

Design and Error Analysis of a Scale Free CORDIC Unit with Corrected Scale Factor

N Prasad, Ayas Kanta Swain, and K. K. Mahapatra
Dept. of Electronics and Communication Engineering
National Institute of Technology, Rourkela, India - 769008

Abstract –This paper presents architecture of CORDIC, embedded with a scaling unit that has only minimal number of adders and shifters. It can be implemented in rotation mode as well as vectoring mode. The purpose of the design is to get a scaling free CORDIC unit preserving the design of original algorithm. The proposed design has a considerable reduction in hardware when compared with other scaling free architectures. The analysis of error for different word lengths and different input ranges for fixed word length gives a better choice to choose the parameters. The error in rotation mode for 16 bit data path, obtained for ordinate equivalent input is 0.073% and for abscissa equivalent input is 0.067%. The proposed design implemented in Xilinx XC3S500E-4FG320 FPGA, fabricated in 90 nm process technology, consumes 503 slices and 984 4-input look up tables (LUTs). The proposed design has a maximum frequency of operation of 75.593 MHz and a slice delay product of 104.645.

Index Terms – Scaling free CORDIC, FPGA, DSP, slice delay product.

I. INTRODUCTION

Coordinate Rotation Digital Computer (CORDIC) unit has become an essential and inevitable hardware block in modern engineering and scientific applications. It serves many applications such as solving trigonometric and transcendental equations, in digital signal processing (DSP) for Fourier basis based orthogonal transforms, in computer technology for 3D graphics, in digital communication systems for modulation and demodulation of the signals, etc. [1] – [4]. As the purpose of CORDIC to be used in many applications has increased, more hardware friendly and computational friendly architectures are designed. These designs have either lesser computation in Z data path or lesser number of iterations and in turn lesser execution time in XY data path. One family of above designs is scaling free CORDIC design. Many scaling free architectures are designed to improve either execution time or hardware reduction [5] – [8]. Though these designs show better performance in terms of one parameter, they have drawback when considering others.

Very few scaling free architectures follow the algorithm of CORDIC that was initially proposed.

The designs that follow conventional CORDIC algorithm have a constant correction factor based on theoretical value. But, the scaling constant is not same for all combinations of inputs in terms of their range and the word length. The proposed design considers optimum scaling factor, for each input that is X input and Y input. Thus the design proposed in this paper has a better accuracy compared to other scaling free CORDIC units that follow conventional algorithm.

The rest of this paper is aligned as follows. Section II reviews the CORDIC algorithm and its scale free architectural implementation. Section III deals with the proposed design with corrected scale factor and its error analysis. Section IV shows the results of the proposed design and section V concludes the paper.

II. REVIEW OF CORDIC AND ITS SCALING FREE IMPLEMENTATION

The conventional CORDIC was first implemented by Volder, in 1959 [1]. The basic equations of the algorithm for circular coordinate system are shown below.

$$\begin{aligned} X_{i+1} &= [X_i - 2^{-i} Y_i] * K_i \\ Y_{i+1} &= [Y_i + 2^{-i} X_i] * K_i \\ Z_{i+1} &= Z_i - \arctan(2^{-i}) \end{aligned} \quad (1)$$

The above set of equations is considered for positive angle of rotation. If the angle is negative, the arithmetic signs get reversed. The index i represents the number of iteration of the unit since the number of iterations depends on the precision we require. Figure 1 shows the CORDIC stage for i^{th} iteration. The scaling constant for each iteration, K_i is formulated as below.

$$K_i = \cos(\arctan(2^{-i})) \quad (2)$$

The congruence constant, obtained after all iterations is shown as

$$K = \prod_i K_i \quad (3)$$

Where $i = 0$ to $N-1$, N is the number of bits in the xy data path or the precision of the inputs. Thus, mathematical value of K approximates to be

$$K = 0.60725 \quad (4)$$

To eliminate the hardware required to compute the above constant after performing the algorithm, many have proposed alternate algorithms [5] – [8]. The work proposed in [5], with parallel compensation of scale factor, has shown two methods namely double rotation method and bit analysis method, compensates the scaling factor in parallel while carrying out the algorithm. Though the scale factor is compensated in parallel, additional hardware such as multiplexers and adders gets added in each stage of iteration making the architecture a bulky one. The one presented in [6], called MSR_CORDIC algorithm carries out computations fastly compared to conventional CORDIC but it also has a drawback of additional shifters ($2i+1$ shifters) and adders which increase hardware.

