UNIVERSITE LIBRE DE BRUXELLES
Faculté des Sciences Appliquées
Laboratoires d'Electronique industrielle et d'Automatique

THE CONCEPT AND DESIGN OF INCREMENTAL COMPUTERS

ERRORS OF COMPUTATION

Thèse de doctorat

Hassan SEIEDRAZI
Avril 1969

UNIVERSITE LIBRE DE BRUXELLES
Faculté des Sciences Appliquées

Laboratoires d'Electronique industrielle et d'Automatique

THE CONCEPT AND DESIGN OF INCREMENTAL COMPUTERS

ERRORS OF COMPUTATION

Thèse de doctorat

Hassan SEIEDRAZI
Avril 1969

Thèse présentée à la Faculté des Sciences appliquées de l'Université Libre de Bruxelles pour l'obtention du grade de Docteur en Sciences Appliquées

Avril 1969.

Acknowledgment

It is my pleasant duty to express my thanks to Professor J. Hoffmann for welcoming me into his department, to do my doctorate degree.

The author owes his interest in automatic control to Professor Peretz, whose interest contributed greatly to the performance of my research work.

The author is greatful for the guidance, encouragment, active interest and participation of Professor J.Florine in the realization of this thesis. The new type of incremental computer designed by the author has been developed and constructed at the Industrial Electronics Laboratory of the " Université Libre de Bruxelles ", directed by Professor J. Florine.

The author benefitted from the intimate cooperation and understanding of the laboratory's engineers, particularly Miss Anne Leussler and Mr. Michel de Hagen.

The author thanks the secretary, Miss Michèle Ducci, the technicians, Mr. R. Meyer, Mr. C. Hubert for their contribution in constructing the computers.

I am indebted to the " Ministre de l'Education Nationale et de la Culture de Belgique, section Relations Culturelles et Internationales " for their assistance and facilities which they put at my disposal in order to present this doctorate thesis.

Brief outline of thesis topics.

The research work carried out since september 1965 in the laboratories of industrial electronics and automatic control of Brussels University, has had as its goal to investigate the new techniques of hybrid and incremental computation for modern control system engineering. This has led to the present doctorate thesis in the applied sciences.

The first step of this investigation was the study of existing digital and analog computation, especially the hybrid computation and the design of electronic transistor computers.

The second step and the main aim was the study of a new method of incremental computation in automatic control. This investigation led us to elaborate "multiple incremental" computation which has the advantages of speed, versatility and flexibility over the unitary increment computation which is the base of Digital Differential Analyzers.

The incremental computation was first employed in Digital Differential Analizers (D.D.A), for integral operation. The veiwpoint purposed here is that a D.D.A. is a special member of the more general class of machines which are known as incremental computers. The essential difference between an incremental computer and a digital computer is that, an incremental machine accompishes information transfers between storage cells on a fractional word rather than a whole word basis.

The first D.D.A. was built in the U.S.A in January 1950. In this machine every calculation was refered to integration, and the unitary increment could have only two fixed value +1 and -1. In the later "Ternary" machine, improvement was obtained by increasing the number of possible value to +1, O, -1. This machine was used primarally for scientific and technical calculation associated with solution of systems of differential equations.

The limitation of fixed increment (±1,O) of D.D.A., led to the development of an incremental computer which could have five fixed values ±I ±32, O, that was suggested by S. Shackell and J.A. Tryon. With his method the initial solution of a new problem was delivered with the reasonable promptness so that the changes in variables were processed in each computation cycle, since the computer must have been move promptly from whatever state it finds itself into the state demanded by the problem (the time required for such motion is known as slewing time).

Further development of incremental computation is the task of this thesis, the study of "multiple increment" computation that increments can have any desired value between $0, 2^0, 2^1, 2^2, 2^3 \ldots$ 2^h is the aim of this investigation. In this system the largest permissible increment is larger than any accepted change in any input, intermediate value or result and increments are expressed with a sufficient number of digits to flow any rapid and jump function.

The new type of transistor incremental computer which is designed and developed by the author in the Industrial electronic laboratories of Brussels University, performs the integration on the basis of unirary and multiple increment computation.

In addition, this machine is capable of doing all the basic mathematical operations and other combined operations.

The interconnection between the integrators, was realized normally by the stored programme or by patch panel with two lead for unitary incremental computation. We developed a new algorithm which permit to interconect the integrators by only one lead on the patch panel.

The computation time for a integral operation in the general purpose digital computer is about 50 ms (with clock frequency of 1 M hertz) in unitary incremental computer, (with 500 k hertz clock frequency) is 40Γ sec, so the speed of this machine is 2500 higher than general purpose. Still the multiple increment computer increases the speed of integration by 2^4 (same clock frequency), so the speed of integration is increased by $2500 \cdot 2^4 = 40.000$ compared to general purpose digital computer. Morever, when the incremental computer are provided with multiple increments, the slewing time is reduced at the price of equipment, the result has the advantage of very high speed computation high dynamic quality in automatic control, and very good capacity for repetitive calculation upon continuous quantities.

Because the incremental computers work on discrete values of the variation of a function at particular instants of time, they are associated with the error of computation. This study of error in incremental computation is an important factor in the performance of the computer and the choice of algorithms of computation in most economical and convenient way.

The error analysis of D.D.A. has been done by some authors (among them are particulary D.E. Skabelwnd of the university of Utah U.S.A., F.B. Hill at M.I.T. and O. Hange in Germany).

To our best knowledge, they claculated only the error of method and round off, but they did not deal with the quantization and transmission error. Moreover, their computation were applied only in the particular and simplest case of unitary incremental computation (D.D.A).

The viewpoint of our investigation is to present new effective methods of calculating all the errors (method, round off, quantization and transmission) in the general form of multiple incremental computation. The "unitary increment" becomes then a particular case of the general theory .

This permits us to compare the various errors in both types of incremental computation which is necessary in order to choose the algorithms of machine in the most economical and convenient way with the desired accuracy in relults.

The calculation of error of method in the integration process, lead us to choose the most convenient quality and degree of approximation for unitary and multiple increment computation.

Computing the quantization error for different methods of integration in unitary and multiple increment computation and the way of minimizing them, gives the idea of the choice of the register's length and the speed of incremental computer.

The study of round off and transmission error shows the way of minimizing them in incremental computation.

The calculation of the total error enabled us to compare the error for different methods of integration in unitary and multiple increment computation.

From the comparaison of these errors, we deduced the choice of algorithm for unitary and multiple increment computer, with the desired accuracy.

Regarding application of these computations, the incremental computers with unitary increments as D.D.A. has the advantage of high computation speed, very good capacity for repetitive calculation, small volume, low wight and reliability compared to general purpose digital computers. This design is suitable for real time control problem e, g, in control of industrial process autopilot and guidance systems.

However the unitary incremental computer cannot be used where fast slewing is required. This eliminates any problem in which it must produce results immediatly after the first datas are applied.

When the incremental computer is provided with multiple increment so that slewing time is reduced at the price of more equipment, the method has of course the same advantages as D.D.A. and in addition it posses short slewing time. The design is therefore appropriate where both high computation frequency and short slewing time are needed for dynamic response. Suitable problem appears in the problems of Direct Digital Control (D.D.C), optimization and simulation of automatic control, missiles aerodynamics; navigation and aviation.

CONTENT

CHAPTER I

CONCEPT OF INCREMENTAL COMPUTERS.

Introduction.

The current tendency in control field towards high degree of accuracy, reliability, as well as decision making and compatibility, has placed emphasis on the digital techniques. The increasing size and complexity of control systems, necessarily involves, the development of digital automatic control systems. In this aim, the general purpose digital computers played the principal role in the first stage.

In parallel with the development of the general purpose, went the development of many special purpose techniques, which were found useful for implementation of specialized devices for control computation and information processing.

Automatic process control is now a well established discipline encompassing a variety of techniques and methods, and in which computer techniques are of increasing importance. Direct Digital Control (D.D.C.) is opening new possibilities of accuracy quality and economy of automatic process control.

Computer techniques offer practically unlimited possibilities for accuracy, speed and sophistication of integrated control systems.

In view of requirement of modern control systems, the speed of general purpose computer is completely insufficient for real time computation. Even an extremely large general purpose computer cannot handle the computation necessary for real time control systems. The time used in setting up and programming a problem may amount to weeks or even months. For most practical applications, where the problem is solved in accordance with a previously prepared program, a general purpose computer is not neccessary. In fact, in terms of the stated problem, such a computer is unnecessarily, complex and relatively inefficient. Large computers should be built only for large computing centers in which effective use of such computers is possible. Thus, technical and economical factors dictate the use of simpler, more reliable, economical and compact special purpose digital computers, for use in many applications.

During the last years, a new type of computer, based on the principle of digital integration, has been found increasingly wider application. Such computers, combining the advantages of digital and analog machines, were first refered to digital differential analyzers (D.D.A.). Further development of digital differential analyzers evolved a class of incremental computers, based on the principle of summation of increment. There are two types of incremental computers; the former is the incremental computer with unitary and fixed increments $\Delta x = \pm 1$ and $\Delta y = \pm 1$, which includes the digital differential analyzers (D.D.A.), the latter, which is a development of (D.D.A.), is new generation of incremental computer with multuple increments. (The increments in the computation may take the multiple quantities of $\pm 2^0$, $\pm 2^1$, $\pm 2^2$, ... $\pm 2^h$).

The high computing speed and operating efficiency of the incremental computer result from the fact that :

a) the computer operates with increments of input quantities and not with the quantities themselves, as it is the case in the general purpose computers. This permits considerable increases in computing speed and in switching integrators,

b) due to the use of multiple increments, the speed of integration in increment computer is multiplied by the factor 2^h compared with the unitary increment (h = 1) which is used in Digital Differential Analyzer (D.D.A.),

c) by using integration as a basic operation, operations of integration, differentiation, multiplications, divisions, extractions of a rooth, logarithm calculations, and so on, take a time equivalent to two or three times of addition operation. This time is much smaller than in a general purpose computer,

d) there is no need to store the operation codes and memory addresses in the internal memory of increment computer for use of integration, differentiation, multiplication and division of functions.

Consequently, the solution of a relatively complex problem in the increment computers does not require an internal memory of large capacity,

e) with the increase in complexity and nonlinearity of the problem, the increment computer becomes an even more effective machine, because the amount of equipment does not increase in proportion to the complexity of the problem. The accuracy of increment computer does not decrease with an increase in complexity of the problem,

f) in addition the increment computer realizes the basic mathematical operation, as addition, substraction and multiplication of several values in one time of addition. It also performs the other basic mathematical and logical operations as general purpose computer.

Therefore the incremental computers are much more rapid, economical, compact and efficient than general purpose digital computer, and they have the advantage of both analog and digital computers.

However, because of the discrete nature of incremental computer's operations, an incremental computer realizes the approximated value of integration and not the original one.

In chapter one, we are going to explain the principal of increment computers and their operation.

