A CORDIC Algorithm with Improved Rotation Strategy for Embedded Applications

Kui-Ting Chen
Research Center of Information, Production and Systems, Waseda University, Fukuoka, Japan
Email: nore@aoni.waseda.jp

Ke Fan, Xiaojun Han, and Takaaki Baba
Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan
Email: fanker0806@asagi.waseda.jp; alberthan0811@ruri.waseda.jp; tbaba@waseda.jp

***Abstract*—Coordinate rotation digital computer (CORDIC) is an iterative algorithm to calculate various complex mathematical functions such as trigonometric, hyperbolic, logarithmic functions and so on. The iterative procedure of the conventional CORDIC algorithm is inefficient due to its rotation strategy. This research presents a resourceful rotation strategy to reduce the unnecessary iteration times. The proposed rotation strategy divides the conventional rotation approach to two rotation functions which are named coarse rotation function and precise rotation function. The simulation results prove that can drastically reduce unnecessary iteration times compared with the conventional approach. In addition, the proposed approach is a hardware-oriented algorithm for embedded applications when compared with other CORDIC algorithms.**

***Index Terms*—coordinate rotation digital computer, greedy algorithm, hardware-oriented algorithm**

I. Introduction

Coordinate rotation digital computer (CORDIC) is a well-known and powerful iterative algorithm. It only exploits the operation of addition and bit-shifting to perform various mathematical functions such as sine, cosine, square root, logarithm and so on [1] and [2]. The execution time of the CORDIC algorithm only requires tens of the clock cycles. Hence, the CORDIC algorithm can be applied to various real-time applications [3]. However, the achieved calculation speed of the conventional CORDIC algorithm [1]-[3] is still not fast enough, since the calculation tasks of the modern real-time applications are become more and more complex. The drawback of the conventional CORDIC algorithm is that each angles of an elementary angle list must be used for each iterative procedure. If the total angle number (N) of the elementary angle list is increased, the calculation precision becomes more accurate. As the trade-off, the total iteration times must be increased to N. To reduce the total iteration times, the rotation strategy of the conventional CORDIC algorithm has to be improved.

Y. H. Hu and S. Naganathan [4] proposed an angle recoding algorithm to reduce the calculation time of the conventional CORDIC algorithm. The angle recoding algorithm includes two features which are angle recoding and greedy algorithm. The angle recoding can rebuilt an elementary angle list of the CORDIC algorithm. The greedy algorithm can select an appropriate rotation angle from the elementary angle list for each iterative procedure. The simulation results proved that their proposed approach can reduce the total iteration time by at least $N/2$ of N elementary angles without sacrificing the calculation precision. However, the total iteration times of their approach exceed $N/2$ in most of the situation. The total iteration times of a few results are quite close to the total angle number (N). Hence, their proposed rotation strategy still needs to further improve.

C.-S. Wu and A.-Y. Wu [5] proposed a modified vector rotational CORDIC algorithm (MVR-CORDIC) which is based on the research of Y. H. Hu and S. Naganathan [4]. The MVR-CORDIC can overcome the above drawback of the angle recoding algorithm. It limited the total iteration times of the iterative procedure to ensure that the CORDIC algorithm can perform its function with a fixed iteration times. However, the total iteration still needs $N/2$ times for calculating the mathematical functions. Moreover, the hardware design of the MVR-CORDIC becomes more complex as a trade-off.

In order to overcome the above difficulties, a CORDIC algorithm with improved rotation strategy is proposed to provide a high calculation ability. The proposed approach divides the conventional rotation strategy into two new rotation functions named coarse rotation function and precise rotation function. The coarse rotation function can approximate the target angle with once iteration. Then, the precise rotation function exploits greedy algorithm to perform the micro-rotation for minimizing the deviation between the target angles and rotated angle. The proposed approach can drastically reduce the total iteration times without increasing the complexity of the CORDIC algorithm. In addition, the proposed approach is more suitable for hardware implementation.

