AN ABSTRACT OF THE THESIS OF

Raynard Edward McCormick, Jr. (Name) for the MASTER OF SCIENCE (Degree)

in Electrical and Electronics Engineering (Major) presented on December 17, 1971 (Date)

Title: A COMPARISON OF DIGITAL DIFFERENTIAL ANALYZER INTEGRATION TECHNIQUES

Redacted for Privacy

Abstract approved: L. J. Weber

When selecting an approximate integration technique for use in a digital differential analyzer, such factors as cost, parts count, power dissipation and solution accuracy must be considered. This paper considers the application of the classical rectangular and trapezoidal integration techniques and a new two cycle method of integration, all implemented using commercially available components, in the sine/cosine generator and compares the integration techniques in terms of cost, power dissipation, and accuracy. The two cycle integration technique is shown to have a constructional simplicity near that of rectangular integration and the accuracy of trapezoidal integration.

A Comparison of Digital Differential Analyzer Integration Techniques

by

Raynard Edward McCormick, Jr.

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1972

APPROVED:

Redacted for Privacy

Professor of Electrical and Electronics Engineering

in charge of major

Redacted for Privacy

Head of Department of Electrical and Electronics Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented December 17, 1971

Typed by Clover Redfern for Raynard Edward McCormick, Jr.

TABLE OF CONTENTS

Chapter		Page
I.	INTRODUCTION	1
II.	THE DDA INTEGRATOR	3
III.	FLOW CHARTS AND SCALING	9
	Flow Charts	9
	Flow Charting the Sine/Cosine Generator	10
	The Numbering System	12
	Scaling	13
IV.	RECTANGULAR INTEGRATION	16
	Simple Integration	16
	Bias Error	18
	Implementation of the Sine/Cosine Generator	19
	Parameters of Comparison	21
	Cost and Power Dissipation	22
	Sine/Cosine Generator Solution Form and Accuracy	22
V.	TRAPEZOIDAL INTEGRATION	34
	The Algorithms of Trapezoidal Integration	34
	Implementation of the Sine/Cosine Generator	38
	Cost and Power Dissipation	39
	Sine/Cosine Generator Solution Form and Accuracy	39
	Modified Trapezoidal Integration	48
VI.	TWO CYCLE INTEGRATION	51
	The Technique and Its Algorithms	51
	Implementation of Sine/Cosine Generator	53
	Cost and Power Dissipation	54
	Sine/Cosine Generator Solution Form and Accuracy	55
VII.	SUMMARY AND CONCLUSIONS	64
	BIBLIOGRAPHY	78
	APPENDIX: System Control Logic	80

LIST OF FIGURES

Figure		Page
2.1.	Integration with respect to time.	4
2.2.	The generalized integrator.	4
2.3.	Digital integrator with incremental output and scale bit.	6
2.4.	The incremental integrator.	8
3.1.	The integrator.	9
3.2.	The integrator with sign change.	9
3.3.	DDA integrator with sine and cosine nomenclature.	11
3.4.	The sine/cosine generator.	11
3.5.	Formation of a negative number.	12
3.6.	Integrator with scaling constants.	13
3.7.	Five bit sine/cosine generator scaled by 1/2.	14
3.8.	An example of fractional scaling.	15
4.1.	Rectangular integration.	16
4.2.	Scaling with unbiased integrators.	19
4.3.	Implementation of the sine/cosine generator by rectangular integration.	20
4.4.	Definition of symbols.	22
4.5.	Comparison of cos θ and C_n, the DDA approximation of cos θ using rectangular integration.	27
4.6.	Comparison of sin θ and S_n, the DDA approximation of sin θ using rectangular integration.	28
4.7.	Comparison of the unit circle and the approximation obtained using rectangular integration.	29

Figure		Page
4.8.	Difference between cos θ and its rectangular approximation.	31
4.9.	Difference between sin θ and its rectangular approximation.	32
5.1.	Trapezoidal integration.	35
5.2.	Sine/cosine generator using trapezoidal integration.	37
5.3.	Comparison of cos θ and C_n, the DDA approximation of cos θ using trapezoidal integration.	42
5.4.	Comparison of sin θ and S_n, the DDA approximation of sin θ using trapezoidal integration.	43
5.5.	Comparison of the unit circle and the approximation obtained using trapezoidal integration.	44
5.6.	Difference between cos θ and its trapezoidal approximation.	46
5.7.	Difference between sin θ and its trapezoidal approximation.	47
5.8.	Sine/cosine generator using modified trapezoidal integration.	49
6.1.	Two cycle integration.	52
6.2.	Implementation using two cycle integration.	53
6.3.	Additional control logic for two cycle integration.	54
6.4.	Comparison of cos θ and C_n, the DDA approximation using two cycle integration.	58
6.5.	Comparison of sin θ and S_n, the DDA approximation of sin θ using two cycle integration.	59
6.6.	Comparison of the unit circle and the approximation obtained using two cycle integration.	60

Figure | Page

6.7. Difference between cos θ and its two cycle approximation. 61

6.8. Difference between sin θ and its two cycle approximation. 62

Appendix

A.1. Two phase clock restrictions. 81

A.2. System control logic diagram. 82

LIST OF TABLES

Table		Page
1.	Component price and power dissipation list.	65
2.	Sine/cosine generator using rectangular integration response to 1/16 radian reversals.	66
3.	Sine/cosine generator using rectangular integration response to 1/4 radian reversals.	67
4.	Sine/cosine generator (trapezoidal integration) response to 1/16 radian reversals.	69
5.	Sine/cosine generator (trapezoidal integration) response to 1/4 radian reversals.	70
6.	Sine/cosine generator (modified trapezoidal integration) response to 1/16 radian reversals.	72
7.	Sine/cosine generator (modified trapezoidal integration) response to 1/4 radian reversal.	73
8.	Sine/cosine generator (two cycle integration) response to 1/16 radian reversals.	75
9.	Sine/cosine generator (two cycle integration) response to 1/8 radian reversals.	76

A COMPARISON OF DIGITAL DIFFERENTIAL ANALYZER INTEGRATION TECHNIQUES

I. INTRODUCTION

The exact definition and origin of the term differential analyzer is difficult to assess, since the term has been applied to a variety of mechanical, electro-mechanical and electronic machines. The term is used to define a class of machines each used in the solution of a particular set of differential equations within a range of coefficients, as opposed to the simulator which includes all machines attempting to provide a working model of a physical system (3). A digital differential analyzer is an array of digital integrators used in the solution of differential equations. In the literature this term is sometimes misused and applied to the digital integrator itself.

Conceptually and practically the electronic digital differential analyzer (DDA) has been available for sometime. However, only with the advent of high density MOS (metal oxide semiconductor) technology have the use of the DDA systems become feasible in many space limited applications (aircraft and missiles). The DDA element (adder) considered in this thesis is commercially available and when used in conjunction with a pair of appropriate length shift registers, forms a digital integrator. If a dual 16 bit shift register is considered, the digital integrator, formed using two MOS flatpacks, would require

approximately 450 discrete devices. Since even a special purpose system might easily require ten to twenty integrators, the need for high density packaging is apparent.

This thesis seeks to investigate the application of a commercially available DDA in a simple system, the sine/cosine generator and to propose an alternative integration technique, which is simple to implement and greatly improves system accuracy. The investigation will start with a brief review of DDA operation, mapping and scaling. The effects of various integration techniques on system operation will then be studied. While investigation of the sine/cosine generator may seem trivial, the equivalent of many such generators may often be found in even small systems, particularly in systems designed for aircraft and missile control.

In the study of the sine/cosine generator only the parallel (simultaneous) type machine will be considered. A parallel machine is one consisting of several integrators all of which update simultaneously as opposed to a sequential machine which processes all its data sequentially through a central integrator. The parallel machine offers the advantages of higher speed and greater flexibility.

II. THE DDA INTEGRATOR

The DDA integrator is an incremental device with its principle of operation based on the hypothesis that physical systems change only by a small amount in a small interval of time. If the interval ΔT is always small, then

$$\sum_i Y_i \Delta T \approx \int Y dT$$

The time interval of importance in the parallel DDA is the machine iteration period or the time to completely process one word of information. When using a 16 bit shift register and a shifting frequency of 400 KHz, the iteration time would be 16 x 2.5μsec=40μsec.

Consider the typical function of time shown in Figure 2.1. Allow each horizontal step to represent one machine iteration period and allow each vertical step to be some unit in the magnitude of the function. During any one iteration period the change in the integral is approximately the rectangular area $Y_i \Delta T$ and if ΔT is sufficiently small,

$$f(T_N) = \int_0^{T_N} f(T) dT \approx \sum_{i=0}^{N} f(T_i) \Delta T$$

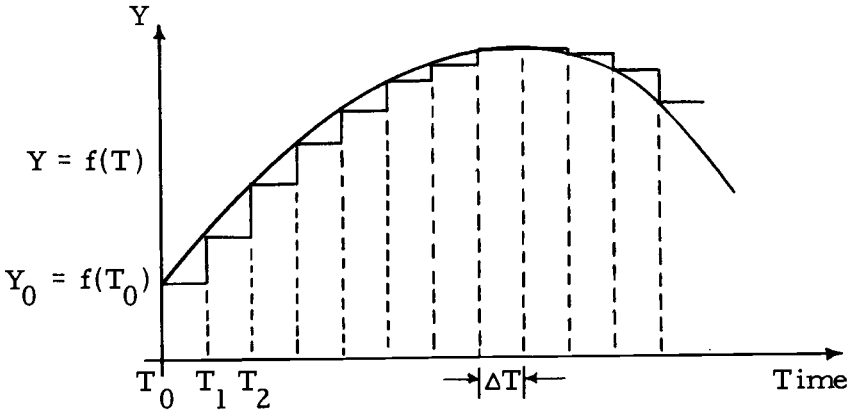
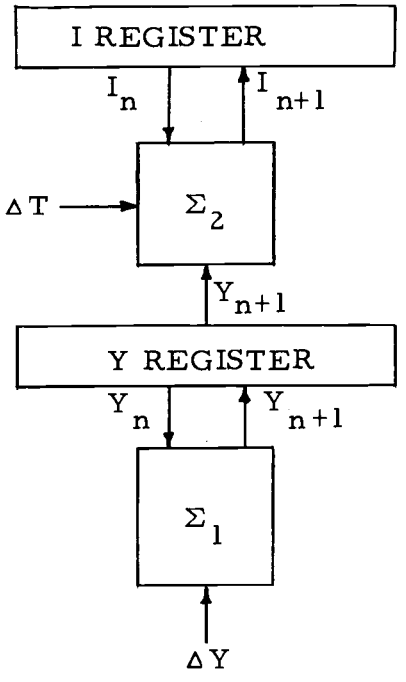


Figure 2.1. Integration with respect to time.

Integration using the above approximation is known as rectangular or Euler integration. Mechanization is not difficult and may be visualized as shown in Figure 2.2. All commercially available DDA's known to the author operate on this principle.



Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$I_{n+1} = I_n + Y_{n+1}\Delta T$$

Figure 2.2. The generalized integrator.

The algorithms of Figure 2.2 define the update sequence. It will be noted that Y is updated first and then multiplied by ΔT and added into the integral. The reasoning behind this update sequence will become apparent later.

Hidden in the above discussion is a clue to DDA integrator operation. It has been required that ΔT be one iteration period. Since ΔT is always one, multiplication is reduced to simple addition. To extend the application of the DDA integrator to a variable (x) other than time, it is necessary only that the rate of change $\Delta x/\Delta T$ be restricted such that Δx never changes by more than one quantum during any iteration period. Conceptually the system need not even change. The second adder (Σ_2) is simply required to handle multi-signed (+, -) inputs.

All commercially available units utilize a two wire (+, -) ternary system. The adder (Σ_2) is required to process positive, negative and zero inputs.

In order to use digital integrators in the solution of differential equations, it is essential that they be interconnected, that is, some incremental output must exist for use as an input to subsequent integrators. To accomplish this purpose the integral (I) register of Figure 2.2 is rather arbitrarily divided into most significant and least significant halves. The least significant half is defined as the R register and is required for ease in implementation (timing) to be

the same length as the Y register. Its overflows and underflows form the incremental outputs, Δz (ternary), which are sign summed into the remaining half of the I register to yield the integral value.

Since in a parallel system all integrators update simultaneously, the Δz outputs must be stored until the next iteration to be of use to subsequent integrators. To facilitate its use as either Δx or ΔY variables in the following integrators, the logic state of the Δz output is stored and held throughout the next iteration period.

Unlike Δx which provides multiplication ($Y\Delta x$) gating, the ΔY input is intended to add data into the least significant bit of the Y register. This is accomplished by gating the input Δ's with a bit time or word weight marker termed the scale bit. (See Figure 2.3.)

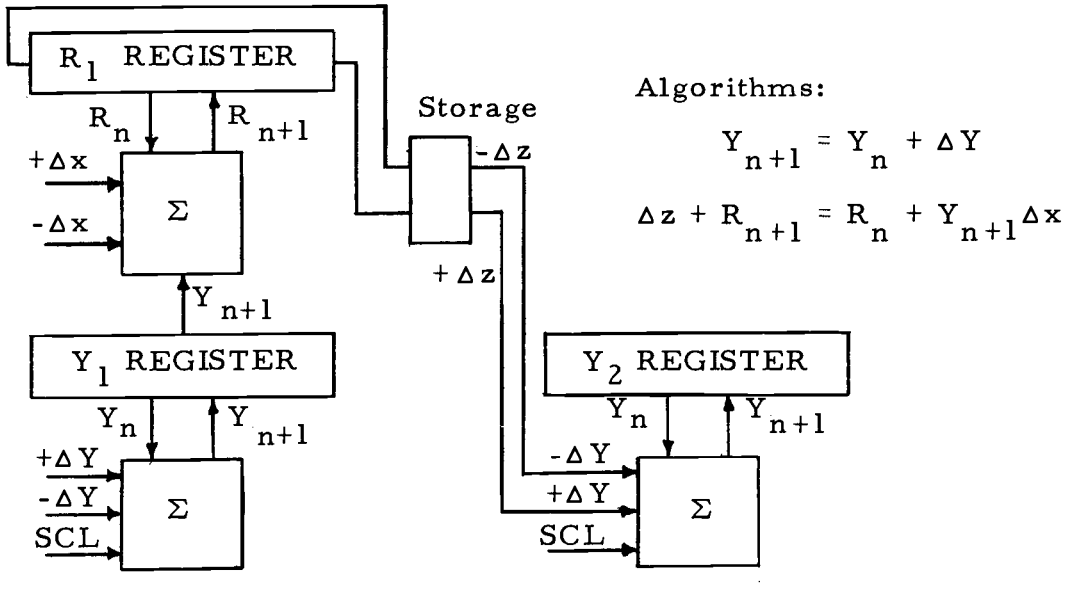


Figure 2.3. Digital integrator with incremental output and scale bit.

The digital integrator thus far constructed consists of Y and R registers, Δx and ΔY inputs and Δz outputs. The most significant half of the integral is formed by summing the Δz outputs. Since that sum is updated during the iteration period following that in which overflow occurred, it is clear why algorithms of the type shown in Figures 2.2 and 2.3 are used.

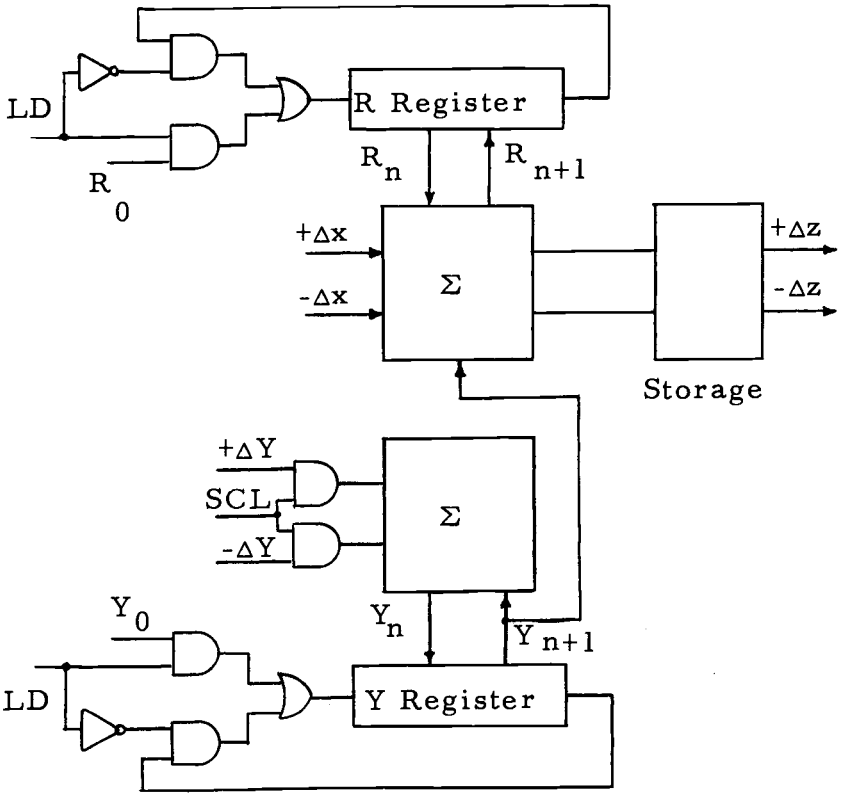
Since the true value of the integral has been truncated by its least significant half, all calculations making use of the integral value suffer from inaccuracy. Some work has been done on improving Y register resolution. One approach provides multi-outputs (Δz) with word weights 1, 1/2, and 1/4 (13). Another accomplishes the same goal by adding partial words to an extended Y register (7). Both units have exhibited the effect of improved resolution, reduced error. Units such as these are not currently available commercially.

While the value of the Δx input is restricted to one during any iteration period, no such limitation is placed upon the ΔY input. A ΔY input originating from any single source will have a value of one quantum during any iteration period, since the supplying R register could only overflow (underflow) once during that period. The ΔY input, however, is not limited to inputs from one source and therefore, may take on value greater than one quantum.

Integration, whether real or approximate, must start from some point. To complete the digital integrator it is, therefore,

necessary to provide gating for presetting initial conditions in the Y and R registers.

The completed DDA integrator is shown in Figure 2.4.



Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$\Delta z + R_{n+1} = R_n + Y\Delta x$$

Figure 2.4. The incremental integrator.

III. FLOW CHARTS AND SCALING

Flow Charts

In flow charts the DDA integrator is represented diagrammatically. No consistant standard for representation of the DDA integrator is found in the literature. The basic shape of the symbol remains roughly the same but with various additions and omissions. This paper will represent the integrator as shown in Figure 3.1. This representation allows one to model the general DDA integrator consisting of one Δx input, several ΔY inputs and with the Δz outputs capable of multiple connection. The small vertical slash on the ΔY_i inputs is used during scaling for noting the word weight of the scale bit. A small ring on the output of an integrator will be used to signify a sign change on the output (Figure 3.2).

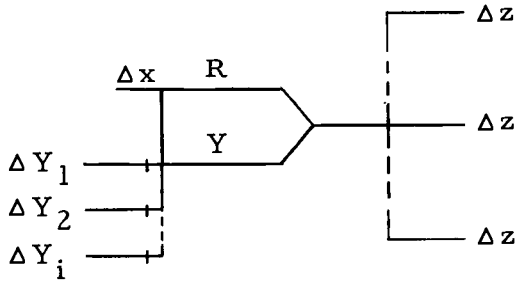


Figure 3.1. The integrator.

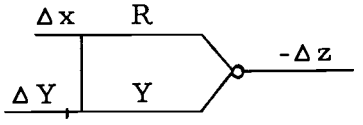


Figure 3.2. The integrator with sign change.

In general some numbering scheme is used to identify the individual integrators. However, the problem under discussion in this paper is not sufficiently complex to warrant numerical integrator designation. The integrators will be labeled according to the contents of their Y registers.

