ANALYSIS OF COMPUTING METHODS FOR THE TERRIER FIRE CONTROL SYSTEM

James Ray Williamson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Analysis of Computing Methods
for the
TERRIER Fire Control System

by

James Ray Williamson

Thesis Advisor: V. Michael Powers

December 1972

Approved for public release; distribution unlimited.

Analysis of Computing Methods
for the
TERRIER Fire Control System

by

James Ray Williamson
Lieutenant Commander, United States Navy
B. S., University of Illinois, 1959

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1972

ABSTRACT

Future modifications in the TERRIER missile fire control system will be restricted by limitations in unallocated core and by problem solution time in the system's digital computer.

This thesis is an analysis of methods by which reductions in core storage requirements and in problem solution time could be achieved. A determination of those functions requiring the most computer resources is made and alternative methods of computing the functions are analyzed. Comparisons of implementation of the functions by software in the fire control computer versus other devices is made, and the tradeoffs required by each method are presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL DESCRIPTION

The TERRIER missile fire control system is currently being modernized to replace the present MK 119 analog computer with a MK 152 (UNIVAC 1219B) digital computer for calculation of launcher, missile, radar, and weapon direction system quantities. Figure 1 shows the relationship of the MK 152 computer to the other system components. The fire control system (FCS) elements include the radar, launcher, and missile. The FCS elements communicate through the Signal Data Converter (SDC), where analog-to-digital conversions are accomplished. The MK 152 computer receives information from and passes information to the FCS elements through the SDC. The MK 152 computer also receives from the SDC outputs of other shipboard elements, such as the gyrocompass, anemometer and the pitometer log. The MK 152 computer can communicate with the MK 152 computer of the adjacent fire control system. Either of the pair of MK 152 computers can be switched to a teletype (I/O console MK 77). Both computers have access to the Digital Data Recorder magnetic tape unit. One additional communication path, not shown in figure 1, is between the MK 152 computers and the digital Naval Tactical Data System (NTDS).

B. PROBLEM DESCRIPTION

In the implementation of the digital computer in the TERRIER missile system, bounds have been placed on the capabilities of the system by:

1. Limitations in Storage

The software program being implemented will require most of the available 40K words of storage. Advanced proposals for future system

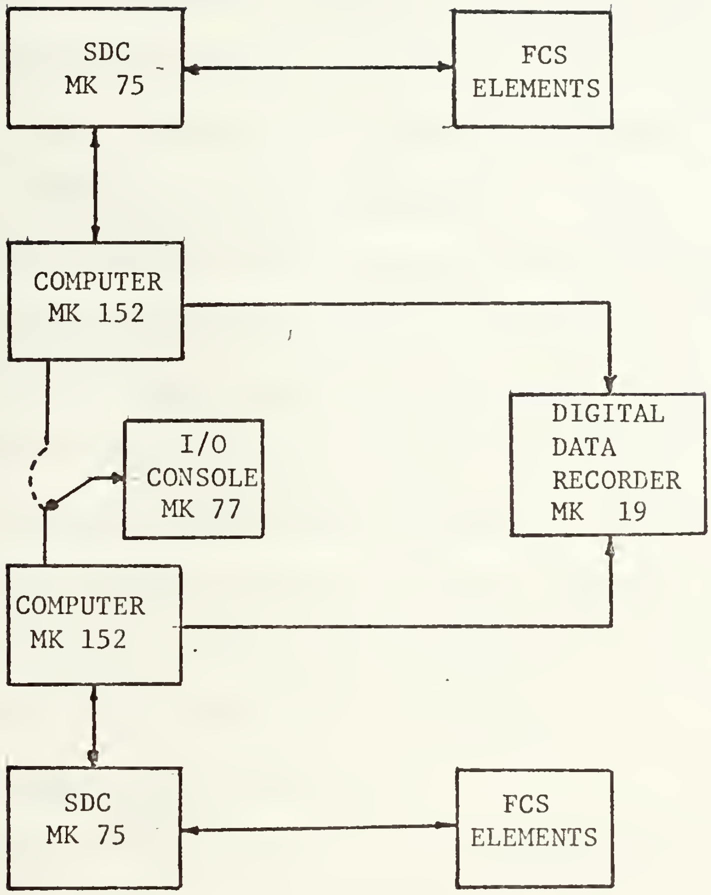

Figure 1

TERRIER Missile System Block Diagram

changes indicate that a requirement will exist for additional core storage in the MK 152 computer to implement these changes. The present high usage of available core will severely constrain these future changes by requiring these changes to either fit in the remaining available core space or by reducing some of the system's present capabilities.

2. Limitations in Computing Time

The TERRIER missile, missile launcher, and part of the fire control radar require analog inputs; therefore the quantities computed in the MK 152 computer must be converted from digital to analog. The "smoothness" of the resulting signals, then, will be a function of the speed of solution in the digital computer. Reference 1 indicates that two primary sampling rates are used for input and output - 16 and 32 times per second. The problem solution time for one iteration will determine if these sampling rates can be met.

3. Limitations on Accuracy

As indicated previously, conversion of the data to and from the FCS elements must be accomplished by the SDC. The accuracy of the data input to the MK 152 computer, then, will be limited by the accuracy of the data after conversion by the SDC. Conversely, the accuracy of the data to the FCS elements will be a function of the accuracy of the output data from the MK 152 computer after conversion by the SDC. For both communication paths, two formats of words are used. The words contain an address and either 10 or 12 bits of data depending on the source or destination. With these limitations, it is desirable that minimal degeneration of accuracy occur in the MK 152 computer during manipulation of the data to insure that the output will have the maximum accuracy possible.

C. SYSTEM PARAMETERS

1. Equations for the MK 152 Computer

Reference 1 contains the organizational relationships of the system components and the signal flow between these components. The equations used in the MK 152 computer to generate the signals to other components are contained in Ref. 2. The validity and efficiency of the equations in Ref. 2 are not questioned in this report.

2. MK 152 Computer Characteristics

Appendix A is a partial list of the capabilities of the MK 152 computer. A complete description of the organization, characteristics, and operations of the MK 152 computer may be found in Ref. 3.

D. SELECTION OF OPERATIONS FOR ANALYSIS

In section II, the computer operations are separated into two broad categories - logic operations and arithmetic operations. A comparison is made of the frequency of execution, speed of execution, and core storage requirements of the algorithms within each category to determine which algorithms use the most computer resources. Only those operations using the most computer resources are further analyzed, because reductions in core and computing time in these operations would provide the most overall computer resource savings.

E. ANALYSIS OF OPERATIONS

The methods available for implementing the functions are compiled in section III. A comparison is made in sections IV and V of the methods to determine those applicable to this particular problem. An analysis is done in those sections as to which method would provide the greatest reductions in storage and computation times.

F. IMPLEMENTATION OF FUNCTIONS

In section VI, a comparison is made of the present method of implementing the functions within the MK 152 against the method selected in section IV with the trade-offs incurred in core storage, speed of execution and accuracy. Consideration is given to the fact that the savings in core and speed might be less than that required for implementing future changes; therefore two alternative methods of implementing the functions are considered in section VI. The alternatives consist of adding additional devices to the system which would operate in conjunction with the MK 152 computer. This would enable elimination of the functions from the MK 152 computer, and parallel computations by the auxiliary device. The two devices considered are a microprogrammed computer and a hardware function generator.

G. GENERAL APPLICABILITY

Although the initial research was concentrated on the TERRIER missile system, other Navy systems utilize the same or similar functions during their computations. The MK 86 gun system, the TARTAR missile system, and the TALOS missile system all use the MK 152 computer for fire control system calculations. All of these shipboard systems, for instance, require coordinate conversion matrices to convert the radar line-of-sight to stable deck coordinates; therefore, the analysis conducted herein of that process would have equal applicability to these other systems.

H. SUMMARY OF INTRODUCTION

This thesis is an analysis of methods by which the core usage and computation times of the MK 152 digital computer in the TERRIER missile fire control system could be reduced in order to increase the flexibility of the system for future changes. This analysis determined that:

If these future changes require only a moderate increase in core storage without a concurrent decrease in computing time, this can be achieved by implementing polynomial evaluations for the trigonometric functions within the MK 152 computer program;

If a decrease in computing time is the primary requirement for future changes, then implementation of the trigonometric functions by a hardwired device would be desirable;

If both core and time savings are required, then the addition of a microprogrammed computer to the system would provide the most savings in both areas.

Thus, alternatives are available for providing reductions in the limiting areas of the TERRIER missile system computer program, so that future changes may be made without system degradation.

II. SELECTION OF OPERATIONS FOR ANALYSIS

A. CATEGORIZATION AND COMPARISON OF OPERATIONS

The algorithms contained in Ref. 2 were divided into two primary categories: logic operations and arithmetic operations. The operations on both lists were compared to determine which of these operations had the highest storage requirements and total computer use time during one problem iteration.

B. RESULTS OF COMPARISONS

The comparison and elimination processes resulted in the following operations being classed as the primary constricting operations in each category:

1. Logic Operations

None of the logic operations had long execution times or high storage requirements. The most often repeated operation was a compare and branch on condition function. The MK 152 computer can process this type of statement with a comparative mask and jump.

2. Arithmetic Operations

The list of operations in this category was narrowed to two functions: integration and trigonometric function evaluation. In the present MK 152 program, integration is accomplished by a rectangular approximation method. The trigonometric functions are evaluated by dividing the digital representation of the input angle into three parts. The most significant bits are used to determine the quadrant of the angle. The remaining bits are evaluated by using the trigonometric identity for the sum of two angles. The trigonometric value of the

major portion of the angle is obtained by table lookup, and the trigonometric value of the minor portion of the angle is obtained by a Taylor series polynomial expansion.

Table I is a compilation of the number of repetitions per computation cycle, speed of execution, and core requirements for each of the above operations. In the TERRIER system, separate computation paths are required depending on the mode of operations. Table I was constructed from the normal air mode with a semi-active homing missile computation as being a representative path. As can be seen from Table I, the trigonometric calculations placed the most demands on the MK 152 computer.

OPERATIONS	NUMBER OF REPETITIONS	EXECUTION TIME (SEC)	CORE (WORDS)
Trigonometric Functions			
SIN/COS	13	251-349	315
ARCTAN	7	235 avg.	150
ARCSIN	5	208-348	134
Integration	18	30	4
Conditional Branch	27	16	9-12

Table I
Operational Requirements

C. SELECTION OF THE TRIGONOMETRIC FUNCTIONS

As can be seen from Table I, considering the number of repetitions to be fixed, the greatest possibility of achieving reductions in executions time and core storage would occur with the trigonometric functions; therefore only the trigonometric functions were selected for analysis in this thesis.

III. AVAILABLE TRIGONOMETRIC COMPUTATION METHODS

A. POLYNOMIAL EVALUATION METHODS

References 4 through 8 describe various methods of computing the trigonometric functions. These methods were compared in order to determine which would provide the desired accuracy with a limited word size, and at the same time minimize the execution time and core requirements. The following methods appeared feasible:

1. A Telescoping Power Series

In this method [4, 5], a function is first expanded by a Taylor series. For instance, for the sine:

$$\mathrm{SIN}(\pi x/2) = a_1x - a_3x^3 + a_5x^5 \ . \ . \ .$$

The series is truncated to the point that the error is slightly greater than that desired. In this case, the absolute value of the error in terminating the expansion at three terms for the SIN is less than or equal to 0.00468 over the range of x from 0 to 1. This would mean 7-8 bits of accuracy in the MK 152 computer, which is slightly less than that desired. The expansion is then carried one term further and then telescoped by replacing the last term of the expansion with the Chebyshev expansion for that last term. For this example, the Chevyshev expansion is:

$$x^7 = 1/64(b_5x^5 - b_3x^3 + b_1x + T_7)$$

The magnitude of T_7 never exceeds one; therefore the absolute value of the error is bounded. After replacing the last term with the Chebyshev expansion and regrouping terms, the error will be determined by the magnitude of the coefficient of the last term in the telescoped series.

For SIN(π x/2), the maximum value of the error is $\pm$ 0.0000731529, which is 13-14 bits of accuracy. The telescoping process has enabled reduction of the error as compared to a Taylor series for a given number of terms.

2. A Chebyshev Expansion

The Chebyshev polynomials are generated from a sequence of cosine functions using the transformation $\theta = \cos^{-1} x$ to obtain $T_n(x) = \cos n\theta$. Repeated application of the trigonometric identity

$$\cos n\theta = 2 \cos\theta \cos(n-1)\theta - \cos(n-2)\theta$$

will yield higher-order Chebyshev polynomials. The advantage of the Chebyshev method is that the maximum error does not occur at the end points of the function, but rather at intervals in the range of the function [4,5].

3. Newton's Divided Difference Polynomial

This method [6] uses a pre-defined table for interpolating values of a function. The function is evaluated as follows:

$$f(x) = P_n(x) + R_n(x)$$

where $P_n(x) = f(x_0) + (x-x_0)f(x_1,x_0) + (x-x_0)(x-x_1)$

$$f(x_2,x_1,x_0) + \ldots + (x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

$$f(x_n,x_{n-1},\cdots,x_0)$$

and where $R_n(x) = \prod_{i=1}^{n}(x-x_i) f(x, x_n, x_{n-1},\ldots,x_0)$, the error.