The rest of this paper is organized as follows. In Section II, the concept of CORDIC algorithm and its improvement are briefly introduced. In Section III, the

Manuscript received January 30, 2015; revised April 16, 2015.

© 2015 Journal of Industrial and Intelligent Information
doi: 10.12720/jiii.3.4.274-279

details of the proposed CORDIC algorithm are presented. In Section IV, the simulation results of the proposed approach and conventional approach are provided. Finally, Section V comprises a summary and the conclusions of this research.

II. CORDIC ALGORITHM

The concept of the CORDIC algorithm was firstly introduced in 1956 [1], which can calculate the trigonometric functions, hyperbolic functions, exponential and logarithm. Then, many different types of the CORDIC are proposed for different mathematical calculation such as [6]-[8]. J. S. Walther [6] proposed a unified CORDIC algorithm to combine different CORDIC algorithms into one CORDIC algorithm. It can work on three different coordinate systems and two different modes for calculating various mathematical functions, as shown in Table I. The equations of the unified CORDIC algorithm are given by:

$$X_{i+1} = X_i - m\sigma_i 2^{-i} Y_i \tag{1}$$

$$Y_{i+1} = Y_i + m\sigma_i 2^{-i} X_i \tag{2}$$

$$Z_{i+1} = Z_i + \sigma_i \theta_i \tag{3}$$

$$\theta_i = \arctan(2^{-i}) \quad \text{if} \quad m = 1 \tag{4}$$

$$\theta_i = \text{arctanh}(2^{-i}) \quad \text{if} \quad m = -1 \tag{5}$$

$$\theta_i = 2^{-i} \quad \text{if} \quad m = 0 \tag{6}$$

$$K_N = \frac{1}{\sqrt{1 + m2^{-2N}}} \tag{7}$$

where X and Y are the coordinate components; Z is the angle accumulator; $m=1$, 0 and -1 correspond to circular, linear and hyperbolic coordinate system respectively; θ_i is the elementary angle of rotation for iteration i in the selected coordinate system as shown in (4), (5) and (6). σ_i means the direction of the rotation in the each CORDIC rotation. The value of the σ_i is 1 or -1, which is decided by the situation of iteration i. In (7), K_N is known as scale factor applied at the starting or at the end of the iteration to normalize the result. N means the total number of iteration.

Y. H. Hu and S. Naganathan [4] originally exploit the greedy algorithm to reduce the total rotation times of the conventional CORDIC algorithm. The simulation results proved that their proposed approach can reduce at least 50% of the iteration times. C.-S. Wu and A.-Y. Wu [5] report that such approach is not suitable for hardware implementation, since the total iteration becomes unstable [5]. They propose the MVR-CORDIC algorithm to avoid the above problem. In the MVR-CORDIC algorithm, a new rotation approach is proposed, which is named semi-greedy algorithm. It combines the greedy algorithm and an exhaustive algorithm [5], which can avoid the unnecessary iterative procedure to further enhance the performance of the greedy algorithm and reduce the deviation between the target angle and rotated angle. The deviation can achieve 10^{-4} of the calculation precision. However, the semi-greedy algorithm still requires several iteration times to approximate the target angle during the early stage of the iterative procedure. A new rotation strategy is sincerely required to approximate the target angle within a few iteration times.

TABLE I. FUNCTIONS THAT CAN BE CALCULATED BY CORDIC

Coordinate system (*m*)	CORDIC Modes	
	Rotation Mode ($Z_N = 0$)	Vectoring Mode ($X_N = 0$ or $Y_N = 0$)
Circular ($m=1$)	$f(x) = \sin x$ $f(x) = \cos x$ $f(x) = \tan x$	$f(x) = \arcsin x$ $f(x) = \arccos x$ $f(x) = \arctan x$
Linear ($m=0$)	$f(x,y) = xy$	$f(x,y) = x/y$
Hyperbolic ($m=-1$)	$f(x) = \sinh x$ $f(x) = \cosh x$ $f(x) = \tanh x$ $f(x) = e^x$	$f(x) = \text{arcsinh}\, x$ $f(x) = \text{arccosh}\, x$ $f(x) = \text{arctanh}\, x$ $f(x) = \ln x$ $f(x) = \sqrt{x}$

