'89 KACC 1989. 10. 27~28

A calculation algorithm of transcendental functions on a Digital Signal Processor

Tsuyoshi EBINA Rokuya ISHII

Division of Electrical and Computer,Engineering,
Yokohama National University

Abstract: A Digital Signal Processor (abbreviated to DSP) is used not only for digital signal processing but also for kinematic controls[1]. Then applications to these fields are expected to be developed. We propose a function calculation method on DSP which occupies no table memory. By using these functions, more fast or more accurate control will be achieved without using function table.

1.INTRODUCTION

DSP is developed in order to process signal. Recently, DSP tends to be used in various fields instead of micro processor. So, DSP is used not only to calculate multiplication and summations, but also to calculate transcendental functions. But the suitable calculation algorithm for DSP is not considered. Until now, transcendental functions are implemented by table look-up method. In these cases, to calculate these function require a large table memory on DSP. Therefore we can't employ this method for calculating more than two transcendental functions or calculating transcendental functions in high accuracy.

There are many well-known algorithms. Among them CORDIC (COordinate Rotation DIgital Computer) algorithm is the most popular for micro-processor. But we can not apply this method on DSP. Because DSP's internal structure is far more complex than that of micro-processor and this algorithm can't make use of hardware capacity. Moreover, since multi-pipelined instruction code is employed on some DSPs, CORDIC programming on DSP which adopt multi-pipelined instruction code is not necessarily desirable. As the results, CORDIC algorithm is not efficient for these calculations.

In this paper, we describe the calculation method of transcendental function using multiplier. The shorter the total computation time is, the more desirable processing is. So, we espoused chebyshev polynomial approximation. By using chebyshev polynomial approximation, functions will be computed without using table memory on DSP.

We implemented on MSM6992. MSM6992 is a second generation DSP, developed in OKI electric Industry Co.,LTD. Its data is expressed with 16 bits as mantissa and 6 bits as exponent.

2.Calculation of transcendental function

This section describes the calculation method for transcendental functions. Subscripts m and e of the variable denotes mantissa and exponent, respectively. That is, if

$$X = X_m \cdot 2^{X_e},$$

X_m means mantissa part of X, and X_e means exponent part of X. X_m is normalized in one of the following three region

$$0.5 \leq X_m < 1.0,$$

$$-1.0 < X_m \leq -0.5$$

and

$$X_m = 0$$

2-1.Division

Suppose that X is a divisor and Y is a dividend. Then division Y by X will be expressed with the form

$$\frac{Y}{X} = \frac{1}{X_m} \cdot Y_m \cdot 2^{Y_m - X_m} \tag{1}$$

The exponent of the result of the division is computed by using DSP's scaling function.

Mantissa Y_m / X_m of the division is derived by multiplying Y_m to $1/X_m$. $1/X_m$ is obtained by using the chebyshev polynomial approximation. At first, the value approximated by chebyshev polynomial is obtained. And assign the value to Z_0. That is

$$Z_0 \cong 1/X_m \qquad (2)$$

We derived Z_0 in the following polynomial.

$$Z_0 = \sum_{K=0}^{3} C_K X^K \qquad (3)$$

$C_0 = 5.656846$ $\quad C_2 = 10.64818$

$C_1 = -11.75737$ $\quad C_3 = -3.549393$

In order to minimize error, we employ the newton iterative procedure expressed with the formula

$$Z_1 = (2.0 - X_m \cdot Z_0) \cdot Z_0 \qquad (4)$$

Z_1 is more accurate value than Z_0. Only a few iteration practice will achieve most accurate value regarding to the number of bits.

The above procedure requires from 34 to 36 steps on MSM6992. This means that division will be practiced within 4.0 μ sec.

2-2.Sinusoid

In calculating sin(X) and cos(X), we employ mini-max polynomial approximation[2] since these functions is periodic. Once sin(X) is calculated, cos(X) will be calculated too. Because cos(X) satisfies the following relation.

$$\cos(X) = \sin(X + \pi/2) \qquad (5)$$

So we can get cos(X) by executing sin(X+ π /2).