Table II is an example of a divided difference table that could be used for finding the cosine of an angle θ. Evaluating the cosine using this method, the absolute value of the error is less than 10^{-4}, which provides 13-14 bits of accuracy. Generally, the error is less than this maximum bound. For instance, the cosine evaluated at θ equal to 0.25 has an error of 1.77×10^{-5}.

i	θ_i Radians	$\cos \theta_i$	$f_1(\ )$	$f_2(\ )$	$f_3(\ )$	$f_4(\ )$	$f_5(\ )$
0	0.0	1.0000					
			-0.09967				
1	0.2	0.9801		-.49211			
			-0.24730		0.03711		
2	0.3	0.9553		-.47727		0.03967	
			-0.34275		0.06091		-0.00297
3	0.4	0.9211		-.45291		0.03760	
			-0.47863		0.07971		
4	0.6	0.8255		-.42102			
			-0.60493				
5	0.7	0.7648					

Table II
Divided Difference Polynomial Table

4. Least Squares Fit

This method minimizes the average square difference between the polynomial and the true trigonometric function. The difference is squared and integrated over the range of the function. Partial derivatives are taken with respect to each of the polynomial coefficients, and each of the resultant equations set equal to zero to find the minimum of the function. If n equals the number of terms in the polynomial then the result will be n equations in n unknowns. All of the equations will be linear and in terms of the polynomial coefficients. This method will not produce the closest approximation to the function. Using this method, for example to obtain the coefficients for a polynomial representation of COS($\pi x/2$) results in the following equations:

The polynomial representation is:

$$\mathrm{COS}(\pi x/2) = \sum_{r=0}^{n} a_{2r} x^{2r}$$

In the first equation that follows, then, the a's are the coefficients to be determined.

$$f(x) = \int_0^1 \left(\sum_{r=0}^{n} a_{2r} x^{2r} - \mathrm{COS}(\pi x/2) \right)^2 dx$$

$$= \sum_{r=0}^{n} \frac{a_{2r}^2}{2r+1} + 2 \sum_{s=r+2}^{n+2} (-1)^{r+1} a_{2r} a_s +$$

$$2 \sum_{s=0}^{r} (-1)^s \frac{2r!\, a_{2r}}{(2r-2s)!(\pi/2)^{2s+1}}$$

$$\frac{\partial f}{\partial a_0} = 2 \sum_{r=0}^{n} \frac{a_{2r}}{2r+1} (-1)^r + 2A = 0$$

$$\frac{\partial f}{\partial a_2} = 2 \sum_{r=0}^{n} \frac{a_{2r}}{2r+3} (-1)^r + 2B = 0$$

$$\begin{array}{ccc} \cdot & \cdot & \cdot = \cdot \\ \cdot & \cdot & \cdot = \cdot \\ \cdot & \cdot & \cdot = \cdot \end{array}$$

$$\frac{\partial f}{\partial a_{2n}} = 2 \sum_{r=0}^{n} \frac{a_{2r}}{2r+2n-1} (-1)^r + 2C = 0$$

Where A, B,..,C are constant terms resulting from the last summation in the f(x) equation.

B. TABLE LOOKUP METHODS

The trigonometric functions can be evaluated by pre-storing the sine and cosine values in a table in computer memory. Obviously, not all values of the function over the entire range of the function can be stored because of physical limitation on table size; therefore some interpolation method must be employed.

1. Linear Interpolation

The values in the table above and below the input angles can be extracted. An interpolation increment would be determined by a comparison of the magnitude of the difference in the input angle from adjacent table indices and the difference in the extracted values, i.e.,

$$\sin(A_i+a) \approx \sin(A_i) + \frac{a}{A_{i+1}-A_i}\,(\mathrm{SIN}(A_{i+1})-\mathrm{SIN}(A_i))$$

2. Approximations

The input angle can be separated into the sum of two or more angles. This separation is done in order to be able to use approximations for smaller portion of the angle. Table entries will only have to be provided for the major portion of the angle; therefore the size of the table will be less than that required to represent the entire angle. The trigonometric identities for the sine and cosine of the sum of angle can be used for the evaluations after the table value has been extracted. For instance, for the sine the trigonometric identity is:

$$\mathrm{SIN}(A_i+a) = \mathrm{SIN}(A_i)\mathrm{COS}(a) + \mathrm{COS}(A_i)\mathrm{SIN}(a)$$

where A_i represents the major portion of the angle, and a the minor part of the angle.

IV. ANALYSIS OF POLYNOMIAL COMPUTING METHODS

A. PURPOSE OF ANALYSIS

The series approximations to the trigonometric functions given in Refs. 4 through 8 provide high degree of accuracy in the evaluations. This accuracy is attained, in part, by precisely defining the polynomial coefficients to a large number of decimal places. None of the references, however, examined the effect of rounding off the coefficients in order to apply the polynomials to a limited word size computer such as the MK 152. In addition, the polynomials were written for applicability in a floating point arithmetic mode of operation, which is not presently implemented in the MK 152 computer. This analysis was conducted to determine if:

1. The polynomial coefficients could be rounded to a degree expressible as single precision numbers in a limited word length machine without serious degradation of accuracy.

2. The errors generated by fixed point arithmetic operations of the polynomial would not seriously degrade the accuracy.

These problems could be reduced by the use of double precision (36bits) arithmetic and/or floating point; however the increased computation times and additional core storage requirements for these modes would be unacceptable in the fire control system.

B. ERROR ANALYSIS PROCEDURES

A set of polynomials from Ref. 4 with input ranges of 0 -$\pi/2$ and a set of polynomials generated by the least square error method with input ranges of 0 - 1 were used for evaluations. These two sets of polynomials were chosen as representing the largest variation in coefficient magnitudes and input ranges; therefore they would probably produce the greatest difference in errors due to rounding operations. An analysis was first conducted using a gross error criterion in order to obtain an estimate of the number of terms required in the trigonometric polynomials. From the results of this initial analysis, a detailed simulation was done to:

1. Achieve a true bound on the error.

2. Determine the errors due to coefficient rounding alone.

3. Establish the total error because of rounding and arithmetic operations and determine if the resulting error reduced the accuracy below that required in the MK 152 computer.

C. SINGLE FUNCTION ERROR ANALYSIS

1. Determination of Gross Error Bounds

Reference 7 provided a method of determining the maximum error bounds due to roundoff in the coefficients and due to roundoff in arithmetic operations. It was found, however, that the computations required using this method were difficult and cumbersome. The method contained in Appendix B was developed in order to simplify the calculations. A computer algorithm was generated using this method to determine the gross errors generated in rounding the polynomial coefficients. In successive runs, the coefficients were rounded to 3,4 and 5 decimal

places to simulate the word lengths of small machines. In each case, the polynomial was nested according to Horner's rule. As noted in Ref. 4, for polynomials with small numbers of terms, this form provides the minimum number of arithmetic operations. The results of these evaluations for the sine and cosine were that the gross error was less than $\pm 10^{-2}$ after the rounding operations for 3,4, and 5 term polynomials, except when the 4 and 5 term polynomial coefficients were rounded to 3 decimal places. Polynomials with more than 5 terms could not be represented, because the coefficients of the least significant terms rounded to zero with 5 or fewer decimal places. On the basis of these results, further analysis was limited to polynomials with 5 or fewer terms.

2. Refinement of Error Bounds

The polynomial expansions used were:

$$COS(Z) = a_0 - a_2Z^2 + a_4Z^4 - a_6Z^6$$

$$SIN(Z) = a_1Z - a_3Z^3 + a_5Z^5 - a_7Z^7$$

The coefficients from Ref. 4 and those of the least square error expansions differ because of the magnitude of the input values. The trigonometric functions from Ref. 4 are of the form:

$COS(X) \qquad 0 \leq X \leq \pi/2$

$SIN(X) \qquad 0 \leq X \leq \pi/4$

Note: the range for SIN was the maximum available in Ref. 4. This method of evaluation will be referred to as method 1. The second method, which will be referred to as method 2, was generated from the least mean square error, and has the form:

$COS(\pi X/2) \qquad 0 \leq X \leq 1$

$SIN(\pi X/2) \qquad 0 \leq X \leq 1$

Using both methods 1 and method 2, the effects of rounding the coefficients alone on the approximation of the polynomial functions was determined. Then, the effects of the rounding of coefficients plus the affects of arithmetic operations on errors was determined.

a. Coefficient Roundoff Effects Alone

A computer program was generated to compare the effect of roundoff of the polynomial coefficients alone with the double precision trigonometric functions in the IBM 360/67 library. Figures 2 through 8 show the errors resulting from the rounding operations. The increasing error in the functions with decrease precision clearly illustrates the poorer approximations resulting from the rounding of the coefficients. In all of the figures, it can be seen that the rounding to 3 decimal digits greatly magnified the error. Thus, a word size of at least 12 bits is required in a digital computer if polynomial evaluations of the trigonometric functions are desired.

b. Coefficient Roundoff Plus Arithmetic Effects

A digital computer program was written to simulate the arithmetic operations in a fixed point arithmetic logic unit (ALU) of a digital computer. Within the program, two's complement arithmetic was used. Algorithms were generated for the following operations:

(1) Multiply. Booth's algorithm [9] was used because of the high speed of the algorithm and the ability to multiply signed numbers with no special manipulation.

(2) Add/Subtract. Adds were accomplished in normal two's complement form. Subtracts were done by complementing the subtrahend and adding.

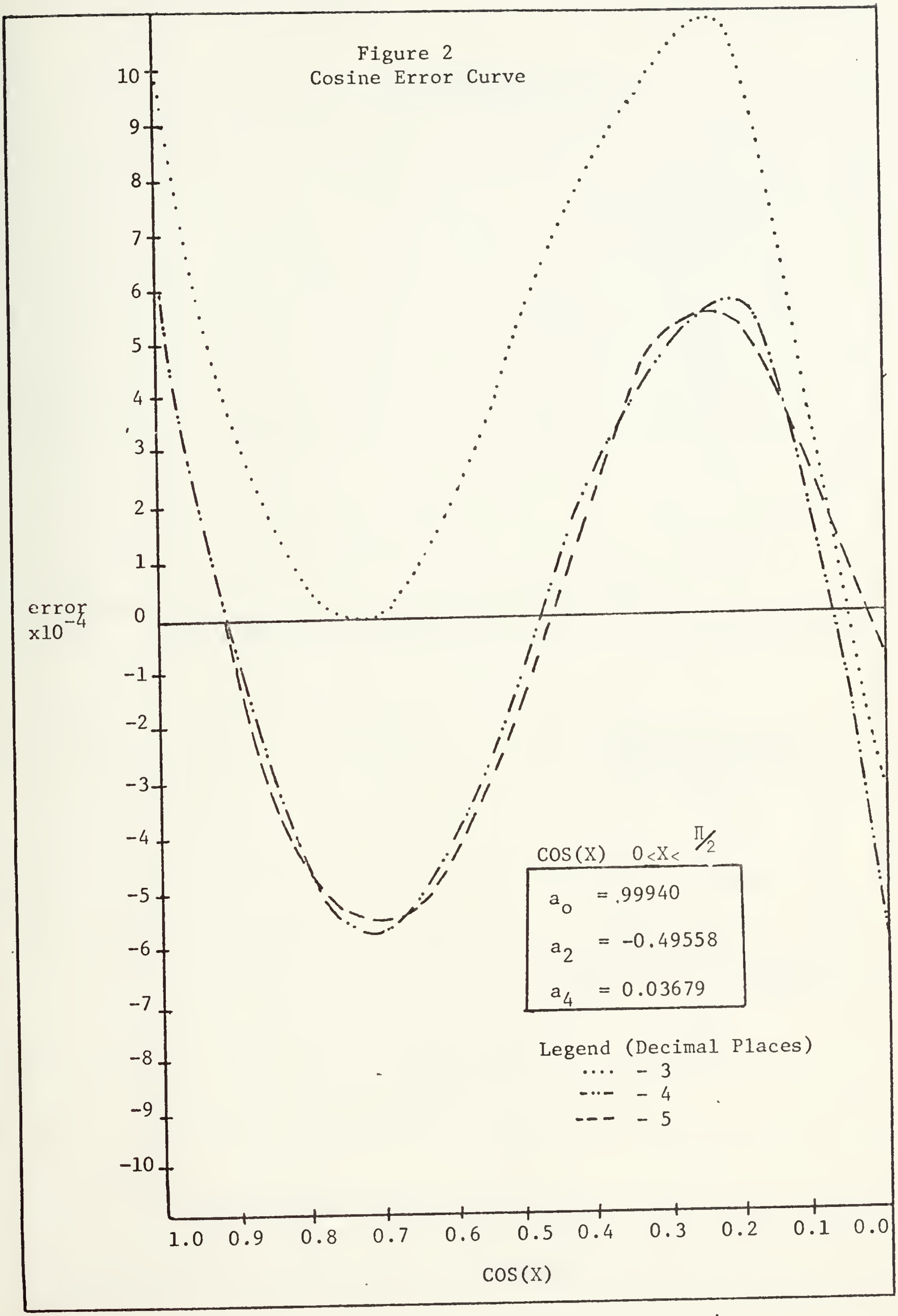

Figure 2
Cosine Error Curve
10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
error
x10^{-4}
COS(X) 0<X<$\pi/2$
a_o = .99940
a_2 = -0.49558
a_4 = 0.03679
Legend (Decimal Places)
.... - 3
-..- - 4
--- - 5
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
COS(X)