Flow Charting the Sine/Cosine Generator

The differential equations required in the formation of the sine/cosine generator are as noted below:

$$d \sin \theta = \cos \theta \, d\theta \tag{3.1}$$

$$d \cos \theta = -\sin \theta \, d\theta \tag{3.2}$$

These equations are approximated by the difference equations

$$\Delta \sin \theta = \cos \theta \, \Delta \theta \tag{3.3}$$

$$\Delta \cos \theta = -\sin \theta \, \Delta \theta \tag{3.4}$$

The difference equations express mathematically the incremental changes in $\sin \theta$ and $\cos \theta$ respectively. If now the Y registers of two DDA integrators are allowed to be $\sin \theta$ and $\cos \theta$ and the Δx inputs are allowed to be $\Delta \theta$ in units of radian measure, then the Δz outputs would represent increments of $\sin \theta \, \Delta \theta$ and $\cos \theta \, \Delta \theta$ respectively (Figure 3.3). According to Equations 3.4 and 3.3, these outputs represent incremental changes of $\cos \theta$

and $-\sin\theta$. Using the negative Δz output of the sine integrator and interconnecting the outputs with the appropriate ΔY inputs results in the sine/cosine generator of Figure 3.4.

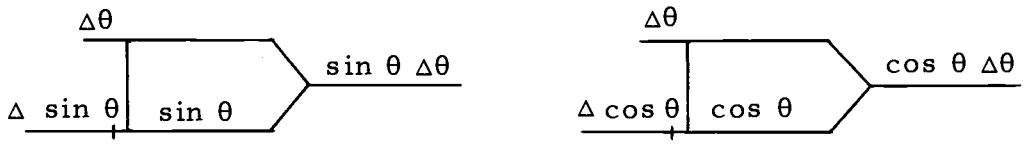


Figure 3.3. DDA integrator with sine and cosine nomenclature.

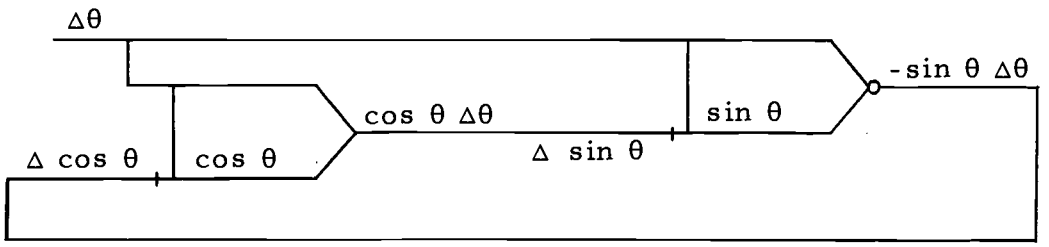


Figure 3.4. The sine/cosine generator.

The $\Delta\theta$ input may be an incremental output from another source or if a generator in the true sense of the word is desired, it may be simply plus or minus logic one allowing the generator to oscillate as a function of scaling and its iteration rate. If the machine has an iteration rate of 40 μ sec, and if each $\Delta\theta$ input represented 1/32 radian, the output period would be about 8.04 msec or the frequency of oscillation would be approximately 120 Hz.

The Numbering System

The numbering system most commonly used in DDA integrators is the two's complement system. The closed counting sequence below where $N = 2^i$ and i is a positive integer defines all the numbers used in the two's complement system.

$$-1, -1 + \frac{1}{N}, -1 + \frac{2}{N}, \ldots, -1 + \frac{N-1}{N}, 0, \frac{1}{N}, \ldots, \frac{N-1}{N}$$

It can be seen that the most negative number is -1 and the most positive $1/N$ less than 1.

The system utilizes a sign bit of word weight 2^0 and a sequence of binary digits weighted 2^{-1} through $2^{-i} = 1/N$. A positive number is simply the binary representation of that number. A negative number is formed by taking the binary representation of the positive number, complementing it and adding $1/N$ (Figure 3.5). The two's complement arithmetic system allows addition and subtraction to be performed without the necessity of word modification.

	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	
14/16	0.	1	1	1	0	
Compliment	1.	0	0	0	1	
Add 1/N	0.	0	0	0	1	
	1.	0	0	1	0	$= -\frac{14}{16}$

Figure 3.5. Formation of a negative number.

Scaling

If the numbering system is to restrict the value of Y register representations in the range $-1 \leq Y \leq 1-\frac{1}{N}$, it will generally be necessary to scale the variables in order to obtain meaningful results. Assume that the problem variables dx, dy and y are represented in the machine by Δx, ΔY, and Y. Let the two sets of variables be related by the equations

$$dx = 2^{a}\Delta x$$

$$dy = 2^{b}\Delta Y$$

$$y = 2^{c}Y$$

The difference between the scaling constants on Y and ΔY defines the word weight of the scale bit, which must be 2^{b-c}. The output Δz scaling is defined by the product $Y\Delta x$ (Figure 3.6).

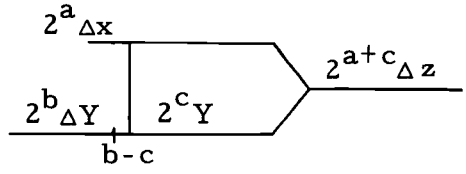
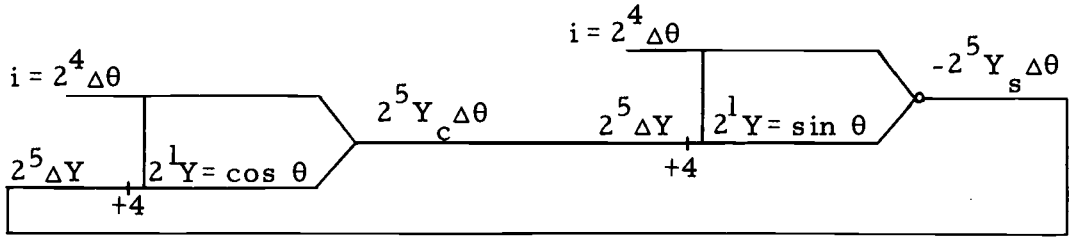


Figure 3.6. Integrator with scaling constants.

As a rule the literature notes that the value of the Y register should be less than one and proceeds through example to show how the problem is solved when some scaling of the form 2^{-i}, where i is

integer, is used. It is assumed that this is done because of the binary nature of the system. An example of scaling by one-half (2^{-1}) is shown for the sine/cosine generator in Figure 3.7 utilizing five shift register positions.



Scaling by 1/2 requires $c = 1$
Five bit system requires $b - c = 4$
$\therefore \quad b = 5$
Since the output of one integrator is the input of the next,

$$b = a + c, \quad a = 4$$

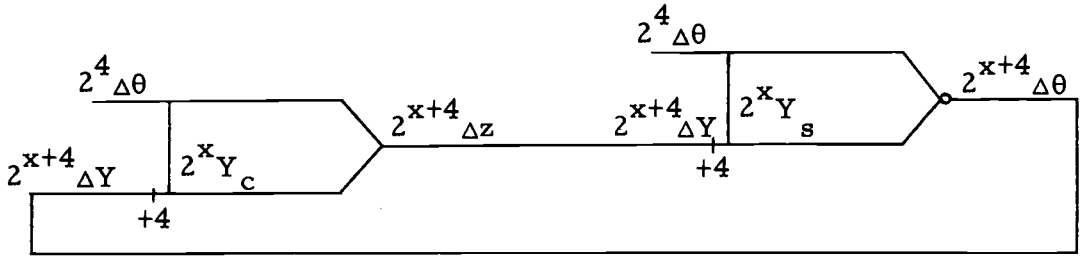
Figure 3.7. Five bit sine/cosine generator scaled by 1/2.

It should be noted that for a given Y register length, scaling by one-half is eliminating 50% of the Y register resolution and thereby, reducing solution accuracy. It is generally desirable to allow the maximum value of the Y register variable to come as near to filling the register as possible. When attempting this, care must be taken to account for integrator inaccuracies.

In order for the maximum value of the variable in the Y register to more nearly approximate one, it is necessary that c take on a value between zero and one. Because of this, the author uses the term fractional scaling. Compare Figure 3.8 below which

uses fractional scaling with Figure 3.7. Observe that in this case the Y register resolution is 1/14 as opposed to 1/8 in the previous case.

$$2^x = \frac{14}{16}$$



$$b - x = 4$$
$$a + x = b = 4 + a$$
$$a = 4$$

Figure 3.8. An example of fractional scaling.

IV. RECTANGULAR INTEGRATION

Simple Integration

It is educational to consider rectangular integration in obtaining the area under the curve (Figure 4.1) defined by $y = mt + b$ where $m = \Delta Y/\Delta T$. Since the DDA integrator operates according to the algorithms

$$Y_{n+1} = Y_n + \Delta Y$$

$$M\Delta z_{n+1} + R_{n+1} = R_n + Y_{n+1}\Delta T$$

where M is the weighting constant for Δz equal to the capacity of the R register, summing the Δz overflows results in the equation

$$\sum_{i=1}^{N} M\Delta z_i = R_0 - R_N + \sum_{i=1}^{N} Y_i(\Delta T = 1) \qquad (4.1)$$

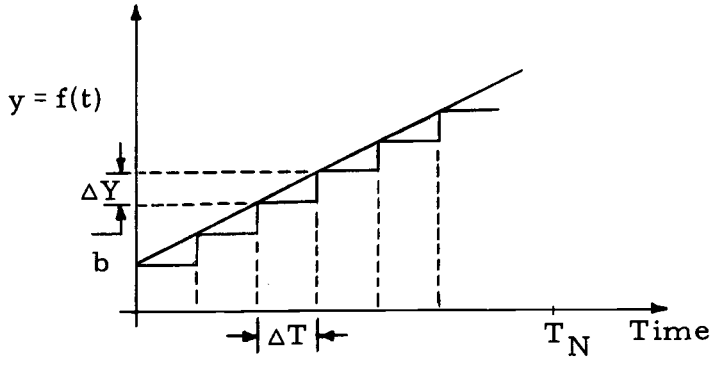


Figure 4.1. Rectangular integration.

In the second chapter it was noted that the algorithms were selected to account for a one iteration time delay in providing the ΔY input to the accumulating Y register. If this is so, then according to the algorithm for Y updating, $Y_1 = Y_0$ since no ΔY could be available initially. Therefore, Equation 4.1 may be rewritten as

$$\sum_{i=1}^{N} M\Delta z_i = R_0 - R_N + \sum_{i=0}^{N-1} Y_i \tag{4.2}$$

If it is further defined that

$$\Delta z_i = Z_i - Z_{i-1},$$

then

$$M(Z_N - Z_0) = R_0 - R_N + \sum_{i=0}^{N-1} Y_i \tag{4.3}$$

Due to the nature of the curve selected, the true integral may be exactly represented by the trapezoidal area.

$$\therefore \quad \int_0^{T_N} ydt = \frac{1}{2}\Delta T(Y_N - Y_0) + \Delta T \sum_{i=0}^{N-1} Y_i \tag{4.4}$$

Again allowing for $\Delta T = 1$, substituting for $\sum_{i=0}^{N-1} Y_i$ in Equation 4.3 and rearranging

$$(Z_N - Z_0) = \frac{1}{M}\int_0^{T_N} ydt - \frac{(Y_N - Y_0)}{2M} + \frac{R_0 - R_N}{M} \tag{4.5}$$

Equation 4.5 shows that the rectangular integral (Z_N-Z_0) differs from the exact integral.

The term $(Y_N-Y_0)/2M$ is the truncation error caused by sampling the curve at discrete instances in time. In general the term would be more complex. However, the most significant contribution will be due to neglect of the second order ΔT^2 term in the Taylor expansion of the integral.

The term $(R_0-R_N)/M$ is a rounding error. It will be remembered (Chapter II) that the integral register was split into most significant and least significant halves. This term represents that portion of the integral in the least significant half. Since the value in the R register can only range from zero to M, this error should not be greater than one bit in the Y register.

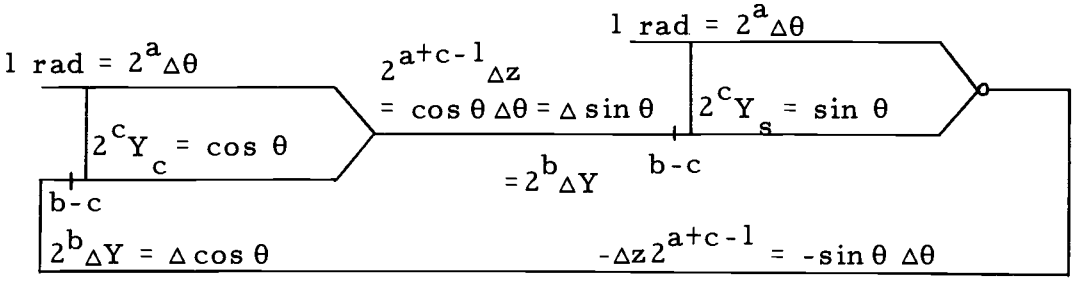
Bias Error

Consider yet another source of error. Consider an R register about to overflow and update the following Y register. At this point in time, based on the split integral concept, the Y register, which may be used in subsequent calculations, is in error by approximately one bit. Furthermore, except when the R register is zero, the round off error is always in the same direction.

To prevent accumulation of errors of this type some manufacturers have provided unbiased error outputs. The error is statistically unbiased and may be anywhere in the range +1 to -1

quantum during any given iteration period. Unbiased error systems are achieved by allowing the data in the R register to propagate into the sign bit and rescaling by 1/2 to provide for the associated reduction in overflows.

The sine/cosine generator shown below (Figure 4.2) has been rescaled to accommodate the use of unbiased outputs. Hereafter in this paper, the basic system scaling will remain the same and only unbiased operation will be considered.



Let $2^x = 14/16 = 2^c$ $\quad c = x$

For a 5 bit system $\quad b - c = 4 \quad b = 4 + c$

$b = a + c - 1$

$a = 5$

Figure 4.2. Scaling with unbiased integrators.

Implementation of the Sine/Cosine Generator

An interconnection block diagram of the system of Figure 4.2 is presented in Figure 4.3. All of the control logic inputs were mentioned earlier (Chapter II) except for the sign bit (SN) marker and

clocking (ϕ_1, ϕ_2). Data is processed serially in the DDA integrator least significant bit first. The sign bit marker indicates the end of one iteration period and the beginning of the next. It negates any data stored in the adder carry logic and controls the output flip-flops.

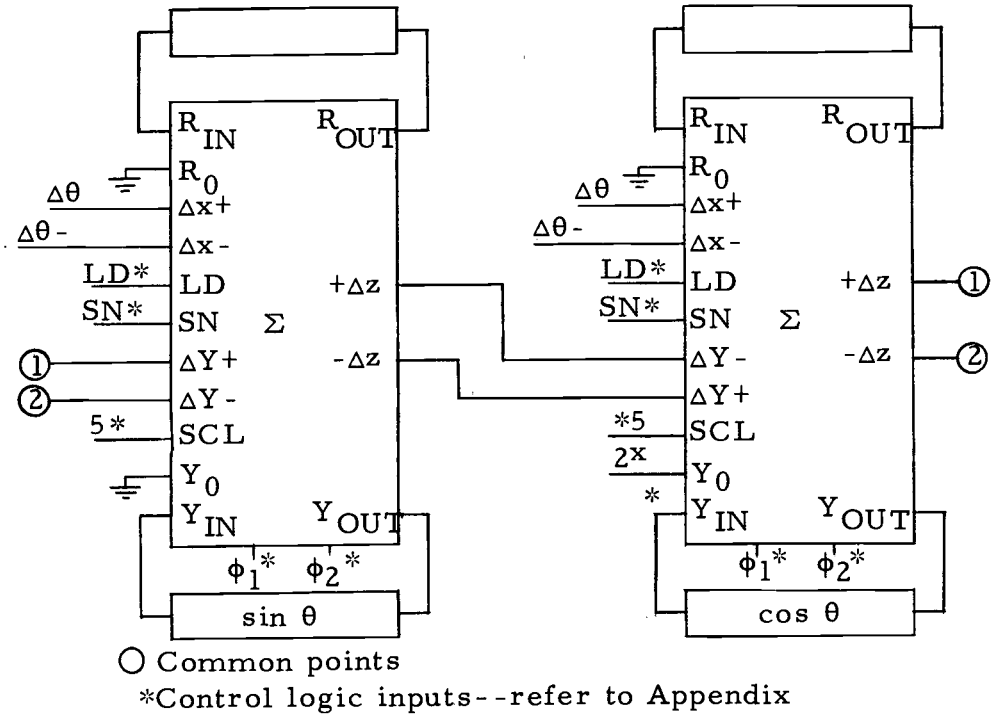


Figure 4.3. Implementation of the sine/cosine generator by rectangular integration.

A typical set of control logic inputs are presented in the Appendix. Two phase 400 KHz clocking is derived from a basic 4.8 MHz clock. This allows a system using dual sixteen bit shift registers to operate with an iteration rate of 25 KHz.

Parameters of Comparison

In determining which integrator techniques are to be used in a given application, accuracy, power dissipation and relative cost will be at least three of the factors considered. For this reason these three parameters will be utilized in comparing the different integration techniques. Accuracy will be considered later in the chapter. In regard to the other two, only the integrator power dissipation and cost will be considered as the control logic will remain roughly constant for all the integrators. No assembly or layout cost will be included as again, this cost is reasonably constant and hence will not effect the cost differences.

A list of the part types required in the integrators under consideration along with their power dissipation and cost is provided in Table 1 (pg. 65). The power dissipation and cost figures were determined by averaging the respective quantities on similar parts provided by different manufacturers.

For simplicity and availability the shift registers of Figure 4.3 will be assumed to be of the dual sixteen bit variety, even though this greatly exceeds the scaling requirements of Figure 4.2. In reality scaling accuracies yielding more resolution than those of Figure 4.2 would normally be used. Hence, the choice of the extended register probably makes the power dissipation and cost figures for the

integrator more realistic.

Cost and Power Dissipation

Referring to Table 1, it may be observed that the average integrator (DDA adder element and shift register) costs about $105 and dissipates on the order of 195 mw. The cost and power dissipation figures for the sine/cosine generator employing two such integrators is then $210 and 390 mw.

Sine/Cosine Generator Solution Form and Accuracy

Thus far, it has been stated that the integrator system of the type shown in Figure 4.2 is a sine/cosine generator but no mathematical evidence has been offered to support this claim. Figure 4.4 defines the symbols to be used in the analysis to follow.

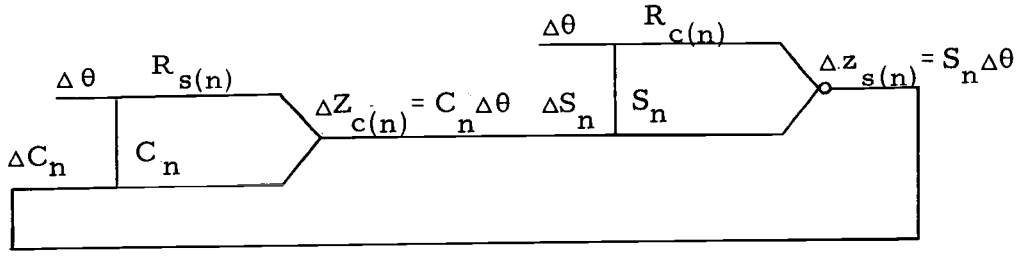


Figure 4.4. Definition of symbols.