Therefore we need to obtain the polynomial approximation only for sin(X) in the domain $[-\pi/2, \pi/2]$. The enough order of chebyshev approximation polynomial is 7 in the case of MSM6992. Though it's high order, sin(X) is odd function and even-order terms of the polynomial equal to zero. Therefore, the calculation complexity is not so high in practice. The approximation polynomial is as follows.

$$\sin(\pi/2 \cdot X) = \sum_{K=1}^{7} C_{2K+1} X^{2K+1} \qquad (6)$$

$C_1 = 1.570794852$ $\quad C_5 = 0.079487663$

$C_3 = -0.645920978$ $\quad C_7 = -.004362476$

The above procedure is achieved under the assumption that X satisfies $[-\pi/2, \pi/2]$ since sin(X) is periodic. So X must be normalized in the region $[-\pi/2, \pi/2]$.

The step number of the above total computation is at least 24 steps on MSM6992. The total program steps required for sinusoid calculation increase proportionally with arguments.

2-3.Tan(X)

Tan(X) takes from the value $-\infty$ to ∞ when x is changed from $-\pi/2$ to $\pi/2$. Then such function can't be approximated with polynomial of order 9. The approximation for tan(X) is achieved by 9 rational function of order 9. There are some rational approximations. We employed the continued

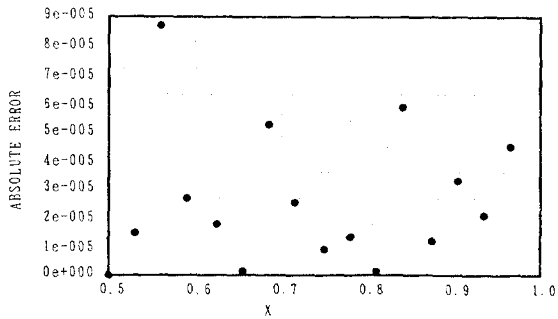

Fig.1. Division error

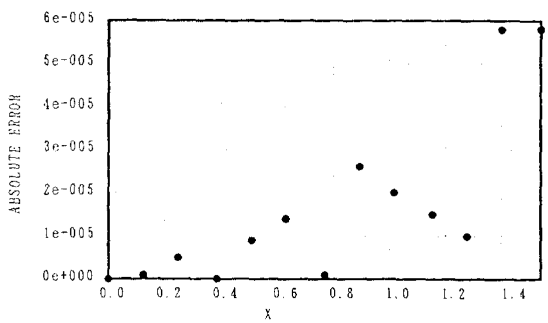

Fig.2. Sin(X) error

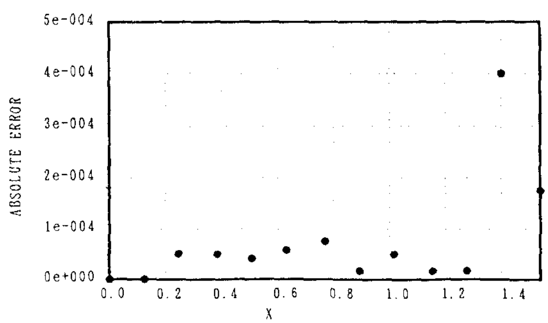

Fig.3. Tan(X) error

fraction approximation among them. But there is one problem. To calculate continued fraction approximation requires many division operations. Division on DSP is not so easy in comparison with multiplication. Then the continued fraction approximation of tan(X) is translated to a rational function whose divisor and dividend are expressed with polynomials. Employing this formulation, the number of division is decreased to only one. The corrected formula is as follows.

$$\tan(X)=\frac{X^4-5.46059X^3-22.09259X^2+91.70779X}{8.18130X^3-35.94090X^2-22.11515X+91.71003} \quad (7)$$

The calculation of the rational function with MSM6992 needs at least 61 steps. We show calculation error in Fig.3.

2-4.Atan2(Y,X)

Arctangent function (abbreviate to atan2(Y,X)) is expressed with 2 arguments X and Y. We describe the relation between them in Fig.4. On the other hand, we define the arctangent function with one argument (abbreviate to atan(Z)). The value atan(Y,X) is in only the region $(-\pi, \pi]$. The value of atan(Y,X) is symmetrical with the signs of X and Y. So, considering the signs of X and Y, it is necessary to express atan2(Y,X) with signs of X and Y. If $X > 0$ and $Y \geqq 0$, $Z = Y / X$ is positive or zero. Now we can express atan2(Y,X) with atan(Z) if $X > 0$ and $Y \geqq 0$.