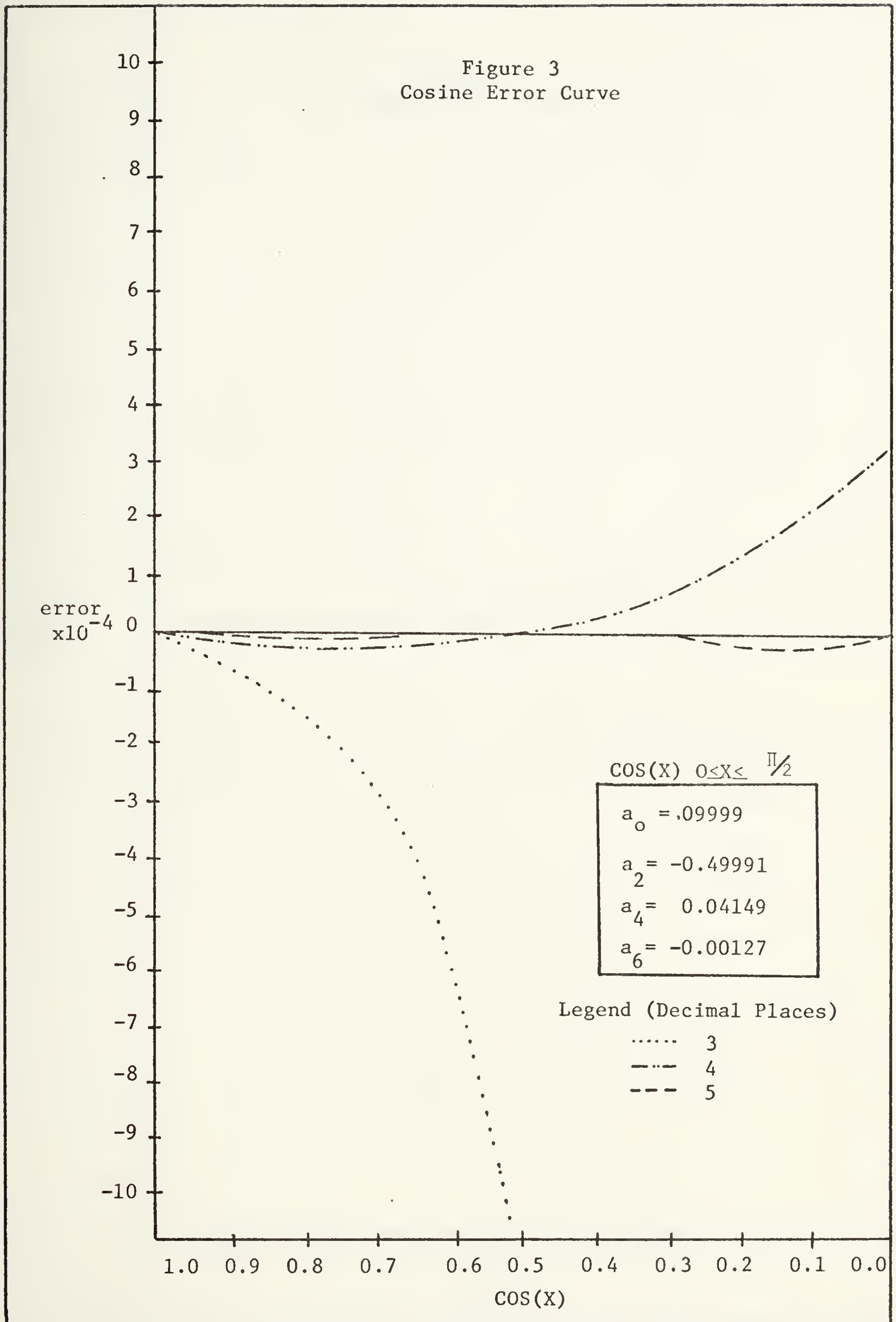

Figure 3
Cosine Error Curve

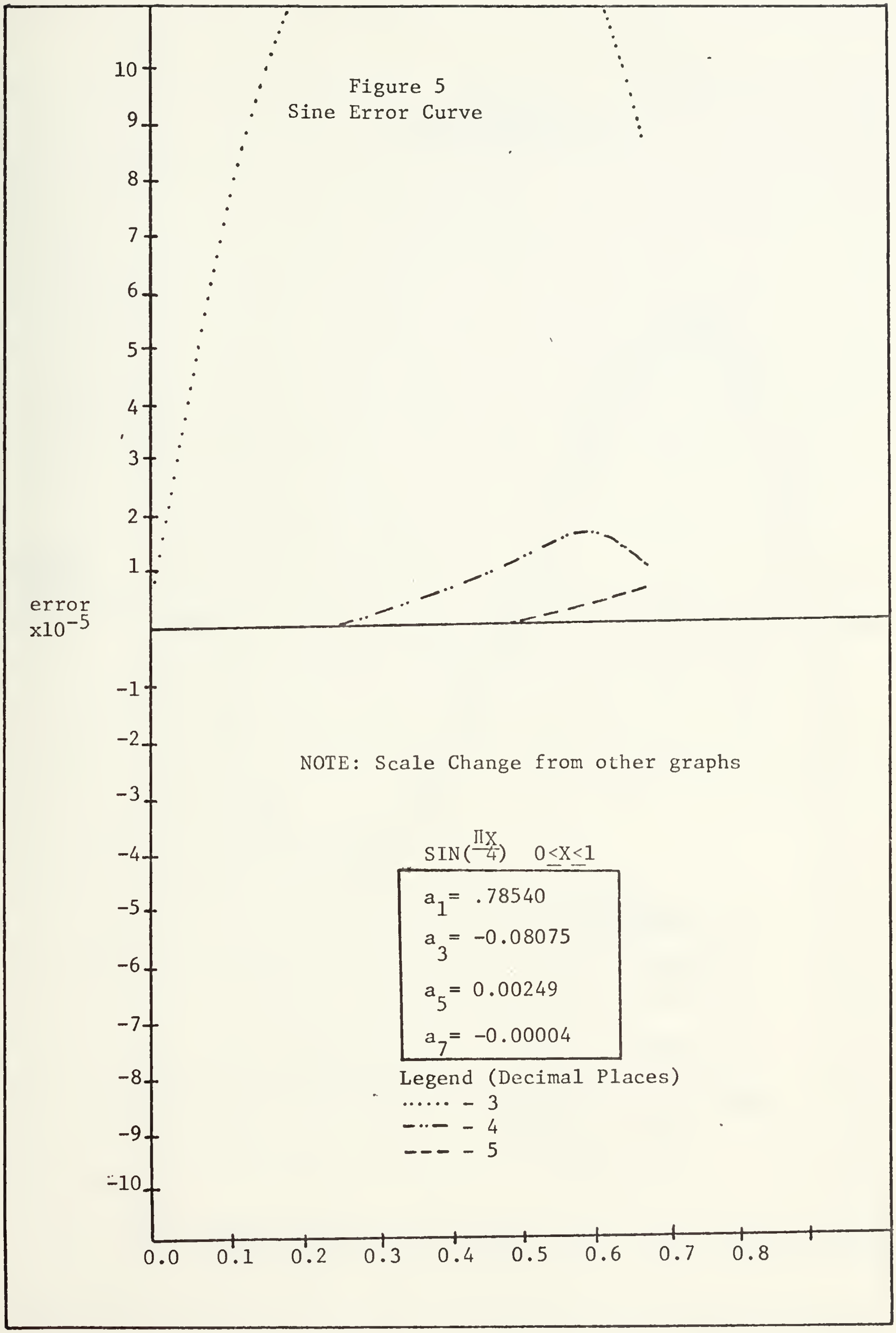
Figure 5
Sine Error Curve
error
x10^-5
10
9
8
7
6
5
4
3
2
1
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
NOTE: Scale Change from other graphs
SIN(ΠX/4) 0≤X≤1
a1= .78540
a3= -0.08075
a5= 0.00249
a7= -0.00004
Legend (Decimal Places)
...... - 3
-..- - 4
--- - 5