Recall the overflow algorithm

$$M\Delta z_{n+1} + R_{n+1} = R_n + Y_{n+1}\Delta x$$

and note that the term $M\Delta z$ provides the inputs for the following Y registers. If now the initial values of the C and S registers are

$$C_0 = k \cos \theta$$

$$S_0 = k \sin \theta$$

then the desired result after n overflows would be

$$C_{n+1} = k \cos (\theta + n/M) \tag{4.6a}$$

$$S_{n+1} = k \sin (\theta + n/M) \tag{4.6b}$$

Applying the overflow algorithm to the integrators of Figure 4.4 and noting that $\Delta\theta$ has a value of one quantum results in

$$M\Delta z_{c(n+1)} + R_{s(n+1)} = R_{s(n)} + C_{n+1} \tag{4.7a}$$

$$-M\Delta z_{s(n+1)} + R_{c(n+1)} = R_{c(n)} + S_{n+1} \tag{4.7b}$$

Substituting

$$\Delta z_{s(n+1)} = C_{n+2} - C_{n+1}$$

$$\Delta z_{c(n+1)} = S_{n+2} - S_{n+1}$$

and rearranging yields

$$S_{n+2} + \frac{R_{s(n+1)}}{M} - (S_{n+1} + \frac{R_{s(n)}}{M}) = \frac{C_{n+1}}{M} \tag{4.8a}$$

$$C_{n+2} + \frac{R_{c(n+1)}}{M} - (C_{n+1} + \frac{R_{c(n)}}{M}) = -\frac{S_{n+1}}{M} \tag{4.8b}$$

Let

$$X_{n+1} = S_{n+1} + \frac{R_{s(n)}}{M} \tag{4.9a}$$

and

$$Y_{n+1} = C_{n+1} - \frac{R_{c(n)}}{M} \tag{4.9b}$$

Equations 4.9 simply provide for considering the entire integral value, (both most significant and least significant halves). Substituting Equations 4.9 into Equations 4.8 results in

$$X_{n+1} - X_n - \frac{Y_n}{M} = \frac{1}{M^2} R_{c(n)} \tag{4.10a}$$

$$Y_{n+1} - Y_n + \frac{X_n}{M} = \frac{1}{M^2} R_{s(n)} \tag{4.10b}$$

The solution to the linear difference Equations (4.10) is exactly analogous to the solution of simultaneous linear differential equations. The general solution is the sum of a particular solution and a complimentary function. The complimentary functions are a set of linearly independent solutions of the homogeneous equations obtained by letting the forcing function be zero.

It can be shown by direct substitution that the complimentary solutions of Equations 4.10 are of the form

$$X_n = AK^n \sin(n\alpha+\phi)$$

$$Y_n = AK^n \cos(n\alpha+\phi)$$

where $\tan\alpha = 1/M$ and $K^2 = 1 + 1/M^2$.

Similar results can be obtained by utilizing a z transform approach to the problem (8). The z transform approach initially ignores the round off error and allows that other means may be used in estimating its effects.

This paper will not seek to mathematically analyze the effects of round off error other than to acknowledge its existence in the particular solution to Equations 4.10. In general the solution of the particular equation is not of closed form. The technique of variation of parameters provides a systematic method for solution of the particular integral.

Referring to Equations 4.9 and allowing $X_{P(n)}$ and $Y_{P(n)}$ to represent the particular solutions, the general solutions may be written as follows:

$$S_n = AK^n \sin(n\alpha+\phi) + X_{P(n)} - R_{s(n-1)}/M \qquad (4.11a)$$

$$C_n = AK^n \cos(n\alpha+\phi) + Y_{P(n)} + R_{c(n-1)}/M \qquad (4.11b)$$

Comparing the results noted in Equations 4.11 with those desired as expressed in Equations 4.6, multiple sources of error are observed. The only readily recognizable effect is that of simple

round off. Truncation and round off propagation errors combined in a complex fashion are sources of the remaining errors. Even if the errors noted as simple round off and the particular solution are negligible, the solution form is seen to be exponential. Therefore, if any degree of accuracy is to be maintained, particularly in more complex systems, it will be necessary to limit the range of the input angle.

To assist in visualizing the effects of the error terms in Equations 4.11, rectangular sine/cosine generator state sequence data has been plotted in Figures 4.5, 4.6, and 4.7. Figures 4.5 and 4.6 compare the actual values of sine and cosine with their respective digital approximations. It may be observed in both figures that the magnitudes of the approximate sine and cosine tend to grow and that associated with this growth is a phase shift.

It should also be noted that the author has not allowed Y register data to propagate as it should. If allowed to propagate normally, the value of the cosine at 6.5 radians would have been negative as the Y register overflowed into its sign bit. Overflow could be prevented at any predefined point by rescaling the system.

A plot of the approximate sine and cosine is compared with the unit radius circle in Figure 4.7. The exponential effect is responsible for the outward spiral of the calculated curve. Occasionally systems are designed to sense the zero crossing of the sine and cosine and

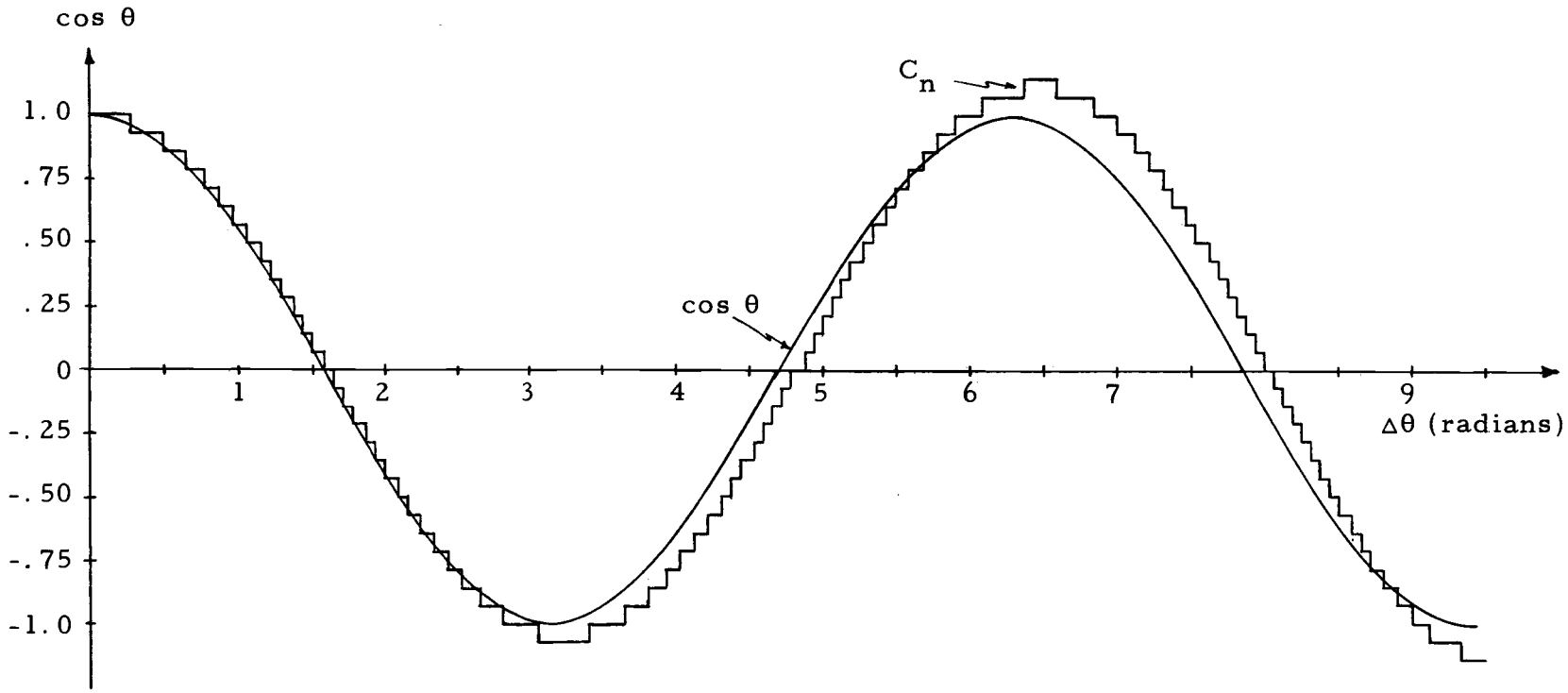


Figure 4.5. Comparison of cos θ and C_n, the DDA approximation of cos θ using rectangular integration.

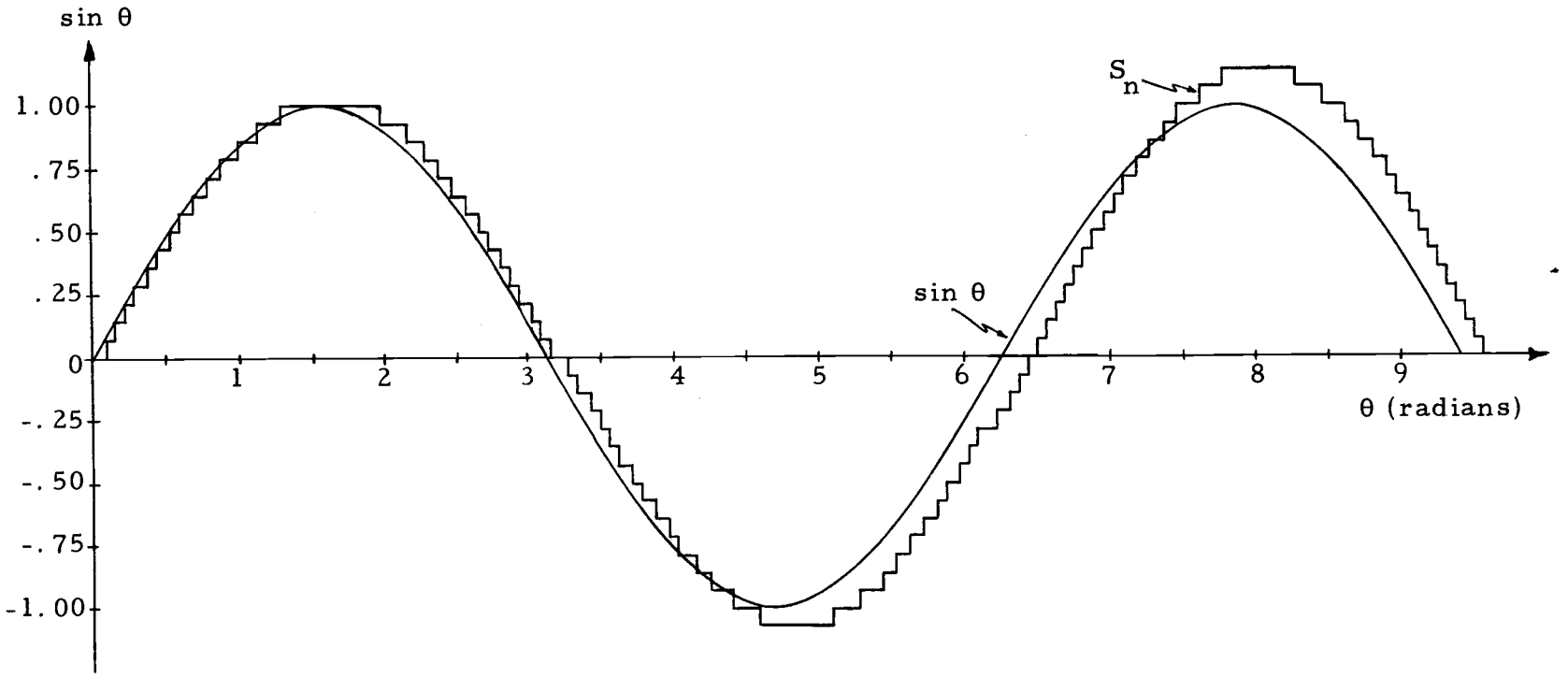


Figure 4.6. Comparison of sin θ and S_n, the DDA approximation of sin θ using rectangular integration.

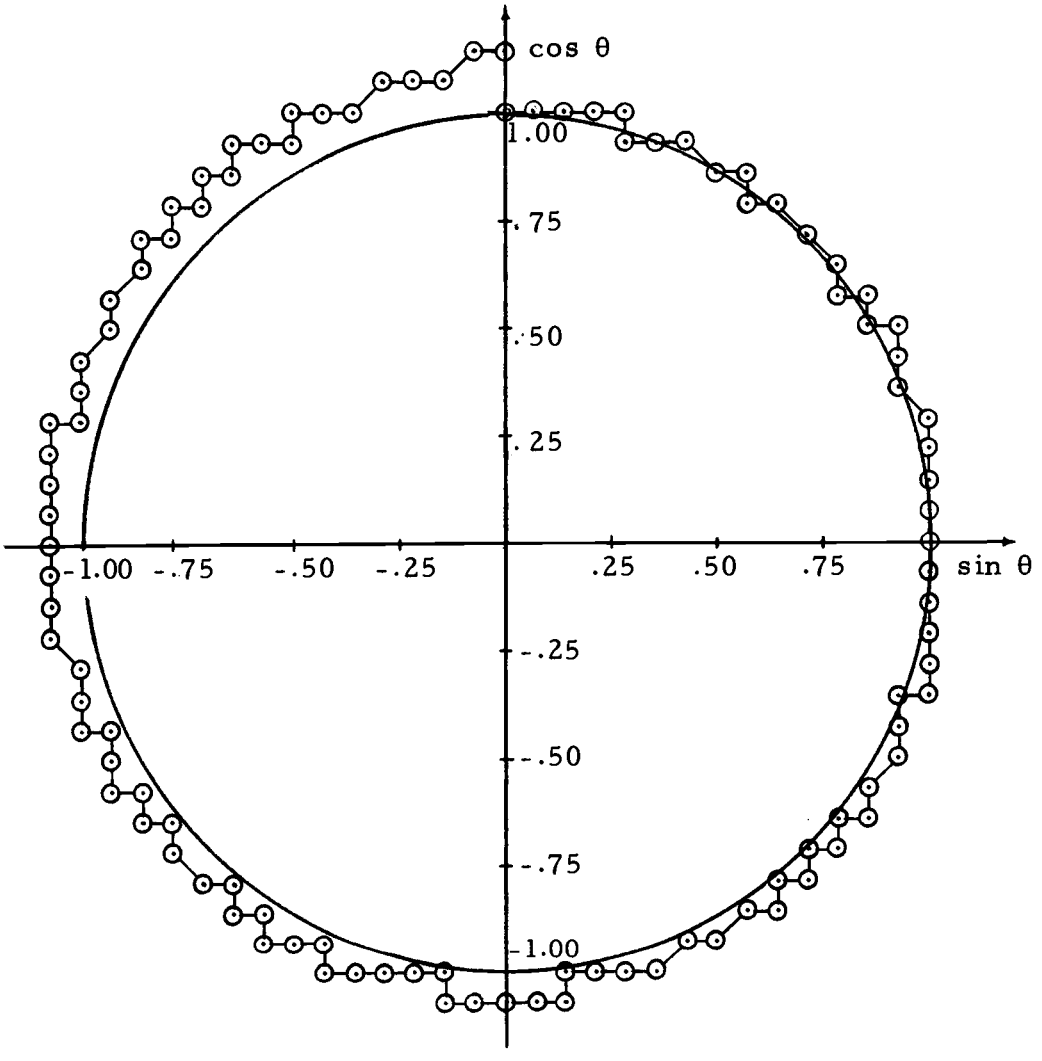


Figure 4.7. Comparison of the unit circle and the approximation obtained using rectangular integration.

reorthogonalize to maintain accuracy over large angle excursions. It may be observed that for the first $\pi/2$ radians, the rectangular approximation is fairly accurate.

Figures 4.8 and 4.9 are plots of the error between sine and cosine and their respective approximations. Simple round off error has been eliminated by utilizing both Y and R registers in computing the values for comparison. It may be observed in Figures 4.8 and 4.9 that the remaining error is increasing in magnitude and oscillatory in nature. The discontinuities in the sine error curve (Figure 4.9) occur as the sine passes through zero. The exact cause of this discontinuity has not yet been determined.

Before leaving rectangular integration and considering improved methods of integration, it is desirable to consider the effect of repeated angular reversals (angular excursions away from and back to zero).

Tables 2 (pg. 66) and 3 (pg. 67) show the effects of angular reversals with excursions of one-sixteenth and one-fourth radians with and without Y register stabilization allowed between inputs. It is noted that an error has accumulated in the R register when the cumulative angle has returned to null that is a function of the number of ΔY inputs. The error is specifically due to the order of algorithm solution, that is, the Y register is updated, then added to or subtracted from the R register. This results in a one

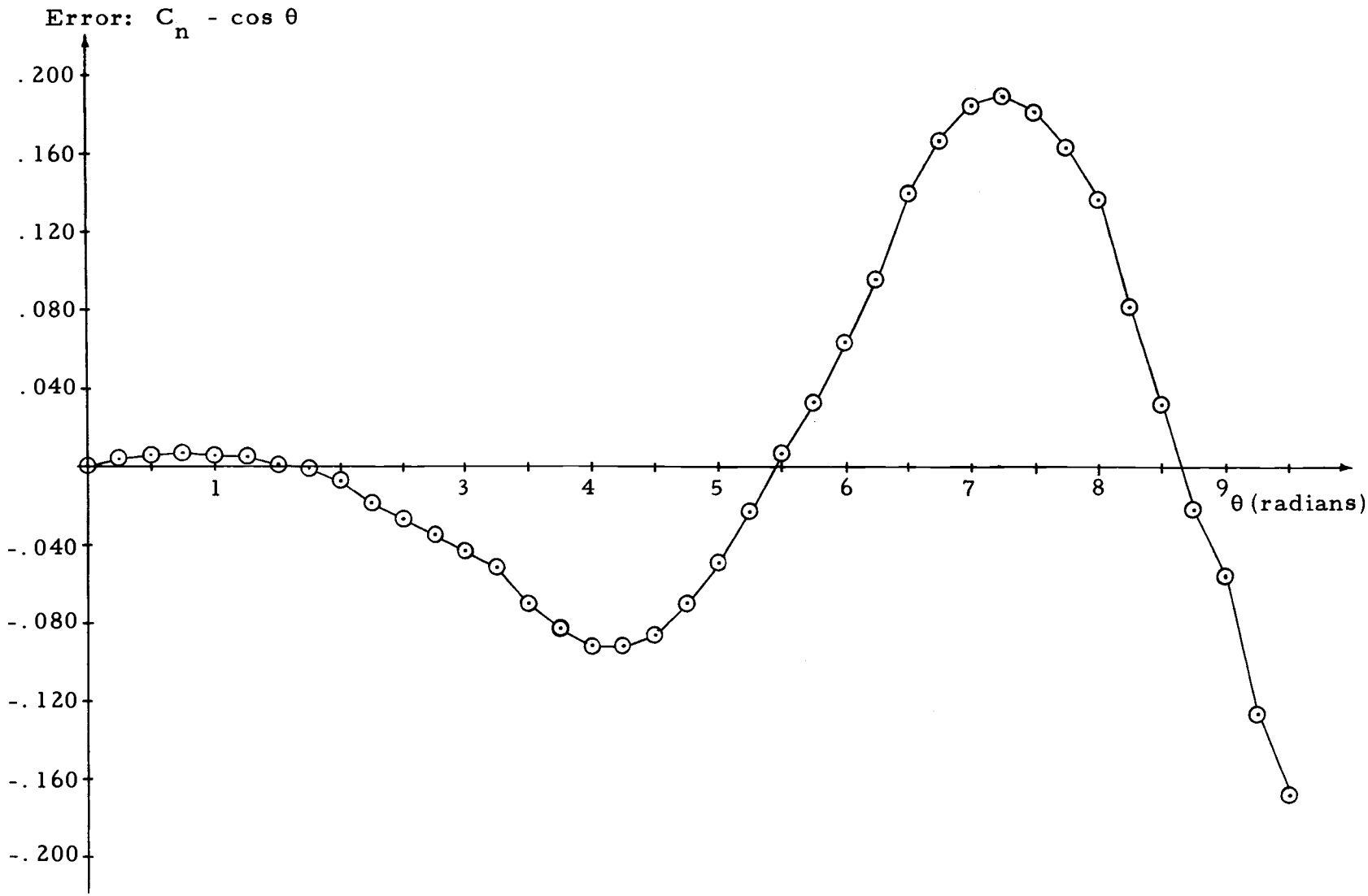


Figure 4.8. Difference between cos θ and its rectangular approximation.