III. A CORDIC ALGORITHM WITH IMPROVED ROTATION STRATEGY

In order to overcome the aforementioned difficulties, an improved rotation strategy is proposed in this paper. The conventional rotation function is divided into two rotation functions which are the coarse rotation function and precise rotation function, as illustrated in Fig. 1. The proposed precise rotation function is also based on the greedy algorithm, since greedy algorithm can select the most appropriate rotation at each iterative procedure.

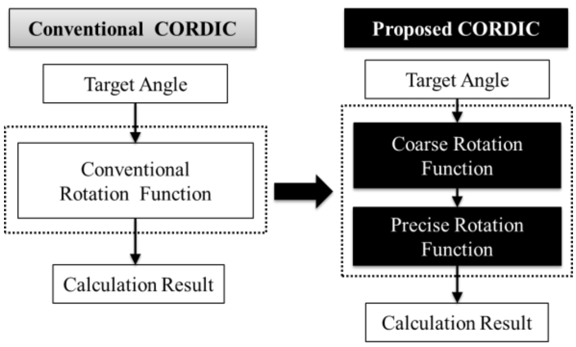

Figure 1. The proposed rotation strategy.

A. Coarse Rotation Function

The conventional CORDIC algorithm is difficult to approximate the target angle during the early rotation stage, even if the greedy algorithm is employed. Several iteration times are required to approximate the target angle with high precision. The reason is that the elementary angle list of rotation is based on the $\arctan(2^{-i})$ or $\text{arctanh}(2^{-i})$. The elementary angles are not a uniform distribution, which is difficult to envelope the angle from 0 degree to 90 degree. The proposed coarse rotation function aims to uniformly disperse the elementary angles from 0 degree to 90 degree, which can approximate target angle within once iteration. In addition, it can

© 2015 Journal of Industrial and Intelligent Information

approximate the target angle with higher precision than conventional approaches.

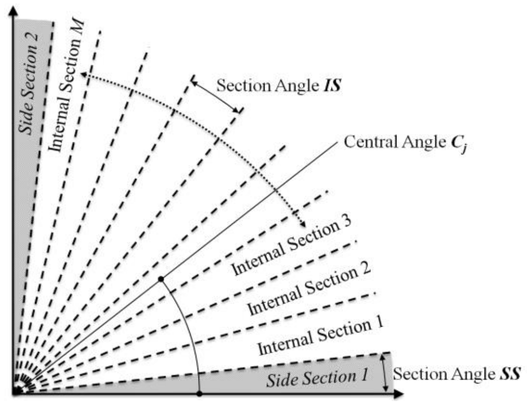

Figure 2. The example of the angle sections.

The proposed coarse rotation function divides the angle rotation range into several angle sections, as shown in Fig. 2. The angle sections include two different kinds of sections which are the side sections and the internal sections. Except for side sections, the internal sections have own central angle degree C_j which is predefined by the (4) to (6) to approximate the required central angle degree C_j. The deviation between the target angle and the central angles can be reduced with increasing the total number of the internal sections. The memory cost of the central angle list must be increased. The equation of C_j is given by

$$C_j = \sum_{i=0}^{n} \theta_i \tag{8}$$

where j is the number of the internal section, θ_i is defined in (4) to (6), n is the total number of θ_i combinations. In the each internal section j, the related value of X_{i+1} and Y_{i+1} are also predefined as an initial value. Z_{i+1} must be calculated by (3) with using $\theta_i = C_j$. The predefined value of C_j, X_0, Y_0 and Z_0 are stored into a central angle list for the first iterative procedure. Once the target angle is located on a particular angle section, the coarse rotation function can rotate the coordinate to the particular angle section within a single iteration. It can drastically reduce the unnecessary iterative procedure.