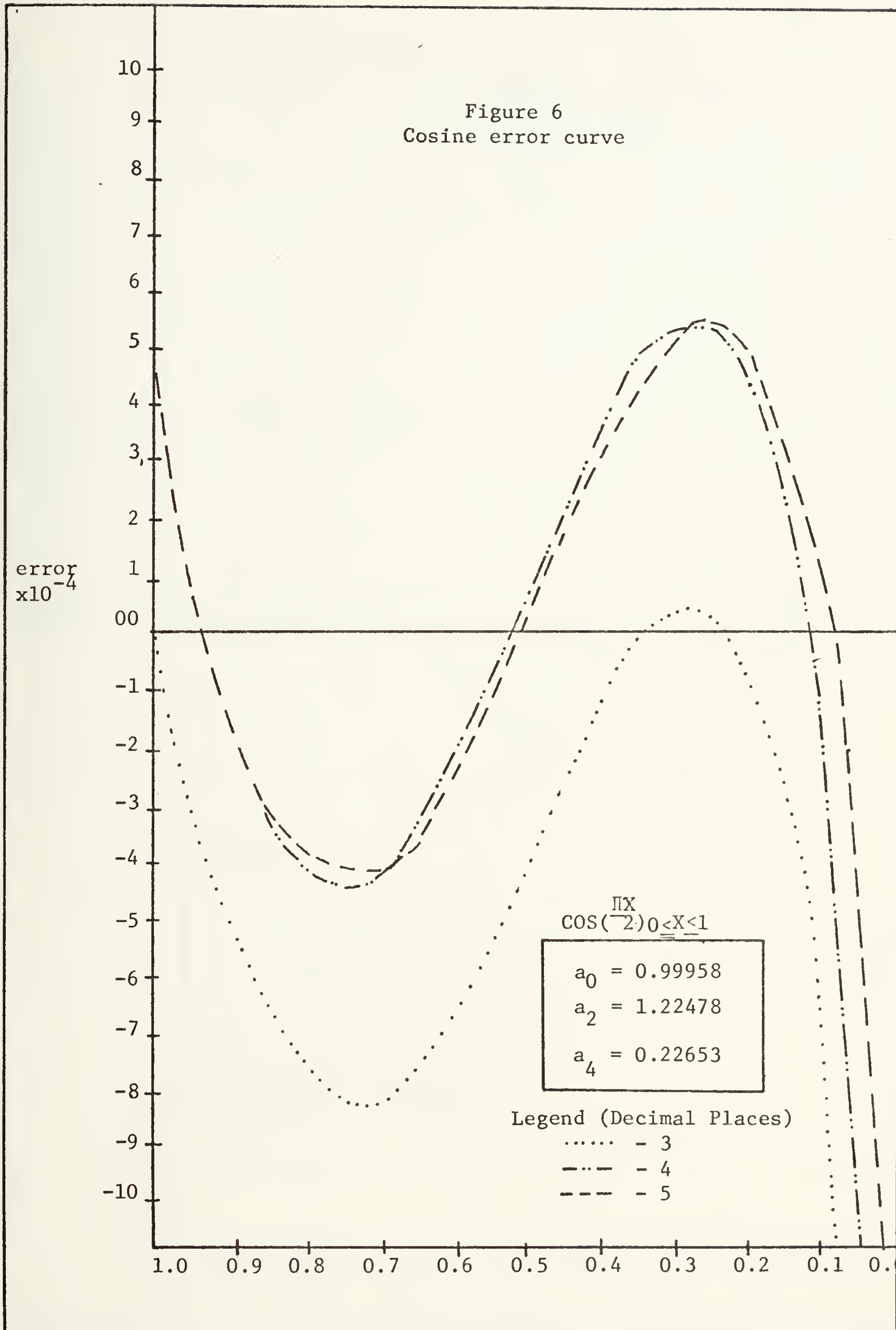

Figure 6
Cosine error curve

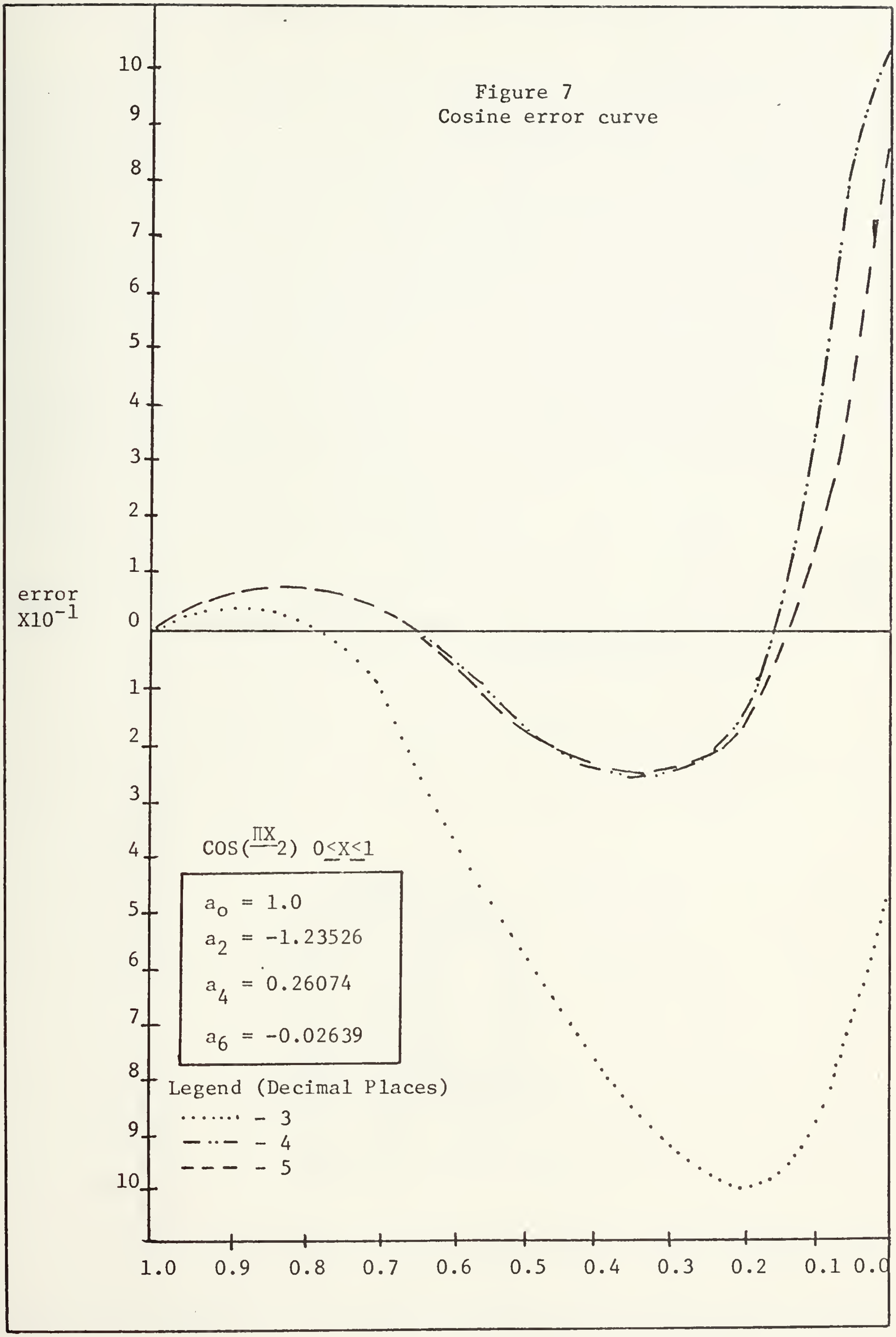

Figure 7
Cosine error curve
error
X10^{-1}
10
9
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
10
$\cos(\frac{\Pi X}{2})$ $0 \le X \le 1$
a_o = 1.0
a_2 = -1.23526
a_4 = 0.26074
a_6 = -0.02639
Legend (Decimal Places)
- 3
- 4
- 5
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

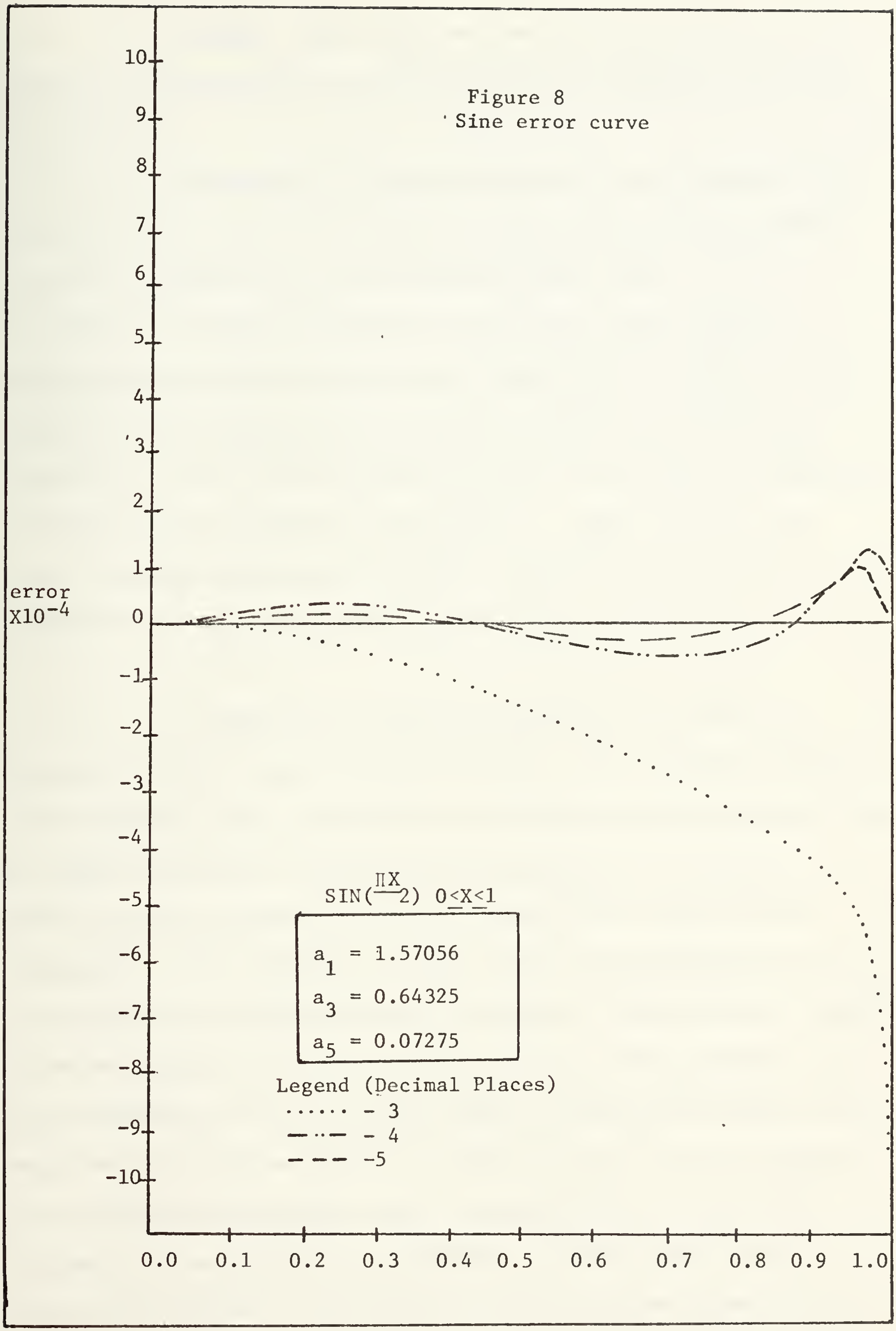
Figure 8
Sine error curve
error
X10^-4
10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
SIN(ΠX/2) 0≤X≤1
a1 = 1.57056
a3 = 0.64325
a5 = 0.07275
Legend (Decimal Places)
······ - 3
—··— - 4
— — — -5

(3) Shift. Left shifts were conducted with a zero fill in the least significant bits. Right shifts were accomplished with sign bit extension.

(4) Complement. Complementation was done by searching the bit string from right to left until the first set bit was encountered. That bit was skipped and all the remaining bits toggled.

The insertion of output statements at the end of each subroutine provided data as to which operations were contributing the most to error generation. The primary error generation occurred in the multiply operations. Horner's form was again utilized to minimize the number of multiplies and the magnitude of the numbers. Using this form, the largest number encountered was $(\pi/2)^2$ in the method 1 evaluations; therefore the binary point could be placed after the third bit.

During the program runs, the sine and cosine were evaluated over the range of $0 - \pi/2$; however, since these functions are "well behaved", only selected points over that range were actually taken. At each of the points, several small increments above and below the point were also evaluated, so that the trend of the error could be established. After completion of runs over the full range for each of the functions, the input range was restricted to where the error appeared to be the largest for each function. Additional runs were made with small increments in the restricted range to establish the maximum error magnitude as closely as possible. Tables III and IV contain selected values obtained during these runs. In each table, the maximum and minimum error points have been included. As can be seen from both tables, the magnitude of the error for the cosine grew larger as the

cosine approached zero. For both methods, the magnitude of the error with three terms for the cosine was unacceptable over the full range, as well as the four term cosine by method one. The four term method two cosine could be acceptable for implementation, because of over 11 bits of accuracy in the interval. The three term sine by method two produced over 12 bits of accuracy on the same interval. Consideration was given to restricting the range to $[0 , \pi/4]$. This would have reduced the error in the polynomial approximation (Figure 5), but the computation time would have been increased depending on the octant of input angle. The amount of error reduction achievable with this range reduction was not computed.

For the method 1 runs, a comparison was made of the effects of rounding versus truncation during the arithmetic operations. In truncation, the least significant bits are dropped regardless of magnitude. In round-up, the 18th bit is always set to one regardless of the magnitude of the least significant bits. In round-off, the 19th bit is examined. If this bit is set, then one is added to the 18th bit. If the 19th bit is not set, the upper 18 bits are left unchanged. As can be seen from Table III, very little difference in accuracy was detected among the errors due to truncation, round-up and round-off. The increase in computation time required for either rounding operation versus the minimal accuracy gained would eliminate rounding as a desirable operation in the fire control system. For the method 2 runs, only truncation was used.

Table III-1

COS(X) - 3 terms, 5 decimal places

OUTPUT COS(X)	Truncation	TOTAL ERROR Round-up	Round-off
0.00000	0.00122	0.00123	0.00123
0.27740	0.00040	0.00040	0.00040
0.36646	0.00018	0.00019	0.00019
0.47942	0.00030	0.00031	0.00031
0.50708	0.00042	0.00047	0.00047
0.52075	0.00074	0.00074	0.00074
0.86586	0.00049	0.00010	0.00059
0.96272	-0.00011	-0.00007	-0.00007
0.98853	-0.00039	-0.00041	-0.00041
0.99585	-0.00042	-0.00042	-0.00042

Table III-2

COS(X) - 4 terms, 5 decimal places

OUTPUT COS(X)	Truncation	TOTAL ERROR Round-up	Round-off
0.00000	-0.00133	-0.00134	-0.00134
0.24740	-0.00399	-0.00400	-0.00400
0.36627	-0.00127	-0.00128	-0.00127
0.47942	-0.00017	0.00031	0.00031
0.50661	0.00047	0.00047	0.00047
0.52075	0.00073	0.00074	0.00074
0.86586	0.00060	0.00059	0.00057
0.96272	0.00042	0.00042	0.00042
0.98893	0.00008	0.00009	0.00009
0.99626	0.00007	0.00007	0.00006

Note: The negative sign on the Error indicate that the computed value was less than the real value.

METHOD ONE ERRORS

Table III

Table IV-1

COS(πX/2)- 3 terms, 5 decimal places

OUTPUT COS(πX/2)	TOTAL ERROR	OUTPUT COS(πX/2)	TOTAL ERROR
0.00000	0.00249	0.83147	0.00035
0.00077	0.00116	0.92388	-0.00002
0.19509	-0.00290	0.92621	0.00009
0.38268	-0.00167	0.98086	-0.00024
0.70711	0.00020	0.99577	-0.00038
0.71143	0.00026	0.99881	-0.00024
0.77301	0.00034	0.99996	-0.00029

Table IV-2

COS(πX/2)- 4 terms, 5 decimal places

OUTPUT COS(πX/2)	TOTAL ERROR	OUTPUT COS(πX/2)	TOTAL ERROR
0.00000	0.00001	0.83147	-0.00024
0.00077	-0.00010	0.92390	0.00000
0.19509	0.00040	0.92621	-0.00021
0.38268	0.00031	0.98086	-0.00009
0.70711	-0.00014	0.99577	-0.00006
0.71141	-0.00005	0.99881	0.00002
0.77301	-0.00024		

Table IV-3

SIN(πX/2)- 3 terms, 5 decimal places

OUTPUT SIN(πX/2)	TOTAL ERROR	OUTPUT SIN(πX/2)	TOTAL ERROR
0.048685	-0.00010	0.63439	0.00000
0.091909	-0.00011	0.70275	0.00000
0.146730	-0.00012	0.70707	-0.00008
0.194714	-0.00013	0.92388	-0.00005
0.377007	-0.00005	0.98079	-0.00019
0.382639	-0.00007	0.99999	0.00000
0.555570	-0.00003	1.00000	0.00000

Table IV

Method Two Error

3. Evaluation of ARCTAN

The inverse tangent can be approximated by a series which is of the same form as that for the sine, only with different coefficients; therefore, the procedures above were repeated for the inverse tangent. The inputs for the inverse tangent are Y divided by X, which goes to zero as Y goes to zero and infinity as X goes to zero. This range can be reduced to 0 - 1 by evaluating the smaller of Y/X or X/Y. The result of this evaluation may have to be rotated 90° depending on the quadrant and whether Y or X is the larger value. As with the sine and cosine, selected values obtained in the computer simulations runs are listed in Table V. The 3 term ARCTAN has an accuracy of more than 10 bits over the range [0 , 1], and the 4 term ARCTAN has an accuracy of more than 12 bits for that range. The addition of the fourth term would require an additional add and multiply. In the MK 152 computer, this would require 3 additional core locations and 21 microseconds additional computation time. Thus, 2 additional bits of accuracy could be obtained with those expenditures.