In chapter two, the different algorithms of integration and their errors are calculated in a general case for unitary or multiple incremental computation, when the independent variable of integral X is equal to, or is a function of the independent variable t of the machine.

In chapter three, we will study the quantization process and the quantization error in unitary or multiple incremental computation, when the independent variable of integral X is equal to or is a function of the independent variable t.

In chapter four and five, we calculate the round off error, transmission error, the total error and the way of their minimization by the appropriate choice of algorithms which are applied to the machine.

In chapter six, we explain the design, development and construction of a new generation of incremental computers, that the author have developed in the industrial electronics laboratory of the University of Brussels.

1.1 The basic operation of a new type of incremental Computers.

The need for simple, compact digital computers suitable for solving differential equations, automatic control simulation and optimization led to the development of a special type of control computer which is called the incremental computer.

In ordinary computation, the function must be evaluated anew for each value required. This computation method conducts to complex and time-consuming procedures.

Another approach is to compute just the increment of the function from a1 evaluation to the next. Two characteristics of this approach are :

The value of the increment between successive evaluations are smaller than the values of the function itself.

The variation of a function is simpler than the function itself.

These characteristics make possible some very simple computers, in terms of hardware and logic. Any function can be determined by its initial value and its variation in time, which is called the increment of function.

For instance, the function y (x), can be determined by its initial condition (x_0, y_0) and its increments δx and δy.

As it is seen from figure (1 - 1) the function y (x) can be completely determined in time as :

In initial condition x_0 y_0

" Instant $x_1 = x_0 + \delta_1 x$ $y_1 = y_0 + \delta_1 y$

" " $x_2 = x_1 + \delta_2 x$ $y_2 = y_1 + \delta_2 y$

" " $x_i = x_{i-1} + \delta_i x$ $y_i = y_{i-1} + \delta_i y$

As incremental computers are digital machines, instead of using the initial values (x_0, y_0), and the increments $\delta_i x, \delta_i y$, they use their quantized values (x_{0Q}, y_{0Q}) and $\delta_{iQ}x$, $\delta_{iQ}y$, as it is shown in figure (1 - 2).

Therefore there is an error ε_{iQx}, ε_{iQy} between the original values of function y_i (x) and its approximated quantized values y_{iQ} which is defined as :

$$\begin{cases} \varepsilon_{iQx} = x_i - x_{iQ} \\ \varepsilon_{iQy} = y_i - y_{iQ} \end{cases} \qquad (1 - 1)$$

The quantized function y_{iQ} (x) is determined with the initial values (x_{oQ}, y_{oQ}) and the quantized increment $\delta_{iQ}x$, $\delta_{iQ}y$:

$$\begin{cases} x_{iQ} = x_{(i-1)Q} + \delta_{iQ}x \\ y_{iQ} = y_{(i-1)Q} + \delta_{iQ}y \end{cases} \qquad (1 - 2)$$

There are two kinds of incremental Computers : unitary increment computer and multiple increment computer.

The unitary increment computers includes the Digital

Differntial Analyzer (D.D.A.). In this kind of computers, the increments $\Delta_i x$ and $\Delta_i y$ are fixed and limited to ± 1 or 0. So it is not possible to treat functions which varies rapidly in time, because the function y (x) can only change by the quantum Δy for each interval Δx. Fig. (1 - 3).

The multiple increment computers are a new type of machines which operate on multiple or variable increments $\delta_{in} x$, $\delta_{in} y$. So it is possible to treat functions which varies rapidly in time, because the function y (x) can change by the $\delta y = 2^r \cdot \Delta y$ for each interval $\delta x = 2^r \cdot \Delta x$, ($h > r > 0$). Fig. (1 - 4).

Therefore these kind of computation have a great advantages over the unitary increment computers because of their flexibility and ability to operate with any rapid function.

One of the principal operation of incremental computers is the integration, which can be down with unitary or multiple increments.

The integral operation by unitary incremental computation is the basic operation of digital differential Analyzer (D.D.A.). In this case, the step of integration is the quantum Δx, which can have the logical value ± 1 or 0.

Therefore the approximated value of integral $S_Q^*(t)$ is :

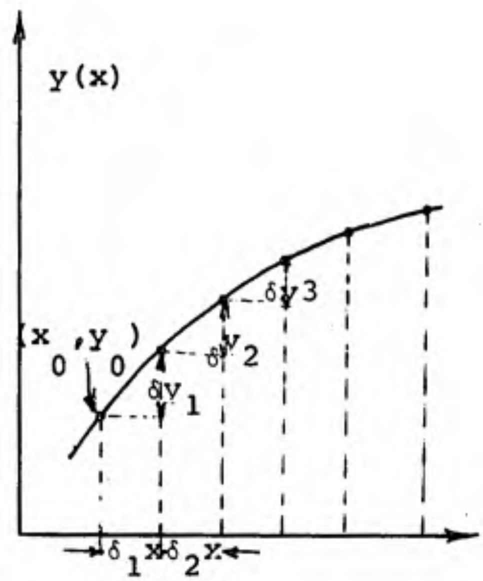


Fig 1-1

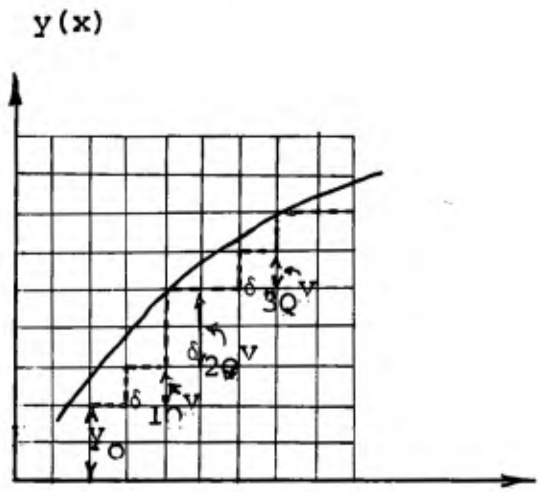


Fig 1-2

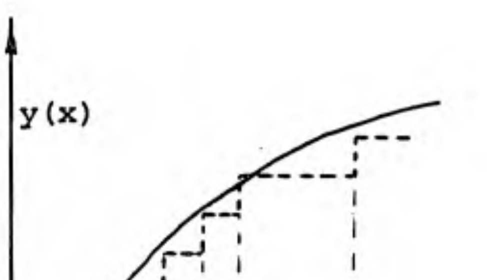


Fig1 -3

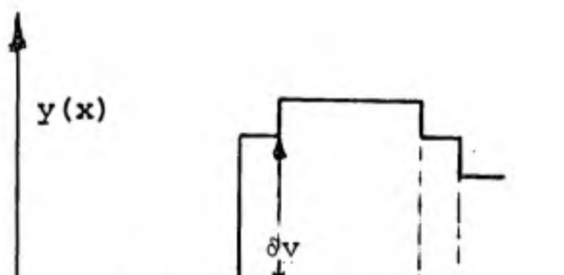


Fig 1-4

$$S^{*}_{Q}(t) = \sum_{i=1}^{k} y_{i\ \acute{e}q.} \cdot \Delta_{iQ}\, x \qquad (1 - 3)$$

Where

$$\Delta x = \pm 1 \text{ or } 0$$

$$y_{\acute{e}q.} = f\ [(\delta_{iQ}x,\ \delta_{iQ}y),\ (\delta_{(i-1)Q}x,\ \delta_{(i-1)Q}y), \ldots$$

The value of $y_{i\ \acute{e}q.}$ is chocsen in such a way that when multiplied by $\Delta_{iQ}x$, it give the integral of the function in interval x (x_i, x_{i+1}) with any desired accuracy, (fig. 1.5.).

The new method of integral operation is based on the principal of multiple increment Computation. In this case instead of using small step of integration equal to the quantum Δx, we use a large step $\delta x = 2^r \cdot \Delta x$. Therefore the speed of integration will increase by the factor 2^r, compared to D.D.A. The integral function $\delta_{iQ}S^{*}$ in interval x (x_i, x_{i+1}) will be :

$$\delta_{iQ}S^{*} = y_{i\ \acute{e}q.} \cdot \delta_{iQ}x$$

Where

$$\begin{cases} \delta_{iQ}x = 2^r \cdot \Delta_{iQ}\, x \\ \delta_{iQ}y = 2^r \cdot \Delta_{iQ}\, y \\ \delta_{iQ} = 2^r \cdot \Delta_{iQ}\, S \\ y_{i\ \acute{e}q.} = f\ [\delta_{iQ}x,\ \delta_{iQ}y,\ \delta_{(i-1)Q}x,\ \delta_{(i-1)Q}y, \ldots] \end{cases}$$

The value of $y_{i.\ éq.}$ is choosen in such a way that, when multiplied by $\delta_{iQ}x$, it gives the integral of the function in interval $x \in (x_i, x_{i+1})$ with any desired accuracy, Fig. (1 - 6). We will study later on, the value of $y_{i.\ éq.}$ and the approximate function of integral in more detail.

The multiple increment computation, has the great advantages of speed, versatility and flexibility over the unitary increment computation. In following discussion we shall treat the general case : The multiple increment computation. The basic operation of D.D.A. is a special case of multiple increment for which $r = 0$

The arithmetic unit which realize the integration on the basis of multiple increment computation is shown in fig. (1 - 7).

The input increments are added in block II in order to find the value of function at each instant t_i as :

$$Y_{iQ} = Y_{oQ} + \sum_{i=1}^{\kappa} \delta_{iQ} Y$$

Block I, recieves the information ($\delta_{iQ}x$, $\delta_{(i-1)Q}x$, $\delta_{iQ}x$ ...) .($\delta_{iQ}y$, $\delta_{(i-1)Q}y$,... $\delta_{iQ}y$) , and according to the choosen algorithm of machine gives the value of $y_{éq}$, and transfer to the $y_{éq}$ register . After multiplication by the step of integral $\delta_{iQ}x$, the result is added to the rest of integral $S_{0(i-1)}$ (from former iteration i-1) and transfered to the S register. I f n and h are the number of bits in $y_{éq}$ and δx register, then the number of bits in S register will be n + h . The most significant bits of S register, from n + 1 to n + h are taken as the approximated rounded off increment of integral δS^{*}_{iQM} t).

The output $\delta S^{*}_{QM}(t)$ transmitted to the input δ x and δ y of other integrators, or is memorized in the increment memory. The rest of integral $S_{0(i)}$ which is in the S register (bits 1 to n) is memorized in the computer memory and will be used in next iteration.

This operation is shown by the following equation.

(1 - 5)

iteration i $\quad S_{0(i-1)} + y_{i\ éq.} \cdot \delta_{iQ} \cdot x = \delta_{iQM} S^{*} + S_{0i}$

iteration i+1 $\quad S_{0i} + y_{(i+1)éq} \cdot \delta_{(i+1)Q}{}^{x} = \delta_{(i+1)QM} S^{*} + S_{0(i+1)}$

By this method, the integral operation is done on the basic of multiple incremental computation, with the input quantities $\delta_{iQ}{}^{x}$, $\delta_{(i-1)Q}{}^{x}$ .... , $\delta_{iQ}{}^{y}$, $\delta_{(i-1)Q}{}^{y}$, ..., and the output is the increment of integral $\delta_{iQ} S^{*}(t)$.