Atan(Z) satisfies the following formula

$$\mathrm{atan}(Z)=\frac{\pi}{4}+\mathrm{atan}\frac{Z-1}{Z+1} \quad (8)$$

By (8), the region $[0, \infty)$ of the value Z is transformed to the region [-1,1]. Then, we approximate atan(Z) by the mini-max approximation in the region [-1,1]. The approximation polynomial is expressed as follows:

$$\sum_{K=1}^{5} C_{2K-1} X^{2K-1} \quad (9)$$

$C_1 = 0.9998660$ $C_7=-0.0851330$

$C_3=-0.3302995$ $C_9= 0.0208351$

$C_5= 0.1801410$

Though coefficient of the mini-max approximation polynomial is difficult to calculate, we employ the method proposed by [2].

But in practice, the influence of difference between the mini-max approximation and the chebyshev approximation is not so far in comparison with influence of DSP's word length. Consequently the chebyshev approximation can be used instead of mini-max approximation.

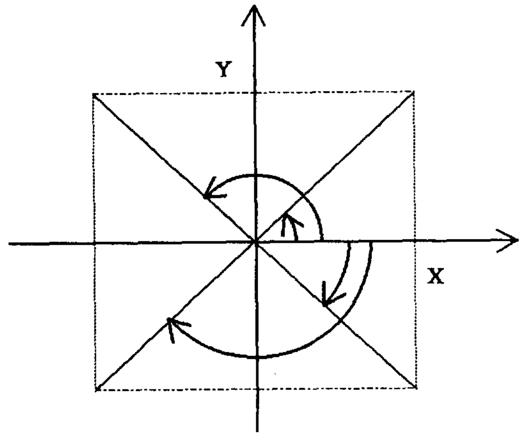

Fig.4. The relation between (X,Y) and atan2(Y,X)

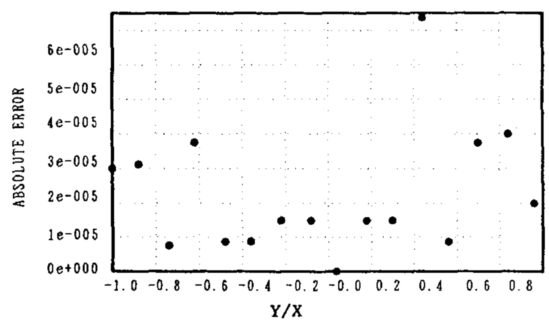

Fig.5. Atan2(Y,X) error

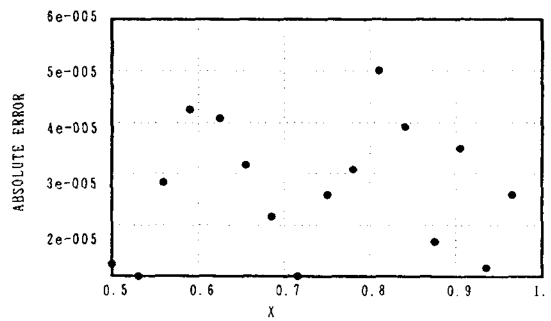

Fig.6. Square-root error

2-5.Square-root

Suppose that argument of square-root function is $X = X_m \cdot 2^{X_e}$. Then, $\sqrt{2^{X_e}}$ is computed by a right shifting of X_e. If the least significant bit of X_e is zero, square-root of 2^{X_e} is obtained by a right shifting. Otherwise, we must multiply $\sqrt{2}$ to the right shifted data. These discussion is expressed as follows.

$$\sqrt{X} = \begin{cases} \sqrt{X_m} \cdot 2^{[X_e/2]} & X_e\text{:even} \\ \sqrt{X_m} \cdot \sqrt{2} \cdot 2^{[X_e/2]} & X_e\text{:odd} \end{cases} \tag{10}$$

[X] denotes the Gauss' notation. But, this procedure is hard to realize on MSM6992. Because bit sequence translation from the mantissa to the exponent is difficult to implement. Then we have to use table memory for giving $\sqrt{2^{X_e}}$. However, if DSP has a assembly command which make mantissa bit sequence move to exponent bit sequence, we need not to use table memory.

We employ 4th chebyshev approximation polynomial for calculation of the mantissa as follows.

$$\sqrt{X_m} \cong \sum_{K=1}^{4} C_K X_m^{K} \tag{11}$$

$C_0 = 0.2308020$ $C_3 = 0.4825752$
$C_1 = 1.290867$ $C_4 = -0.1153099$
$C_2 = -0.8889398$

The square-root of X can be obtained by the product of $\sqrt{X_m}$ and $\sqrt{2^{X_e}}$.

2-6.Exp(X)

Exponential function(abbreviated to exp(X)) is derived in the following.