4. Evaluation of ARCSIN

The evaluation of the inverse sine becomes difficult as the input approaches the value of 1. There is no simple solution for this problem as with the ARCTAN. The ARCSIN can be evaluated over the full range by using the trigonometric identity

$$\mathrm{ARCSIN}(X) = \mathrm{ARCTAN}\left[\frac{X}{\sqrt{1-X^2}}\right]$$

To be able to utilize this effectively in the fire control system, the evaluation of the square root would have to be rapid. An analysis of the square root was not conducted as part of this thesis; however, it

Table V-1

ARCTAN(X) - 3 terms, 5 decimal places

OUTPUT ARCTAN	TOTAL ERROR	OUTPUT ARCTAN	TOTAL ERROR
0.78538	0.00051	0.12051	-0.00051
0.71883	-0.00069	0.10894	-0.00054
0.64350	0.00005	0.09348	-0.00046
0.46365	0.00034	0.06242	-0.00029
0.46052	0.00040	0.05853	-0.00030
0.35877	-0.00025	0.03124	-0.00023
0.24495	-0.00069	0.02734	-0.00024
0.18535	-0.00066	0.01562	-0.00012
0.12435	-0.00058	0.01172	-0.00012

Table V-2

ARCTAN(X) - 4 terms, 5 decimal places

OUTPUT ARCTAN	TOTAL ERROR	OUTPUT ARCTAN	TOTAL ERROR
0.78538	0.00002	0.12051	-0.00015
0.71883	0.00016	0.10894	-0.00018
0.64350	-0.00007	0.09348	-0.00010
0.46365	0.00010	0.06242	-0.00016
0.46052	0.00005	0.05853	-0.00006
0.35877	0.00000	0.03124	-0.00011
0.24494	-0.00008	0.02734	-0.00012
0.18535	-0.00017	0.01562	-0.00012
0.12435	-0.00009	0.01172	-0.00012

Table V

ARCTAN Evaluation

appears that the 190-196 microseconds [10] now required for the square root evaluation could be shortened considerably by optimizing the starting value of the Newton-Raphson iteration by the method in Ref. 11. In the present implementation, the ARCSIN requires 208-348 microseconds for evaluation, so implementation by the ARCTAN should be less than or equal to this time. The implementation of a common subroutine for both functions would provide a saving of about 130 storage locations.

D. MULTIPLE FUNCTION ERROR ANALYSIS

In the fire control problem, the greatest number of operations on a set of quantities, and hence the greatest error, is in the evaluation of the coordinate transformation matrices:

$$\begin{bmatrix} \cos A \sin B \\ \cos A \cos B \\ \sin C \end{bmatrix} = \begin{bmatrix} \cos Zo & 0 & -\sin Zo \\ \sin Zo \sin Eio & \cos Eio & \cos Zo \sin Eio \\ \sin Zo \cos Eio & -\sin Eio & \cos Zo \sin Eio \end{bmatrix} \times \begin{bmatrix} \cos W \sin X \\ \cos W \mathrm{coo} X \\ \sin W \end{bmatrix}$$

The largest error for the single trigonometric functions was 0.00040 for the cosine of 78.75° (Table IV-2). In the 3x3 matrix, Eio is ship's pitch angle and Zo is ship's roll angle; therefore, as these angle never exceed about 45°, the cosine of these values will never be evaluated at the maximum error point. In fact, Eio is generally limted to less than a few degrees, which is the range where the cosine error is minimum. The quantities W and X are obtained from the FCS elements. Both inputs can attain the maximum error simultaneously. To determine how much error would occur if both W and X were at 78.75°, the product of cosWcosX was first taken. The resultant error for the product was 0.00032, which is less than for the cosine alone. Thus, the maximum error is not at the point both functions individual errors are maximum.

The maximum error would occur when either X or Y was at 78.75° and the other value was 0°. At that point, the error would be 0.0004, which is better than 11 fractional bits of accuracy. The outputs of the matrix, cosAsinB and cosAcosB are used as the inputs for the inverse tangent evaluation. As can be seen from the matrix, the product of cosWcosX doesn't enter into the calculation of cosAsinB. The maximum error, therefore should be less than above. The calculated maximum error is 0.00031. With similar reasoning as above for cosAcosB, the maximum error will be 0.00040. Thus both inputs to the ARCTAN subroutine retain more than 11 bits of accuracy for the full range of these values. After evaluation of the ARCTAN with four terms, more than 11 bits of accuracy were still available. The sinC result will have a maximum error of 0.00020, which is better than 12 bits of accuracy. This is used as an input to the ARCSIN subroutine. If the procedure noted in the section on ARCSIN evaluation is used, then the accuracy achievable in the subroutine will depend on the accuracy of the square root calculation.

E. POLYNOMIAL EVALUATION SUMMARY

The use of a 4 term cosine and 3 term sine by method 2 and a 4 term ARCTAN will produce results with sufficient accuracy for implementation in the MK 152 computer.

It should be noted that the binary terms used in the polynomials may not be the optimal set for maximum error reduction. After determining the magnitude of the coefficients in decimal form, the terms were converted to binary. As a true binary representation was not possible, some deviation in binary representation can be expected. During the simulation runs, some of the least significant bits were changed to

determine if the resultant error was improved or degraded by the change. Sufficient bit changing was done to insure that the functions were "reasonably" accurate; however optimization was not attempted. A minor increase in accuracy could probably be achieved by the use of an optimization program.

V. ANALYSIS OF TABLE LOOKUP METHODS

A. GENERAL CONSIDERATIONS

1. Form of Inputs

The values input to the MK 152 computer are scaled in Binary Angular Measurements (BAMS). As described in Ref. 12, the high order bit is normally equal to 360° in BAMS, and each succeeding bit is equal to one-half the value of the preceding bit. In implementation in the MK 152 computer, however, this form is modified so that the high order bit equals 180°. This limits the expression of angular values to the range from zero degrees to slightly less than 360°.

2. Range Reductions

Although the inputs can span the range from 0° to 360°, it is desirable to reduce the range to some smaller span so that the corresponding table size can be reduced. This reduction can be accomplished by extraction of the high order bits and processing them separately. For instance, the extraction of the two highest order bits in an input angle to the MK 152 would reduce the range table to 0° to 90°. After extraction of the trigonometric value of the angle from the table, the two bits extracted would have to be evaluated to determine the rotation required to place the trigonometric value in the proper quadrant. Each bit extraction reduces the range to one-half the size of the previous range, and consequently reduces the table size by one-half. The computation time, though, is increased by each extraction because of the necessity of providing separate evaluations for the values of the extracted bits. So the primary considerations in a table lookup program

are the tradeoff in table size versus the speed of conversion. Figure 9 is a graph of the core storage requirements for a table of sine or cosine values plotted against the maximum difference in the trigonometric values between two adjacent storage locations for either a 45° or a 90° range. It can be seen from the graph that increases in table value accuracy rapidly becomes expensive in terms of core storage requirements.

3. Use of Approximations for Table Size Reductions

Figure 9 shows that to achieve 10 bits of accuracy (.001 maximum difference between adjacent storage locations) would require a large table if either a 45° or 90° range is used. After extraction of the upper bits for range reduction, the remainder of the word can be subdivided into a major angle portion and one or more minor angle portions. The major angle portion would be evaluated by table lookup and the minor portion obtained by some other method such as additional tables, polynomial evaluations, or by interpolation. With the extraction of the low order bits for separate evaluation, the number of trigonometric values required to be stored in the table is reduced. Again referring to Figure 9, if the angle increments in the major portion of the input angle are increased so that 6×10^{-3} vice 1×10^{-3} is required between the maximum adjacent table values, then the table could be reduced from 1526 words to 256 words.

The evaluation of the major portion of the angle can be made very fast by using the angle input as the entry address in the table. The speed of evaluation of the minor portion of the angle will depend on the evaluation method used. If polynomials or additional tables are used, then trigonometric identities must be employed to establish the trigonometric value of the total angle. If interpolation is used, then the

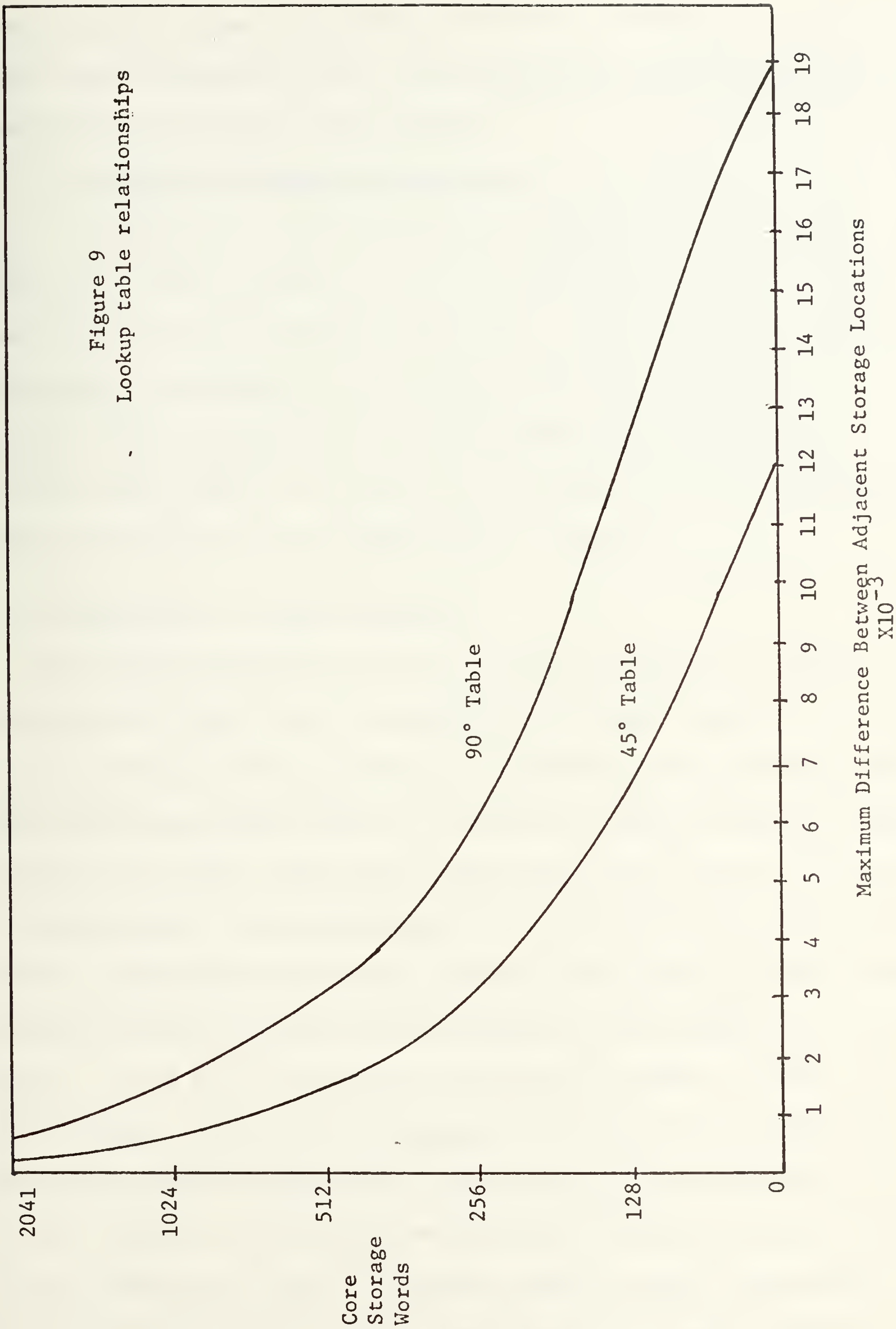

Figure 9
Lookup table relationships

ratio of input values to the table values must be computed. All of these minor angle evaluations utilize some approximation method; therefore some degradation of accuracy can be expected.

4. Summary of General Considerations

Reductions in table size can be accomplished by dividing the angle into several sections. The extraction of the upper bits permits range reductions for the table evaluations. Extraction of the lower bits for separate processing increases the interval between stored table values. The reductions in table size accomplished by these methods requires more complex calculations, which causes increases in computation times, and results in an attendant loss of accuracy.

B. DETERMINATION OF RANGE REDUCTIONS

The number of storage locations required to represent a full 360° for the sine and cosine would be prohibitive; therefore, some range reduction must be utilized. The repetition of the trigonometric values by quadrant with only sign changes, makes it desirable to reduce the range for table lookup to 0° to 90°. This would require extraction of the top 2 bits of the input angle. For implementation in the MK 152 computer, the additional computations would require adding 14 instruction and would increase computation time by 16 to 28 microseconds over that required for a full table. For 10 bits of accuracy, this operation reduces the table size to 1536 words. For the same accuracy, the extraction of an additional bit would reduce the range to 45° and the table size to 768 words. This further reduction would require an additional 7 instructions and 28 microseconds computation time. For further range reductions, the symmetric properties of the sine and cosine used above are no longer applicable; consequently, computation time becomes large.

Thus, the selection of range reduction will depend on whether storage or computation time is the most critical factor.

C. DETERMINATION OF APPROXIMATIONS

After performing the range reduction, the remainder of the input angle can be divided up into two or more subdivisions. The upper bits will represent the major portion of the angle, and will be referred to as angle A. The lower bits, which will constitute the remainder of the subdivisions, will represent a minor portion of the angle. This lower portion will be referred to collectively as angle B, regardless of the number of subdivisions. The trigonometric value of angle A will be obtained by table lookup. Three methods were considered for obtaining the trigonometric values:

1. Linear Interpolation

In this method, the input angle is separated as follows:

quadrant	angle A	angle B

The trigonometric value of angle A is obtained from a table. This value is pertubated by adding an interpolated trigonometric value for angle B. The equation to accomplish this is:

$$\mathrm{SIN}(A_i + B) = \mathrm{SIN}(A_i) + \frac{B}{A_{i+1} - A}\,(\mathrm{SIN}(A_{i+1}) - \mathrm{SIN}(A_i))$$

where A_i is the table entry for angle A and A_{i+1} is the next adjacent table entry.