Fig 1-5

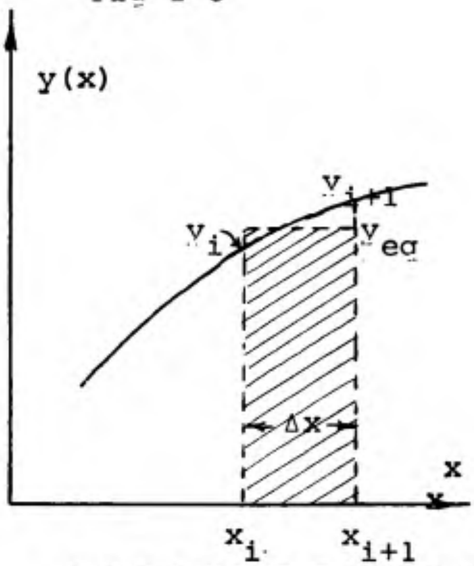
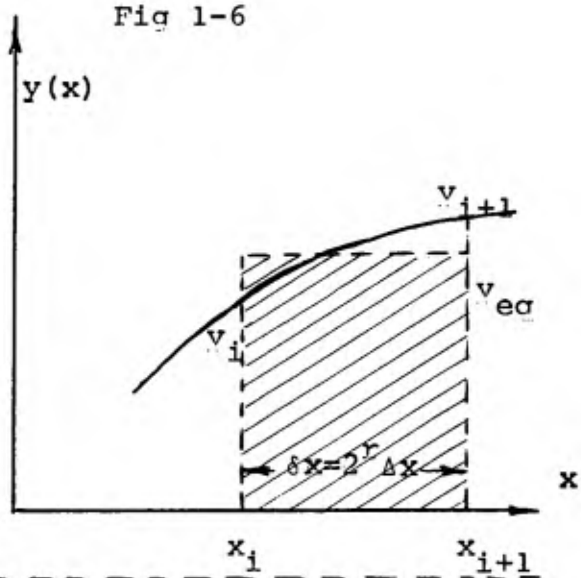


Fig 1-6



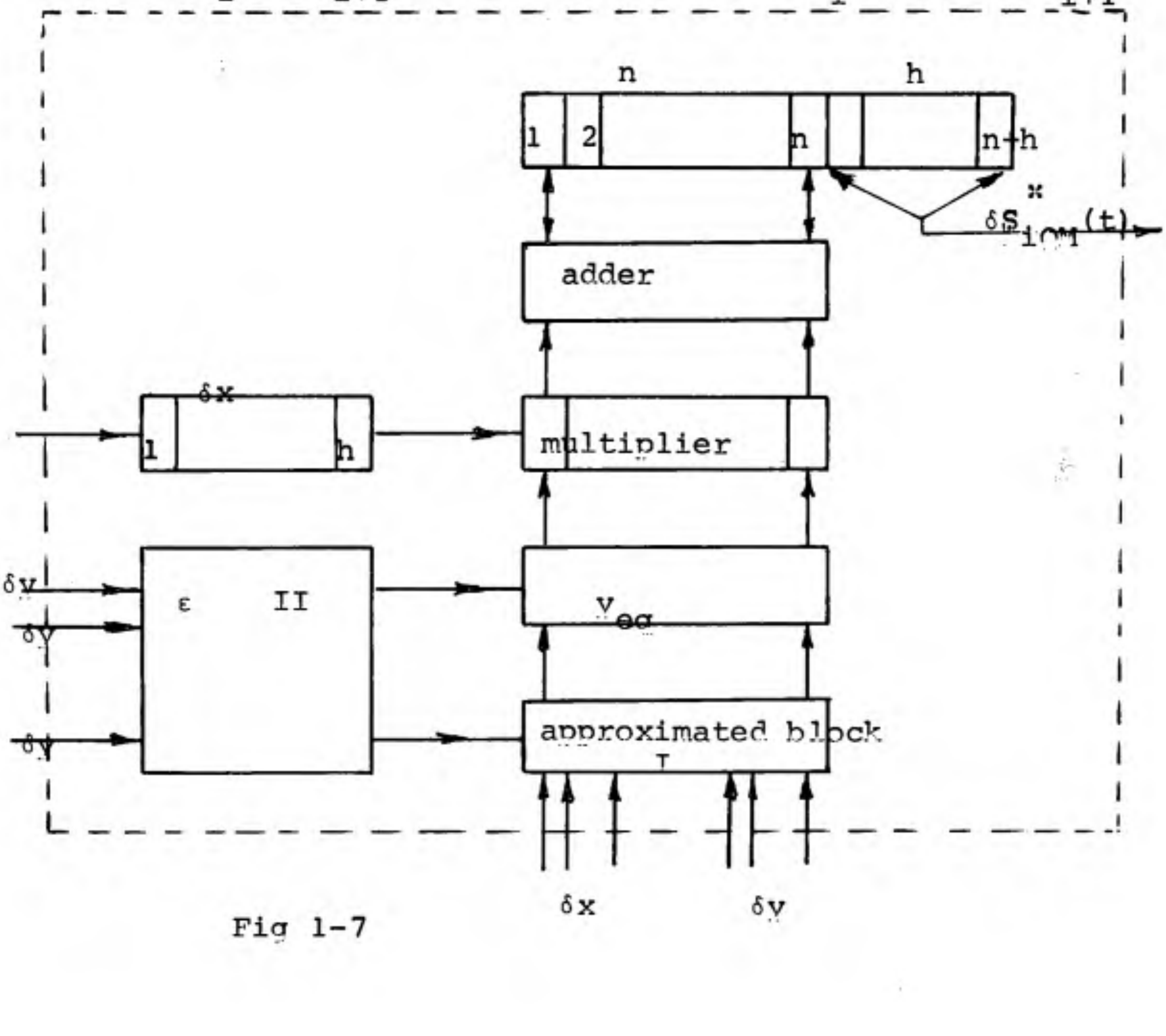


Fig 1-7

The organization of the new serial type of incremental computer is as following :

1 - Arithmetic unit

2 - Memory unit

3 - Control unit

4 - Programing unit

5 - Input output unit

The arithmetic unit of this incremental computer operate the integral operation on the basis of unitary or multiple incremental computation. By a new method it performs the multiplication of two functions with higher accuracy. This unit also performs the basic mathematical operations as addition, substraction, multiplication, etc ... in one time of addition and is also capable of decision making.

There are two memories for memorization of the values of the function y_{iQ} ; the rest of integral S_{0i} or other intermediate results of computation.

The control unit gives all the control pulses for arithmetic unit, memory unit, programming unit and input output unit.

The new method of programming of incremental computer is the patch panel using only one lead for transmitting the information between the integrators in unitary or multiple increment

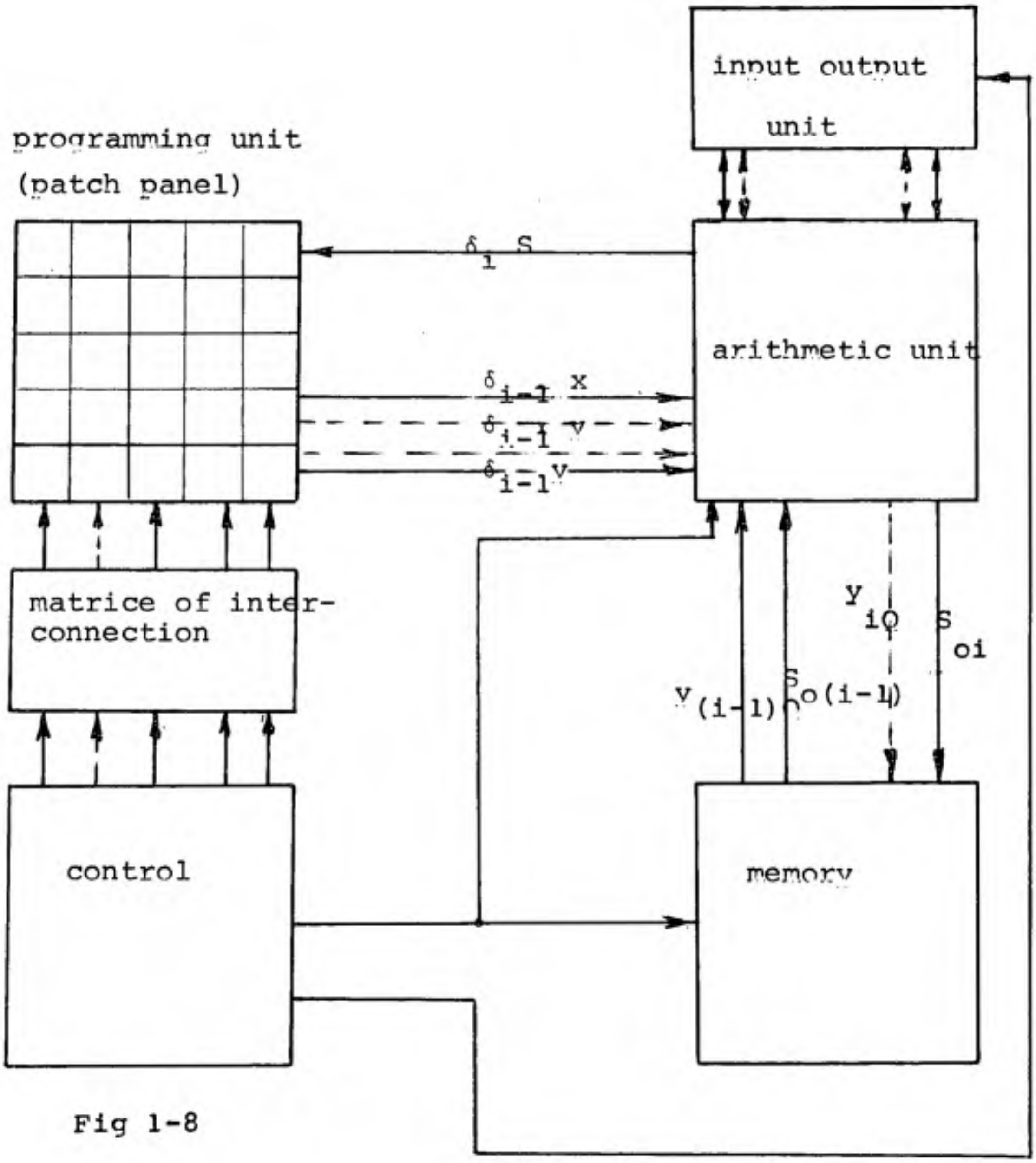

input output
unit
programming unit
(patch panel)
$\delta_i S$
arithmetic unit
$\delta_{i-1} x$
$\delta_{i-1} y$
$\delta_{i-1} y$
matrice of inter-
connection
y_{iO}
S_{oi}
$y_{(i-1)}$
$S_{o(i-1)}$
control
memory

Fig 1-8

computation. Therefore a problem can be programmed on the patch panel exactly as on the analog computer.