The design proposed in [7], uses additional shifters and adders compared to conventional architecture. This design even depends on a parameter called basic shift, which limits the angle of rotation and more care has to be taken while mapping the angle to entire coordinate space. The above design is called modified virtually scaling free CORDIC. Another architecture using generalized micro-rotation selection is proposed in [8]. In this, they have approximated the angles of \sin and \cos using Taylor series expansion, as shown below.

$$\begin{aligned} \sin \alpha &= \alpha - (3!)^{-1} \cdot \alpha^3 + (5!)^{-1} \cdot \alpha^5 \dots \\ \cos \alpha &= 1 - (2!)^{-1} \cdot \alpha^2 + (4!)^{-1} \cdot \alpha^4 \dots \end{aligned} \quad (5)$$

This recursive architecture though has better performance compared to that of others in same family has hardware overhead compared to conventional CORDIC. The advantage of this architecture is that it has lesser slice delay product compared to that of other scaling free architectures.

In enhanced scaling free CORDIC proposed in [9], they have used radix 4 booth encoding to perform the algorithm. The disadvantage of this is that it performs rotation only in one direction. From this architecture, it is evident that even this has higher hardware compared to conventional CORDIC but the advantage lies in faster computation of the vector rotation.

Thus, the architectures mentioned above for obtaining scale free CORDIC mostly have much hardware overhead compared to conventional CORDIC which makes the designers to concentrate

on designing scale free architectures which have lesser or comparable hardware overhead to that of conventional CORDIC. Though latency is another issue in these designs, pipelined designs always have better latency compared to fully dedicated architectures.

The proposed design has been implemented using the stages mentioned in figure 1.

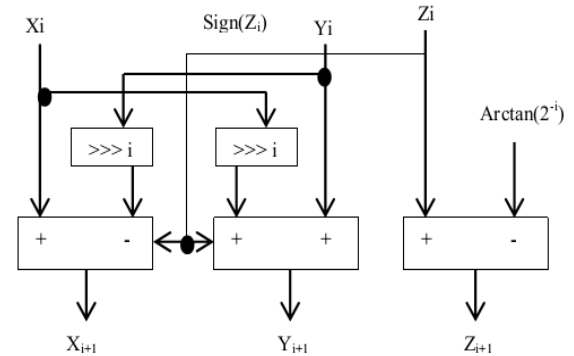


Fig. 1. CORDIC stage for i^{th} iteration

Figure 2 shows the CORDIC stage used in [7] for first half iterations.

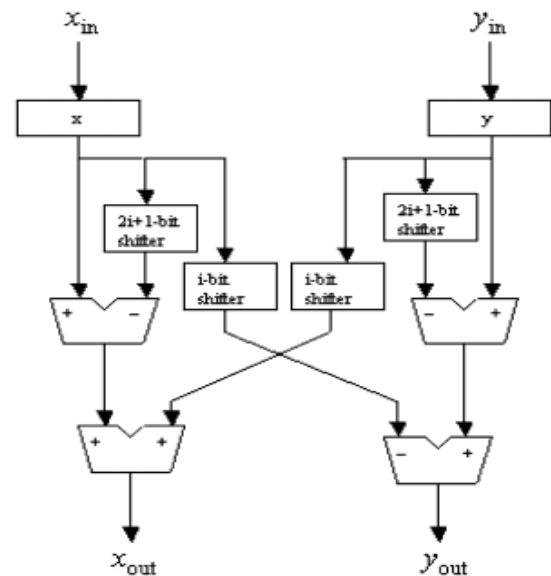


Fig. 2. CORDIC stage for $i < b/2$ [7]

III. DESIGN AND ANALYSIS OF PROPOSED ARCHITECTURE WITH CORRECTED SCALE FACTOR

From (4), the value of the congruence constant is 0.60725. This is theoretical value and practical values deviate from the mentioned value and the deviation is different for different ranges of inputs. Table I shows the values of congruence scale factor

for different ranges of inputs. For analysis, we considered the data path width of 16 bits.