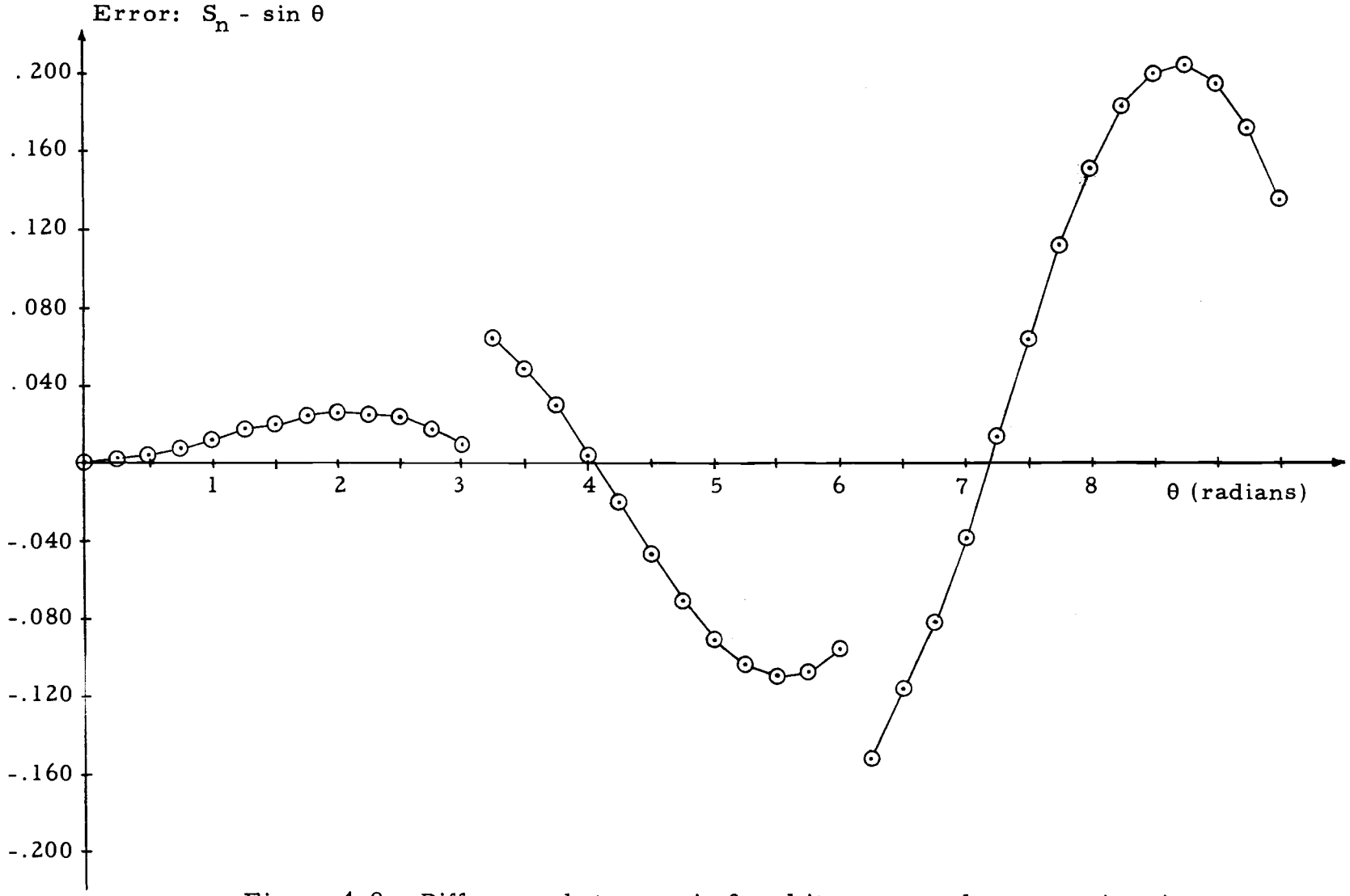


Figure 4.9. Difference between sin θ and its rectangular approximation.

quantum error in the R register per Y input at the time of reversal.

In summary it can be noted that the DDA sine/cosine generator using rectangular integration is simple to construct but subject to substantial cumulative errors when the input angle is large or prone to reversal. It should further be noted that if the incremental value of $\Delta\theta$ is reduced, improved accuracy would be obtained. However, the errors noted are inherent in the mechanization and would simply require a longer time to propagate into significant value.

V. TRAPEZOIDAL INTEGRATION

The Algorithms of Trapezoidal Integration

Consider that given the initial values of $\sin\theta_0$ and $\cos\theta_0$, one increment in angle occurs. The correct value of sine and cosine is now $\sin(\theta_0+\Delta\theta)$ and $\cos(\theta_0+\Delta\theta)$. Expanding these by use of the sums of angle formulae, it is observed that

$$\sin(\theta_0+\Delta\theta) = \sin\theta_0 \cos\Delta\theta + \sin\Delta\theta \cos\theta_0$$

$$\cos(\theta_0+\Delta\theta) = \cos\theta_0 \cos\Delta\theta - \sin\theta_0 \sin\Delta\theta .$$

If now the series sum representation is used for the $\Delta\theta$ terms, the equations take the form

$$\sin(\theta_0+\Delta\theta) = \sin\theta_0\left(1-\frac{\Delta\theta^2}{2!}+\frac{\Delta\theta^4}{4!}-\ldots\right)+\cos\theta_0\left(\Delta\theta-\frac{\Delta\theta^3}{3!}+\frac{\Delta\theta^5}{5!}-\ldots\right) \tag{5.1a}$$

$$\cos(\theta_0+\Delta\theta) = \cos\theta_0\left(1-\frac{\Delta\theta^2}{2!}+\frac{\Delta\theta^4}{4!}-\ldots\right)+\sin\theta_0\left(\Delta\theta-\frac{\Delta\theta^3}{3!}+\frac{\Delta\theta^5}{5!}-\ldots\right) \tag{5.1b}$$

It may be observed in Equations 5.1 that if the functions of $\Delta\theta$ are approximated by only first order terms, rectangular integration formulas result. If second order terms are included, the process accuracy is increased and the result is trapezoidal integration.

To verify that inclusion of the second order terms does indeed represent trapezoidal integration, and to determine the operating

algorithms, consider Figure 5.1. Trapezoidal integration may be obtained by summing the minor trapezoids. A minor trapezoid would be defined by the equation

$$A_{i+1} = Y_i \Delta x + \frac{1}{2}(Y_{i+1} - Y_i)\Delta x = Y_i \Delta x + \frac{1}{2}\Delta Y_{i+1}\Delta x \qquad (5.2)$$

Since A_{i+1} is a partial integral, it might be replaced in DDA integrator terminology by the least significant half of the integral (R register) and the overflow. Trapezoidal integration would then be expressed in the algorithms

$$Y_{i+1} = Y_i + \Delta Y_{i+1} \qquad (5.3a)$$

$$M\Delta z_{i+1} + R_{i+1} = R_i + Y_{i+1}\Delta x + \frac{1}{2}\Delta Y_{i+1}\Delta x \qquad (5.3b)$$

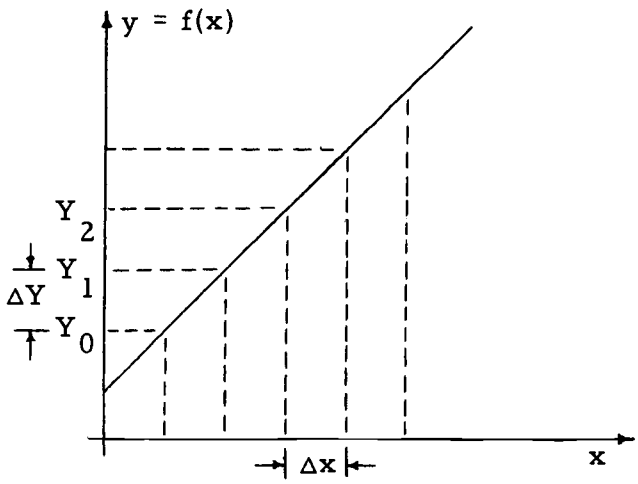


Figure 5.1. Trapezoidal integration.

If the substitutions

$$Y_{(i+1)} = \sin(\theta_0+i\Delta\theta),$$

$$Y_{(i+1)} = \cos(\theta_0+i\Delta\theta)$$

$$\Delta x = \Delta\theta$$

and the approximations

$$\Delta \sin[\theta_0+(i-1)\Delta\theta] \approx \Delta \sin(\theta_0+i\Delta\theta) = \cos(\theta_0+i\Delta\theta)\Delta\theta$$

$$\Delta \cos[\theta_0+(i-1)\Delta\theta] \approx \Delta \cos(\theta_0+i\Delta\theta) = -\sin(\theta_0+i\Delta\theta)\Delta\theta$$

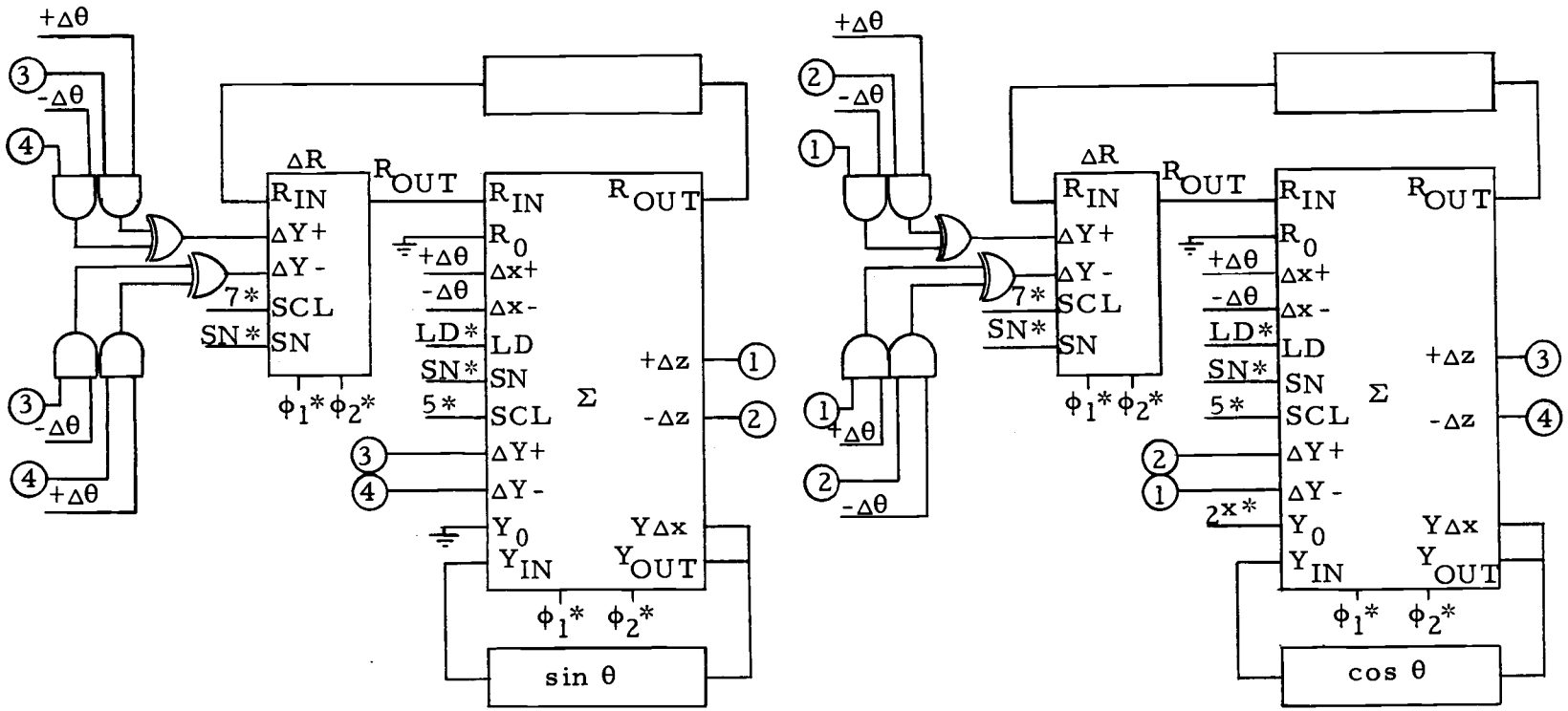
are made in two equations of the form of Equation 5.3b, the Equations 5.4 result.

$$M\Delta z_{c(i)}+R_{c(i)} - R_{c(i-1)} = \sin(\theta_0+i\Delta\theta)\Delta\theta + \frac{1}{2}[\cos(\theta_0+i\Delta\theta)\Delta\theta]\Delta\theta \tag{5.4a}$$

$$M\Delta z_{s(i)}+R_{s(i)} - R_{s(i-1)} = \cos(\theta_0+i\Delta\theta)\Delta\theta - \frac{1}{2}[\sin(\theta_0+i\Delta\theta)\Delta\theta]\Delta\theta \tag{5.4b}$$

If in Equations 5.1 the initial values of sine and cosine are transferred across the equality and the result compared with Equations 5.3, the effects of trapezoidal integration are apparent. It is readily noted that $-M\Delta z_{c(i)}$ should be used when updating the cosine.

Returning to Equations 5.3 it is observed that the addition of a second term into the R register will be required. Conceptually this presents no problem. Practically it requires the addition of an adder in the R register output line (Figure 5.2). This means that data leaving the R register must suffer some propagation delay



*Control circuit inputs - see Appendix
Common points

Figure 5.2. Sine/cosine generator using trapezoidal integration.

before reaching the DDA adder element R input. The additional time lag introduced by the incremental adder is about 250 nsec. This delay would not necessitate a reduction in maximum system operating frequency.

Implementation of the Sine/Cosine Generator

An interconnection block diagram for the sine/cosine generator utilizing trapezoidal integration is shown in Figure 5.2. The incremental adder shown was designed for use in expanding the number of ΔY inputs. It has somewhat more capability than is required in this application and it might be replaced by a full adder and some gating to eliminate word to word carries. The more compact adder offers the advantages of less components, reduced power dissipation and reduced cost.

Timing for the scale input on the incremental adder (ΔR) is normally advanced one count so that all partial sums are already formed when the input data is presented. This serves to maintain minimal propagation times. In this application the incremental scale input must be advanced by a second count to provide for division by two as defined in Equations 5.4. The impact of these advances on the system is that two additional shift register positions are required. (The 5 bit system requires 7 register positions.)

Cost and Power Dissipation

Each integrator is observed to consist of a four "and" gate package, a dual "exclusive or," a dual shift register, an incremental adder and the DDA adder element. Referring to Table 1 (pg. 65) for the cost and power dissipation of each of these components and adding to obtain the integrator figures, the cost per integrator is found to be $183 and the power dissipation 440 mw. The sine/cosine generator system would double both of these figures, since it uses two integrators with the resulting cost being $366 and the power dissipation 880 mw.

Sine/Cosine Generator Solution Form and Accuracy

As was done in the case of rectangular integration, it is desirable to investigate the difference equations and ascertain at least the basic form of the solution. Consider now the trapezoidal integration algorithm (Equation 5.3b) applied to sine and cosine (symbols defined in Figure 4.4) as noted below:

$$M\Delta z_{s(n+1)} = R_{s(n)} - R_{s(n-1)} + C_{n+1}\Delta\theta + \frac{(C_{n+1}-C_n)\Delta\theta}{2} \qquad (5.5a)$$

$$M\Delta z_{c(n+1)} = -R_{c(n)} + R_{c(n-1)} - S_{n+1}\Delta\theta - \frac{(S_{n+1}-S_n)\Delta\theta}{2} \qquad (5.5b)$$

Substituting

$$\Delta z_{s(n+1)} = S_{n+2} - S_{n+1}$$

$$\Delta z_{c(n+1)} = C_{n+2} - C_{n+1}$$

$$\Delta\theta = 1$$

into Equations 5.5 and rearranging yields

$$M(S_{n+2}-S_{n+1}) = R_{s(n)} - R_{s(n+1)} + \frac{3C_{n+1}}{2} - \frac{C_n}{2} \qquad (5.6a)$$

$$M(C_{n+2}-C_{n+1}) = -R_{c(n)} + R_{c(n+1)} - \frac{3S_{n+1}}{2} + \frac{S_n}{2} \qquad (5.6b)$$

The terms $3C_{n+1}/2$ and $2S_{n+1}/2$ are somewhat disconcerting because it is well known that trapezoidal techniques result in summations of the form $\frac{1}{2}(Y_{i+1}+Y_i)$. This anomaly is due to the fact that the last occurring ΔY is being used to predict the next increase in the Y term.

If now Equations 5.6 are transformed using the substitutions

$$X_n = S_{n+1} + R_{s(n)}/M$$

$$Y_n = C_{n+2} - R_{c(n)}/M$$

Equations 5.7 are obtained.

$$X_{n+1} - X_n - \frac{3Y_n}{2M} + \frac{Y_{n-1}}{2M} = \frac{3R_{c(n)}}{2M^2} - \frac{R_{c(n-1)}}{2M^2} \qquad (5.7a)$$

$$Y_{n+1} - Y_n + \frac{3X_n}{2M} - \frac{X_{n-1}}{2M} = \frac{3R_{s(n)}}{2M^2} - \frac{R_{s(n-1)}}{2M^2} \qquad (5.7b)$$

Using the complimentary and particular solution approach discussed earlier, wherein the complimentary solution defines the general form of the result and the particular solution represents an error term, it can be shown that

$$S_{n+1} = A \sin(n\alpha+\phi) + S_{(n+1)P} - R_{s(n)}/M$$

$$C_{n+1} = A \cos(n\alpha+\phi) + C_{(n+1)P} + R_{c(n)}/M$$

where $S_{(n+1)P}$ and $C_{(n+1)P}$ represent the particular solutions, A and ϕ are arbitrary constants and

$$\sin\alpha = \frac{1/M}{1-1/4M^2} = \frac{1}{M}\left[\frac{4M^2}{4M^2-1}\right]$$

Comparing these results with Equations 4.6, it is observed that excluding particular solution and simple round off error terms, the results differ only in the approximation of α. For small increments in α, these differences should be negligible.

The approximated sine and cosine obtained using trapezoidal integration are compared with their respective functions in Figures 5.3 and 5.4. No magnitude or phase shift errors are observed.

