Three Dimensional CORDIC with reduced iterations

C.T. Clarke and G.R. Nudd

Department of Computer Science
University of Warwick
Coventry
CV4 7AL
United Kingdom

Abstract

This paper describes a modification to the three dimensional CORDIC algorithm using an approximation of the Taylor series. The modification has the potential to reduce the number of iterations required for a three dimensional CORDIC operation by at least 25%. The approach used is based upon a modification of the two dimensional CORDIC algorithm originally suggested by H.M Ahmed[1].

Introduction

The COordinate Rotation DIgital Computer or CORDIC was first suggested in 1959 by J. E. Volder [2]. The motivation was a need for accurate calculations for on board an aircraft navigation. The system was capable of rapidly computing vector rotations, and performing Cartesian to circular co-ordinate conversions. The same hardware was also able to multiply, divide, and convert between binary and mixed-radix number systems. The CORDIC algorithm is now widely used for trigonometric evaluation [3, 4].

The basic CORDIC iteration can be described as a rotation and extension of a vector. Ignoring the vector extension, the relationship between the two vectors is:

$$X_{i+1} = X_i \pm 2^{-i} Y_i$$

and

$$Y_{i+1} = Y_i \mp 2^{-i} X_i$$

The ± indicates the fact that the rotation can be in either direction. As the vector is rotated, it is extended by a small amount. The scaling effect is removed by pre or post-scaling of *X*, and *Y*. A third register (conventionally *Z*)is used to accumulate prestored angle values to give the current remaining rotation required.

In 1982 Ahmed suggested a combination of CORDIC, and the Taylor series for vector rotation [1]. The first term of the Taylor series for sine *Z* is simply *Z*, and it is possible to perform the last of our CORDIC iterations using this approximation, rather than 2^{-i}. Hence after the last CORDIC iteration, a final iteration, as follows is performed:

$$X_{m+1} = X_m - Z_m Y_m$$

$$Y_{m+1} = Y_m + Z_m X_m$$

It has been shown by Ahmed that only $m\left(=\dfrac{n+1}{2}\right)$ (where *n* is the number of bits of accuracy required in the output) iterations of the CORDIC algorithm are needed followed by the final iteration as given above. Timmermann et al. extended this work to include vectoring mode [5]. The advantage of these techniques is that they speed up the calculation of the function, but multiplication and division are now required in the final iteration.

The three dimensional CORDIC algorithm

Clarke et. al. [6], and Delosme [7] have shown that the CORDIC algorithm can be extended to three dimensions. This paper considers only the three dimensional algorithm suggested by Clarke et. al. A vector in three dimensional has Cartesian coordinates $\left(X_i, Y_i, Z_i\right)$ and spherical coordinates $\left(R_i, \theta_i, \phi_i\right)$. Note that *Z* is no longer the angle register. The vector can be rotated by an angle α_i around the z-axis and by an angle β_i away from the z-axis to become a new vector which has Cartesian coordinates

$(X_{i+1}, Y_{i+1}, Z_{i+1})$ and spherical coordinates $(R_i, \theta_i + \alpha_i, \phi_i + \beta_i)$. The variables U_i, V_i and W_i defined below must be used to complete the set of iteration equations.

$$U_i = R_i \cos\theta_i \cos\phi_i$$

$$V_i = R_i \sin\theta_i \cos\phi_i$$

$$W_i = R_i \sin\phi_i$$

The iteration equations are:

$$U_{i+1} = \frac{1}{k_i^2}(U_i - X_i b_i 2^{-i} - V_i a_i 2^{-i} + Y_i a_i b_i 2^{-2i})$$

$$V_{i+1} = \frac{1}{k_i^2}(V_i - Y_i b_i 2^{-i} + U_i a_i 2^{-i} - X_i a_i b_i 2^{-2i})$$

$$W_{i+1} = \frac{1}{k_i}(W_i + Z_i b_i 2^{-i})$$

$$X_{i+1} = \frac{1}{k_i^2}(X_i + U_i b_i 2^{-i} - Y_i a_i 2^{-i} - V_i a_i b_i 2^{-2i})$$

$$Y_{i+1} = \frac{1}{k_i^2}(Y_i + V_i b_i 2^{-i} + X_i a_i 2^{-i} + U_i a_i b_i 2^{-2i})$$

$$Z_{i+1} = \frac{1}{k_i}(Z_i - W_i b_i 2^{-i})$$

It should be noted that the scale factor for Z_{i+1} and W_{i+1} is different to that of U_{i+1}, V_{i+1}, X_{i+1} and Y_{i+1}. Clarke et. al. [6] have shown that no repeated iterations are needed for these equations.