TABLE I. VARIATION OF FOR DIFFERENT RANGES OF INPUTS FOR 16 BIT DATA PATH (TAKEN USING SIMULATION)

Xin = Yin	Xout	Yout	Scale factor x	Scale factor y
15	24	26	0.6250	0.5769
63	103	104	0.6117	0.6058
511	837	846	0.6105	0.6040
4095	6718	6768	0.6096	0.6051
16383	26882	27075	0.6094	0.6051

Thus, from table I, the scaling factor approaches to the mathematical value at higher width of data path, which is more complex in terms of hardware. This is also shown in figure 3. Thus, we approximated the scale factor for the data path with width as 16 bits and developed our CORDIC unit with the architecture for corrected scaling factor. In table I, scale factor for x input and for y input is different. Thus our scaling architecture has concentrated to build separate scaling blocks for x data path as well as for y data path. The practical values considered in designing our scaling units are given below.

$$\begin{aligned} K_x &= 0.6128190983 \\ K_y &= 0.6010718683 \end{aligned} \quad (6)$$

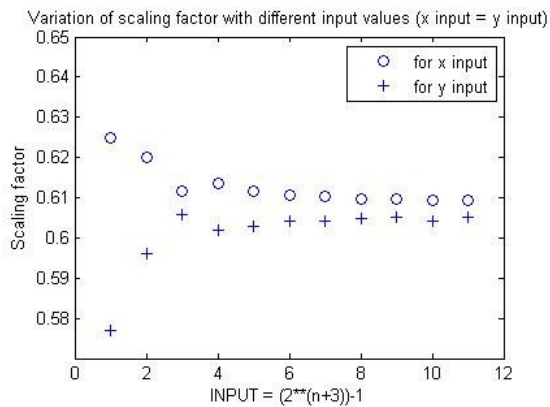


Fig. 3. Variation of scale factor for different ranges of inputs for 16 bit data path

The values given in the equation (6) are rms values of possible scale factor values, thus making it more robust for a particular data path. Since we are following the original algorithm in developing our scale free CORDIC unit, the basic structure of CORDIC unit does not change. After the last stage of iteration, we have introduced the scaling stages, which take one extra clock cycle to scale the coefficients and to give the outputs. Since our design is implemented using pipelining, latency

issues are taken care of. As seen in equation (6), there is much difference between theoretical and practical congruence constant.

The scaling units are designed using hardwired shifters and adders thus minimising latency issues those arise in adding the hardware to the existing design. The scaling units are well approximated to get more accurate outputs.

Figure 4 shows the scaling unit for x data path and figure 5 shows the scaling unit for y data path.

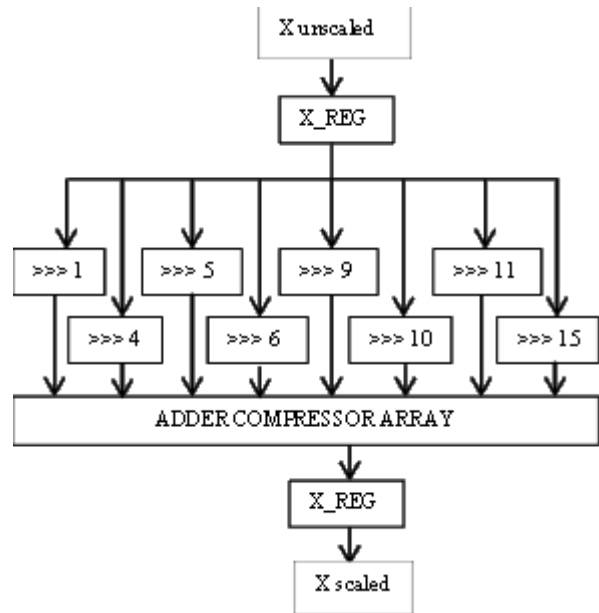


Fig. 4. Scaling unit for x data path (16 bit)