In Figure 5.5 the approximated values of sine and cosine are

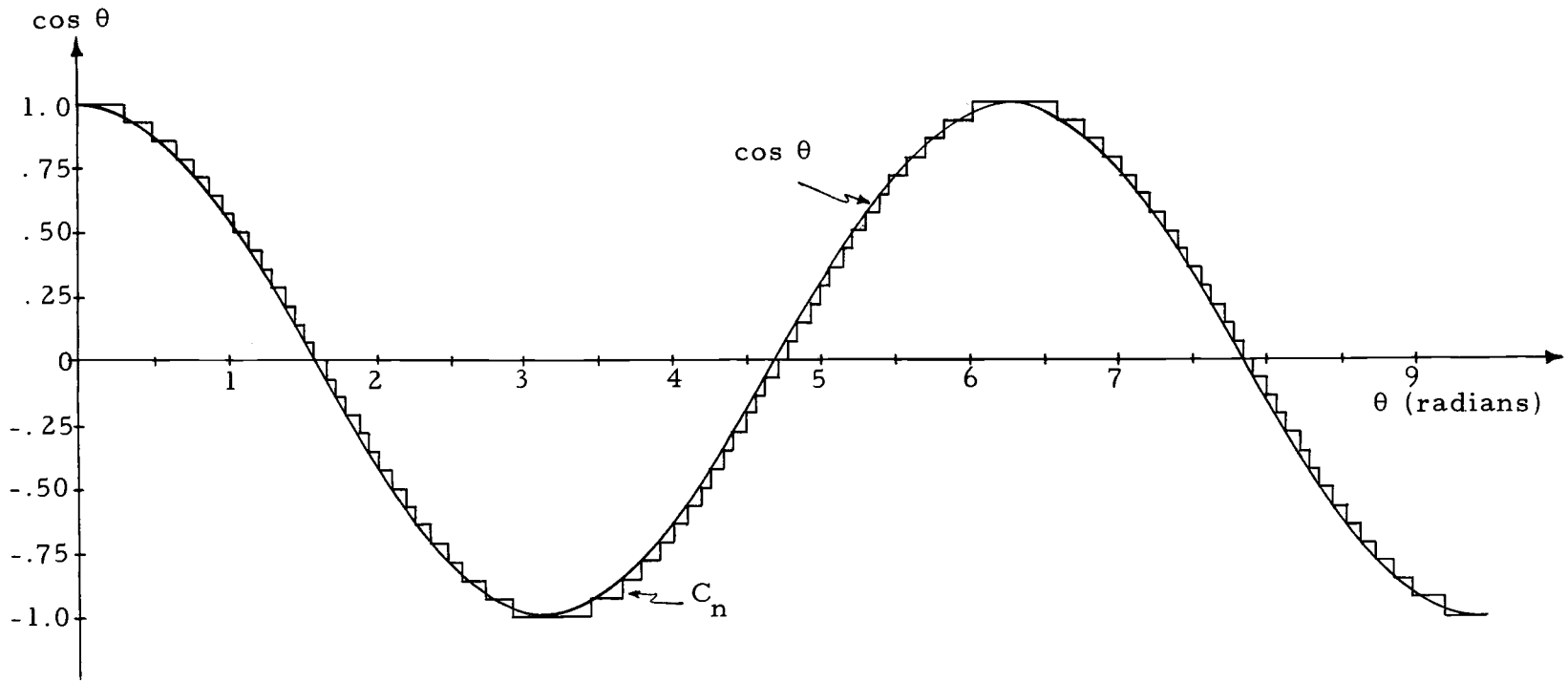


Figure 5.3. Comparison of cos θ and C_n, the DDA approximation of cos θ using trapezoidal integration.

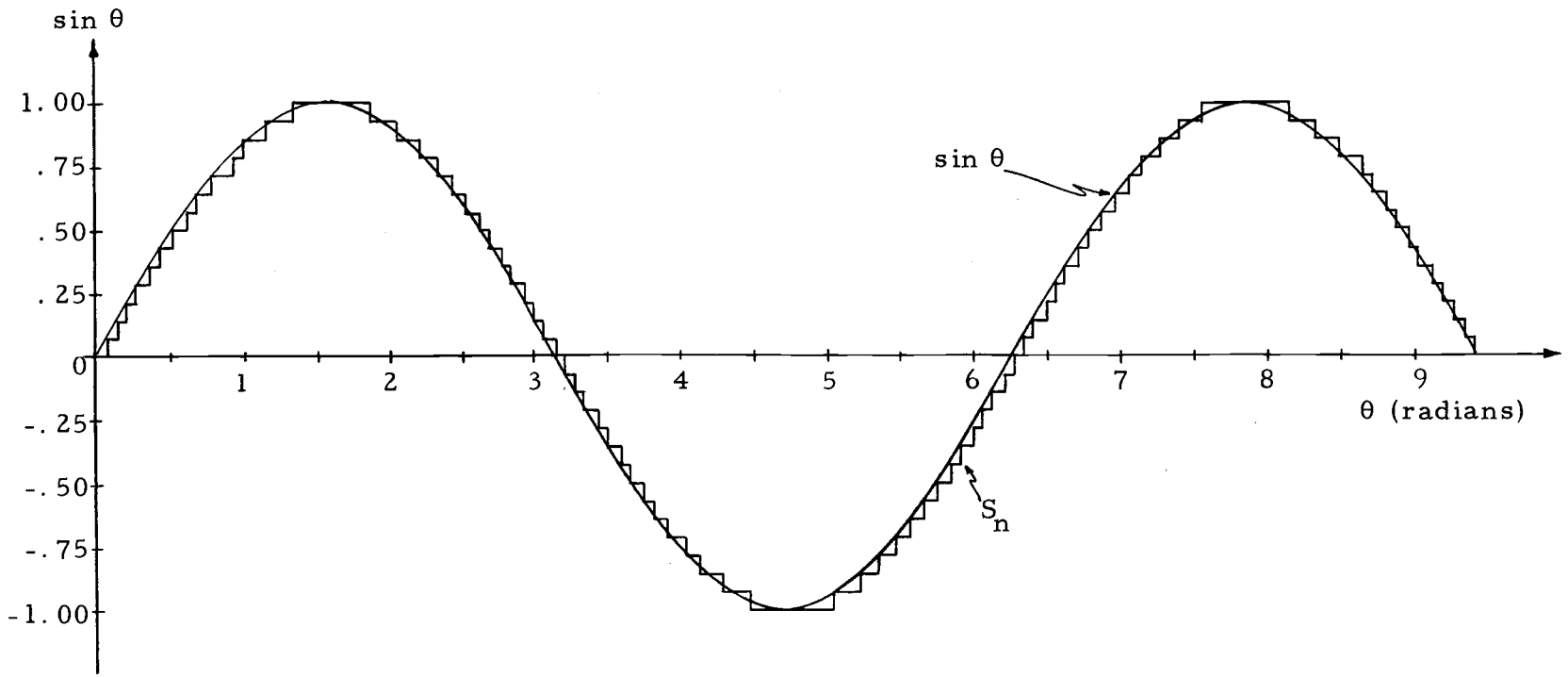


Figure 5.4. Comparison of sin θ and S_n, the DDA approximation of sin θ using trapezoidal integration.

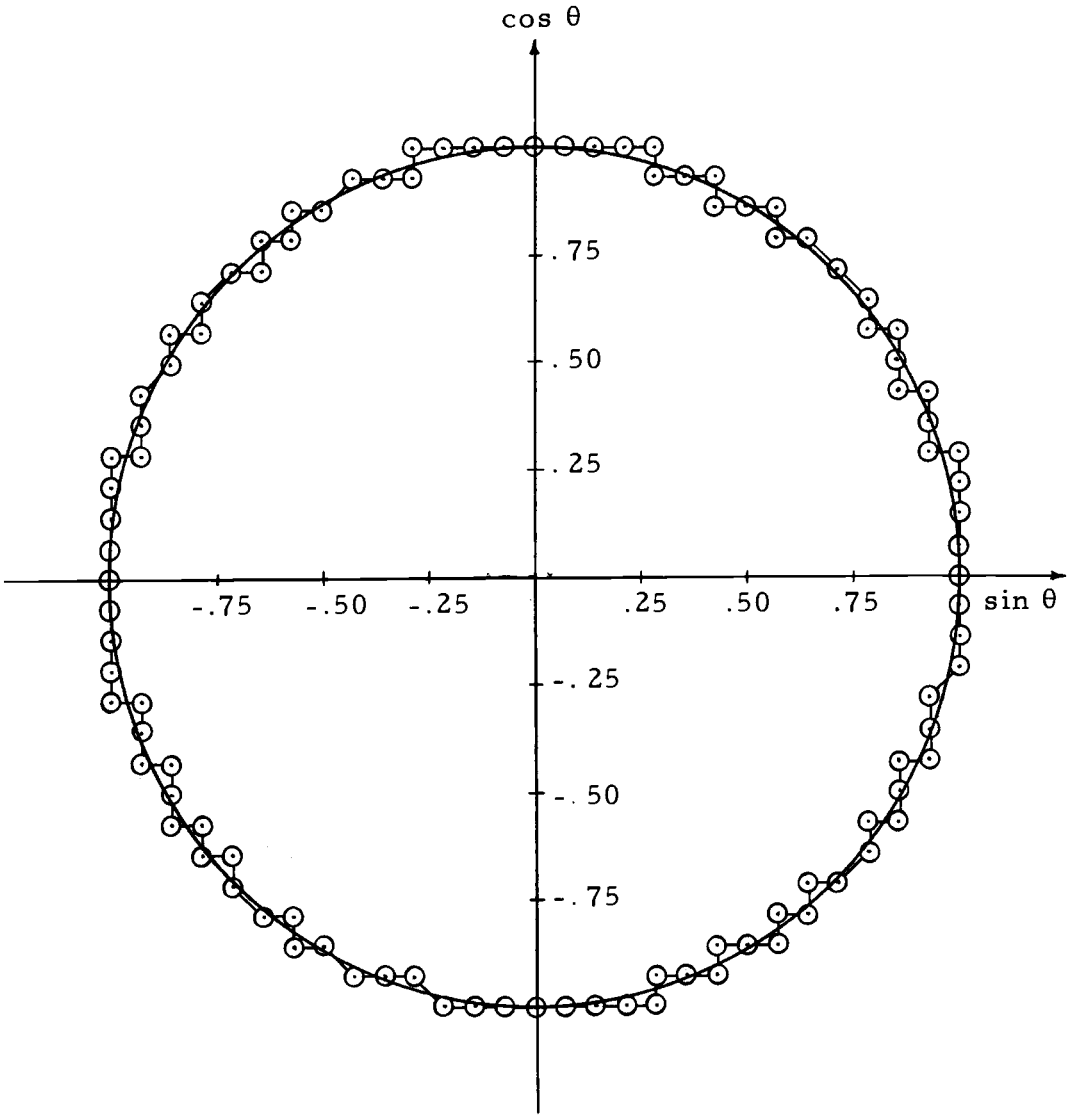


Figure 5.5. Comparison of the unit cycle and the approximation obtained using trapezoidal integration.

compared with the unit circle. It may be observed that a stable radius is maintained.

The differences between the actual values of sine and cosine and those obtained using the trapezoidal sine/cosine generator for selected points are displayed as a function of θ in Figures 5.6 and 5.7. The approximated values were obtained by utilizing data in the appropriate Y and R registers so as to eliminate the effects of simple round off. These differences represent the combined effects of truncation and propagation errors and are observed to be roughly sinusoidal and of the same frequency as their generating functions. The errors are observed to be shifted approximately $\pi/2$ radians relative to their respective functions. This is to be expected since the maximum rate of change in the sinusoidal waveform occurs as it passes through zero.

In rectangular integration input angle reversals were shown to cause error build up in the R register. Tables 4 (pg. 69) and 5 (pg. 70) show the effects of reversals with excursions of one sixteenth and one fourth radians with and without Y register stabilization allowed between inputs. It is observed that in those cases where overflows are allowed to propagate into their respective Y registers prior to reversals that the proposed implementation technique degenerates to the rectangular approximation because the product $\Delta Y \Delta x$ cannot be formed since Δx and ΔY do not

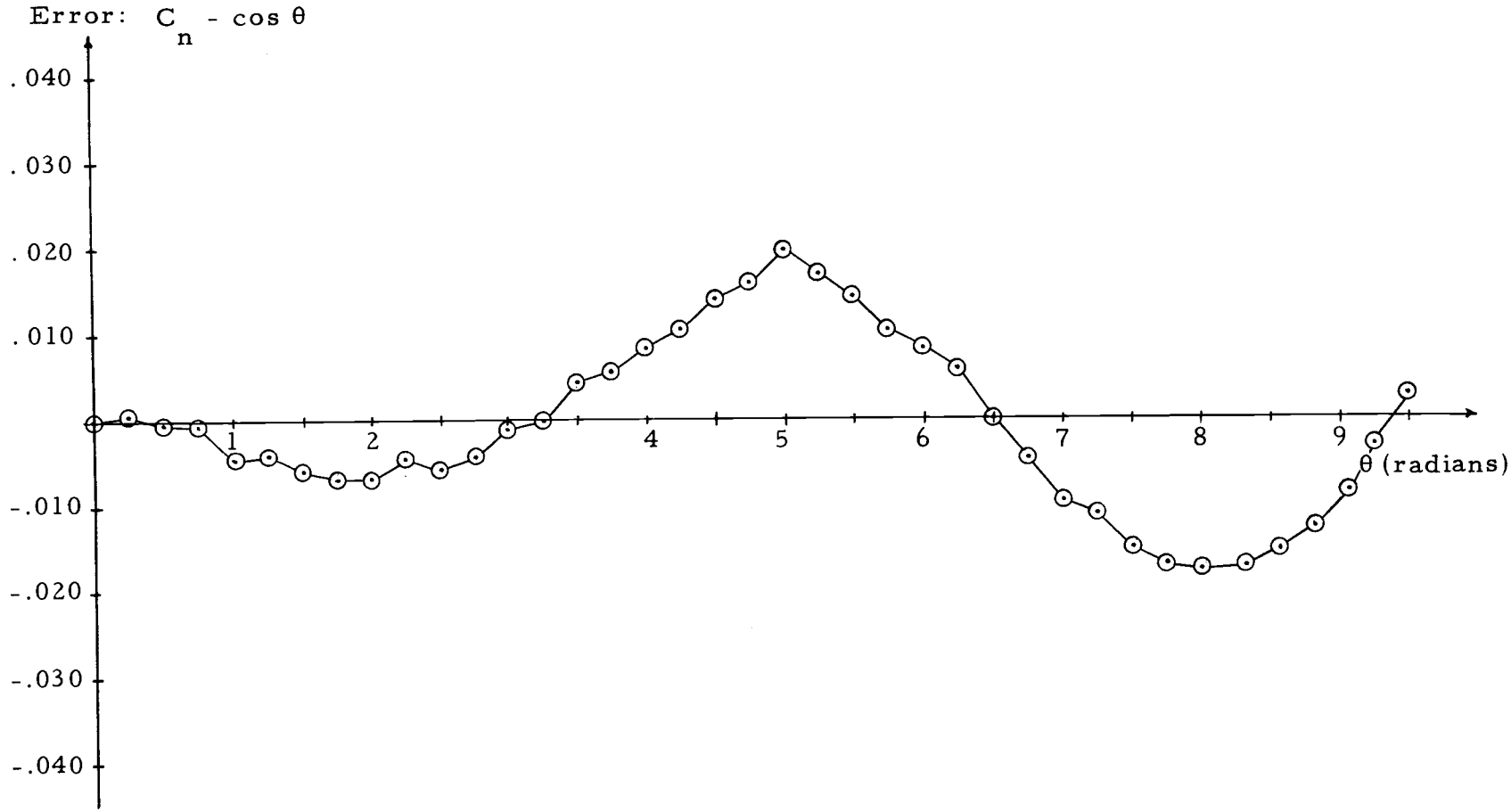


Figure 5.6. Difference between cos θ and its trapezoidal approximation.

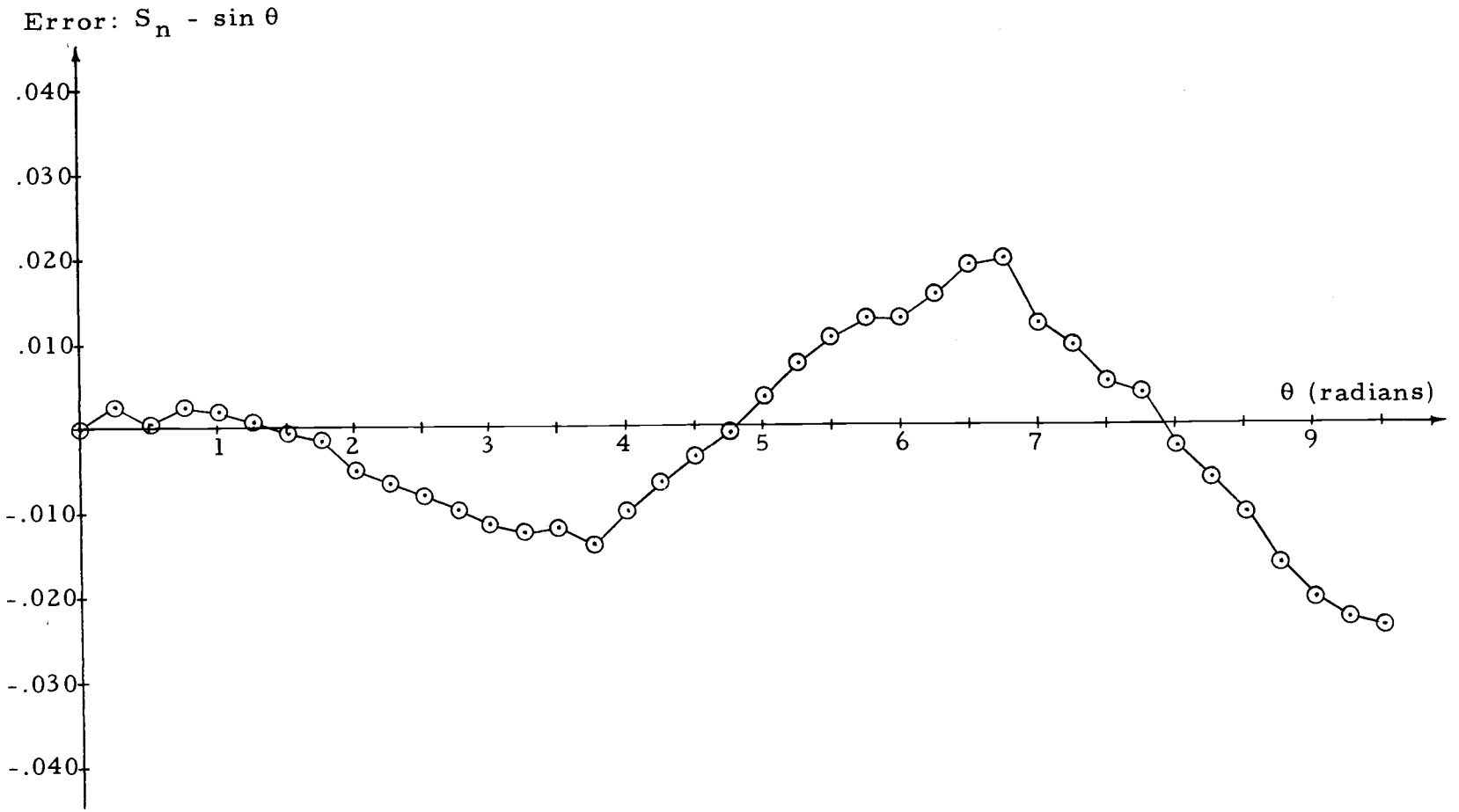


Figure 5.7. Difference between sin θ and its trapezoidal approximation.

exist concurrently. On the other hand in those cases where propagation of overflows is not completed prior to reversal, an error of minus one quantum per reversal is accumulated in the R register.