Considering the midpoint in the interval between adjacent table values as approximately the point of poorest interpolation, the error produced for the sine near 0° was 0.00317. This is less than 9 bits of accuracy; therefore, this method was eliminated from consideration to implement in the fire control system.

2. Table Plus Polynomial Evaluation

This method uses the following trigonometric identities for evaluation of the sine and cosine:

$$\mathrm{SIN}(A+B) = \mathrm{SIN}(A)\mathrm{COS}(B) + \mathrm{COS}(A)\mathrm{SIN}(B)$$

$$\mathrm{COS}(A+B) = \mathrm{COS}(A)\mathrm{COS}(B) - \mathrm{SIN}(A)\mathrm{SIN}(B)$$

The sine and cosine of angle A are obtained by table lookup. The sine and cosine of angle B are obtained from a Taylor series expansion. If the angle B is sufficiently small, then only a one term expansion will be required for the sine and cosine of B to obtain a good approximation. From the single term expansions, the SIN(B) is approximately equal to B and the COS(B) is approximately equal to 1. Inserting these quantities in the original trigonometric identities provides the following simplifications:

$$\mathrm{SIN}(A+B) = \mathrm{SIN}(A) + (B)\mathrm{COS}(A)$$

$$\mathrm{COS}(A+B) = \mathrm{COS}(A) - (B)\mathrm{SIN}(A)$$

This method is presently implemented in the MK 152 computer. In that subroutine, angle A and angle B are each 8 bits. Thus, 256 words are required in the table for referencing by angle A. The maximum value of angle B is 42.0228'; therefore the one term expansions provide a good approximation. Fourteen bits of accuracy were achieved over the range $0 - \pi/2$ for both the sine and cosine in the implementation. Reducing A to 7 bits and increasing B to 9 bits would reduce the required table to 128 words, but double the maximum value of B. Using the one term Taylor series expansion, the error in the approximation reduces the accuracy over the range to less than 9 bits. An additional term was added to both expansions to increase the accuracy; however, the increased number of operations extends the computation time beyond that required for a

polynomial evaluation alone, and a table was still required. Thus, the highest efficiency in terms of time and core storage using this method is as presently implemented.

3. Multiple Tables

The trigonometric identities from the last section were used, only the trigonometric values for angle B were obtained by a table vice approximation. This method was much slower in computation time for evaluation of angle B and required more core storage then using the polynomial approximation for B.

D. COMPARISON OF TABLE LOOKUP METHODS

The present method implemented in the MK 152 computer using a table for the major portion of the angle and one term polynomial approximations for the minor portion of the angle yields the greatest accuracy, for the core and time expended, of any table lookup algorithm of this class.

VI. COMPARISON OF TRIGONOMETRIC EVALUATION METHODS

A. METHOD IMPLEMENTABLE IN THE MK 152 COMPUTER

The primary controlling criterion for the methods analyzed was the accuracy requirement in the TERRIER fire control system. It was found for the polynomial evaluation methods (Section IV) that a 4 term cosine and a 3 term sine would meet the requirement. The 4 term polynomial expansion for the inverse tangent would also meet the accuracy requirements, and, if a sufficiently accurate square root routine were used, the inverse tangent subroutine could be used for the inverse sine. The presently implemented for of table lookup (Section V) using the sum of angle trigonometric identities with a table lookup plus polynomial evaluation provided sufficient accuracy.

B. COMPARISON OF IMPLEMENTATION REQUIREMENTS

1. Sine and Cosine

Table VI is a comparison of the MK 152 computer resources expended by the polynomial evaluation and table lookup methods.

Method	Computation Time (μsec)	Core (words)	Accuracy (bits)
Polynomial	186-204*	55	11-12
Table Lookup	180-194*	315	14

*Time dependent on input angle quadrant.

Table VI

MK 152 Computer SIN/COS Requirements

It can be seen from Table VI that the computation times for the two methods are comparable. Although the accuracy for the polynomial method is slightly less than the table lookup method, even with the degradation

in subsequent operations, the final result is still within accuracy requirements. Thus, a sizeable reduction in storage requirements could be achieved in the SIN/COS subroutines by the implementation of the polynomial evaluation method.

2. Inverse Tangent and Inverse Sine

The ARCTAN and ARCSIN programs were not available during the period of this evaluation, so the standards set forth in Ref. 10 were used as a basis of comparison with the methods analyzed. Table VII is a summary of the MK 152 computer requirements for the polynomial evaluation method for the ARCTAN and the Ref. 10 standards.

Method	Computation Time (μ sec)	Core (Words)	Accuracy (bits)
Polynomial	146-156*	74	12
Ref. 10	235 avg.	150	10-11

* Time Dependent on Quadrant

Table VII

MK 152 Computer ARCTAN Requirements

The polynomial method for ARCTAN provided savings in computation time and core storage with a higher degree of accuracy than the Ref. 10 standards. The use of the ARCTAN subroutine for evaluation of both the ARCTAN and ARCSIN would provide an additional core saving of about 130 words. A savings in computation time could also be achieved for the ARCSIN if the time for the computation of the square root was shortened as noted in Section IV. Even without this reduction, the time for evaluating the ARCSIN by the trigonometric identity for ARCTAN is approximately equal to the upper time bound (348 μ sec) listed in Ref. 10.

VII. ALTERNATIVE IMPLEMENTATIONS OF TRIGONOMETRIC FUNCTIONS

A. ALTERNATIVES CONSIDERED

The trigonometric functions are presently implemented by subroutine programs within the MK 152 computer; therefore consideration was given to other means of implementing these functions in order to achieve more core savings and to enhance the speed of operation. Two primary areas were considered; microprogrammed computers and hardware devices.

1. Microprogrammed Computers

a. General Description

The structure of a microprogrammed computer is similar to that of a conventional computer, except for the implementation of the control section. Figure 10 is a comparison of the two structures.

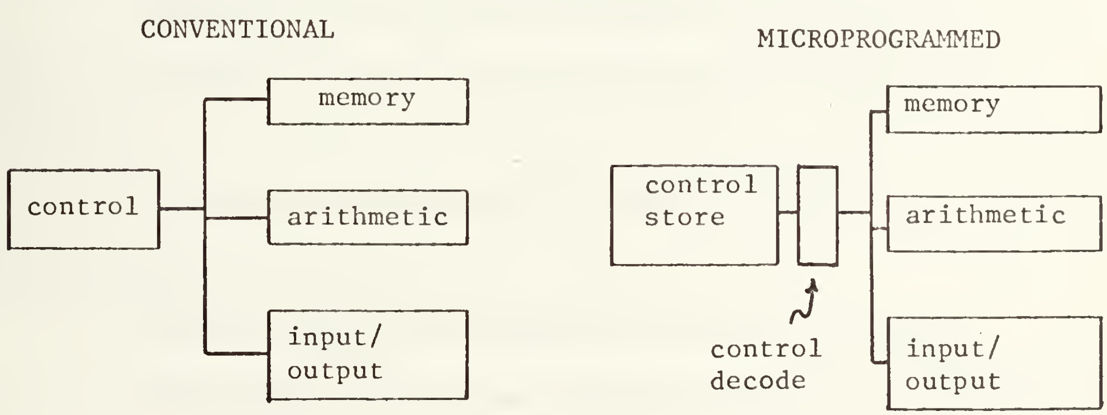

Figure 10

Computer Structures

The control store is usually implemented by read-only memories (ROM). The primary difference in the two control functions above is that in the microprogrammed computer instructions are executed by addressing an entry location in the ROM, which causes execution of a sequence of

micro-operations. For a conventional computer, the control is entirely be means of software programs whose instructions activate certain hardwired logic paths. One of the prime advantages gained by microprogramming is the reduction in the number of hardwired paths for instruction logic. The instruction logic can be easily altered or expanded in a microprogrammed machine by altering the ROM contents (called "firmware"). In a conventional machine an expensive hardware change would be required to achieve the same thing. The microprogrammed computer structure also makes implementation of special instructions easier. The control store structure has a further advantage of being able to execute several micro-operations simultaneously. This increases the speed of operations of these machines.

b. Capabilities of Microprogrammed Computers

The capabilities of microprogrammed computers vary widely [13-18]. For example, some of the capabilities available in these machines are:

(1) Direct implementation of higher level languages such as FORTRAN and ALGOL (HP 2100).

(2) Floating point arithmetic (MICRO 800, HP 2100).

(3) Direct memory access (AMI 7200, UNIVAC 1616, HP 2100, MICRO 800).

(4) Interrupts (AMI 7200, HP 2100, UNIVAC 1616, MICRO 800).

(5) Various memory sizes (MCS 4, UNIVAC 1616, HP 2100, AMI 7200, MICRO 800).

In all of the microprogrammed computers studied, the size and capabilities of the instruction sets would enable implementation of the trigonometric functions in these machines. HEWLETT-PACKARD's microprogrammed computer

(HP 2100) was used as an example microprogrammed computer for implementing the trigonometric functions, so that a comparison of speed of execution and accuracy could be made with the MK 152 computer. Appendix III lists the capabilities of this computer. References 13 and 14 provide a complete description of this computer and its' capabilities.

2. Hardware Devices

The advent of integrated circuits (IC's) have reduced some of the former disadvantages of analog devices. Their small size reduces space requirements greatly, and facilitates easy replacement or changing of devices. The cost of the devices has been steadily declining as the technology of fabrication is improved. The accuracy of analog IC's output has not been significantly improved over discrete devices. A limitation of about ±1% of full scale accuracy may be found with these devices. Two devices presently available on the market were considered for implementing the sine and cosine functions: BURR-BROWN's sine/cosine function generator and OPTICAL ELECTRONIC's analog function module for sine and cosine.

B. IMPLEMENTATION BY ALTERNATIVE DEVICES

1. Microprogrammed Computer

a. Implementation of Single Trigonometric Functions

The HP 2100 could, for example, be programmed to compute the trigonometric functions when passed an angle by the MK 152 computer. The HP 2100, as well as the other microprogrammed computer considered, had a word size of 16 bits. Thus a loss of precision, over that achievable in the MK 152 computer, would occur. The faster instruction execution times, however, would enable the calculations of the trigonometric

functions to be performed faster than with the MK 152 computer. The HP 2100 can perform the sine and cosine calculations in a maximum of 179 microseconds whereas the function in the MK 152 computer would require 204 microseconds. During this period of computation in the microprogrammed computer, the MK 152 could continue processing instructions until receiving an interrupt notification from the microprogrammed computer that the computations were complete. Thus, the time now required for software execution in the MK 152 computer, as well as the core storage requirements for the trigonometric subroutines would be practically eliminated. The only time requirement would be that necessary for passing information, considering no dead time occurs waiting for the computational results. Communication time could be minimized by using spare function codes in the MK 152 computer as the means of activating the microprogrammed functions, and by using a microprogrammed computer, such as the HP 2100, which has a direct memory access.

b. Implementation of Multiple Functions

In addition to implementing just the trigonometric functions in the microprogrammed computer, additional time and core in the main computer could be saved by implementing larger portions of the fire control problem in the microprogrammed computer. The coordinate conversion computations, for instance, could be implemented entirely within the microprogrammed computer. The inputs to the microprogrammed computer could come entirely from the MK 152 computer, or one of the channels of the microprogrammed computer could be coupled directly to the SDC to obtain the gyrocompass outputs. This latter method would enable "continuous" computation of the sine and cosine of the gyro values. Thus

all elements in the 3x3 coordinate conversion matrix would be available whenever the main computer required a coordinate transformation. The ARCTAN and ARCSIN routines could also reside in the microprogrammed computer. With a microprogrammed computer such as the UNIVAC 1616 [17], the implementation of the ARCSIN by the trigonometric identity cited in section IV becomes feasible because of the availability of a high speed, built-in, square root routine. Thus, the major time and core consuming routines in the MK 152 computer would be eliminated. The ability of the microprogrammed computers to perform calculations independently after receiving inputs means that the MK 152 computer could continue with other calculations while the microprogrammed computer processed its information.

c. Additional Possible Applications

A microprogrammed computer could also be tasked with other operations in addition to computing the trigonometric functions. For instance, it could act as a buffer between the fire control computer and the NTDS computer (UNIVAC 642B). As many microprogrammed computers have 16 bit words, the formatting of words to be compatible with the 32 bit words of the NTDS computer would be straightforward. The microprogrammed computer would also be an excellent test vehicle for other elements in the systems.