So the incremental Computer has the advantages of the analog computer for integration and simplicity of programming. It also has the advantages of the digital computer for accuracy, decision making, memorization and all the logical and basic mathematical operation.

This new type of machine has been devised by the author at the industrial electronics laboratory of Brussels University.

1.2. Data processing in incremental computers.

The incremental computers (IC) are one type of specialized control devices. This pertains to the computers in which the results of a given mathematical operation are transmitted for use in another mathematical operation by means of increments.

All the quantities and the transformations in incremental computers are merely increments of initial quantities, while at the completion stage of these transformations, the results quantities are obtained by the summing of the increments.

Therefore, any variable in incremental computer can be represented as the sum of increments.

$$\begin{cases} x = \sum\limits_{i=1}^{n} a_{xi} \cdot \delta_i x \\ y = \sum\limits_{i=1}^{n} a_{yi} \cdot \delta_i y \\ z = \sum\limits_{i=1}^{n} a_{zi} \cdot \delta_i z \end{cases} \qquad \text{Fig. } (1-6)$$

The δx, δy, ... δz, are the increments of the functions x, y,.. ... z and the coefficients a_{xi}, a_{yi}, ... a_{zi}, are the sequence orders of increments δx, δy, and δz, which depend on the functions x, y, ...z. The coefficients a_{xi}, a_{yi}, ... a_{zi} have one of the three values 0, ±1 that determines whether the increments $\delta_i x$, $\delta_i y$, ... $\delta_i z$ should be added (+1), substracted (-1), or ineffective, to the former value of x_{i-1}, y_{i-1}, ... z_{i-1}, to form x_i, y_i, ... z_i.

For example, the function y (x) which is replaced by the approximated interpolated quantized function $f_{iQy}(x)$ is represented by:

$$\begin{cases} y_{iQ}(x) = y_{(i-1)Q}(x) + a_{iy} \cdot \delta_{iQ}y \\ x_i = t_i = x_{(i-1)Q} + \delta_{iQ}x \end{cases} \tag{1-7}$$

or

$$\begin{cases} y_{iQ}(x) = y_{oQ} + \sum_{i=1}^{n} a_{iy} \cdot \delta_{iQ}y \\ x_i = t_i = x_o + \sum_{i=1}^{n} \delta_i x \end{cases} \tag{1-8}$$

$$a_{ix} = +1$$

The approximated quantized value y_{iQ} and the order sequences of increments a_{iy} are represented in figure (1.9).

In the same way any function x_{iQ}, y_{iQ}, ... z_{iQ} can be approximated by:

$$\begin{cases} x_{iQ} = x_o + \sum_{i=1}^{n} a_{ix} \cdot \delta_{iQ}x \\ y_{iQ} = y_o + \sum_{i=1}^{n} a_{iy} \cdot \delta_{iQ}y \end{cases} \tag{1-9}$$

$$z_{iQ} = y_o + \sum_{i=1}^{n} a_{iz} \cdot \delta_{iQ} z$$

So the value of each quantity at particular instant , is obtained by accumulating the individual increments generated by the system throughout the time of its operation, on the basis of separate increments arriving in time at the input of the system which has the inherent delay of τ with respect to the original continuous functions.

According to Shanon theory, any complex differential equation:

$$f_k \left[x, y_1, y_1', \ldots\ldots y_1^{a_{k1}}, y_2, y_2', \ldots\ldots y_2^{a_{k2}}, \ldots\ldots y_i', \ldots y_i^{a_{ki}}, \ldots\ldots y_e, y_e', \ldots\ldots y_e'^{a_{kl}} \right] = 0 \qquad (1\text{-}10)$$

can be solved by the (IC), provided, it can be transformed to the following equation.

$$\frac{dy_k}{dy} = \sum_{i,j=0}^{n} a_{ijk} \cdot y_i \frac{dy_j}{dy} \qquad (1\text{-}11)$$

$$k = 2,3,\ldots n$$

where $y_o = 1$ (introduced to make notation compact), y_1 is the independent variable and $y_2, y_3, \ldots\ldots y_n$ are the dependent variables.

The equation (1-11) can be written in the following form:

$$dy_k = \sum_{i,j=0}^{n} a_{ijk} \cdot y_i \cdot dy_j$$

$$\begin{cases} y_i = f\,(x,\ y_1,\ y_2,\ \ldots\ldots\ y_1',\ y_2',\ \ldots\ldots\ y_1^{(k)},\ldots,y_2^{(k)},\ldots) \\ y_1 = x \qquad\qquad k = 2,3,\ldots\ldots n \\ y_o = 1 \qquad\qquad (k \text{ is the number of integrator}) \end{cases} \tag{1-12}$$

In order to solve the equations (1-12) by the (IC), we should transform it to the input and output quantity of the (IC). With this aim, it is assumed:

$$\begin{cases} a_{ijk} \cdot y_i \cdot dy_j = d\xi_{ijk} \\ d\xi_{ijk} = a_{ijk} \cdot dz_{m1} \\ dz_{m1} = y_m \cdot dy_1 \end{cases} \tag{1-13}$$

The first equation (1-13) can be written in three equations:

$$\begin{cases} dy_k = \sum_{j,i=0}^{n} d\xi_{ijk} \\ d\xi_{ijk} = a_{ijk} \cdot dz_{m1} \\ dz_{m1} = y_{eq} \cdot dy_1 \end{cases} \tag{1-14}$$

in this equation, $dz_{m1} = y_{eq} \cdot dy_1$ is the output of the integrator, (y_{eq} is the equivalent value of y). It is connected to the input of the integrator k by the programme matrice a_{ijk} ($a_{ijk} = 0$ or ± 1 is determined by the programme), and the equation $dy_k = \sum_{i,j=1}^{n} d\xi_{ijk}$

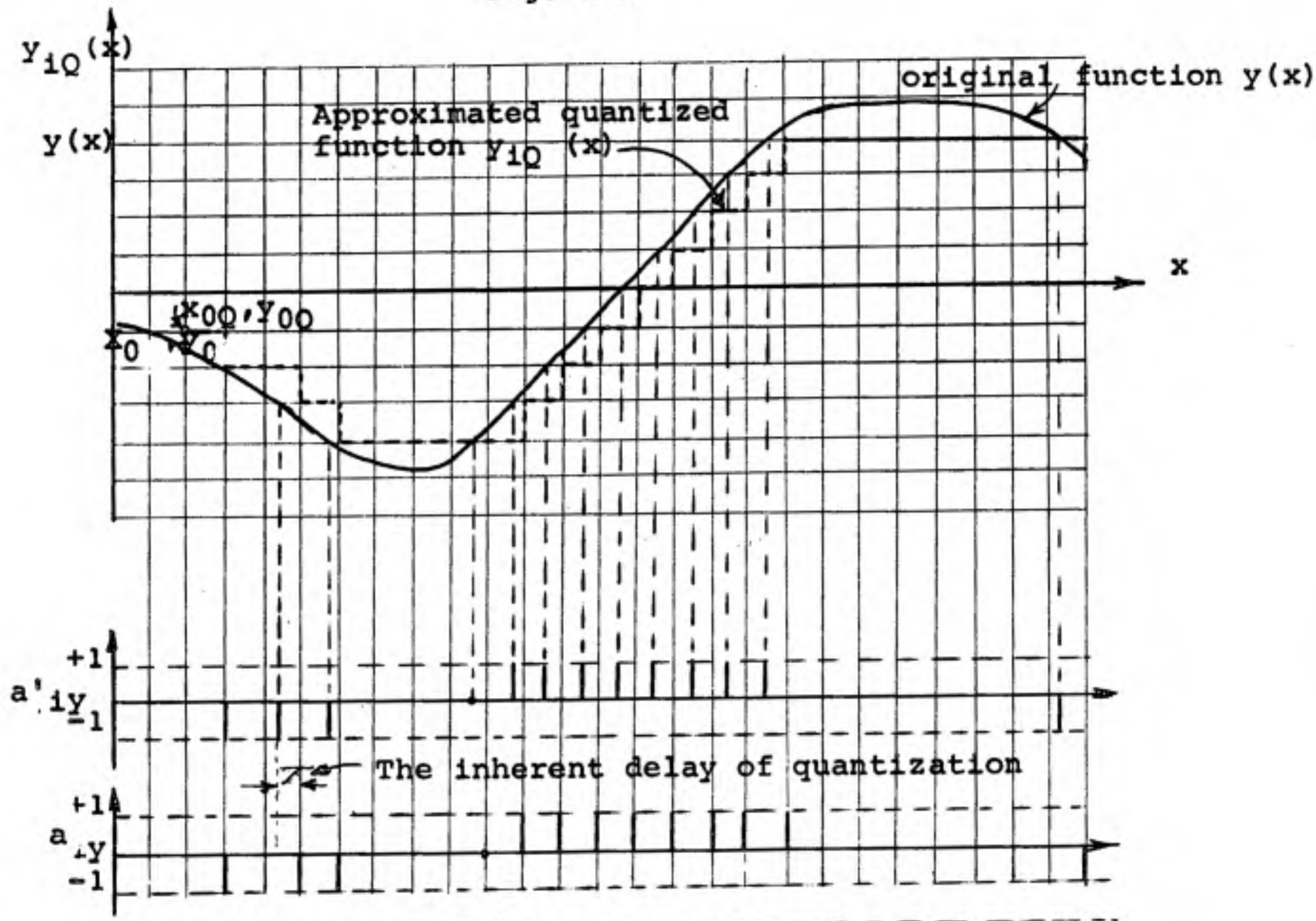
gives the input variable y_k of (IC), which is the sum of the outputs of the other integrators. So, the model of (IC), for solving the differential equation, is shown in figure (1.10).

As it is seen from the figure (1.10), the output of the other integrators dz_{ml} are connected to the input $dy_{1(k)}$, $dy_{2(k)}$, ... $dy_{j(k)}$ of the integrator (k), by the programme unit, which determine the matrice a_{ijk} for the interconnection between the integrators.