Modified Trapezoidal Integration

Consider the R register update algorithm as noted below:

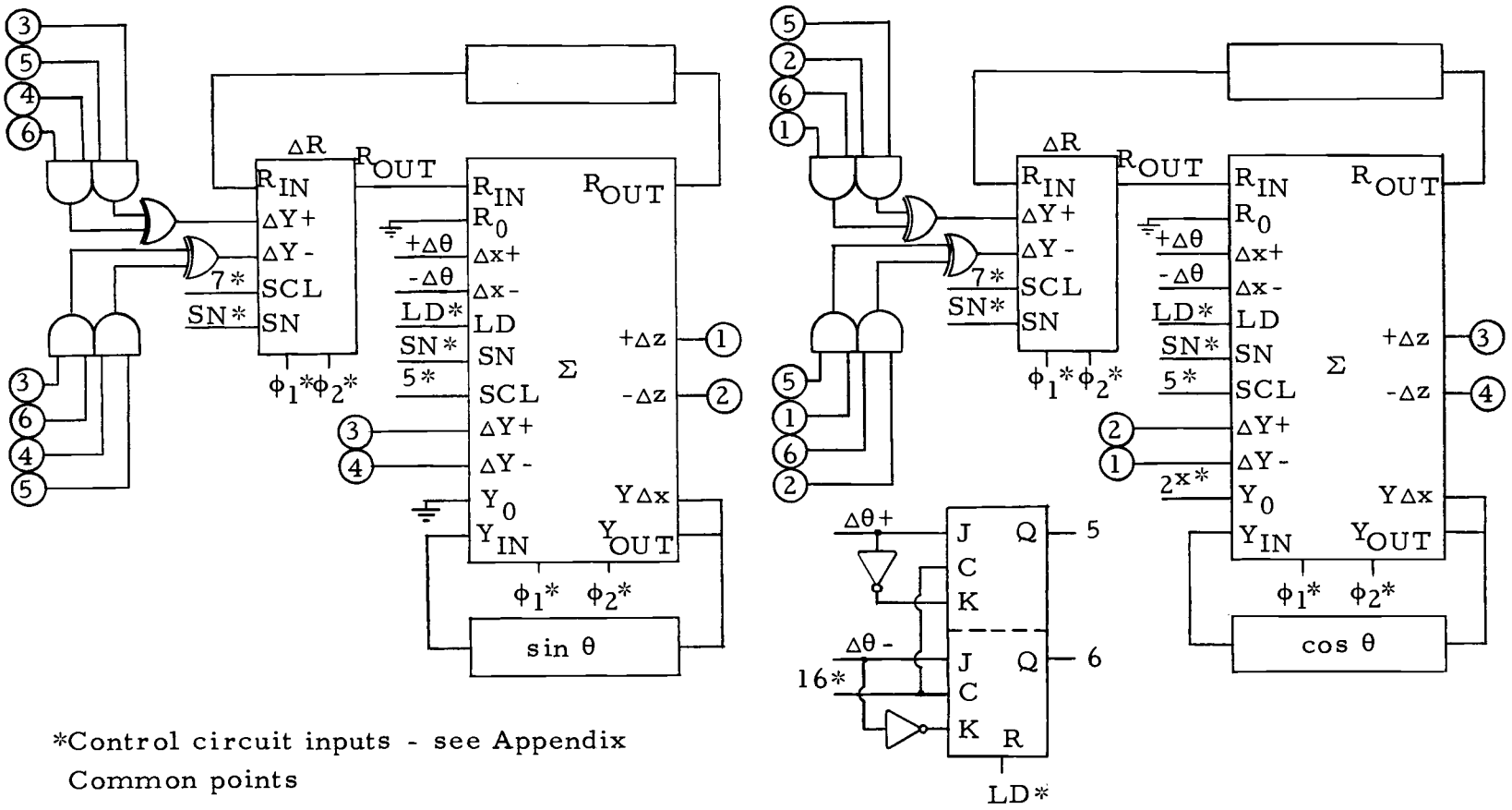
$$M\Delta z_{n+1} + R_{n+1} = R_n + Y_{n+1}\Delta x_{n+1} + \frac{1}{2}\Delta Y_n \Delta x_{n+1}$$

The last term in this expression is the second order term and was generated by the previous Δx signal. Strictly speaking, then, the update algorithm should be:

$$M\Delta z_{n+1} + R_{n+1} = R_n + Y_{n+1}\Delta x_{n+1} + \frac{1}{2}\Delta Y_n \Delta x_n$$

Tables 6 (pg. 72) and 7 (pg. 73) demonstrate that the susceptability of the trapezoidal integrator to reversal error is eliminated when the above noted correction is made.

An interconnection block diagram of the sine/cosine generator using the modified trapezoidal technique is shown in Figure 5.8. The modification is accomplished with the addition of a storage flip-flop and two inverting gates. The addition increases generator cost by $25 and power dissipation by 147 mw. When the modified technique is used, the sine/cosine generator cost and power dissipation are



*Control circuit inputs - see Appendix
Common points

Figure 5. 8. Sine/cosine generator using modified trapezoidal integration.

increased to $391 and 1.027 watts respectively. All previous comments regarding the accuracy of the calculations remain valid since no reversals in $\Delta\theta$ were involved.

In summary it must be noted that the trapezoidal integration technique is substantially more accurate than the rectangular technique. However, increased cost, power dissipation and a small increase in iteration time are the price.

VI. TWO CYCLE INTEGRATION

The Technique and Its Algorithms

One of the basic error contributing factors in the DDA integrator sine/cosine generator is the fact that unlike its analog counter part, sine and cosine do not update simultaneously and continually. Calculations are based on what has happened during the preceding cycle rather than on what is happening during the current cycle.

To compensate for errors of this type a predictor/corrector (two cycle) approach to integration has been developed. During the predictor cycle rectangular integration is used to predict overflows that might occur during the current iteration period. These predicted overflows are used during the second or corrector cycle to form the integral using an equivalent of the trapezoidal technique.

Figure 6.1 may be used to more easily visualize the modified trapezoidal technique. Initially some value of Y will be known. During the predictor cycle predicted overflows $(\Delta z \rightarrow \Delta Y)$ are observed. During the corrector cycle these overflows are given a scale weighting of one half the normal (in other words they represent increments of value $\Delta Y/2$) and are added into an imaginary Y register. This Y register value is used in forming the integral by the standard rectangular technique and results in the summation of rectangles of the form

$$\Delta x(Y_n + \frac{\Delta Y}{2}) = (Y_n + \frac{Y_{n+1} - Y_n}{2})\Delta x = (\frac{Y_{n+1} + Y_n}{2})\Delta x \qquad (6.1)$$

which may be observed to be exactly the form obtained by trapezoidal integration (Equation 5.2). Overflows obtained during the corrector cycle when the rectangles defined by Equation 6.1 are summed into the R register are used to update the Y register at normal (ΔY) weighting during the predictor cycle.

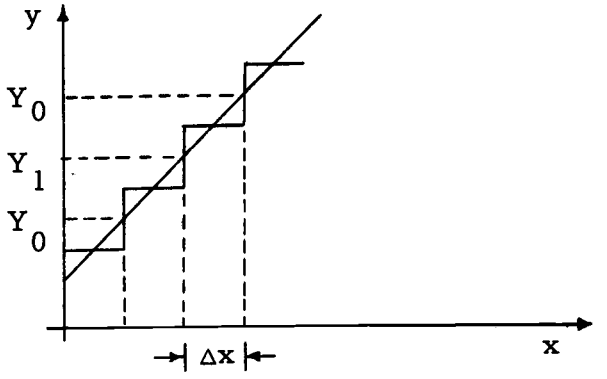


Figure 6.1. Two cycle integration.

To perform integration in the style noted above it is necessary to implement the following algorithms:

Predictor

$$Y_{n+1} = Y_n + \Delta Y \qquad (6.2a)$$

$$M\Delta z_{(n+1)p} + R_{(n+1)i} = R_{(n)i} + Y_{n+1}\Delta x \qquad (6.2b)$$

Corrector

$$Y_{(n+1)i} = Y_{n+1} + \Delta Y/2 \qquad (6.2c)$$

$$M\Delta z_{(n+1)c} + R_{(n+1)} = R_n + Y_{(n+1)i}\Delta x \qquad (6.2d)$$

The subscript i is used to denote the imaginary Y and R register values. The values are imaginary in that they exist only for the defined calculation and are not stored in their respective registers.

Implementation of Sine/Cosine Generator

The interconnection/logic diagram for the sine/cosine generator using the two cycle integration technique is shown in Figure 6.2. It is observed that the integrators themselves are modified simply by the inclusion of several "and" and "exclusive or" gates.

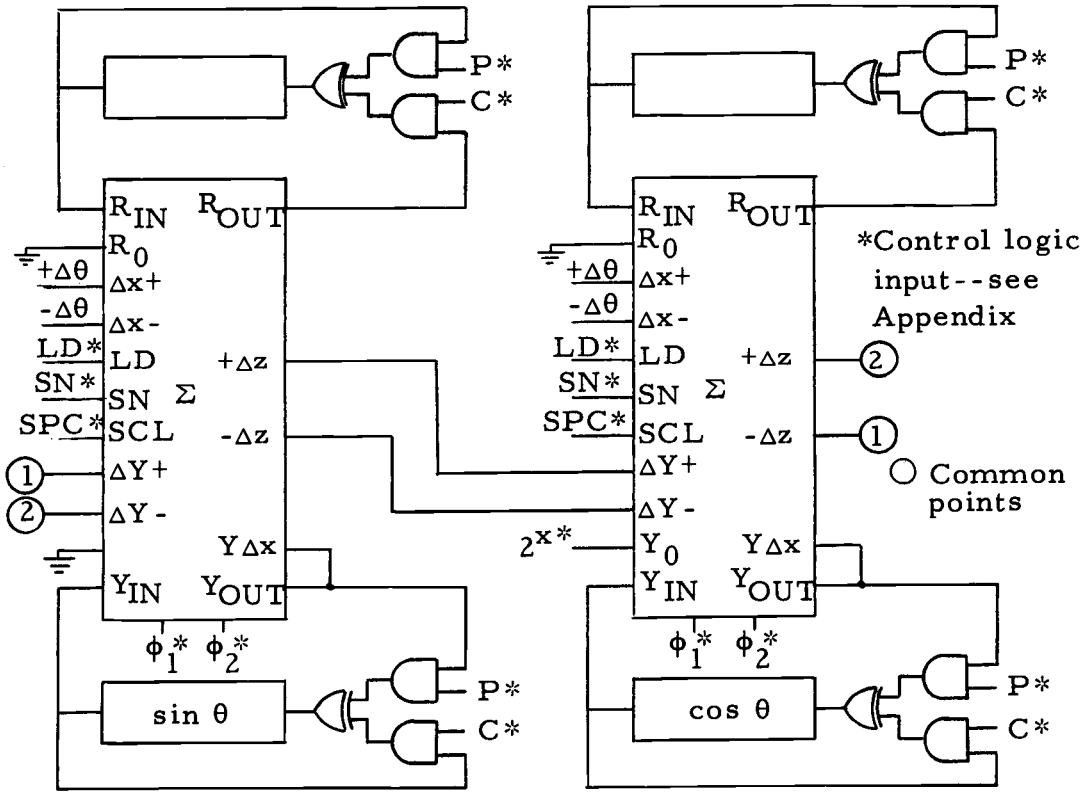


Figure 6.2. Implementation using two cycle integration.

The additional control circuit logic required for this type implementation has been taken from the Appendix and presented in Figure 6.3 for convenience. The flip-flop is simply a cycle counter and the gates are required for proper modulation of the scale bit. While these modifications to the control logic might appear extensive they need be done only once, which greatly reduces the per integrator cost in more complex systems.

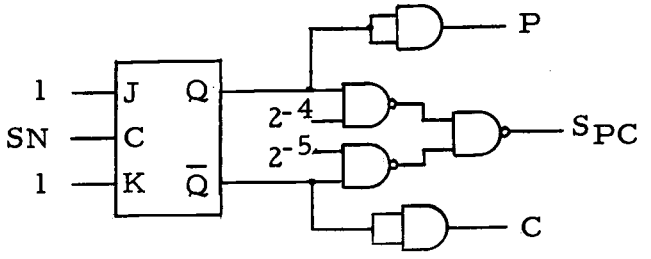


Figure 6.3. Additional control logic for two cycle integration.

Cost and Power Dissipation

Referring to Table 1 (pg. 65) for the cost and power dissipation figures of the various components required in the integrator, it may be determined that the two cycle integrator costs $133 and dissipates 280 mw. Allowing another $35 and 100 mw, for the control circuit modifications (not included in Table 1), the sine/cosine generator using two cycle integration would be $301 and dissipate 660 mw.

Sine/Cosine Generator Solution Form and Accuracy

As before it is desirable to investigate the update algorithms and determine the basic form of the solution. Since only the corrector R register overflows are actually used in updating the stored Y register, only this algorithm need be studied. Rewritting Equation 6.2d in the nomenclature of Figure 4.4 and allowing that $\Delta\theta = 1$, results in the following equations:

$$M\Delta z_{s(n+1)} = R_{s(n)} - R_{s(n+1)} + C_{n+1} + \frac{\Delta C_{n+1}}{2} \qquad (6.3a)$$

$$M\Delta z_{c(n+1)} = -R_{c(n)} + R_{c(n+1)} - S_{n+1} - \frac{\Delta S_{n+1}}{2} \qquad (6.3b)$$

If all the incremental (Δ) functions are replaced by a state difference defined as the next state less the incremental state ($\Delta z_i = Z_{i+1} - Z_i$), Equations 6.4 are obtained.

$$M(S_{n+2} - S_{n+1}) = R_{s(n)} - R_{s(n+1)} + \frac{C_{n+2} + C_{n+1}}{2} \qquad (6.4a)$$

$$M(C_{n+2} - C_{n+1}) = -R_{c(n)} + R_{c(n+1)} - \left(\frac{S_{n+2} + S_{n+1}}{2}\right) \qquad (6.4b)$$

It may be noted that the general form of Equations 6.4 in terms of the Y register variables is clearly trapezoidal.

If now

$$X_n = S_{n+1} + R_{s(n)}/M$$

$$Y_n = C_{n+1} - R_{c(n)}/M$$

are substituted into Equations 6.4, then

$$X_{n+1} - X_n - \frac{Y_{n+1}}{2M} - \frac{Y_n}{2M} = \frac{R_{c(n+1)}}{2M^2} + \frac{R_{c(n)}}{2M^2} \qquad (6.5a)$$

$$Y_{n+1} - Y_n + \frac{X_{n+1}}{2M} + \frac{X_n}{2M} = \frac{R_{s(n+1)}}{2M^2} + \frac{R_{s(n)}}{2M^2} \qquad (6.5b)$$

Again allowing that the solution to the difference Equations (6.5) consists of complimentary and particular parts, it can be shown that

$$X_n = A \sin(n\alpha+\phi) + X_{P(n)} \qquad (6.6a)$$

$$Y_n = A \cos(n\alpha+\phi) + Y_{P(n)} \qquad (6.6b)$$

where $X_{P(n)}$ and $Y_{P(n)}$ represent the particular solutions, A and ϕ are arbitrary constants and α is defined by

$$\sin\alpha = \frac{1/M}{1+1/4M^2} = \frac{1}{M}\left[\frac{4M^2}{1+4M^2}\right]$$

Equations 6.6 may be rewritten as

$$S_{n+1} = A \sin(n\alpha+\phi) + X_{P(n)} - R_{s(n)}/M \qquad (6.7a)$$

$$C_{n+1} = A \cos(n\alpha+\phi) + Y_{P(n)} + R_{c(n)}/M \qquad (6.7b)$$

Excluding the particular solution and simple round off error terms, the form of Equations 6.7 is clearly similar to those sought after, as defined in Equation 4.6. The only difference is in the definition of the angle α. Errors caused by this source should be negligible, so long as M is reasonably large or to word it another way, so long as the sample width is small. Even for resolutions as course as those used in the examples of this paper, this error source contributes less than 10^{-3} radians of error per radian input.

The approximate sine and cosine obtained using the two cycle integration technique are compared with their respective functions in Figures 6.4 and 6.5. As was the case with trapezoidal integration, none of the anomalies observed with rectangular integration are in evidence.

In Figure 6.6 the approximate sine and cosine are compared with the unit circle. Stability of the radius is observed.

In Figures 6.7 and 6.8 the differences between the approximate values of sine and cosine and their respective functions after simple round off errors have been eliminated are shown as a function of θ. The errors are seen to be somewhat sinusoidal in nature and lead their generating functions by roughly $\pi/2$ radians. The maximum error obtained during the test cycle is approximately one-half the

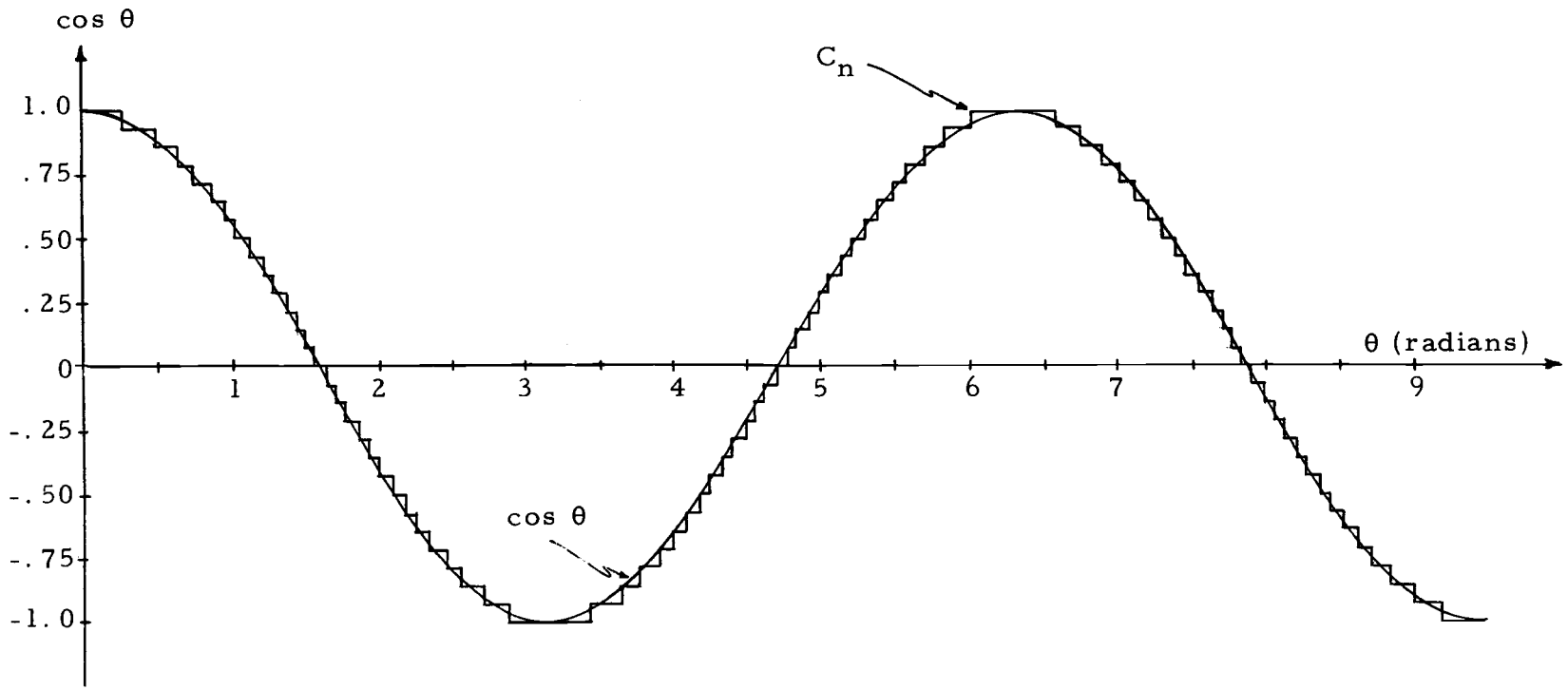


Figure 6.4. Comparison of cos θ and C_n, the DDA approximation using two cycle integration.