However, it is difficult for the internal sections to perform well than the greedy algorithm when the target angle approximates to 0 degree. The greedy algorithm can easily select an appropriate rotation angle from the elementary angle list, since most of the elementary angles are distributed near the 0 degree. In contrast situation, the greedy algorithm also performs well when the initial rotation angle set as 90 degree. To overcome the above drawback of the internal sections, two side sections are proposed, as shown in Fig. 2.

The purpose of the side sections is slightly different with the internal sections. The side sections are not used to rotate the coordinate to the central angle (C_j) of the particular angle section. Once the target angle is located in two side sections, the coarse rotation function set the initial rotation angle to 0 degree or 90 degree for the precise rotation function. If the target angle is located on *Side Section 2*, the coarse rotation function sets the initial rotation angle to 90 degree. Otherwise, it sets the initial rotation angle to 0 degree while target angle is located on *Side Section 1*. The definition of the section angles is defined as

$$2 * SS + M * IS = 90^{\circ} \tag{9}$$

where, SS is the section angle of the side sections, IS is the section angle of the internal sections. M is the number of the internal sections. The deviation between the target angle and the central angles can be reduced with increasing the total number of the internal sections. Hence, the memory cost of the central angle list must be increased.

The proposed coarse rotation function can approximate the target angle within once iteration which is much faster than the greedy algorithm. As a trade-off, the central angle list is required for the iterative procedure. The memory cost is slightly larger than the conventional approaches. However, the proposed approach can promise a higher calculation precision and less iteration times.

B. Precise Rotation Function

The execution time of the greedy algorithm is slightly longer than rotation strategy of the conventional CORDIC algorithm, since each elementary angle has to be sequentially inspected by the greedy algorithm for selecting the appropriate rotation angle. Namely, the execution time has to be increased more while the number of elementary angles is increased. However, the work of [5] presented a hardware can overcome the above difficulty. In order to reduce the unnecessary iteration, the greedy algorithm is also employed into precise rotation function.

Since the operation of the coarse rotation function is finished, the rotation begins from a first rotated angle (C_j, 0 degree or 90 degree). In order to ensure the deviation between the first rotated angle and target angle is absolutely less than $IS/2$, the relationship between the IS and SS is given by

$$SS \le \frac{IS}{2} \tag{10}$$

Equation (10) led to that some the elementary angles are never to be selected by greedy algorithm when the elementary angles are larger than $IS/2$. Therefore, the elementary angle list has to be redefined by

$$\theta_r = \arctan(2^{-i-k}) \quad \text{if} \quad m = 1 \tag{11}$$

$$\theta_r = \text{arctanh}(2^{-i-k}) \quad \text{if} \quad m = -1 \tag{12}$$

$$\theta_r = 2^{-i-k} \quad \text{if} \quad m = 0 \tag{13}$$

where, k is the number of the θ_i which is smaller than $IS/2$. In the assumption of the i=16 and k=4 for (11), the elementary angle list is redefined by replacing $\arctan(2^{-1})$ to $\arctan(2^{-4})$ with $\arctan(2^{-17})$ to $\arctan(2^{-20})$, which can achieve higher calculation precision of the CORDIC than its conventional approach with exploiting the redefined elementary angle list. Furthermore, the memory cost of the elementary angle list is still the same with conventional approach.

The greedy algorithm terminates only when the deviation between the target angle and rotated angle cannot be further improved. It leads to that the total iteration number is not fixed to constant iteration times. In such situation, the hardware architecture is become more complex [5]. The terminated conditions have to be employed into the greedy algorithm to reduce the unstable iteration times. Hence, the greedy algorithm with a terminated condition is also applied in the proposed precise rotation function to reduce the unnecessary iteration. In the precise rotation function, the greedy algorithm is terminated when the deviation between the target angle and rotated angle is less than a specified threshold such as 10^{-3} or smaller number. The threshold is based on the requirement of the applications.