The values δy_k, $\delta \xi_{ijk}$, δz_{ml}, can be calculated by integrating the equation in interval $x \in (x_i, x_{i+1})$:

$$\left| \begin{aligned} \delta_i\, y_k &= \sum_{i,j=0}^{n} \int_{x_i}^{x_{i+1}} d\xi_{ijk} \\ \delta_{ijk}\xi &= \int_{x_i}^{x_{i+1}} d_{ijk}\xi \\ &= \int_{x_i}^{x_{i+1}} a_{ijk}\, dz_{ml} \\ \delta z_{ml} &= \int_{x_i}^{x_{i+1}} y_{eq} \cdot dy_l \end{aligned} \right. \qquad (1\text{-}15)$$

fig. 1-9



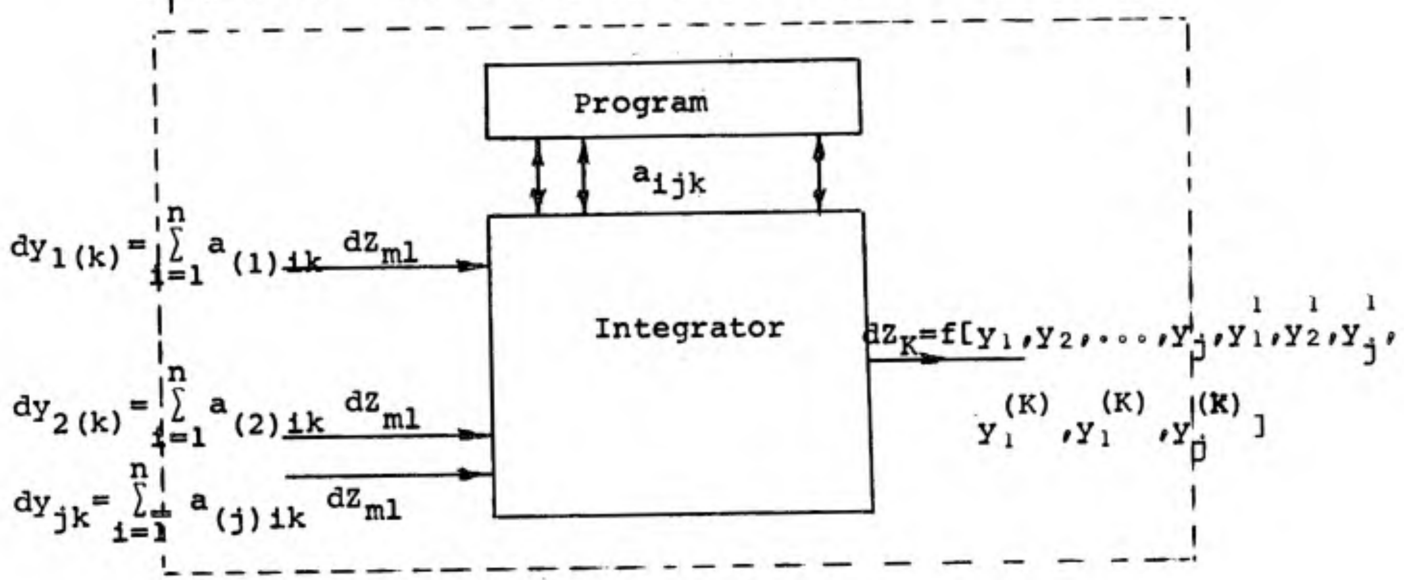


fig. 1-10

$$\left\lfloor y_1 = x \qquad\qquad k = 2,3,\ldots\ldots m\right.$$

In order to calculate the integral of input quantities dy_{jk}, it would be necessary to have the informations of $d\xi_{ijk}$, $a_{ijk}\,dz_{ml}$ and $y_{eq} \circ dy_1$ in i interval. But in practice, in the i^{th} iteration of (IG), the only informations which exist, are the informations of former iterations, 1, 2, ... (i-1), which are in the memory. Therefore, all the data have a delay of one machine cycle T, with respect to the quantized value of information. Of course, this delay is appeared in the transmission of data, to the inputs.

As it was seen, the quantization process produces an inherent delay τ with respect to the continuous function. Here it is shown that in the transmission of data, there will be an iteration delay time T in the input data of the integrator, compared with the quantized data which should be available in the i^{th} iteration. The total delay of data is $\tau + T$ or $T\,(1 + \frac{\tau}{T}) = T\,(1 + \lambda)$, by assuming $\lambda = \frac{\tau}{T}$.

Consequently, the informations that are available in the input of the integrators, from the above discussion, can be written as:

$$\begin{cases} (x_{oQ},\ \delta_{1Q}x,\ \delta_{2Q}x,\ \ldots\ ,\ \delta_{(i-\lambda-1)Q}x) \\ (y_{oQ},\ \delta_{1Q}y,\ \delta_{2Q}y,\ \ldots\ ,\ \delta_{(i-\lambda-1)Q}y) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ (W_{oQ},\ \delta_{1Q}W,\ \delta_{2Q}W,\ \ldots\ ,\ \delta_{(i-\lambda-1)Q}W) \end{cases} \qquad (1\text{-}16)$$

The block diagram of (IC) will be as in the figure (1.11).

Now we will explain the resolution of differential equations, by the incremental computer. As it was seen in order to solve the equation (1-10) by incremental computer, it should transform to the following equations:

$$\begin{cases} dy_{ik} = \sum_{i,j=0}^{n} d\xi_{ijk} \\ d\xi_{ijk} = a_{ijk} \cdot dz_{iml} \\ dz_{iml} = y_{im} \cdot dy_{il} \end{cases} \qquad (1\text{-}17)$$

The third equation (1-17), is the integrator action, the second one is the multiplier action by the constant coefficients a_{ijk}, and the first one are the summation of increment, which give the desired output dy_k. The equation (1-17) can be programmed on the incremental computers, as it is shown in figure (1.12).

In the figure (1.12), the incremental computer first operates all the integrations, then it operates all the multiplications, and at last it operates the summations.

But as it was discussed earlier, because of discret nature of the operation of the incremental computer, and the delay of T $(1+\lambda)$ which is introduced, there will be an error in each operation.

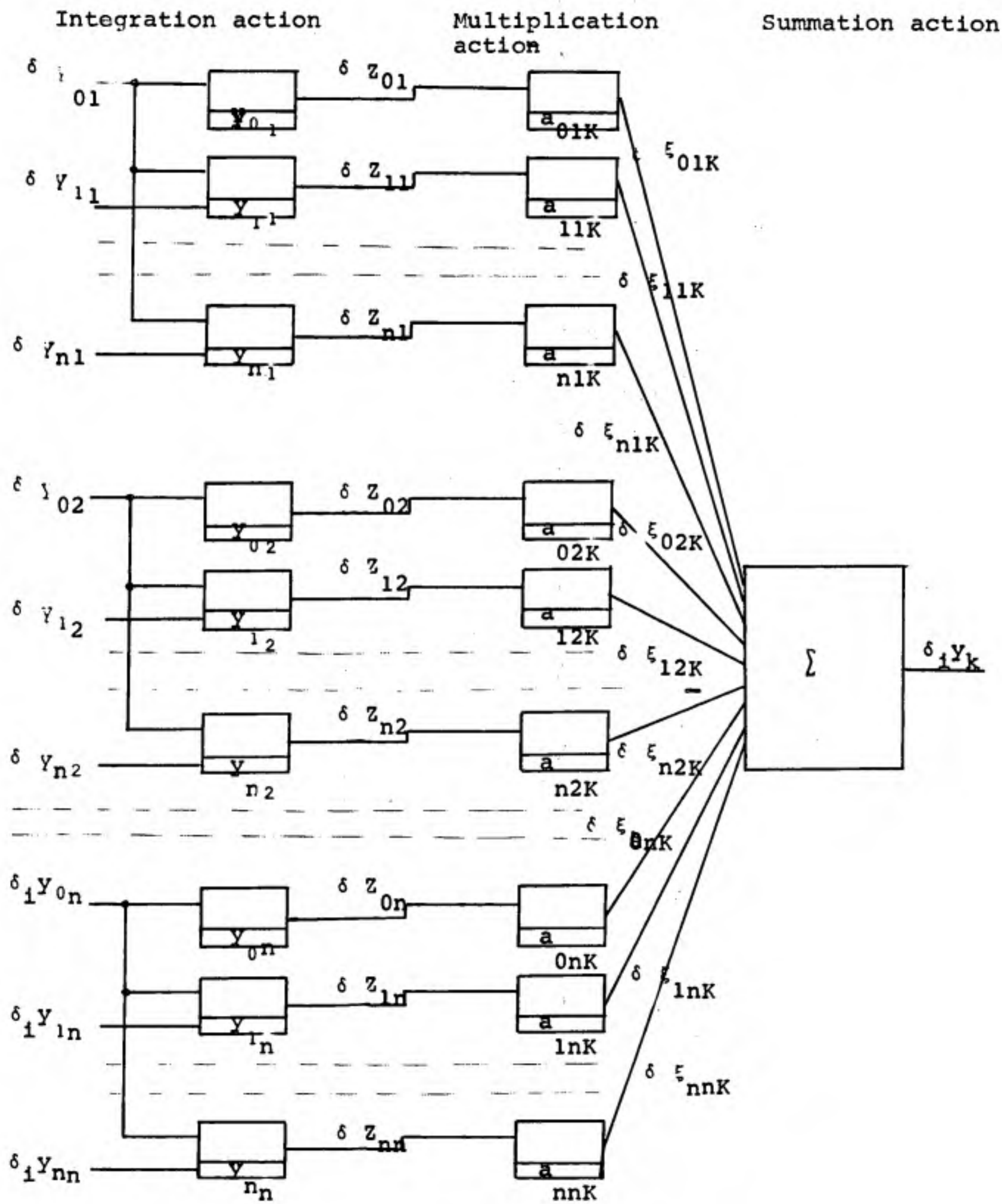

Integration action
Multiplication action
Summation action

Fig. 1-12

In following chapters, we shall calculate the error in unitary and multiple increment computation for various methods of integration.

1.3. The numerical integration in incremental computers.

Numerical integration is the study of how the numerical value of an integral can be found. There is the method of approximate integration, where an integral is approximated by a linear combination of the values of the integrand.

$$\int_a^b y \, dx = \int_a^b f(x) \cdot dx \tag{1-18}$$

$$= \omega_1 f(x_1) + \omega_2 f(x_2) + \dots + \omega_k f(x_k)$$

$$a < x < b$$

The $x_1, x_2, \dots x_k$ are k points usually chosen to lie so in the interval of integration, and the numbers $\omega_1, \omega_2, \dots \omega_K$ are k weights accompanying these points. Occasionally, the values of the derivatives of the integrand appear on the right hand side of the equation (1- 18). Let's suppose that $y = f(x)$ is a bounded function on the finite interval (a, b). Partition the interval (a, b) into k subintervals by the points:

$$a = x_o < x_1 < x_2 < \quad \ldots\ldots < x_k = b$$

let ξ_i be any points in the subintervals $x_i < \xi_i < x_{i+1}$. Then the sum of:

$$\sum_{i=1}^{k} f\ (\xi_i)(x_{i+1} - x_i)$$

is called Rejmand sum.

The approximate form of integration in interval $x \in (a, b)$ is:

$$\int_a^b y\ dx = \sum_{i=1}^{k} f\ (\xi_i)(x_{i+1} - x_i) \qquad (1\text{-}20)$$
$$k \to \infty$$

where $f\ (\xi_i)$ is an approximate function of $y_i \approx f\ (\xi_i)$ for interval $\xi \in (x_i, x_{i+1})$, as it is shown in figure (1.13).

