60, no.2, :314-326.

[4] S. Y. Park, Y. J. Yu. (2012) Fixed-Point Analysis and Parameter Selections of MSR-CORDIC with applications to FFT Designs. IEEE Trans. Signal Process., vol.60, no.12, :6245-6256.

[5] H. Huang, L. Y. Xiao. (2013) CORDIC Based Fast Radix-2 DCT Algorithm. in IEEE Signal Process. Lett., vol.20, no.5, :483-486.

[6] J. Lee, T. Lang. (1991) SVD by constant-factor-redundant-CORDIC. Proc.10th Symp. Comput. Arithmetic, :264-271.

[7] M. D. Ercegovac, T. Lang. (1990) Redundant and on-line CORDIC: application to matrix triangularization and SVD. IEEE Trans. Comput., vol.39, no.6, :725-740.

[8] S. D. Munoz, J. Hormigo. (2015) High-Throughput FPGA Implementation of QR Decomposition. IEEE Trans. Circuits Syst. II, Expr. Briefs, vol.62, no.9, :861-865.

[9] Y. H. Hu. (1992) The quantization effects of the CORDIC algorithm. IEEE Trans. Signal Processing, vol.40, no.4, :834-844.

[10] X. Hu, S. C. Bass. (1993) A neglected error source in the CORDIC algorithm. Proc.1993 IEEE ISCAS ’93, vol.1, :766-769.

[11] J. A. Harding, T. Lang, J. A. Lee. (1991) A comparison of redundant CORDIC rotation engines. Conf. Computer Design 91, :556-559.

[12] D. Timmermann, H. Hahn, B. J. Hosticka. (1992) Low latency time CORDIC algorithms. IEEE Trans. Comput., vol.41, no.8, :1010-1015.

[13] J. A. Lee, T. Lang. (1992) Constant-factor redundant CORDIC for angle calculation and rotation. IEEE Trans. Comput., vol.41, no.8, :1016-1025.

[14] J. A. Lee, K. Kim. (1992) Fully-pipelined VLSI architectures for the kinematics of robot arm manipulators. Proc.11th AlPC, :80-86.

[15] J. A. Lee, M. Ahmad. (1994) VLSI implementation of CORDIC angle units. Proc. 4th GLSV ’94, :144-149.

[16] E. Antelo, J. D. Bruguera, T. Lang, E. L. Zapata. (1997) Error analysis and reduction for angle calculation using the CORDIC algorithm. IEEE Trans. Comput., vol.46, no.11, :1264-1271.