Use of the Taylor series

The work of Timmermann et. al. and Ahmed [5, 4] that reduces the number of iterations required for a given accuracy can be extended to three dimensions. The technique relies on the first term of the Taylor series to remove the need for further iterations. The approximations used are:

$$\cos\delta \approx 1 \text{ and } \sin\delta \approx \delta$$

A final iteration is to be added to the algorithm in rotation mode that uses these approximations. Clearly, if only (x, y, z) needs to be evaluated, then it is only necessary to do the final iteration for those equations. The equations for *u*, *v and w*, are not given because it will not normally be necessary to store these variables. If they are required, it would be a simple matter to calculate the equations for them.

$$X_n = X_{n-1} - Y_{n-1}r_\theta + U_{n-1}r_\phi - V_{n-1}r_\theta r_\phi$$

$$Y_n = Y_{n-1} + X_{n-1}r_\theta + V_{n-1}r_\phi + U_{n-1}r_\theta r_\phi$$

$$Z_n = Z_{n-1} - W_{n-1}r_\phi$$

The variables r_θ and r_ϕ are the remaining angles to be rotated through. They are the remainder stored in the angle registers. To get an overall accuracy of *m* bits, it is necessary to use the standard CORDIC algorithm to obtain *m* bit variables to an accuracy of *n* bits and then one iteration as described above. The relationship between *m* and *n* is

$$n = \left(\frac{m-1}{2}\right)$$

This result is found directly from the second term of the Taylor series (the others do not need to be taken into account). Bit level simulation accuracy results are shown in table 1 for both conventional arithmetic and redundant arithmetic types over 10,000 random vector rotations for each result. Because the remaining angle must be small and the number of bits to which the angles are stored will be the same as the coordinates, the bits that can be anything other than zero will only be in the lower half of the word. It is therefore possible to use a fairly small multiplier (in the region of m bits by $\frac{m}{2}$ bits) for the most of the multiplications. With a pipelined Booths multiplier this will take $\frac{m}{4}$ steps. The combination of the reduced CORDIC iterations, and the Taylor series expansion is roughly 25% fewer iterations than a full three dimensional CORDIC operation.

		Output Accuracy (bits)				
		18	19	20	21	22
It	9	13	14	14	14	14
er	10	13	15	16	16	16
at	11	13	14	16	16	17
io	12	13	14	15	16	17
ns	13	13	14	15	16	17

a)

		Output Accuracy (bits)				
		18	19	20	21	22
It	9	13	14	15	15	15
er	10	13	14	15	16	16
at	11	13	14	15	16	17
io	12	13	14	15	16	17
ns	13	13	14	15	16	17

b)

Table 1. The accuracy of a 3D Hybrid CORDIC for a) conventional, and b) redundant arithmetic. Numbers of iterations are CORDIC iterations.

Conclusions

This paper has described a modification to the three dimensional CORDIC algorithm of Clarke et. al. [6]. The modification reduces the number of iterations required for a given output accuracy. The final iteration is more complex than in the standard algorithm, but this is more than outweighed by the reduction in the overall iteration count.

References

[1] "Signal Processing Algorithms and Architectures", H.M. Ahmed, PhD thesis, Stanford University, June 1982.

[2] "The CORDIC Trigonometric Computing Technique", J.E. Volder, IRE Trans. Electron. Comput. EC-8, pp. 330-334, 1959.

[3] "A unified algorithm for elementary functions", J.S. Walther, Spring Joint Computer Conf., 1971, pp. 379-385.

[4] "INTEL's Floating-Point Processors", A.K. Yuen, Proceedings of Electro/88, 1988, pp. 48/5/1-6.

[5] "Modified CORDIC Algorithm with Reduced Iterations", D. Timmermann, H. Hahn, B. Hostika, Electronics Letters, Vol. 25, No. 15, July 1989, pp. 950-951.

[6] "A Redundant Arithmetic CORDIC System With A Unit Scale Factor", C.T. Clarke, Research Report RR234, University of Warwick Dept of Computer Science, December 1992.

[7] "CORDIC algorithms: Theory and Extensions" J.M. Delosme, SPIE Vol. 1152 Advanced Algorithms and Architectures for Signal Processing IV, 1989, pp. 131-145.