If the independent variable of integral X is a function of the independent variable t of the machine, then the functions Y (t) and X (t) are replaced by the approximated interpolated functions f_{ix} (t) and f_{iy} (t) as following:

$$\begin{cases} X\ (t) \approx f_{ix}\ (t) & t \in (t_i, t_{i+1}) \\ Y\ (t) \approx f_{iy}\ (t) & \end{cases} \qquad (1\text{-}21)$$

where

$$\begin{cases} f_{ix}\ (t) = f_{ix}\ (x_i,\ x_{i-1},\ x_{i-2},\ \ldots\ldots\ t) \\ f_{iy}\ (t) = f_{iy}\ (y_i,\ y_{i-1},\ y_{i-2},\ \ldots\ldots\ t) \end{cases} \tag{1-22}$$

Then the general formula of integral S (t),

$$S\ (t) = \int_a^b Y\ (t)\ d\ \frac{X(t)}{dt}\ dt \qquad t \in (a,\ b) \tag{1-23}$$

is replaced by the approximated interpolated function of integral $S^{*}(t)$ as:

$$S^{*}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy}\ (t) \cdot d\ \frac{f_{ix}(t)}{dt}\ dt \qquad t \in (t_i,\ t_{i+1}) \tag{1-24}$$

In general it is possible to interpolate the functions f_{ix} (t), f_{iy} (t) with any interpolation formula as Newton, Reiman, Stirting, Lagarangian, and so on, with any degree of accuracy, in interval $t \in (t_i,\ t_{i+1})$.

The polynomial formula of interpolation is much used in physical and engineering problems, specially in the digital computer, whose functions are transformed to the approximated polynomial functions. But in incremental computer, the informations which are transmited and operated on, are in the increment forms δx, δy, δt. Therefore, the first condition of interpolation function for incremental computer is that, it must use the increments of functions.

The Newton interpolation formula with forward informations in interval $\xi \in (x_i, x_{i+1})$, can be represented as:

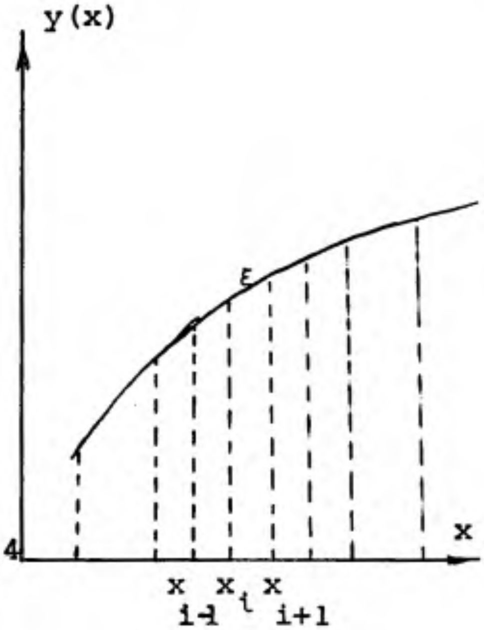
$$f_i(\xi) = y_i + \xi \cdot \delta_i y + \frac{\xi(\xi-1)}{2!} \delta_i^{II} y + + \frac{\xi(\xi-1)(\xi-2)}{3!} \delta_i^{III} y + \cdots + \frac{\xi(\xi-1)\ldots(3-n+1)}{n!} \delta_i^{(n)} y \qquad (1-25)$$

and

$$\begin{aligned} \delta_i^{I} y &= f(x_{i+1}) - f(x_i) \\ \delta_i^{II} y &= \delta_{i+1}^{I} y - \delta_i^{I} y \end{aligned} \qquad (1-26)$$

As it is seen from equations (1-25), (1-26) and figure (1.14), the interpolation formula for each interval $x \in (x_i, x_{i+1})$ depends on the information of the points x_i, x_{i+1}, x_{i+2}, ... x_{i+n}.

The formula is called the Newton's interpolation formula with forward differences. This formula is useful, when we have forward informations of interval ξ (x_i, x_{i+1}), like the physical problem or



ig.1.14

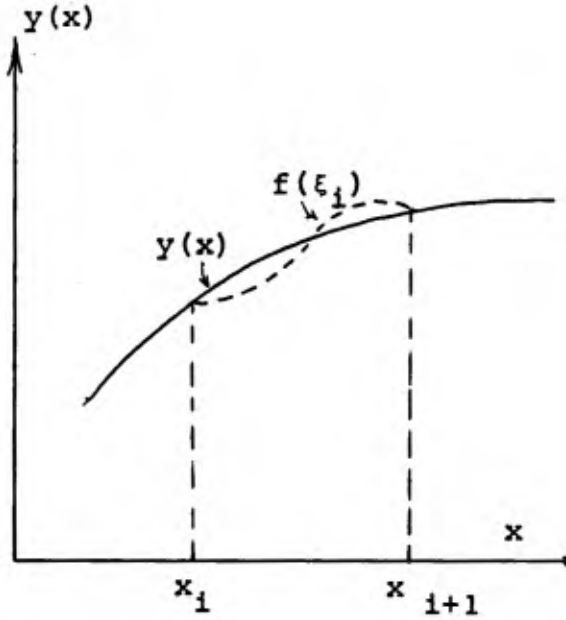


fig.1-13

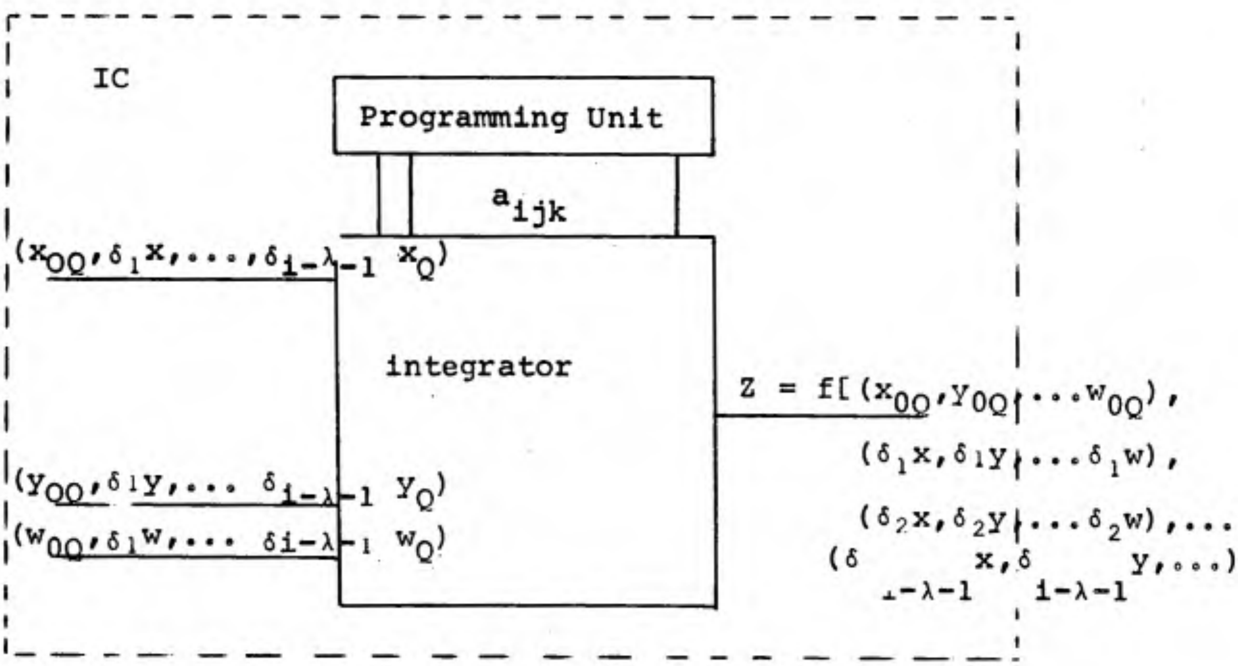


Fig. 1-11

experimental data.

But in incremental machine, in iteration i, the only information that may exist in the memory, is the information of former iteration i-1, i-2, i-3, ... 0,1. Therefore, the second condition of the interpolation formula is that, it should use backward difference data, or the increments of former iteration. The Newton's interpolation formula with backward differences, in interval $\xi \in (x_i, x_{i+1})$ is:

$$f_i(\xi) = y_i - \xi \cdot \delta_i^I y - \frac{\xi(\xi+1)}{2!} \delta_i^{II} y - \frac{\xi(\xi+1)(\xi+2)}{3!} \delta_i^{III} y - \dots - \frac{\xi(\xi+1)(\xi+2)\dots(\xi+n-1)}{n!} \delta_i^{(n)} y \tag{1-27}$$

where

$$\begin{aligned} \delta_i^I y &= f(x_i) - f(x_{i-1}) \\ \delta_i^{II} y &= \delta_i^I y - \delta_{i-1}^I y \end{aligned} \tag{1-28}$$

By using the equation (1-27), the integral of interpolated function in interval $x \in (x_i, x_{i+1})$ will be as following:

$$\delta_i s = -\delta_i x \int_0^{-1} f(\xi)\, d\xi \tag{1-29}$$

$$= - \delta_i x \left| \xi y_i - \frac{\xi^2}{2} \delta_i^{I} y - \frac{1}{2} \left(\frac{\xi^3}{3} + \frac{\xi^2}{2}\right) \delta_i^{II} y - \right.$$

$$- \frac{1}{6} \left(\frac{\xi^4}{4} + \xi^3 + \xi^2\right) \delta_i^{III} y - \frac{1}{24} \left(\frac{\xi^4}{5} + \right.$$

$$\left. + \frac{3\xi^4}{2} + \frac{11\xi^3}{3} + 3\xi^2\right) \delta_i^{III} y - \ldots\ldots \Big|_{0}^{-1}$$

$$= y_i \cdot \delta_i x + \frac{1}{2} \delta_i x \cdot \delta_i y + \frac{1}{12} \delta_i x \cdot \delta_i^{II} y +$$

$$+ \frac{1}{24} \delta_i x \cdot \delta_i^{III} y + \frac{19}{720} \delta_i x \cdot \delta_i^{IV} y + \ldots\ldots \qquad (1\text{-}30)$$

$$x \in (x_i,\ x_{i+1})$$

in the formula (1-30), if we choose the first term of right hand side, we will have:

$$\delta_i s \approx y_i \cdot \delta_i x \qquad (1\text{-}31)$$

that is the approximate integration formula of the rectangular method.

By choosing the first two terms of right hand side of the equation (1-30), we will have:

$$\delta_i s = y_i \cdot \delta_i x + \frac{1}{2} \delta_i x \cdot \delta_i y \qquad (1\text{-}32)$$

$$x \in (x_i,\ x_{i+1}$$

The formula (1-32) is the approximate formula of the trapezoidal method of integration.

The Newton's interpolation formula with backward difference informations, is very useful in interpolating the functions X (t) and Y (t).

In the following chapter we use the Newton's interpolation formula for unitary and multiple incremental computations.

CHAPTER II

THE METHODS AND ERRORS OF INTEGRATION IN INCREMENTAL COMPUTATION

2.1. The methods and errors of integration in incremental computation when the independent variable of integral x is the independent variable t.