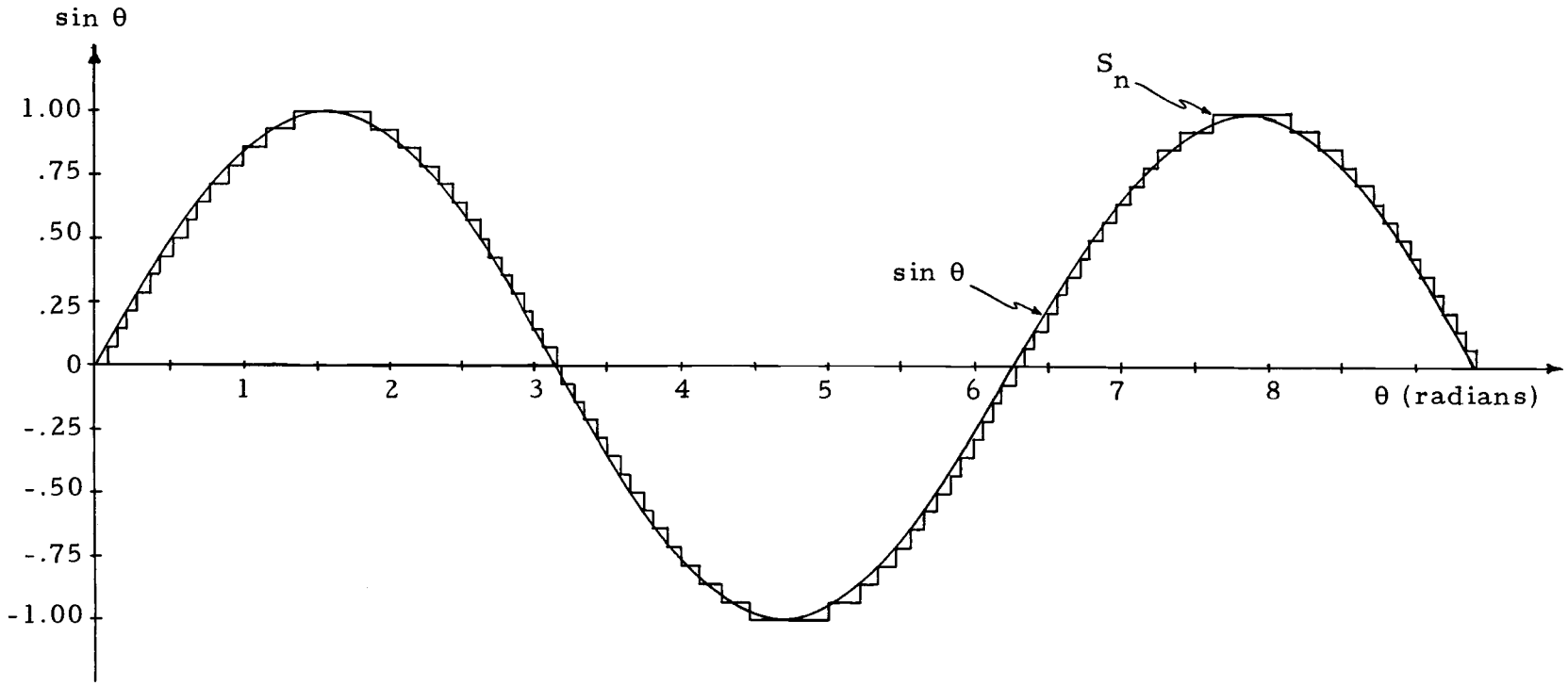


Figure 6.5. Comparison of sin θ and S_n, the DDA approximation of sin θ using two cycle integration.

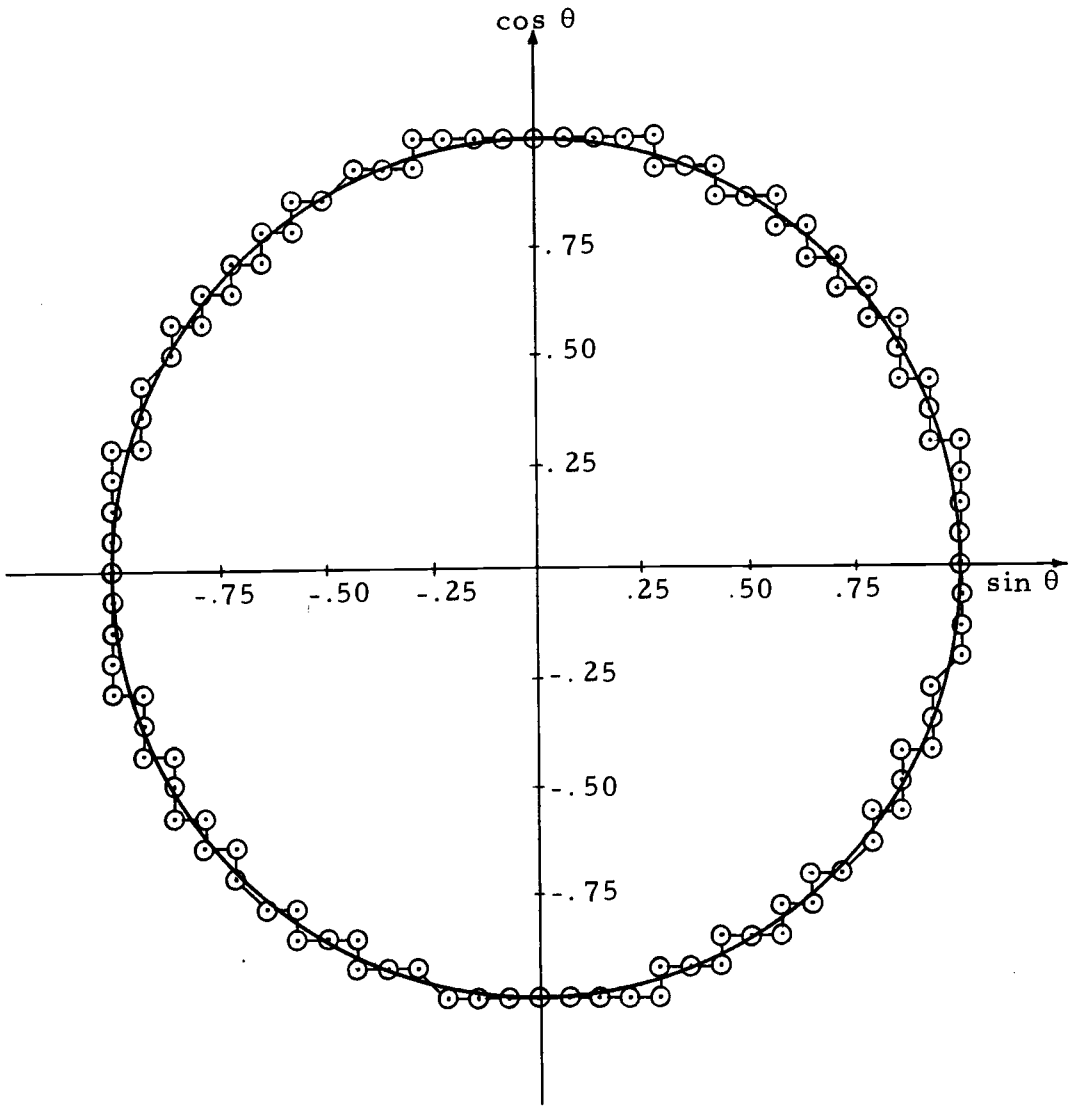


Figure 6.6. Comparison of the unit circle and the approximation obtained using two cycle integration.

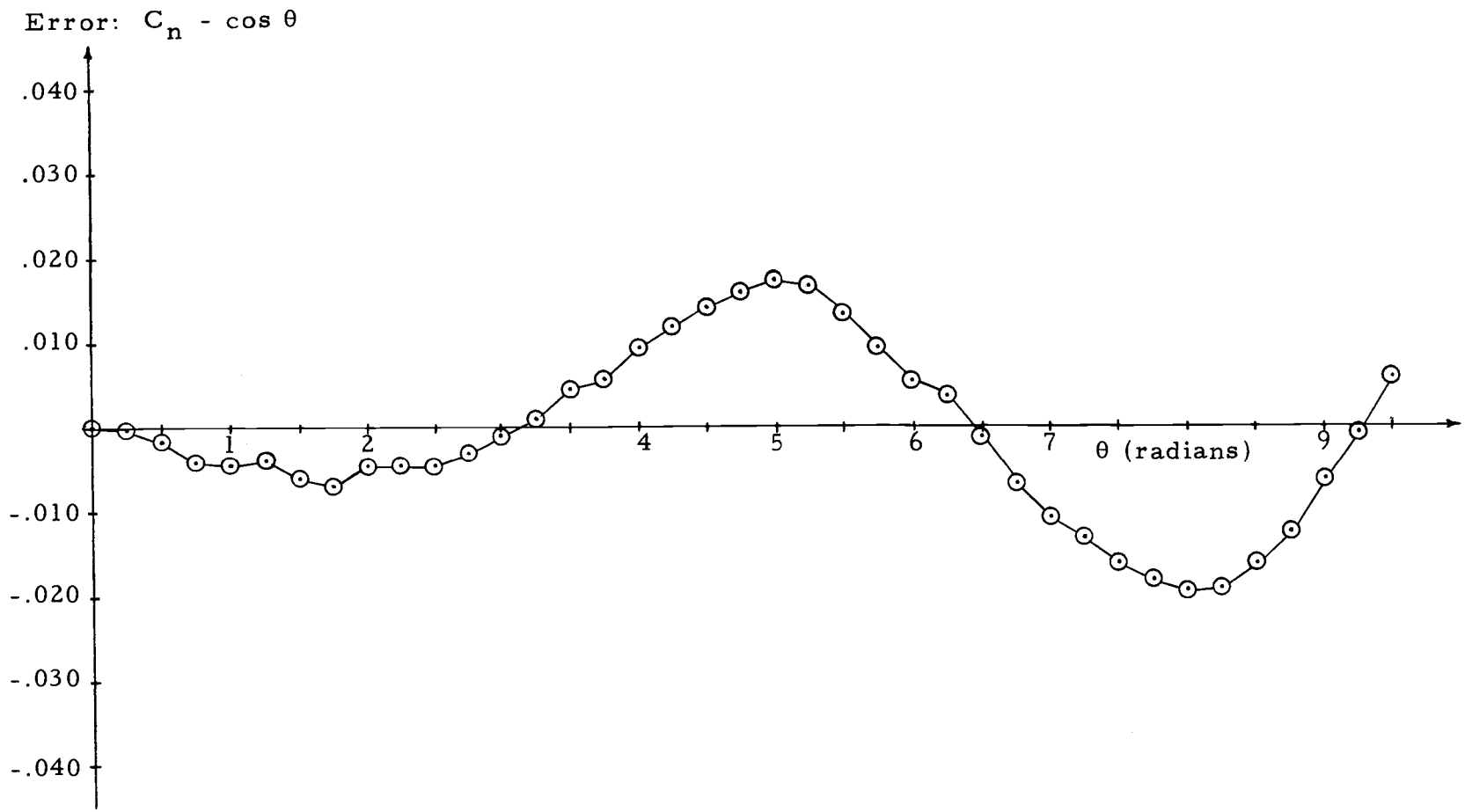


Figure 6.7. Difference between cos θ and its two cycle approximation.

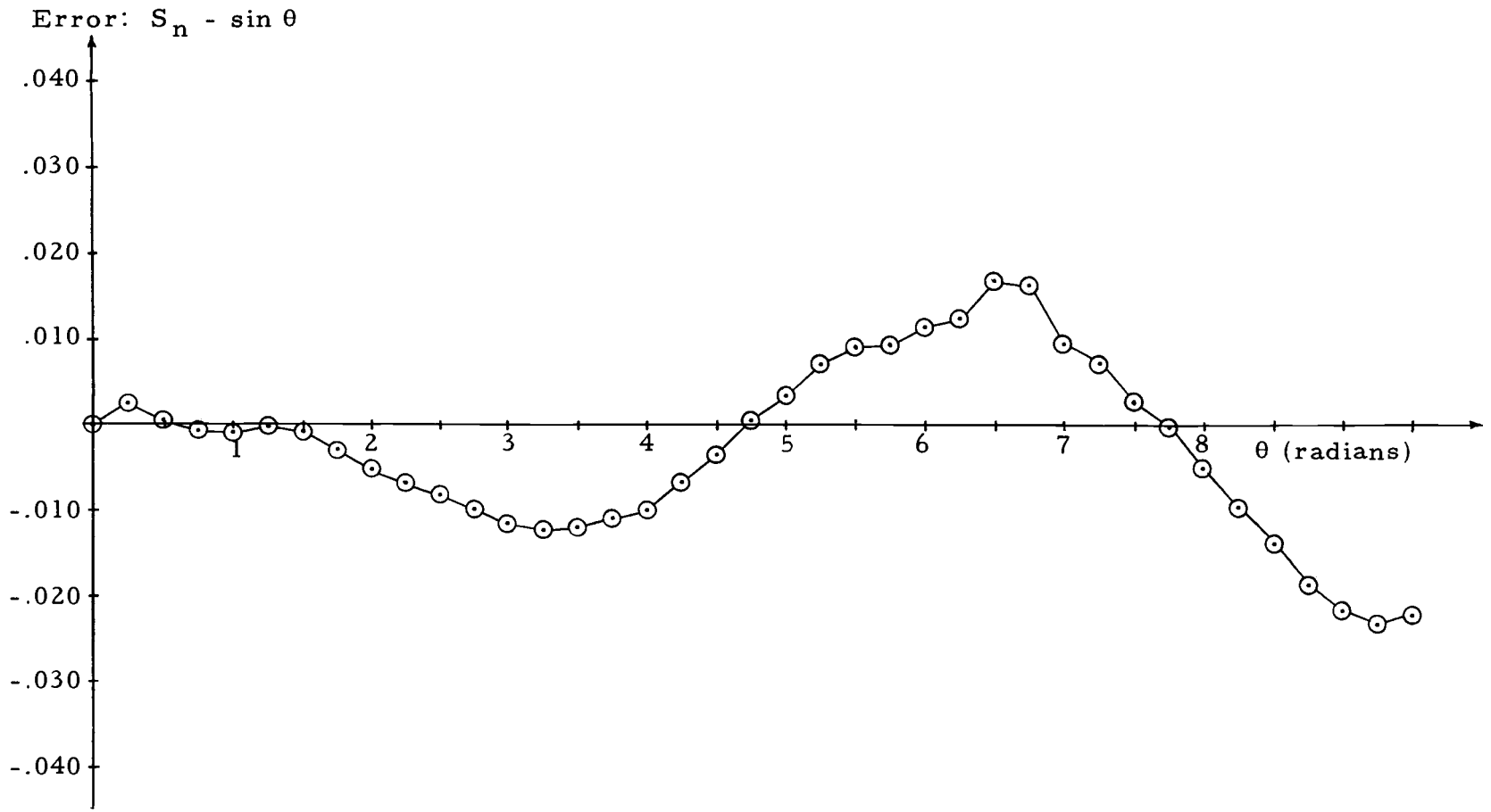


Figure 6.8. Difference between sin θ and its two cycle approximation.

value of the least significant bit in the Y register.

The errors generated using two cycle integration (Figures 6.7 and 6.8) may be observed to be nearly identical with those generated using trapezoidal integration (Figures 5.6 and 5.7). Since two cycle integration uses rectangular techniques to approximate trapezoidal integration, similar error patterns would be expected.

Tables 8 (pg. 75) and 9 (pg. 76) provide data concerning error buildup in the R register for excursions in the input angle of one-sixteenth and one-eighth radian with and without stabilization of Y register data allowed. It may be observed that no error is accumulated.

In summary it may be noted that two cycle integration provides a simple and inexpensive way of improving system performance. Cost and power dissipation are well below those of trapezoidal integration, yet the accuracy is equivalent. It should be noted, however, that a price is paid in time. The iteration period of two cycle integration is twice that of rectangular and nearly twice that of trapezoidal. It is felt that since the error mechanism within the two cycle DDA integrator is the same regardless of scaling, that the system could be rescaled by a factor of two to compensate for the increased iteration time without increasing the error by much more than a factor of two.

VII. SUMMARY AND CONCLUSIONS

Three different DDA integration techniques have been discussed. Implementation of each technique in the sine/cosine generator using commercially available MOS devices has been presented. The results of such implementation with regard to cost, power dissipation and accuracy have been investigated.

The rectangular integration technique is the most easily implemented. Only two parts per integrator are required. As a result cost and power dissipation are minimal. However, accuracy leaves something to be desired. In the sine/cosine generator application substantial errors were accumulated when large excursions or many reversals of the input angle were encountered.

The trapezoidal integration technique was observed to provide the most accurate results. However, it was expensive in terms of cost, parts count (six per integrator) and power dissipation.

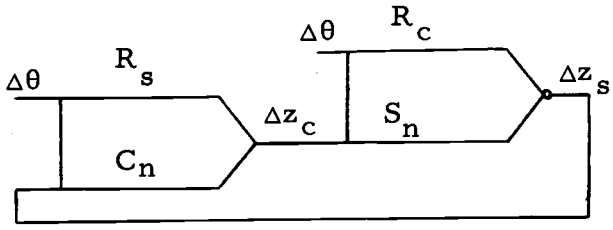
Two cycle integration offers possibly the most balanced approach. It is capable of providing accuracies nearly as good as those obtained using trapezoidal integration but requires only four parts per integrator, thus reducing the associated cost and power dissipation. In large, high density systems where the inaccuracy of rectangular integration cannot be tolerated, but power dissipation and parts count preclude the use of trapezoidal integration, the two cycle integration technique offers a desirable alternative.

Table 1. Component price and power dissipation list.

Component Description	Approximate Price* ($)	Power Dissipation mw
DDA adder element	90	70
Shift register	15	125
Incremental adder (ΔY)	50	160
Quad. "and" gate	10	43
Dual "exclusive or" gate	18	42
Triple "inverting" gate	10	27
Dual J-K flip-flop	15	120

*Approximate price is based on a rough average of the cost various manufacturers units of a given type as of July 1971.

Table 2. Sine/cosine generator using rectangular integration response to 1/16 radian reversals.



Rectangular Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$\Delta z + R_{n+1} = R_n + Y_{n+1}\Delta x$$

θ/32 (rad.)	C_n/14	R_s/224	Δz_c	S_n/14	R_c/224	Δz_s
A: No C_n/S_n Stabilization Before Reversals						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
1	14	+32-28=14	-1	1	-1	0
0	14	0	0	0	-1	0
1	14	14	0	0	-1	0
2	14	-32+28=-4	1	0	-1	0
1	14	+32-28=14	-1	1	-2	0
0	14	0	0	0	-2	0
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
2	14	-4	0	1	0	0
1	14	+32-28=14	-1	1	-1	0
1	14	14	0	0	-1	0
0	14	0	0	0	-1	0
1	14	14	0	0	-1	0
2	14	-32+28=-4	1	0	-1	0
2	14	-4	0	1	-1	0
1	14	+32-28=14	-1	1	-2	0
1	14	14	0	0	-2	0
0	14	0	0	0	-2	0

Table 3. Sine/cosine generator using rectangular integration response to 1/4 radian reversals. (See Table 2 for flow chart and algorithms.)

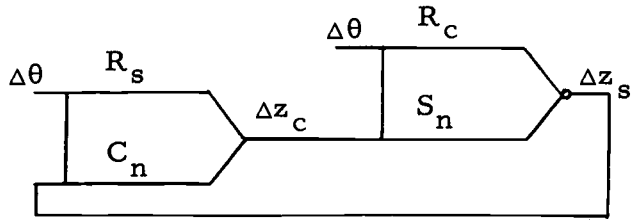
$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
3	14	10	0	1	1	0
4	14	-32+28=-8	1	1	2	0
5	14	+6	0	2	4	0
6	14	-32+28=-12	1	2	6	0
7	14	+2	0	3	9	0
8	14	-32+16=-16	1	3	12	0
7	14	+32-30=2	-1	4	8	0
6	14	-12	0	3	5	0
5	14	32-26=6	-1	3	2	0
4	14	-8	0	2	0	0
3	14	32-22=10	-1	2	-2	0
2	14	-4	0	1	-3	0
1	14	32-18=14	-1	1	-4	0
0	14	0	0	0	-4	0

(continued)

Table 3. Continued.

$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
2	14	-4	0	1	0	0
3	14	10	0	1	1	0
4	14	-32+24=-8	1	1	2	0
4	14	-8	0	2	2	0
5	14	6	0	2	4	0
6	14	-32+20=-12	1	2	6	0
6	14	-12	0	3	6	0
7	14	+2	0	3	9	0
8	14	-32+16=-16	1	3	12	0
8	14	-16	0	4	12	0
7	14	4	-1	4	8	0
7	14	2	0	3	8	0
6	14	-12	0	3	5	0
5	14	32-26=6	-1	3	2	0
5	14	6	0	2	2	0
4	14	-8	0	2	0	0
3	14	32-22=10	-1	2	-2	0
3	14	10	0	1	-2	0
2	14	-4	0	1	-3	0
1	14	32-18=14	-1	1	-4	0
1	14	14	0	0	-4	0
0	14	0	0	0	-4	0

Table 4. Sine/cosine generator (trapezoidal integration) response to 1/16 radian reversals.