2. Hardwired Device

Figure 11 is a four-quadrant sine/cosine generator taken from Ref. 19 by BURR-BROWN Research Corporation. Figure 12 is the sine/cosine generator configuration by OPTICAL ELECTRONICS INC. [20]. These modules could be used to convert the gyrocompass angles to trigonometric values. At the present time, the gyrocompass signals pass through the SDC, where

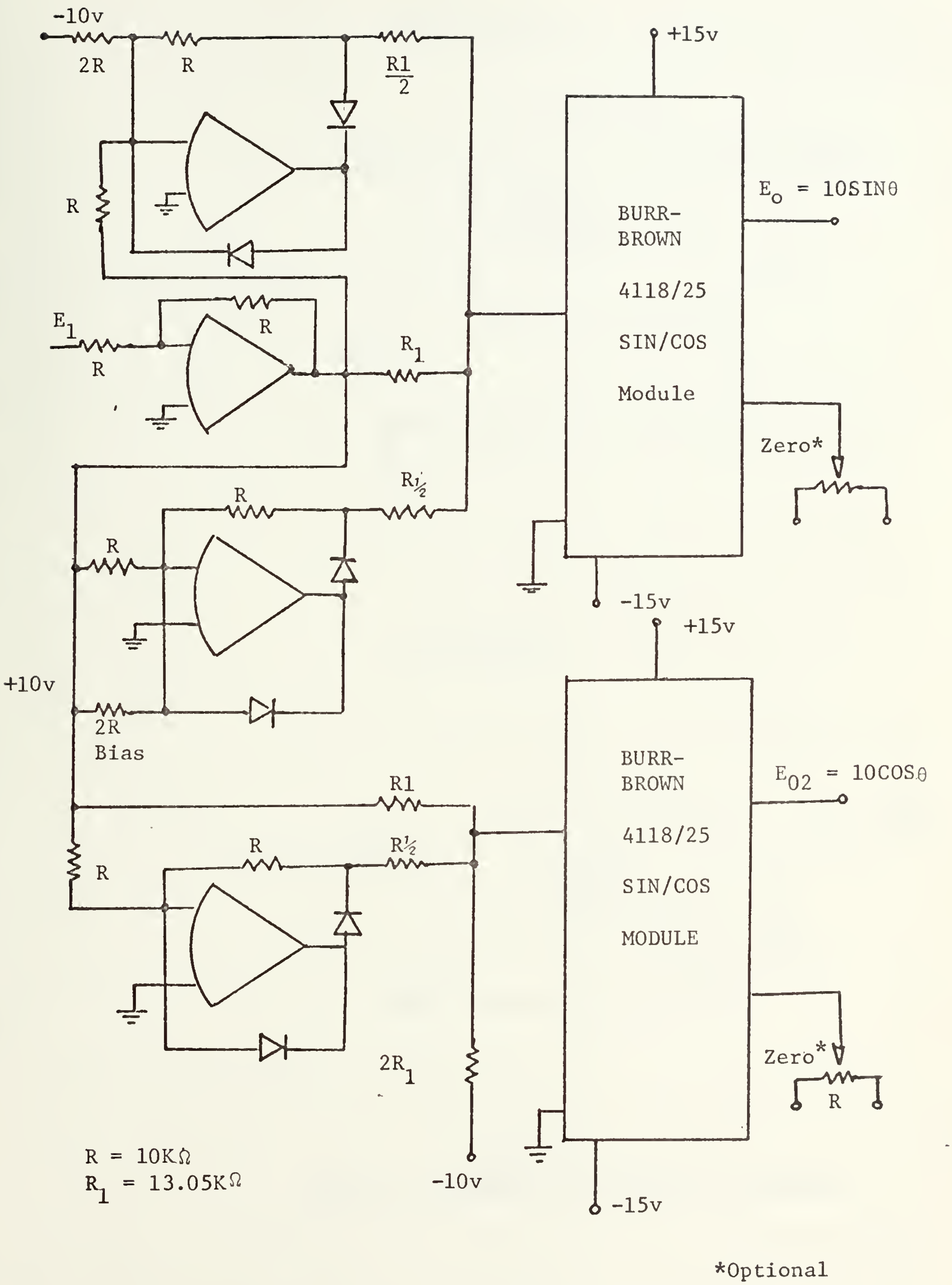

Figure 11
Burr-Brown 4 Quadrant SIN/COS Generator

Figure 12a
Cosine Generator

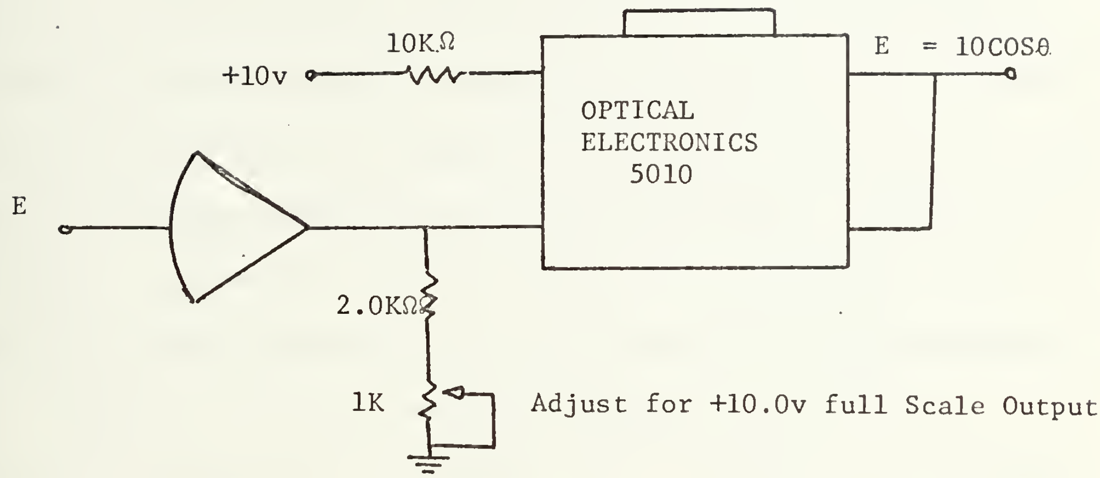

Figure 12b
Sine Generator

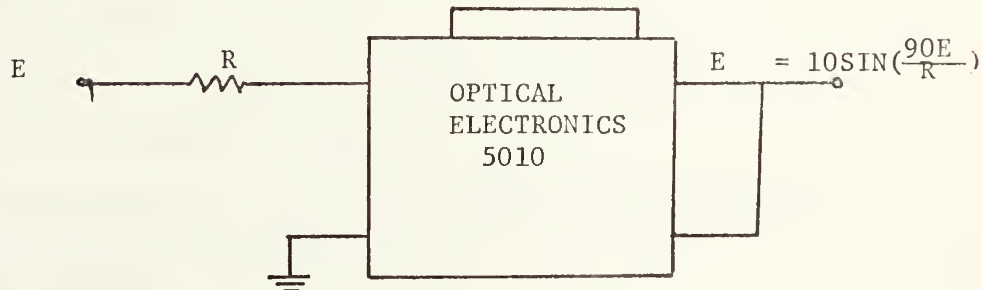

Figure 12
Optical Electronics Sine/cosine Generator

A/D conversion is accomplished. The computer algorithms within the fire control program only use the sine and cosine of these inputs; therfore, each time a new input is received a call to the SIN/COS subroutine is necessary. The gyro signals could be converted to sines and cosines by a hardwired device before A/D conversion in the SDC, thus eliminating that step within the MK 152 computer. This would provide no core savings, because the SIN/COS subroutine would still be necessary in other parts of the problem. The hardware devices could also be coupled with Integrated Circuit A/D and D/A converters to do all of the sine and cosine evaluations. The use of IC's for conversions enable a speed of conversion as fast as 100 nanoseconds to be attained [21]. This method would eliminate the SIN/COS subroutine from the MK 152, and would enable rapid computation times. The accuracy, however, would be limited to the ± 1% accuracy of the converters and the sine/cosine generators. The advantage of this method of implementation are:

a. The small size of the modules would enable installations to be made in existing equipment.

b. The cost of the modules is small - about $100 per generator unit (commercial small-quantity retail).

c. As the generators have plug-in components, changing of failed components would be fast and simple.

C. CONCLUSIONS ON ALTERNATIVE METHODS

The use of software programs in the MK 152 computer provides the most accuracy for the trigonometric functions, but is costly in terms of core storage and time expenditure requirements. The use of a microprogrammed computer would speed up the computation time and reduce the core storage

requirements, but results in some loss of accuracy and increased cost. The use of hardwired devices, in the manner discussed, would be cheaper and easier to install than a microprogrammed computer, but would provide less accuracy. The high speed achievable by hardwired devices would be offset by the limited functions that can be performed by the devices. The limitation on functions necessitates retaining most of the computing capability internally within the MK 152 computer, which means little core saving is achieved.

Table VIII is a general comparisons of the tradeoffs incurred by the use of each method considered for the trigonometric functions.

Implementation	Accuracy	Core Savings	Speed Increase
software (polynomial)	good	moderate	none
microprogrammed computer	good	high	good
hardware device	fair	none	high

Table VIII

Implementation Comparisons

The means of implementing the trigonometric functions, then, would depend whether core savings, or speed, or accuracy needed to be given the primary consideration.

VIII. SUMMARY

It was found in the TERRIER missile fire control system that the calculation of the trigonometric functions is a highly repetitive operation. Each subroutine for the trigonometric functions requires relatively long computation times and uses a large amount of storage compared to the other functions in the fire control system program.

An analysis was conducted of alternative methods of calculating the trigonometric functions to establish whether another method would provide reductions in core and computation requirements from the presently implemented method, yet maintain the systems accuracy requirement. The analysis revealed two general classes of functions for evaluating the trigonometric functions that were applicable in the fire control problem - table lookup and polynomial evaluation. It was found that polynomial evaluations would provide a reduction in storage requirements with about the same accuracy and computation times as the present method.

Alternative equipments were also considered as means to provide larger reductions in core and execution times than could be achieved by changes to the MK 152 computer software programs. The use of a microprogrammed computer in conjunction with the MK 152 computer would enable complete elimination of the trigonometric subroutines from the MK 152. It was found that this would reduce both the computation time and the core requirements for the system. Consideration was given to expanding the function of the microprogrammed computer to include other operations. These expansions resulted in additional savings in core and computing time for the MK 152 computer.

If future installations in the fire control system exceed the storage and/or time limitations of the MK 152 computer, then the addition of a microprogrammed computer to the system would provide the greatest relaxation of the restrictions.

APPENDIX A

I. CHARACTERISTICS OF THE MK 152 COMPUTER

A. CONTROL SECTION

The instruction set for the MK 152 (UNIVAC 121 B) computer has 102 single address instructions of two formats:

1. Format I

Format I is used for arithmetic operations and memory reference instructions. The word construction is:

17 - 12	11 - 0

←——18 bits——→

Bits 12 through 17 contain the function code, and bits 0 through 11 are used either as a constant or as an address. If the low order bits are used as an address, the address can be modified by any one of eight index registers. The index registers are reserved core storage locations. Only one index register can be active during an operation. The activation and deactivation of the index registers is accomplished under program control by instructions sent to a 3 bit hardware index control register (ICR). In general, Format I instructions, with or without address modification, require 4 microseconds for execution.

2. Format II

Format II is used for register-to-register transfers and for control of input and output operations. The word construction for Format II instruction is:

17 - 12	11 - 6	5 - 0

This format is distinguished from Format I by the setting of bits 12 through 17 to 50 octal. Bits 6 through 11 are used for the function code, and bits 0 through 5 are used for channel designators during input/output operations. Most of the Format II instructions are executed in 2 microseconds.

B. ARITHMETIC SECTION

The arithmetic operations are accomplished using parallel one's complement subtractive hardware with fixed point arithmetic. Either single precision (18 bits) or double precision (36 bits) operations may be performed. Five 18 bit flip-flop registers are used in the arithmetic section for data manipulation:

X - Adder input register.

D - Second adder input register.

AU - Adder output register.

AL - Second adder output register.

W - Shift register.

The X register is used in conjunction with the AU register and the W register is used in conjunction with the AL register for shift operations. The AU and AL registers are connected so that 36 bit shifts may be accomplished.Table A-I contains some typical arithmetic execution times. The times include insturction and data fetches plus indexing.

Table A-I

MK 152 Computer Arithmetic Execution Times

Operation	Execution Time
add/subtract (single precision)	4μsec
multiply/divide	14μsec
add/subtract (double precision)	6μsec
compare/masked compare and branch	6μsec
shifts (n=shift count)	2+.5nμsec

C. MEMORY

Memory is constructed of magnetic core arrays with word lengths of 18 bits. The core is divided into two sections; a control memory and main memory. The control memory is a rapid access (300 nanoseconds) section used for index registers, clock cells, input/output buffer control and interrupt registers. The memory cycle time for this 256 word section is 500 nanoseconds. Main memory is a 40960 word storage for program and data. The access time to this section is 750 nanoseconds with a total cycle time of 2 microseconds.

APPENDIX B

I. UPPER BOUND ERROR ANALYSIS OF POLYNOMIALS

A. ERROR CLASSIFICATION

In evaluating Taylor series and other polynomials by digital computer, the accuracy of the answer obtained is dependent on the magnitude of the following two types of error:

1. Approximation Errors

This type of error is partially caused by using a truncated representation of a long or infinite series. The remainder of this type of error is caused by coefficient and input term round-off or truncation due to precision limitations of a digital computer.

2. Arithmetic Errors

Values may be rounded off or truncated during arithmetic operations. The magnitude of the errors generated by this operation will be a function of the procedures used in a digital computer for conducting arithmetic operations [22]. Both approximation and arithmetic errors will be propagated through successive operations; therefore, algorithms with a larger number of operations are more susceptible to loss of accuracy. Thus, it is often desirable to obtain an estimate of the upper bound on the errors when consideration is being given to including polynomials in computer algorithms. The determination of the upper bound on the errors will show if a particular polynomial provides a reasonable approximation to a function.

B. POLYNOMIAL NESTING

One of the ways to reduce errors generated in the ovulation of polynomials is to reduce the number of arithmetic operations by nesting the polynomial according to Horner's rule [7]. The polynomial of the form

$$P\,(X) = \sum_{i=0}^{n} a_i X^i$$

is rewritten in the following form:

$$C_n = a_n$$

$$C_m = X(C_{m+1}) + a_m \qquad m = n-1, n-2, \cdots, 0$$

$$P_n(X) = C_0$$

where a_n and a_m are the polynomial coefficients.