If we have the continuous function $y(x)$, where the independent variable x is the variable t of machine , the integral $S(x)$ in interval $x \in (x_o, x_k)$ will be:

$$S(x) = \int_{x_o}^{x_k} y(x)\ dx \qquad (2\text{-}1)$$

Considering the y (x) function in the interval $x \in (x_o, x_k)$. There are infinit points (x,y,) between $x \in (x_o, x_k)$ which are necessary for calculating the exact value of integral in the interval $x \in (x_o, x_k)$. But in practice, because it is the time consuming for calculating the infinit points (x,y) in interval $x \in (x_o, x_k)$. and also it is too expensive to construct the machine for calculating the infinit points in this interval with infinit capacity of memory, therefore we are obliged to use some points (x,y) in interval $x \in (x_o, x_k)$, let us say x_i, y_i $(i=o,1,2,\ldots k)$ fig. (2-1). So there is an error between the exact value of function y(x) and the interpolated function $f_{iy}(x)$ which use the finit point $x_i, y_i, \ldots (i=1,2,\ldots k)$.

The error ε_{ix} , ε_{iy} between the actual function y(x) and interpolated function f_{iy} (x) in each internal $x \in (x_i, x_{i+1})$ are:

$$\begin{cases} \varepsilon_{iy} = f_{iy}(x) - y(x) \\ \varepsilon_{ix} = f_{ix}(x) - x(x) \end{cases} \qquad (2-2)$$

as $x = t$ then $\varepsilon_{ix} = o$ (2-3)

by putting the value of y(x) and ε_{ix} from equations (2-2) and (2-3) in equation (2-1), we will have:

$$S(x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} (f_{iy}(x) - \varepsilon_{iy})\, dx \qquad (2-4)$$

$$= \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iy}(x) - \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iy}\, dx \qquad (2-5)$$

$$= S^{*}(x) + \Gamma(x) \qquad (2-6)$$

or

$$\begin{bmatrix} S^{*}(x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iy}(x)\, dx \\ \Gamma(x) = -\sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iv}\, dx \end{bmatrix} \qquad (2-7)$$

where $S^{*}(x)$ is approximated interpolated formula of integration and $\Gamma(x)$ is the error of method in the process of integration. The error of method $\Gamma(x)$ depends to the degree of interpolation function $f_{iy}(x)$ which is used. In the following paragraphs we will calculate the error of method $\Gamma(x)$ for different method of integration.

2.1.1. Integration by the interpolated rectangular formula in unitary or multiple increment computation.

The simplest method of integration is the rectangular method, when the independent variable of integration x is same as the independent variable t of machine.

In the rectangular method of integration the interpolation function f_{iy} (x) which is replaced to y(x) in interval $x \in (x_i, x_{i+1})$ is the first term of Newton interpolation formula. Therefore the interpolation function as it is shown in figure (2.2) will be:

$$f_{iy} = y_i \qquad x \in (x_i, x_{i+1}) \tag{2-8}$$

The error ε_{iy} between the actual function y(x) and interpolated function f_{iy} can be find by the following expression:

$$\varepsilon_{iy} = f_{iy}(x) - y(x) \tag{2-9}$$

The formula of integration $\delta_i s$ in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_i s = \int_{x_i}^{x_{i+1}} y(x)\, dx \tag{2-10}$$

if we put the equation (2-9) in equation (2-10) we will have:

$$\delta_i s = \int_{x_i}^{x_{i+1}} [f_{iy}(x) - \varepsilon_{iy}]\, dx \tag{2-11}$$

The integral formula s (x) for all k interval will be:

$$s(x) = \sum_{i=1}^{k} \delta_i s = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} [f_{iy}(x) - \varepsilon_{iy}]\, dx \tag{2-12}$$

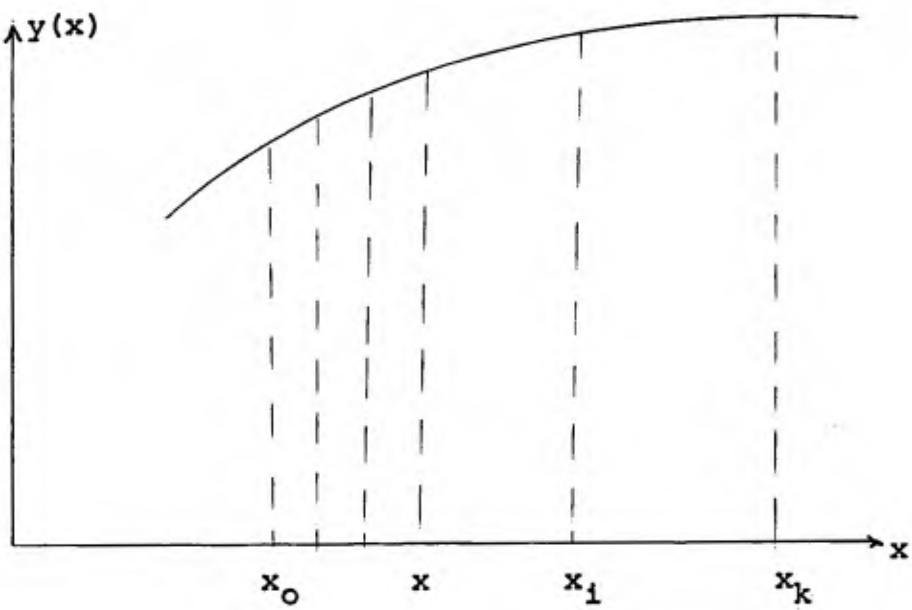


fig. 2.1.

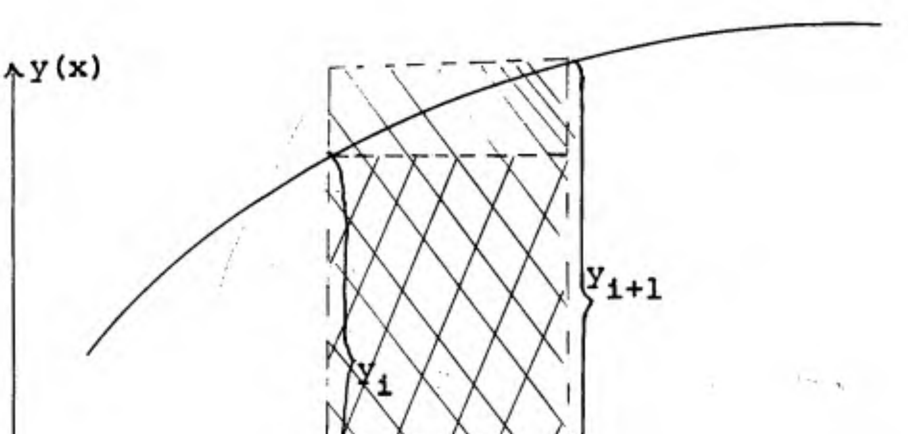


fig. 2.2.

$$= \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iy}\ (x)\ dx - \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iy}\ dx \qquad (2\text{-}13)$$

$$= s^{*}\ (x) + \Gamma\ (x) \qquad (2\text{-}14)$$

where s^{*} (x) is the approximated interpolated function of integral, and $\Gamma(x)$ is the error of method.

In order to calculate the error of method $\Gamma(x)$, we should have ε_{iy}, $y(x)$ and f_{iy} (x). The exact value of function y(x) in internal $x \in (x_i, x_{i+1})$ can be found by the infinit points $(x_i, y_i, \ldots\ i=1,2,\ldots\infty)$ from Newton interpolation formula as following:

$$y(x) = y_i - \xi\delta_i^{I}y - \frac{\xi(\xi+1)}{2!}\delta_i^{II}y - \frac{\xi(\xi+1)(\xi+2)}{3!}\delta_i^{III}y \ldots$$

$$\frac{\xi(\xi+1)(\xi+2)\ldots\qquad(\xi+n-1)}{n!}\delta_i^{(n)}y - \ldots\ldots \qquad (2\text{-}15)$$

from equation (2-8), (2-9) and (2-15) the ε_{iy} will be

$$\varepsilon_{iy} = f_{iy}\ (x) - y(x) \qquad (2\text{-}16)$$

$$= \xi\ \delta_i^{I}y + \frac{\xi(\xi+1)}{2!}\delta_i^{II}y + \frac{\xi(\xi+1)(\xi+2)}{3!}\delta_i^{III}y + \ldots$$

$$\ldots\ldots+\frac{\xi(\xi+1)(\xi+2)\ldots\qquad(\Xi+n-1)}{n}\,\delta_i^{n}y+\ldots \quad (2\text{-}17)$$

in practice we can neglect the second and others terms of ε_{iy} in equation (2-16) and (2-17) with respect to the first one so ε_{iy} will be as following:

$$\varepsilon_{iy} = \xi.\ \delta_i^I\ y \qquad (2\text{-}18)$$

Putting the equation (2-8) and (2-18) in equation (2-13), and tacking the integral in interval $x \in (x_i, x_{i+1})$ or $\xi \in (-1,0)$, then we will have:

$$s(x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iy}(x)dx - \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iy}dx \quad (2\text{-}19)$$

$$= -\sum_{i=1}^{k} \delta x \int_{0}^{-1} y_i\ d\xi + \sum_{i=1}^{k} \delta x \int_{0}^{-1} \xi\cdot\delta_i^I y\ d\xi \quad (2\text{-}20)$$

$$= \sum_{i=1}^{k} y_i\cdot\delta_i\ x + \sum_{i=1}^{k} \frac{1}{2}\ \delta y_i\cdot\delta x_i \quad (2\text{-}21)$$

$$= s^{*}_{i}\ (x) + r_{i}(x) \quad (2\text{-}22)$$

From equations (2-19), (2-20), (2-21) and (2-22), the approximated interpolated formula of integral S_i^* (x) which is the algorithm of machine is:

$$S_1^{*}(x) = \sum_{i=1}^{k} y_i \cdot \delta_i x \tag{2-23}$$

and the error of method $\Gamma_1(x)$ is:

$$\Gamma(x) = \sum_{i=1}^{k} \frac{1}{2} \delta_i x \cdot \delta_i y \quad \text{as} \quad \delta_i y = y'(\xi)\, \delta_i x \tag{2-24}$$

$$\Gamma_1(x) = \sum_{i=1}^{k} \frac{1}{2} \delta_i^2 x \cdot y'(\xi) \tag{2-25}$$