At first, exponent X_e is separated from X. If $X_e \leq 0$, X exists in the region [-1,1]. Therefore, we approximate exp(X) in the domain [-1,1]. The value can be calculated with chebyshev approximation.

$$\exp(X) = \sum_{K=0}^{5} C_K X^{K} \tag{12}$$

$C_0 = 1.000061$ $C_3 = 0.1665497$
$C_1 = 1.000000$ $C_4 = 0.04379272$
$C_2 = 0.4991913$ $C_5 = 0.008635521$

The approximated value is assumed to be Y.

If $X_e > 0$, $\exp(X_m)$ is calculated by the above method because of definition of X_m. Then, the 2^{X_e}-th powers of the value $\exp(X_m)$ is exp(X). Exp(X) is derived in above procedure.

The step number required for this procedure is at least 33 steps on MSM6992.

2-7.Log(X)

We assume log(X) as natural logarithm in this section. Log(X) is expressed with

$$\begin{aligned} \log(X) &= \log(X_m \cdot 2^{X_e}) \\ &= \log(2) \cdot (\log_2(2X_m) + X_e - 1) \end{aligned} \tag{13}$$

Since log(2) is constant, log(X) is derived after calculating $\log_2(2X_m)$. So, we approximate $\log_2(2X_m)$ with chebyshev polynomial. The reason why we employ approximating function as $\log_2(2X_m)$ instead of $\log_2(X_m)$ is that the theoretical error given by $\log_2(2X_m)$ is smaller than that of $\log_2(X_m)$. We show polynomial approximation in the following.

$$\log_2(2X_m) \cong \sum_{K=1}^{4} C_K X^{K} \tag{14}$$

$C_0 = -2.498353$ $C_3 = 5.008257$
$C_1 = 8.058423$ $C_4 = -1.255051$
$C_2 = -8.313341$

This implementation on MSM6992 needs 37 steps.

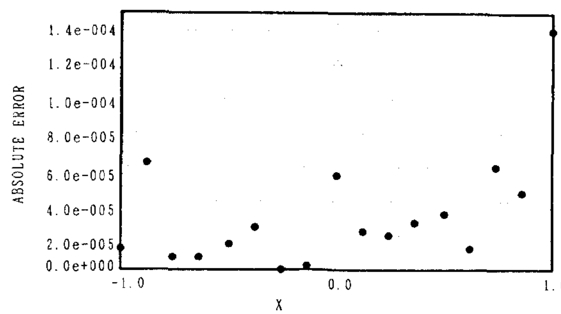

Fig.7. Exp(X) error

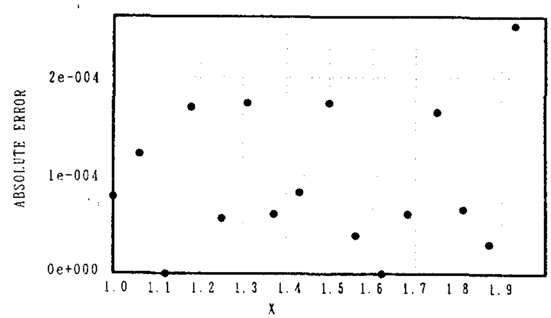

Fig.8. Log(X) error

3. Estimation for speed and error

We describe minimum step number and maximum error in some limited region for these transcendental functions in Table.1. We can see that each function is calculated faster. Though it takes more computation time in comparison with table look-up method, it may be suitable in view point of memory occupation.

Table.1. maximum error and minimum steps

function	maximum error	minimum steps
Y/X	9.0E−5	32
sin(X)	6.1E−5	24
tan(X)	4.0E−4	61
atan2(Y,X)	7.0E−5	77
square-root	5.0E−5	27
exp(X)	1.2E−4	33
log(X)	2.5E−4	37

4. Conclusion

In this paper, we proposed the algorithms for calculating transcendental functions on DSP. By using the proposed procedures, we can conclude that transcendental functions are calculated faster on DSP.

These procedures proposed in this paper can be realized on any other DSPs if floating point data calculation can be practiced on these DSPs. In such case, the order of polynomial must be chosen carefully considering data bit length on DSP.

Reference

[1]R.Ishii,N.Hamaguchi and T.Ebina: " A COMPUTATION OF A ROBOT MANIPULATOR BY USING DIGITAL SIGNAL PROCESSOR", The 27th SICE Annual

[2]Hastings : "APPROXIMATIONS FOR DIGITAL COMPUTERS",Princeton University Press,1955