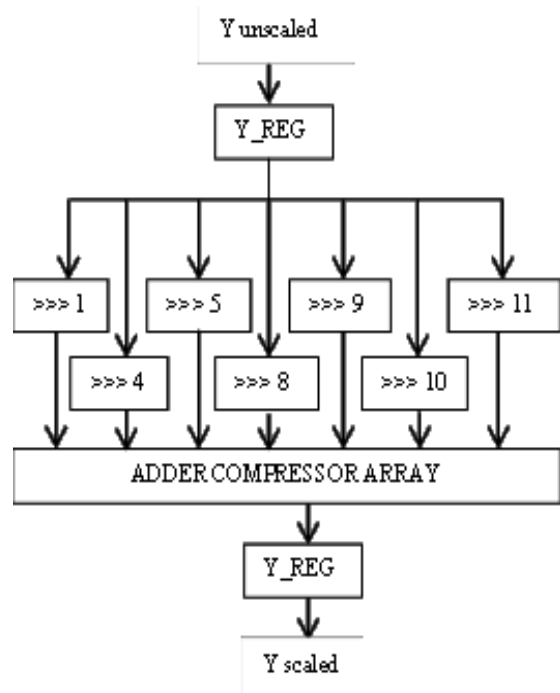


Fig. 5. Scaling unit for y data path (16 bit)

In figures 4 and 5, '>>> i' indicates right shift by i bits. The adder compressor array can be designed using carry save architecture or ripple carry architecture, depending on speed and area requirements.

The schematic of the top module of the proposed design is shown in figure 6, that shows where the scalers of figures 4 and 5 fit in the design. The sub module, CORDIC_OC_PIPELINE, in the figure 6 is a cascade of 16 stages of CORDIC units, each looks like the one mentioned in figure 1. The architecture of sub module SCALER_X_10102012 is based on the one in figure 4 and that of SCALER_Y_10102012 is based on the one in figure 5.

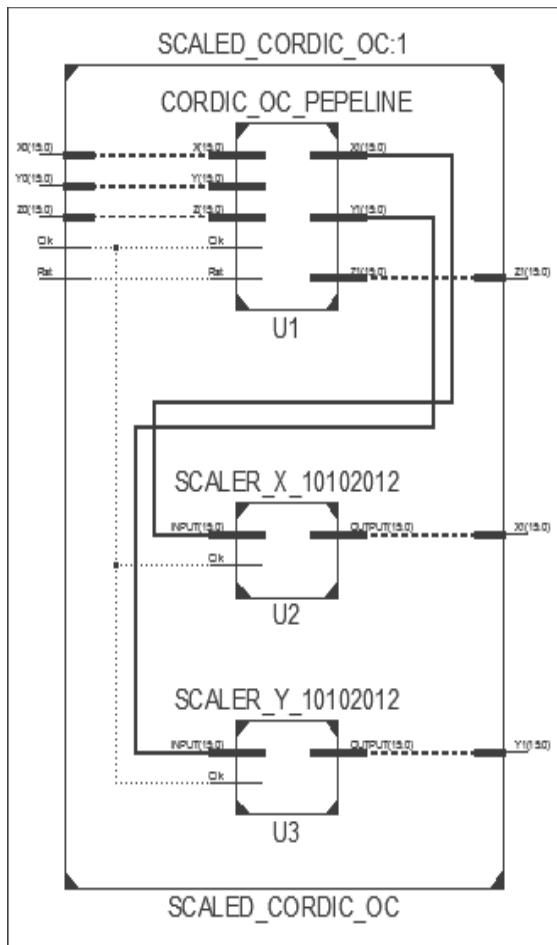


Fig. 6. Schematic of top module of proposed design (taken using Xilinx ISE)

IV. RESULTS OF THE PROPOSED DESIGN

The error analysis is done for the above scaling units after embedding them to the unscaled CORDIC unit. Table II compares output and input for different range of inputs, showing the accuracy of scaled CORDIC unit. The same is shown in figure 7.

TABLE II. COMPARISON OF X AND Y OUTPUTS WITH INPUTS FOR 16 BIT DATA PATH (TAKEN USING SIMULATION)

Xin = Yin	Xout	Yout	% error in x	% error in y
15	14	13	6.67	13.33
63	61	61	3.17	3.17
255	252	252	1.18	1.18
1023	1019	1018	0.39	0.49
4095	4090	4089	0.12	0.15
16383	16372	16371	0.067	0.073

Thus from table II, it is evident that, increase in range of inputs increases the accuracy in scaled outputs. For the proposed design, the error between inputs and scaled outputs, for maximum range of inputs is 0.067% for abscissa equivalent input (x data path) and 0.073% for ordinate equivalent input (y data path). This can be further reduced in increasing the width of the data path.