IV. Simulation Results

The proposed CORDIC algorithm is implemented by C++ language on a PC with Intel Core i7 2.3 GHz and 8 GB RAM. The required parameters of the simulation are shown in Table II. In order to evaluate the performance of the proposed approach, the experiments of rotation performance and calculation precision are carried out. In both of the performance evaluations, the proposed CORDIC is compared with the conventional approaches which include conventional CORDIC algorithm [4] and CORDIC with greedy algorithm [6]. The total number of the angle sections is 13 which include two side sections and eleven internal sections. Section angles of *SS* and *IS* are 3.75 degree and 7.5 degree, respectively. All parameters of the CORDIC algorithm set the same condition for elementary angle number. The target angle is changed from 0.0 degree to 90.0 degree by increasing 0.1 degree. In this paper, 900 times simulations are carried out. The terminated thresholds are set as 10^{-3} and 10^{-6} to compare the rotation performance with conventional approach [4].

TABLE II. Summary of Simulation Parameters

Parameters	Values
Number of elementary angles	32
Number of side sections	2
Number of internal sections	11
Number of iterations	32
Section angle *SS*	3.75 degree
Section angle *IS*	7.5 degree
Number of target angle	900
Target angle increased by	0.1 degree
Interval of target angle	[0.0, 90.0] degree

A. Rotation Performance

The rotation performance of the proposed CORDIC is compared with the conventional approaches [4], [6]. The relationship between the iteration and average deviation is shown in Fig. 3. The definition of average deviation is the deviation between the target angle and rotated angle during the each iteration over 900 simulation results. After the first iteration, the average deviations of the conventional CORDIC and CORDIC with greedy algorithm are 22.5 degree and 12.8 degree. Meanwhile, the proposed coarse rotation function can approximate the target with the average deviation of 1.8 degree which is much better than the conventional approaches, as illustrated in Fig. 3. The precise rotation function can minimize the average deviation to $9.16*10^{-10}$ which is much better than conventional approaches which can only achieve $1.33*10^{-8}$ of the average deviation.

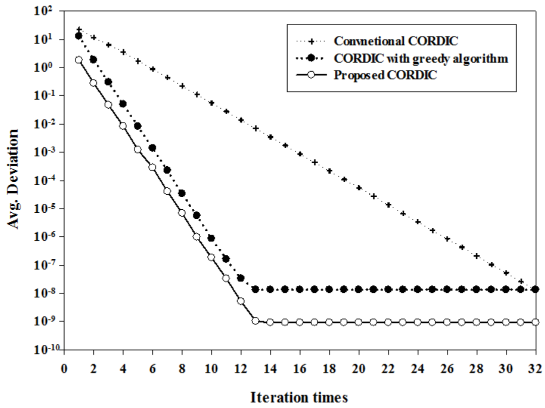

Figure 1. The proposed rotation strategy.

In the Table III and Table IV, the proposed CORDIC can reduce the maximum iteration times from 32 to 7 times. The improved ratio is up to 78.1% (*((32-7)/32)*100%*) at least. According to the simulation results of [4], the improved ratio can only achieve 50% at least when the CORDIC employed greedy algorithm without considering terminated condition. The improved ratio of the CORDIC with greedy algorithm is about 68.8% (*((32-10)/32)*100%*) when the terminated condition is considered. The proposed CORDIC can reduce 1.63 (*6.72-5.09*) iteration times on average from the CORDIC with greedy algorithm. It means that the proposed approach can further improve the rotation performance about 24.3% (*(1.63/6.72)*100%*) when the CORDIC with greedy algorithm is compared. The maximal improved ratio can be up to 30.0% (*((10-7)10)*100%*) when considering the maximal iteration times between the proposed CORDIC and the CORDIC with greedy algorithm. In most of the cases, the proposed CORDIC can finish the calculation within 4 iterations which is a significant improvement for CORDIC algorithm. With increasing the terminated threshold from 10^{-3} to 10^{-6}, the proposed approach can still reduce the iteration times by 1.70 (*9.99-8.29*) on average, as shown in Table IV. The simulation results prove that the proposed CORDIC can achieve higher rotation performance with less iteration times and higher stability than conventional approaches.