Example: Let $n = 2$

$$P_2(X) = X(X(a_2) + a_1) + a_0$$

In this example, nesting has reduced the number of multiplies necessary in a digital computer from 4 to 2.

C. POLYNOMIAL ERROR EVALUATION

To find the approximation of the true value of the polynomial due to errors, the following errors are defined:

1. $R_{a1}, R_{a2}, \cdots, R_{an}$ = the roundoff errors in the polynomial coefficients $a_1, a_2, \cdots, a_n$.

2. R_x = the roundoff error in the variable X.

3. R_m = the roundoff error in one multiplication step.

4. R_s = the roundoff error in one addition step.

R_m and R_s are the maximum errors possible in these operations, which is the reason this is only a upper bound error determination. The approximation of the polynomial can be represented, then by the inclusion of the error terms in Horner's general form:

$$C_n = a_n + R_{an}$$

$$C_m = \left[(X+R_x)C_{m+1} + R_m\right] + \left[a_m + R_{am}\right] + R_s$$

$$P_n(X) = C_0$$

Using the example given above, the expansion becomes:

$$\overline{P}_2(X) = (X+R_x)\left[(X+R_x)\ (a_2+R_{a2}) + R_m + a_1 + R_{a1} + R_s\right] + a_0 + R_{a0} + R_s + R_m$$

The bar over P will be used to indicate that this function is an approximation. The right hand side of the equation can then be expanded. Since the product of error terms will be much smaller than the other terms, they can be omitted in the expansion.

$$\overline{P}_2(X) = a_2X^2 + 2a_2R_xX + R_{a2}X^2 + R_mX + a_1X + R_{a1}X + R_sX + a_1R_x + a_0 + R_{a0} + R_s + R_m$$

Obviously, as n gets large, the number of terms in the expansion prohibits manual manipulation of the expression.

D. SIMPLIFICATION OF POLYNOMIAL ERROR EVALUATION

The following method simplifies the above procedure, and eliminates the necessity of performing a long expansion;

1. Let n equal the maximum subscript for the coefficients, and let m = n-1.

2. Set up a table with two rows and as many columns (n) as polynomial coefficients. Label the column headings with the polynomial coefficients

in descending order from left to right. Label the first row with the variable, and m as the second row heading.

3. Enter in the first row of the tables the values for n,n-1,···,0 successively in each column from left to right. Enter the second row with the values for m,m-1, m-2,···,0 in the same manner. The completed table will appear as in Table B-I with the last row and column entry blank.

	a_n	a_{n-1}	a_{n-2}		a_1	a_0
X	n	n-1	n-2		1	0
m	m	m-1	m-2		0	

Table B-I
Polynomial Error Table

4. For each column, the sum of the coefficient and coefficient error term in that column are multiplied by the variable raised to the power of the number in the first row of that column, i.e. $(a_n+R_{an})X^n$ for the first column.

5. The number in the first row for each column is also the constant that multiplies the product of the coefficient term in that column and the variable error term. This product is multiplied by the variable raised to the power in the second row of that column, i.e., $na_nR_xX^m$ for the first column.

6. For each column in the table, the sum of the multiplication and addition roundoff error terms are multiplied by the variable raised to the power in the second row, i.e., $(R_m+R_s)X^m$ for the first column.

7. The sum of all terms above is the approximate value of the polynomial. If this value is subtracted from the true value of the Function, then the value of the gross error may be found.

E. EXAMPLE OF USE OF PROCEDURE

Again the example used previously with n = 2 will be illustrated.

$$P_2(X) = X(X(a_2) + a_1) + a_0$$

The table is first set up:

	a_2	a_1	a_0
X	2	1	0
m	1	0	

Then the terms in the approximation are found using the procedures above:

$$\overline{P}_2(X) = (a_2+R_{a2})X^2 + (a_1+R_{a1} - 2a_2R_x + R_m + R_s)X + a_0 +$$

$$R_{a0} + a_1R_x + R_m + R_x$$

This example illustrates that with a small amount of practice the evaluation can be done rapidly and with a minimization of the possibility of arithmetic errors. The resultant form is readily adaptable for computer algorithm.

APPENDIX C

I. HP 2100 COMPUTER CHARACTERISTICS

A. CONTROL SECTION

The instruction set of the HP 2100 computer has 80 single address instructions, which are implemented by microprogram, in four formats:

1. Memory Reference

This format is used for arithmetic operations and other memory references. The word construction is:

15	14 - 11	10	9 - 0

←16 bits→

Bits 0 through 9 contain a memory address. To permit more compact addressing, memory is divided into pages of 1024 words. Bit 10 is used to indicate if the address is in the current page or in page zero. Bits 11 through 14 contain the function code. Bit 15 is set for indirect addressing. Execution time for memory reference instructions is 1.96 microseconds.

2. Register Reference

This format is used for rotations and shifts of registers, comparison operations and complementation. The word construction of this format is:

15 - 12	11	10	9 - 0

Bits 0 through 9 contain the instruction to be executed. Groups of up to 8 register instructions can be combined in these lower bits for

simultaneous execution. Bit 10 is used to indicate if the instruction in the lower bits comes from a group of shift-rotate instructions or from a group of alter-skip instructions. Bit 11 indicates which of the two accumulators is being referenced. Bits 12 through 15 indicate that this class of instructions are register reference instructions. Execution of this class of instructions is 1.96 microseconds maximum.

3. Input/output

This format of instruction is used to control input/output devices, transfer data to and from peripherals, and for control of the interrupt system. The word construction of this format is:

15 - 12	11	10 - 6	5 - 0

Bits 0 through 5 reference one of the fourteeen input/output addresses. Bits 6 through 10 contain the input/output instruction. Data can be directly input and output from one of the two accumulators, so bit 11 is used to select one or the other. Bits 12 through 15 denote this is an input/output instruction. Execution time for an instruction of this format is 1.96 microseconds.

4. Extended Arithmetic

These instructions implement all operations which require a double length accumulator, such as a multiply. Two formats are used in extended arithmetic operations:

a. Extended Memory Reference

The extended arithmetic memory reference instructions utilize two words for instruction execution:

15 - 12	11	10 - 4	3 - 0

15	14 - 0

Bits 0 through 3 of the first word are not used. Bits 4 through 10 contain the instruction to be executed. Bit 11 is used to indicate that this is an extended memory reference. The upper 4 bits denote this is an extended arithmetic instruction. In the second word, bits 0 through 14 are used for the memory address and bit 15 is an indirect address bit.

b. Extended Register Reference

The extended arithmetic register reference instructions use one word for shift operations:

15 - 11	10	9 - 4	3 - 0

Bits 0 through 3 indicate the number of shifts to be made and bits 4 through 9 contain the shift/rotate instructions. Bit 10 indicates this is a register reference instruction, and the upper bits labels it an extended arithmetic instruction.

Execution times for this class of instruction vary from 2.9 to 16.7 microseconds, depending on the operation performed.

During program execution, the reference to one of the four classes of instructions above causes execution of a sequence of micro-instructions contained in the ROM control memory. A separate format is used for these instructions:

23 - 21	20 - 18	17 - 13	12 - 9	8 - 5	4 - 0

←—— 24 bits ——→

Bits 0 through 4 control skip functions. Bits 5 through 8 are a special field for execution of functions not covered by other fields. Bits 9 through 12 are mostly used for activating stores into registers from the T bus (the bus structure is explained in the next section). Bits 13 through 17 are a function field that controls operations of the arithmetic logic unit, flag, overflow, shift, and jump logic functions. Bits 18 through 20 causes reads of selected registers to the S bus. Bits 21 through 23 causes reads from selected registers to the R bus.

B. ARITHMETIC SECTION

The HP 2100 uses three buses (R, S, and T) for the transfer of information to and from the arithmetic section. The R and T buses are used within the arithmetic section to provide transfer paths between registers. The S bus provides the main communication path between the four primary sections of the computer (Control, Arithmetic, Memory, and Input/Output). The arithmetic section contains nine 16 bit registers for information processing. Two of these registers are accessible under software program control for use as accumulators. One register contains a program counter, which controls the program flow. The remainder of the registers are manipulated by the microinstruction firmware during program execution. Table C-I indicates the execution times of some of the arithmetic instructions.

Table C-I

HP 2100 Execution Times

Operation	Execution times (μ sec)
add/subtract	1.96
multiply	10.7
divide	16.7
compare	1.96
shift/rotate	1.96 - 7.8 depending on type and length

C. MEMORY

As indicated in Section A, main memory is divided into pages of 1024 words page. Main memory uses a folded planar core which has a 980 nanosecond cycle time. Each word has 17 bits, of which 16 bits are used for data and 1 bit is a parity check bit. Main memory may be expanded from 4K to 32K words by the addition of either 4K or 8K modules. The control memory is comprised of 1024 words of 24 bit semiconductor memory, which has a cycle time of 196 nanoseconds.

BIBLIOGRAPHY

1. Surface Missile Systems Project Officer Report SMS-FS-421, TERRIER Fire Control Digital Computer Program Design Specification (Advanced Development Model), XWS-13058, prepared by The John Hopkins University Applied Physics Laboratory, October, 1970.

2. Surface Missile Systems Project Officer CONFIDENTIAL Report SMS-FS-305, TERRIER Adaptive Fire Control System (AFCS) Fire Control Computer Program Performance Specification (Advanced Development Model) (U), Revision 3, prepared by The John Hopkins University Applied Physics Laboratory, June, 1971.

3. Naval Ordnance Systems Command OP 3514 (PMS/SMS), Part 1, Digital Computer MK 152 Series, Description, Operation and Maintenance, 1 January, 1972.

4. Hart, J. F. and others, Computer Approximations, Wiley, 1968.

5. Hastings, Cecil, Approximations for Digital Computers, Princeton University Press, 1955.

6. Carnahan, Brice; Luther, H.A.; and Wilkes, J. O., Applied Numerical Methods, Wiley, 1969.

7. McCracken, D. D. and Dorn, W. S., Numerical Methods and Fortran Programming, Wiley, 1969.

8. Ralston, Anthony and Wilf, H. S., Mathematical Methods for Digital Computers, Wiley, 1960.

9. Booth, A. D. and Booth, K. H. V., Automatic Digital Calculators, Butterworth LTD, 3rd edition, 1965.

10. UNIVAC Division, Sperry Rand Corporation, UNIVAC 1219B Programmers Reference Manual, UNIVAC PX 4939, date unknown.

11. Moursund, D. G., "Optimal Starting Values for Newton-Raphson Calculation of $\sqrt{x}$", Communications of the ACM, V. 10, p 430-432, July, 1967.

12. Naval Weapons Center Report TP 4577, Digital Fire Control System Software Design, prepared by Lockheed Electronics Company, June, 1968.

13. Hewlett-Packard Corporation, 2100 Computer Programming Guide, 1971.

14. Hewlett-Packard Corporation, HP 2100 Processor Description, July, 1972.

15. INTEL Corporation, MCS 4 Micro Computer Set Users Manual, Revision 3, 1972.

16. MICRODATA Corporation, MICRODATA Handbook, 2nd edition, 1971.

17. Sperry Rand Corporation, UNIVAC Division, UNIVAC 1616 General Description, undated.

18. American Microsystems Incorporated, 7200 Processor, undated.

19. Burr-Brown Research Corporation, Sine Cosine Function Generator, PDS-224B, June, 1972.

20. Optical Electronics Incorporated, Model 5010 Operating and Applications Manual, undated.

21. DATEL Systems Incorporated, Short Form Catalog, Summer 1972.

22. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, 1963.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst Professor V. M. Powers, Code 52 Pw Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor J. R. Ward, Code 52 Wa Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	CDR L. J. Holloway, USN Code 4400 Naval Ship Missile Systems Engineering Station Port Hueneme, California 93043	2
6.	LCDR James R. Williamson, USN 2090 Paralta Avenue Seaside, California 93955	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ATING ACTIVITY (Corporate author) al Postgraduate School terey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

RT TITLE

lysis of Computing Methods for the TERRIER Fire Control System

RIPTIVE NOTES (Type of report and, inclusive dates)

ter's Thesis; December 1972

OR(S) (First name, middle initial, last name)

es Ray Williamson

RT DATE cember 1972	7a. TOTAL NO. OF PAGES 78	7b. NO. OF REFS 22
TRACT OR GRANT NO. JECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

TRIBUTION STATEMENT

Approved for public release; distribution unlimited

PPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

STRACT

Future modifications in the TERRIER missile fire control system will be estricted by limitations in unallocated core and by problem solution time in the ystem's digital computer.

This thesis is an analysis of methods by which reductions in core storage equirements and in problem solution time could be achieved. A determination of hose functions requiring the most computer resources is made and alternative ethods of computing the functions are analyzed. Comparisons of implemention of he functions by software in the fire control computer versus other devices is ade, and the tradeoffs required by each method are presented.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
K 152 Digital Computer						
NIVAC 1219B Digital Computer						
ERRIER Missile System						
rigonometric Function Evaluation						
ross Error Analysis						
ine/Cos Evaluation						
igital Computing Methods						
olynomial Errors						

141456
Thesis
W6245 Williamson
c.1
Analysis of computing methods for the TERRIER fire control system.