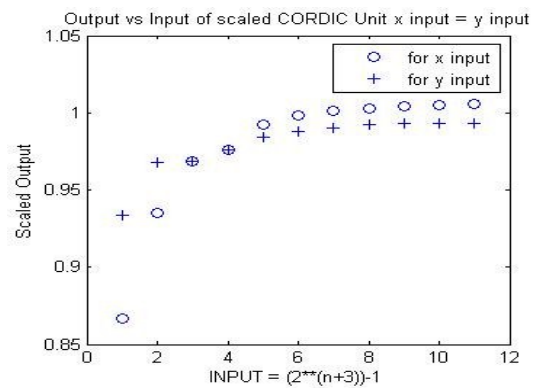


Fig. 7. Plot of input/output vs. input for 16 bit data path

The proposed method is most suitable if the angle of rotation is fixed, since the scaling factor can be well approximated to get accurate outputs. Figure 8 shows the scaled output of CORDIC unit when angle of rotation is 45° .

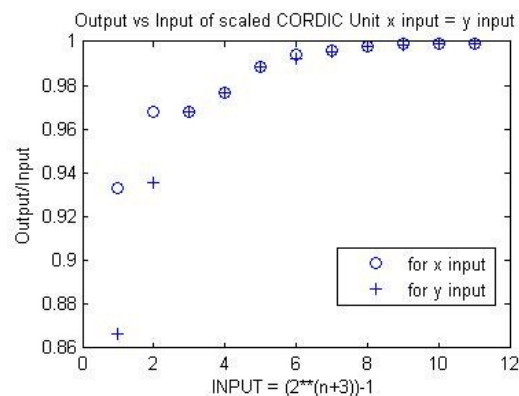


Fig. 8. Plot of output/input vs. input for angle of 45° for 16 bit data path

Thus from figure 8, accuracy of more than 99.99% can be obtained. The figure 8 is plotted using the output data of Xilinx ISim for input range from 15 to 16383. The range is mentioned as x label in figure 8.

Table III shows the device utilization summary when the design is implemented in Xilinx XC3S500E-4FG320 FPGA.

TABLE III. DEVICE UTILIZATION SUMMARY OF PROPOSED DESIGN FOR 16 BIT DATAPATH

Logic utilization	Used	Available	Utilization
Number of Slices	503	4656	10%
Number of Slice Flip Flops	798	9312	8%
Number of 4 input LUTs	984	9312	10%

The maximum frequency of operation is 75.593 MHz. The total number of adders required by the proposed design is 62. Thus the slice delay product (SDP) can be given by

$$SDP = (\text{No. of slices} * \text{worst case iterations}) / \text{freq. in MHz.} \quad (7)$$

As the number of worst case iterations in the proposed design is 16, the slice delay product comes to 106.465.

Figure 9 shows the Xilinx simulation result of scaled outputs, when scaled through mathematical congruence constant. For the set of figures 9 and 10, values in first two rows represent x input and y input respectively. Zero in third row represents the angle of rotation is 0, which is obvious. The corresponding outputs are represented in rows 6 and 7, for x and y, respectively.

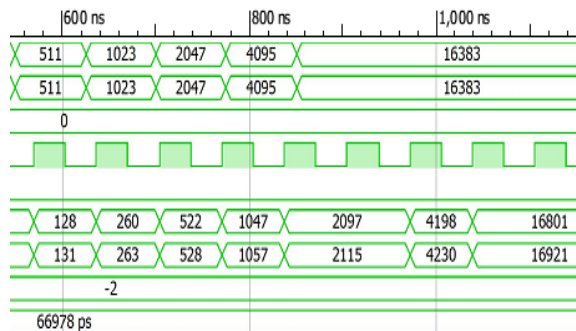


Fig. 9. Xilinx ISE simulation result for scale factor K = 0.60725 (16 bit data path)

Thus from figure 9 the error in the outputs even for maximum range of inputs is more than 3%. It shows that theoretical congruence constant cannot be considered in scaling, when the designs are done based on conventional CORDIC algorithm.

Figure 10 shows the Xilinx simulation output of the CORDIC unit with the proposed scaling units.