TABLE III. Iteration Reduction (Terminated Threshold 10^{-3})

	Conventional CORDIC [6]	CORDIC with greedy algorithm [4]	Proposed CORDIC
Max. iteration	32	10	7
Mean iteration	32	6.72	5.09
Std. deviation	-	1.20	1.05

TABLE IV. ITERATION REDUCTION (TERMINATED THRESHOLD 10^{-6})

	Conventional CORDIC [6]	CORDIC with greedy algorithm [4]	Proposed CORDIC
Max. iteration	32	14	11
Mean iteration	32	9.99	8.29
Std. deviation	-	1.47	1.53

B. Calculation Precision

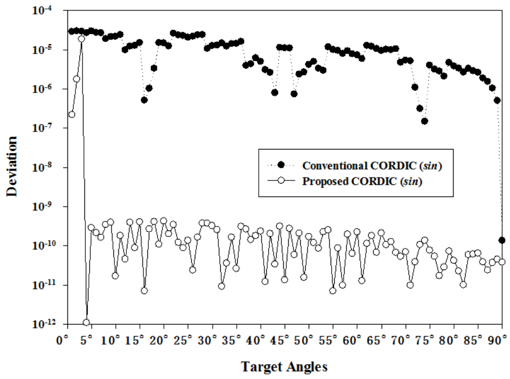

Figure 2. Calculation precision of *sin* functions.

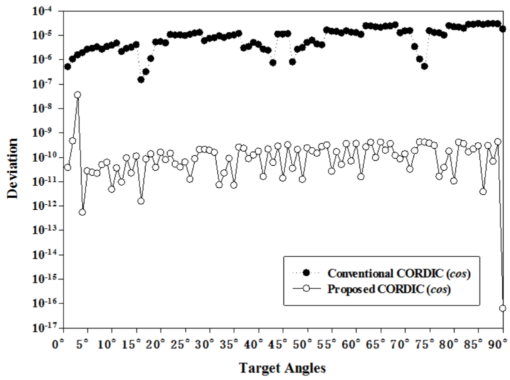

Figure 3. Calculation precision of *cos* functions.

In some hardware development of the CORDIC algorithm [7] and [9], the total iteration times are considered as 16. Since the total iteration times are increased to 32, the hardware resource must to be drastically increased. Hence, the total iteration times is set as 16 during the evaluation of the calculation precision. The *sin* and *cos* functions are used to evaluate the performance of the proposed CORDIC and conventional CORDIC. The performance of the CORDIC with greedy algorithm almost achieves the same performance with the proposed approach at 16 iterations. Therefore, the calculation precision of the CORDIC with greedy algorithm is not discussed in this experiment.

The simulation results of the calculation precision are shown in Fig. 4 and Fig. 5. In most of the target angles, the deviation of the proposed CORDIC can achieve about 10^{-10} for *sin* and *cos* functions. The conventional approach can only achieve the deviation around 10^{-5} in both of *sin* and *cos* functions. The proposed approach can improve the calculation precision about 200% which is a remarkable achievement. The simulation results prove that the proposed CORDIC can perform the superior calculation precision compared with the conventional approach. In addition, the proposed approach is more suitable to do the hardware implementation compared with the other improved CORDIC algorithms such as [4] and [5]. It can be applied into different embedded applications.