Trapezoidal Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$\Delta z + R_{n+1} = R_n + Y_{n+1}\Delta x_{n+1} + \frac{\Delta Y_{n+1}}{2}\Delta x_{n+1}$$

θ/32 (rad.)	C_n/14	R_s/224	Δz_c	S_n/14	R_c/224	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	+1	0	0	0
1	14	14	-1	1	-1.5	0
0	14	0	0	0	-1.0	0
1	14	14	0	0	-1	0
2	14	-4	+1	0	-1	-
1	14	14	-1	1	-2.5	0
0	14	0	0	0	-2	0
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-4	+1	0	0	0
2	14	-4	0	1	0	0
1	14	14	-1	1	-1.0	0
1	14	14	0	0	-1.0	0
0	14	0	0	0	-1.0	0
1	14	14	0	0	-1.0	0
2	14	-4	+1	0	-1.0	0
2	14	-4	0	1	-1.0	0
1	14	14	-1	1	-2.0	0
1	14	14	0	0	-2.0	0
0	14	0	0	0	-2.0	0

Table 5. Sine/cosine generator (trapezoidal integration) response to 1/4 radian reversals. (See Table 4 for flow chart and algorithms.)

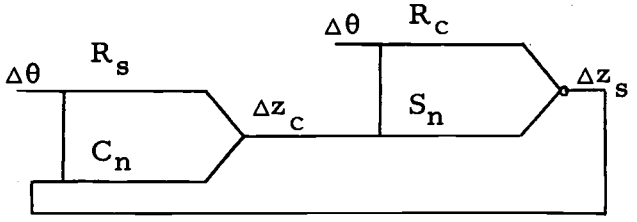
$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	+1	0	0	0
3	14	10	0	1	1.5	0
4	14	-8	1	1	2.5	0
5	14	6	0	2	5	0
6	14	-12	1	2	7	0
7	14	2	0	3	10.5	0
8	14	-16	1	3	13.5	0
7	14	+2	-1	4	9	0
6	14	-12	0	3	6.5	0
5	14	+6	-1	3	3.5	0
4	14	-8	0	2	2	0
3	14	10	-1	2	0	0
2	14	-4	0	1	- .5	0
1	14	+14	-1	1	-1.5	0
0	14	0	0	0	-1	0

(continued)

Table 5. Continued.

$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-4	+1	0	0	0
2	14	-4	0	1	0	0
3	14	10	0	1	1	0
4	14	-8	+1	1	2	0
4	14	-8	0	2	2	0
5	14	6	0	2	4	0
6	14	-12	+1	2	6	0
6	14	-12	0	3	6	0
7	14	2	0	3	9	0
8	14	-16	+1	3	12	0
8	14	-16	0	4	12	0
7	14	4	-1	4	8	0
7	14	2	0	3	8	0
6	14	-12	0	3	5	0
5	14	+6	-1	3	2	0
5	14	+6	0	2	2	0
4	14	-8	0	2	0	0
3	14	10	-1	2	-2	0
3	14	10	0	1	-2	0
2	14	-4	0	1	-3	0
1	14	14	-1	1	-4	0
			0	0	-4	0
0	14	0	0	0	-4	0

Table 6. Sine/cosine generator (modified trapezoidal integration) response to 1/16 radian reversals.



Trapezoidal Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$\Delta z + R_{n+1} = R_n + Y_{n+1}\Delta x_{n+1} + \frac{\Delta Y_{n+1}}{2}\Delta x_n$$

θ/32 (rad.)	C_n/14	R_s/224	Δz_c	S_n/14	R_c/224	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-4	1	0	0	0
1	14	14	-1	1	- .5	0
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
1	14	+32-18=14	-1	1	- .5	0
0	14	0	0	0	0	0
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-4	1	0	0	0
2	14	-4	0	1	.5	0
1	14	14	-1	1	- .5	0
1	14	14	0	0	0	0
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-4	1	0	0	0
2	14	-4	0	1	.5	0
1	14	14	-1	1	- .5	0
1	14	14	0	0	0	0
0	14	0	0	0	0	0

Table 7. Sine/cosine generator (modified trapezoidal integration) response to 1/4 radian reversal. (See Table 6 for flow chart and algorithms.)

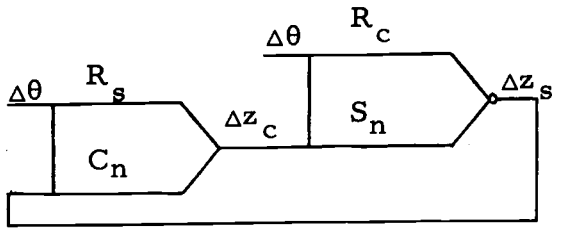
$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
3	14	10	0	1	1.5	0
4	14	-32+24=-8	1	1	2.5	0
5	14	6	0	2	5	0
6	14	-32+20=-12	1	2	7	0
7	14	2	0	3	10.5	0
8	14	-32+16=-16	1	3	13.5	0
7	14	+32-20=2	-1	4	10	0
6	14	-12	0	3	7.5	0
5	14	+32-26=6	-1	3	4.5	0
4	14	-8	0	2	3	0
3	14	+32-22=10	-1	2	1	0
2	14	-4	0	1	.5	0
1	14	32-18=14	-1	1	- .5	0
0	14	0	0	0	0	0

(continued)

Table 7. Continued

$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	14	0	0	0	0
2	14	-32+28=-4	1	0	0	0
2	14	-4	0	1	.5	0
3	14	10	0	1	1.5	0
4	14	-32+28=-8	1	1	2.5	0
4	14	-8	0	2	3	0
5	14	6	0	2	5	0
6	14	-32+20=-12	1	2	7	0
6	14	-12	0	3	7.5	0
7	14	2	0	3	10.5	0
8	14	-32+16=-16	1	3	13.5	0
8	14	-16	0	4	14	0
7	14	32-30=2	-1	4	10	0
7	14	2	0	3	10.5	0
6	14	-12	0	3	7.5	0
5	14	32-26=6	-1	3	4.5	0
5	14	6	0	2	5	0
4	14	-8	0	2	3	0
3	14	32-22=10	-1	2	1	0
3	14	10	0	1	1.5	0
2	14	-4	0	1	.5	0
1	14	32-18-14	-1	1	-.5	0
1	14	14	0	0	0	0
0	14	0	0	0	0	0

Table 8. Sine/cosine generator (two cycle integration) response to 1/16 radian reversals.



Two Cycle Algorithms:

$$Y_{n+1} = Y_n + \Delta Y$$

$$\Delta z + (R_{(n+1)i}) = R_n + Y_{n+1}\Delta x$$

$$(Y_{(n+1)i}) = Y_{n+1} + \Delta Y/2$$

$$\Delta z + R_{n+1} = R_n + Y_{(n+1)i}\Delta x$$

θ/32 (rad.)	C_n/14	R_s/224	Δz_c	S_n/14	R_c/224	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
2	14	(-32+28=-4)	1	0	(0)	0
	(14)	-4	1	(.5)	.5	0
1	14	(+32-18=14)	-1	1	(-.5)	0
	(14)	14	-1	(.5)	0	0
0	14	0	0	0	0	0
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
2	14	(-32-28=-4)	1	0	(0)	0
	(14)	-4	1	(.5)	.5	0
1	14	(32-18=14)	-1	1	(-.5)	0
	(14)	14	-1	(.5)	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
0	14	(0)	0	0	(0)	0

Table 9. Sine/cosine generator (two cycle integration) response to 1/8 radian reversals. (See Table 8 for flow chart and algorithms.)

$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
A: No C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
2	14	(-32+28=-4)	1	0	(0)	0
	(14)	-4	1	(.5)	.5	0
3	14	(10)	0	1	(1.5)	0
	(14)	10	0	(1)	1.5	0
4	14	(-32+24=-8)	1	1	(2.5)	0
	(14)	-8	1	(1.5)	3	0
3	14	(+32-22=10)	-1	2	(1)	0
	(14)	10	-1	(1.5)	1.5	0
2	14	(-4)	0	1	(.5)	0
	(14)	-4	0	(1)	.5	0
1	14	(+32-18=14)	-1	1	(-.5)	0
	(14)	14	-1	(.5)	0	0
	14	0	0	0	0	0

(continued)

Table 9. Continued.

$\theta/32$ (rad.)	$C_n/14$	$R_s/224$	Δz_c	$S_n/14$	$R_c/224$	Δz_s
B: C_n/S_n Stabilization Before Reversal						
0	14	0	0	0	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
2	14	(-32+28=-4)	1	9	(0)	0
	(14)	-4	1	(.5)	.5	0
2	14	(-4)	0	1	(.5)	0
	(14)	-4	0	(1)	.5	0
3	14	(10)	0	1	(1.5)	0
	(14)	10	0	(1)	1.5	0
4	14	(-32+24=-8)	1	1	(2.5)	0
	(14)	-8	1	(1.5)	3	0
4	14	(-8)	0	2	(3)	0
	(14)	-8	0	(2)	3	0
3	14	(32-22=10)	-1	2	(1)	0
	(14)	10	-1	(1.5)	1.5	0
3	14	(10)	0	1	(1.5)	0
	(14)	10	0	(1)	1.5	0
2	14	(-4)	0	1	(.5)	0
	(14)	-4	0	(1)	.5	0
1	14	(32-18-14)	-1	1	(-.5)	0
	(14)	14	-1	(.5)	0	0
1	14	(14)	0	0	(0)	0
	(14)	14	0	(0)	0	0
0	14	(0)	0	0	(0)	0

BIBLIOGRAPHY

1. Callan, J. D. MTOS Integrated Digital Differential Analyzer. General Instrument Corporation, New York, September, 1968. (Microelectronics Application Notes)

2. Forbes, George F. Digital Differential Analyzers. 4th ed. G. F. Forbes, Pacolma, Calif., 1957. 175 p.

3. Gilbert, C. P. The Design and Use of Electronic Analog Computers. Chapman and Hall, London, 1964. 288 p.

4. Kella, J. and Shanni, A. A Note on the Accuracy of Digital Differential Analyzers. IEEE Transactions on Electronic Computers EC-16:230. April, 1967.

5. MacNee, A. B. Some Limitations on the Accuracy of Differential Analyzers. Proceedings of the IRE 40:303-308. March, 1952.

6. Mayorov, F. V. Electronic Digital Integrating Computers: Digital Differential Analyzers. London Iliffe, London, 1964. 273 p.

7. McGhee, Robert B. and Nilsen, Ragner H. The Extended Resolution Digital Differential Analyzer: A New Computing Structure for Solving Differential Equations. IEEE Transactions on Computers C-19:1-9. January, 1970.

8. Nelson, D. J. DDA Error Analysis Using Sample Data Techniques. 1962 Spring Joint Computer Conference, AFIPS Proceedings 21:365-371.

9. Palevsky, M. The Design of the Bendix DDA. Proceedings of the IRE 43:1352-1365. Oct., 1953.

10. Sizer, T. R. H. The Digital Differential Analyzer. Chapman Hall, London, 1968. 201 p.

11. Owen, P. L., Partridge, D. W., and Sizer, T. R. H. The Digital Analyzer and its Realization in Digital Form, Part I. Electronic Engineer 32:614-617. Oct., 1960.

12. Owen, P.L., Partridge, D.W., and Sizer, T.R.H. The Digital Analyzer and its Realization in Digital Form, Part II. Electronic Engineer 32:700-704. Nov., 1960.

13. Turtle, Q.C. Incremental Computer Error Analysis. IEEE Transactions on Communications and Electronics 82:492-498. Sept., 1963.

APPENDIX

APPENDIX

System Control Logic

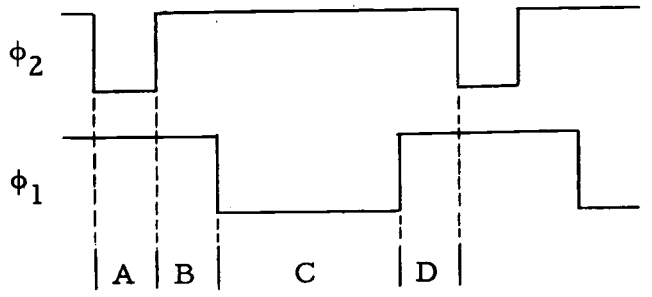
To provide the necessary auxillary commands required for proper DDA operation some control logic is required. The following discussion is generally applicable to all the integration systems discussed in the text. Any features unique to a particular integration technique are so identified. It should be noted that the addition of more integrators to the system would require only minor modification of the control logic if any.

The control functions will be generated using TTL logic and level shift gates. The faster TTL logic (as opposed to MOS) provides a minimum propagation delay in generation of the necessary command functions.

Figure A.1 defines the restrictions placed on the two phase clock required by the DDA integrator components.

Based on these clock restrictions and not wishing to push the limits of operation, a clock frequency of 400 KHz was selected.

Initially some basic clock will be required from which the required two phase clock can be derived. The clock may be a crystal or L-C oscillator, the selection being largely a function of cost, desired stability and the operating environment.



A - .3 μsec minimum, no upper bound
B, D - .3 μsec minimum, .5 μsec maximum
C - 1 μsec minimum, 10 μsec maximum

Figure A.1. Two phase clock restrictions.

The author has elected to use a 4.8 MHz oscillator. The oscillator output (Figure A.2) is shaped by passing it through a TTL gate. To assure symmetry of the input clock frequency a TTL flip-flop is used. This results in division of the clock frequency by two but compensates for any deviations from symmetry produced by the finite threshold level of the squaring gate. Deviations from symmetry are of no significance to the system in question and might, therefore, be ignored. However, it is in general good practice to start with a symmetrical waveform.

The two phase 400 KHz clock is generated by dividing the input 2.4 MHz by six. It is accomplished by using a six state Grey code counter (flip-flops A, B, and D of Figure A.2). The Grey code counter is selected because of its inherent freedom from undesired transients. (The Hamming distance between adjacent words is one.)

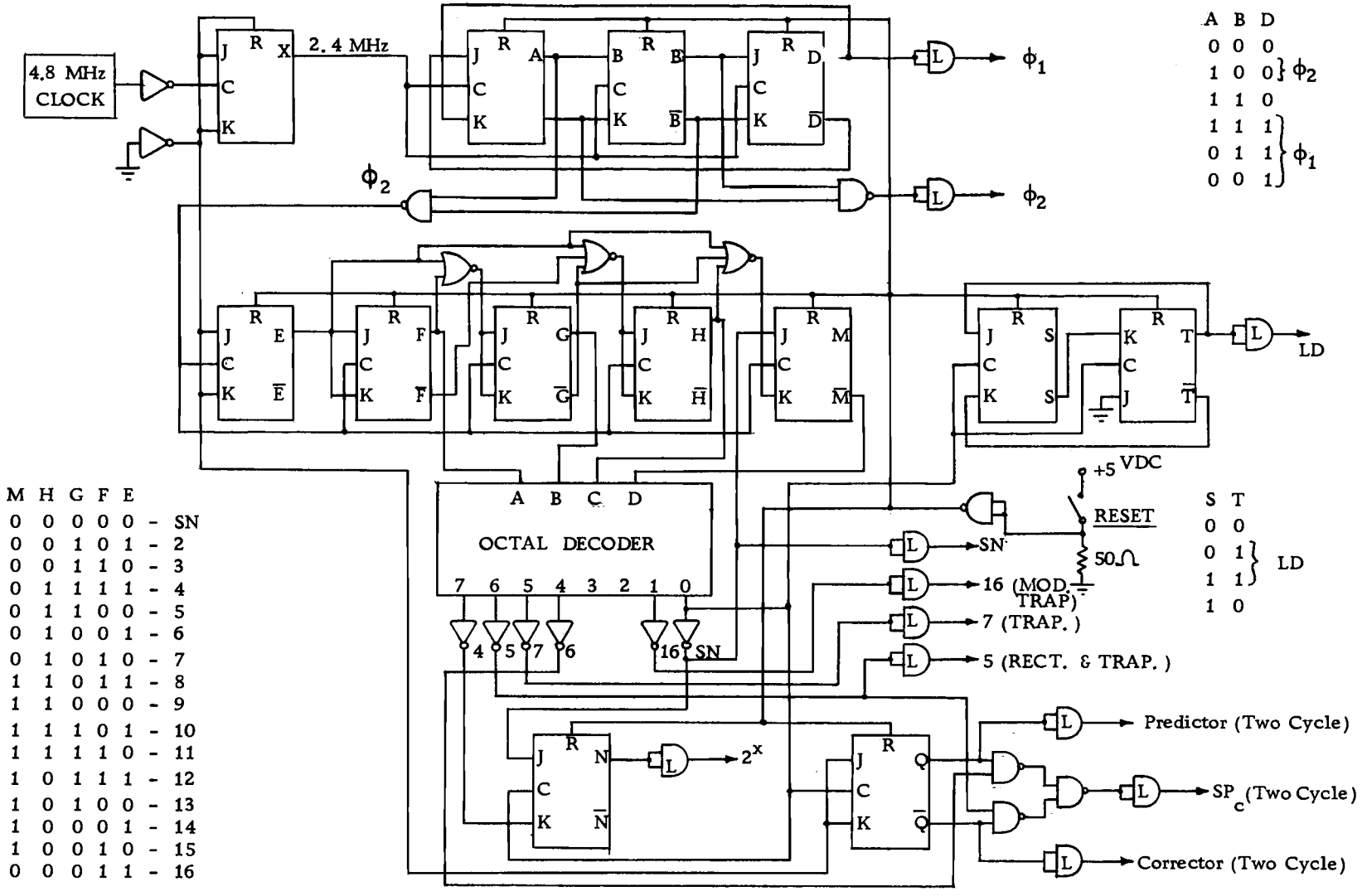


Figure A. 2. System control logic diagram.

The six states each exist for .4167 μsec. Using one state each for the times noted A, B, and D of Figure A.1 and three states for C results in a two phase clock which satisfies all the requirements.

In DDA systems it is necessary to uniquely identify some or all of the register bits (scale and sign bits). Since mechanization of the MOS shift registers is such that their outputs change on the leading edge of the ϕ_2 clock, it is desirable that any identifying bits be synchronous with the leading edge of the ϕ_2 clock as well. This may be accomplished by using a sixteen state Grey code counter (Figure A.2, flip-flops E, F, G, H, M) synchronized to the leading edge of ϕ_2 and determining a particular bit by proper decoding of the counter state. It may be observed in Figure A.2 that bits denoted SN, 4, 5, 6, 7, and 16 have been extracted.

Two cycle integration requires cycle identification in addition to bit identification. Since in the DDA system the sign bit is used to identify the end of one word and the beginning of the next, a flip-flop (Figure A.2, flip-flop Q) triggered off the leading edge of the sign bit may be used for this purpose. Scale bit variation as required in two cycle integration is accomplished by gating the desired scale bit with the appropriate cycle state.

Some logic is always required to generate any non-zero initial conditions required by the system. The proper initial condition in the example used throughout the text is that $\cos\theta$ be 0.1110000 0000 0000

in digital form. The generation of this function is readily accomplished by using a J-K flip-flop triggered off the leading edge of the ϕ_2 clock with the J and K inputs connected to bits SN and 4 respectively (Figure A. 2, flip-flop N).

To load the initial data it is necessary to have a load command which exists for at least the time required for presentation of the non-zero information to the shift registers. It is generally convenient to provide a load command which exists for some number of iteration periods and is synchronized to them. In Figure A. 2 (flip-flops S and T) a four state latching counter is used to provide a load command of two iteration periods. The load command is initiated by a system reset command and after three active iteration periods latched in an inactive state until the system is again reset. A two state counter (flip-flop) would be sufficient for rectangular or trapezoidal integration.