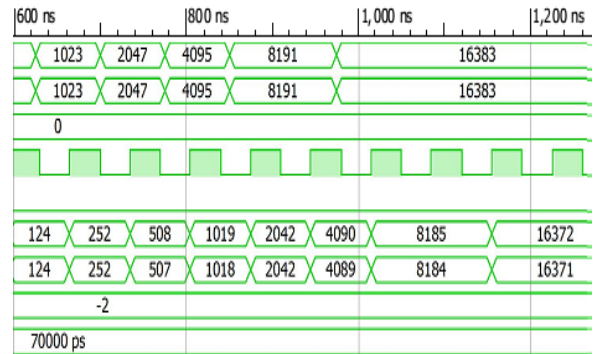


Fig. 10. Xilinx ISE simulation result with proposed scale constants (16 bit data path)

Thus from figure 10, it is evident that, the proposed design of scaling units promises more accurate outputs when better range of inputs are considered. As mentioned earlier, the proposed scaling units can also be implemented while doing vectoring mode operations in circular coordinate system.

Table IV shows the comparison of CORDIC units with proposed method and the one in [10], in terms of hardware, for 20 bits. The table clearly shows that CORDIC unit based on proposed method has lesser hardware as well as higher frequency. Since the design is implemented using pipeline, the count of flip flops has increased.

TABLE IV. COMPARISON WITH DESIGN IN [10] FOR 20 BIT DATA PATH

	[10]	Proposed method
4 input LUTs utilized	1907	1588
Slices utilized	984	812
Max. frequency (MHz)	56.351	61.331

V. CONCLUSIONS

In this paper, we presented new scaling units for x and y data paths separately considering the

practical congrate scaling constants. The proposed approximation is word length dependent and based on requirement of accuracy, the word length of the data paths can be varied. CORDIC unit with the proposed scaling units is implemented for different ranges of inputs and error analysis is done, considering the word length of data path as 16 bits. The error of the CORDIC unit with proposed scaling unit is 0.067% for x data path and 0.073% for y data path, thus making the design more accurate. The hardware requirement of CORDIC unit with proposed scaling units is less or comparable to that of other scale free CORDIC architectures. The maximum frequency, at which the design can be implemented, in Xilinx XC3S500E-4FG320 FPGA, fabricated in 90 nm process technology, is 75.593 MHz and its slice delay product is 106.465.

REFERENCES

- [1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRETrans. Electron. Comput.* vol. EC-8, pp. 330–334, Sep. 1959.
- [2] J. S. Walther, "A Unified Algorithm for Elementary Functions," *Proc. Joint Spring Comput. Conf.*, vol. 38, pp. 379–385, Jul. 1971.
- [3] P. K. Meher, J. Walls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.
- [4] B. Lakshmi and A. S. Dhar, "CORDIC Architectures: A Survey," *VLSI Design*, vol. 2010, pp. 1–19, 2010.
- [5] J. Villalba, J. A. Hidalgo, E. L. Zapata, E. Antelo, and J. D. Bruguera, "CORDIC architectures with parallel compensation of scale factor," *Proc. Application Specific Array Processors Conf.*, pp. 258–269, Jul. 1995.
- [6] Zhi-Xiu Lin and An-Yeu Wu, "Mixed-Scaling Rotation CORDIC (MSR-CORDIC) Algorithm and Architecture for Scaling-Free High-Performance Rotational Operations," *Proc. Acoustics, Speech, and Signal Processing Conf.*, vol. 2, pp. 653–656, Apr. 2003.
- [7] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified Virtually Scaling-Free adaptive CORDIC Rotator Algorithm and Architecture," *IEEE Trans. Circuits Syst. for Video Tech.*, vol. 5, no. 11, pp. 1463–1474, Nov. 2005.
- [8] S. Aggarwal, P. K. Meher, and K. Khare, "Area-Time Efficient Scaling-Free CORDIC Using Generalized Micro-Rotation Selection," *IEEE Trans. VLSI Syst.*, vol. 20, no. 8, pp. 1542–1546, Aug. 2012.
- [9] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L. Zapata, "Enhanced Scaling-Free CORDIC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1654–1662, Jul. 2010.
- [10] M. G. Buddika Sumanasena, "A Scale Factor Correction Scheme for the CORDIC Algorithm," *IEEE Trans. Computers*, vol. 57, no. 8, pp. 1148–1152, Aug. 2008.