V. CONCLUSIONS

In this paper, a CORDIC algorithm with improved rotation strategy is proposed to reduce the unnecessary iteration times. The proposed improved rotation strategy employs a coarse rotation function and a precise rotation function. The coarse rotation function can approximate target angle within a single iteration to reduce the unnecessary iterative procedures. The precise rotation function employs greedy algorithm to select an appropriate rotation angle from the elementary angle list. It can maintain a high calculation precision for various mathematical functions. The simulation results prove that the proposed CORDIC achieves not only 81% iteration times reduction of the conventional CORDIC, but also improve the calculation precision by 200% within 16 iterations. In addition, the proposed CORDIC can further improve the rotation performance by 30.0%, comparing to the CORDIC with greedy algorithm. The proposed CORDIC can perform the mathematical functions with least number of iterations and highest calculation precision, simultaneously. Furthermore, the complexity of the proposed CORDIC algorithm is much simpler than other improved CORDIC algorithms. As the future work, the hyperbolic mathematical functions will be prepared to further evaluate the performance of the proposed approach.

ACKNOWLEDGMENT

This work was supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (No. 26280017).

REFERENCES

[1] J. E. Volder, "Binary computation algorithms for coordinate rotation and function generation," Aeroelectrics Group, Convair Report IAR-1 148, Jun. 1956.

[2] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. on Electronic Computers*, vol. EC-8, no. 3, pp. 330-334, Sep. 1959.

[3] C. C. Wong and C. C. Liu, "FPGA realisation of inverse kinematics for biped robot based on CORDIC," *Electron. Lett.*, vol. 49, no. 5, pp. 332-334, Feb. 2013.

[4] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Trans. on Computers*, vol. 42, no. 1, pp. 99-102, Jan. 1993.

[5] C. S. Wu and A. Y. Wu, "Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp. 548-561, Jun. 2001.

[6] J. S. Walther, "A unified algorithm for elementary functions," in *Proc. Spring Joint Comput. Conf.*, vol. 38, pp. 379-385, May 1971.

[7] R. Andraka, "A survey of CORDIC algorithm for FPGA based computers," in *Proc. ACM/SIGDA 6th Int. Symp. on Field Programmable Gate Arrays*, Feb. 1998, pp. 191-200.

[8] T. Lang and E. Antelo, "CORDIC-based computation of arccos and arcsin," in *Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures and Processors*, July 1997, pp. 132-143.

[9] K.-T. Chen, M. Jia, X. Wu, and T. Baba, "A programmable architecture of CORDIC for accelerating embedded processor,"

Journal of Signal Processing, vol. 17, no. 6, pp. 247-254, Nov. 2013.

Kui-Ting Chen received his M.S. degree and Dr. of Engineering degree from the Graduate School of Information, Production and Systems of Waseda University in 2009 and 2014, respectively. He is a research associate at the Research Center of Information, Production and Systems of Waseda University. His research interests include positioning systems, hardware architecture of particle swarm optimization and relative wireless communication technology. He is a member of IEEE, IEICE and IEEJ.

Ke Fan received his B.S. degree from University of Electronic Science and Technology of China, in 2013. He is now a master student at the Graduate School of Information, Production and Systems of Waseda University. His current research interests include hardware achievement of the optimization algorithms and application of FPGA. He is a student member of IEEE and IEICE.

Xiaojun Han received his B.S. degree from Shenyang Aerospace University of China, in 2012. He is now a master student at the Graduate School of Information, Production and Systems of Waseda University. His current research interests include software computing and hardware achievement of the optimization algorithms. He is a student member of IEEE and IEICE.

Takaaki Baba received his M.S. degree and Dr. of Engineering degree from Nagoya University in 1973 and 1979, respectively. He joined Matsushita Electric Industrial Co., Ltd., in 1973. Between 1983 and 2002, he worked for Matsushita Electric Co. in the US, where he was involved in several strategic projects, such as System LSI and ASIC applications, wireless communication systems and electronic devices. He was a research fellow at UC Berkeley between 1980 and 1982 and at Stanford University between 2002 and 2003. Since 2003, he has been a Professor in the field of system LSI applications at the Graduate School of Information, Production and Systems of Waseda University. He is a member of IEEE and served as an Executive Committee member of IEEE-ISSCC from 1995 to 2003. He is also a member of RISP and IEICE.