In the case of unitary incremental computer, $\delta x = \Delta x = \frac{(x_k - x_o)}{k}$ so the equation (2-25) can be written as:

$$\Gamma_1(x) = \frac{1}{2} \frac{(x_k - x_o)^2}{k^2} \cdot \sum_{i=1}^{k} y'(\xi) \qquad \xi \in (x_i, x_{i+1})$$

assuming

$$y'(\xi_1) = \frac{1}{k} \sum_{i=1}^{k} y'(\xi) \qquad \xi_1 \in (x_o, x_k)$$

then

$$\Gamma_1(x) = \frac{1}{2} \cdot \frac{(x_k - x_o)^2}{k} y'(\xi_1) \tag{2-26}$$

It can be shown that $|y'(\xi)| < \frac{\left|\sum_{j=1}^{b} (\Delta y)_j\right|_{max}}{\Delta x}$ where b is the number of Δy input of the integrator, in our machine b = 7. So the error of method $\Gamma_1(x)$ for unitary incremental computer will be:

$$\Gamma_1(x) < \frac{7}{2} \Delta y \, (x_k - x_o) \tag{2-26*}$$

The equation (2-23) gives the algorithms of rectangular method of integration assuming $f_{iy} = y_i$. But if we assume:

$$f_{iy} = y_{i+1} \tag{2-27}$$

Then the algorithms of rectangular method will be:

$$s_2^{*}(x) = \sum_{i+1}^{k} y_{i+1} \quad \delta_i x \tag{2-28}$$

and the error of this method will be:

$$\epsilon_2 (x) = \sum_{i=1}^{k} \frac{1}{2}(\delta_i x)^2 \; y' \; (\xi) \quad \xi \; (x_i, x_{i+1}) \tag{2-29}$$

The actual value of integral s(x) is between the $s_1^{*}(x)$ and $s_2^{*}(x)$ as:

$$s_2^{*}(x) > s(x) > s_1^{*}(x) \tag{2-30}$$

where:

$$\begin{cases} s_1^{*}(x) = \sum_{i=1}^{k} y_i \cdot \delta_i x \\ s_2^{*}(x) = \sum_{i=1}^{k} y_{i+1} \cdot \delta_i x \end{cases} \tag{2-31}$$

2.1.2. Integration by the interpolated trapezoidal formula in unitary or multiple increment computation.

In trapezoidal method, the interpolation function f_{iy} (x) which is replaced to y (x) in internal $x \in (x_i, x_{i+1})$ is the first two terms of Newton interpolation formula (eq.2-15) in internal $x \in (x_i, x_{i+1})$, as following:

$$f_y(x) = y_i - \xi \cdot \delta_i^T Y$$

$$x \in (x_i, x_{i+1}) \text{ or } \xi \, (o, -1) \qquad (2\text{-}32)$$

As it is seen from figures (2-3,4), in trapezoidal method, the y (x) is interpolated lineary in interval $x \in (x_i, x_{i+1})$, therefore it has less error than the rectangular method. If we assume the error ε_{iy} between the actual function y (x) and the approximate interpolated function f_{iy} (x) in interval $x \in (x_i, x_{i+1})$.

Then we will have:

$$\varepsilon_{iy} = f_{iy}(x) - y(x) \qquad (2\text{-}33)$$

The integral formula, in interval $x \in (x_i, x_{i+1})$, is:

$$\delta_i S = \int_{x_i}^{x_{i+1}} y(x)\, dx \qquad (2\text{-}34)$$

if we put the y (x) from equation (2-33) in equation (2-34), then:

$$\delta_i S = \int_{x_i}^{x_{i+1}} [f_y(x) - \varepsilon_{iy}]\, dx \qquad (2\text{-}35)$$

As we assumed x is independent variable of machine therefore

$$\varepsilon_{ix} = f_x(x) - X(x) = 0$$

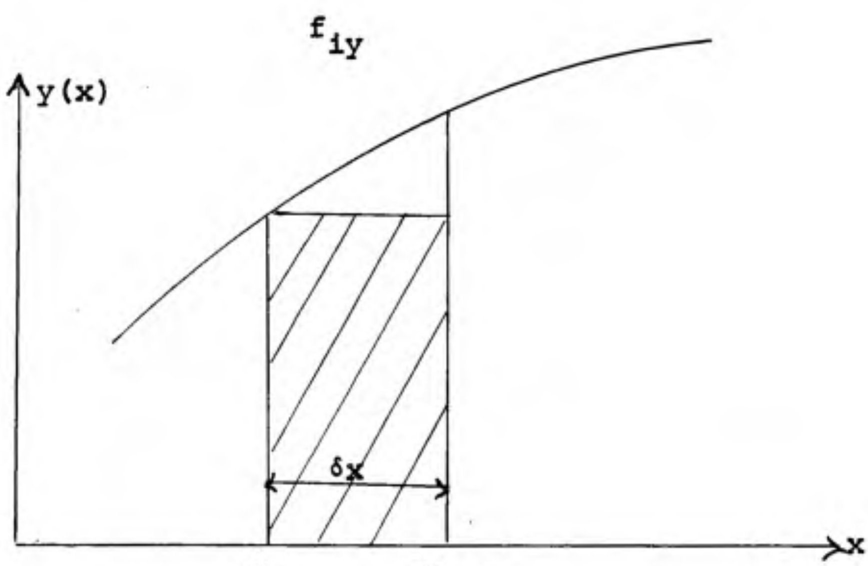


fig. 2.3.

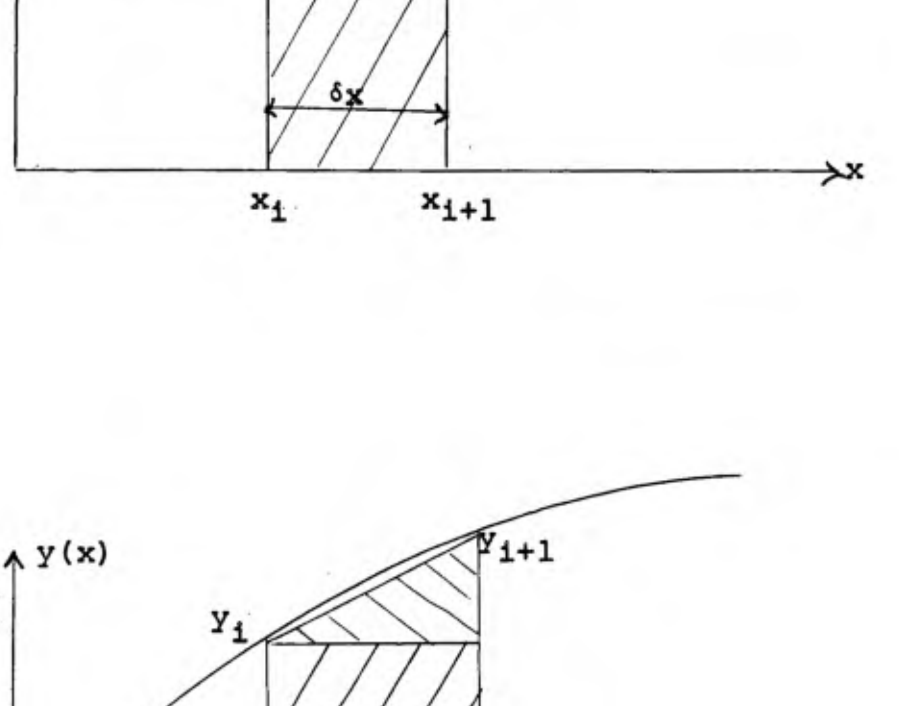


fig. 2.4.

In order to find the value of integral (2-35), we should find ε_{iy}, $y\ (x)$, $f_{iy}\ (x)$.

The exact value of $y\ (x)$ can be found with infinit points $(x_i,\ y_i,\ (i=1,\ 2,\ldots\ldots\infty\ )\ )$. From Newton interpolation formula (eq.2-15).

From equation (2-32), (2-33), and (2-15), the ε_{iy} can be find as following:

$$\varepsilon_{iy} = f_{iy}\ (x) - y\ (x) = \frac{\xi(\xi+1)}{2!}\ \delta_i^{II}\ y + \frac{\xi(\xi+1)(\xi+2)}{3!}\ \cdot$$

$$\cdot\ \delta_i^{III}\ y + \ldots\ldots \qquad + \frac{\xi(\xi+1)(\xi+2)\ldots\ (\xi+n-1)}{n!}\ \delta_i^{(n)} y \qquad (2\text{-}36)$$

In practice we can neglect the second and others terms of ε_{iy}, with respect to the first one. Therefore the ε_{iy} from equation (2-36) will be:

$$\varepsilon_{iy} = \frac{\xi(\xi+1)}{2!}\ \delta_i^{II}\ y \qquad (2\text{-}37)$$

By putting the ε_{iy} from equation (2-37) and $f_{iy}\ (x)$ from equation (2-32) in the integral formula (2-35), we will have:

$$\delta_i\ s = \int_{x_{i-1}}^{x_{i+1}} [y_i - \xi\cdot\ \delta_i^{I}\ y - \frac{\xi(\xi+1)}{2!}\ \delta_i^{II}\ y]\ dx$$

$$x \in (x_i, x_{i+1}) \qquad (2\text{-}38)$$

By changing variable $x \in (x_i, x_{i+1})$, to ξ $(-1,0)$ as before we will have:

$$\delta_i s = -\delta x_i \int_0^{-1} [y_i - \xi \cdot \delta_i^{I} y - \frac{\xi(\xi+1)}{2!} \delta_i^{II} y] \, d\xi \qquad (2\text{-}39)$$

$$= -\delta x_i \left| y_i \cdot \xi - \frac{\xi^2}{2} \cdot \delta_i^{I} y - \frac{1}{2!}(\frac{\xi^3}{3} + \frac{\xi^2}{2}) \cdot \right.$$

$$\left. \cdot \delta_i^{II} (y) \right|_0^{-1} \qquad (2\text{-}40)$$

$$= \delta x_i \left| + y_i + \frac{1}{2} \delta_i^{I} y + \frac{1}{12} \delta_i^{II} (y) \right| \qquad (2\text{-}41)$$

The equation (2-41) can be written as:

$$\delta_i s = y_i \cdot \delta x + \frac{1}{2} \delta_i^{(I)} y \cdot \delta x + \frac{1}{12} \delta_i^{II} (y) \cdot \delta x \qquad (2\text{-}42)$$

If we assume the approximated interpolated integration $\delta_i^{*} s$ in interval $x \in (x_i, x_{i+1})$, be equal to:

$$\delta_i^{*} S = y_i \, \delta x + \frac{1}{2} \delta_i^{(I)} y \cdot \delta_i x \qquad (2\text{-}43)$$

Then the equation (2-42) can be written as:

$$\delta_i s = \delta_i^{*} s + r_i\ (x) \qquad x \in (x_i, x_{i+1}) \qquad (2\text{-}44)$$

In equation (2-44) the $\delta_i s$ is the real value of integral, $\delta_i s^{*}(x)$ is the approximated interpolated value, and the $r_i(x)$ is the error of method in this interval, which can be find from equation (2-42), (2-43) and (2-44) as following:

$$r_i\ (x) = +\frac{1}{12}\ \delta_i^{II}\ (y).\delta x \qquad x \in (x_i, x_{i+1}) \qquad (2\text{-}45)$$

The integral formula for k will be:

$$s(x) = \sum_{i=1}^{k} \delta_i\ s\ (x) \qquad (2\text{-}46)$$

$$= \sum_{i=1}^{k} \delta_i\ s^{*}(x) + \sum_{i=1}^{k} r_i\ (x) \qquad (2\text{-}47)$$

$$= \sum_{i=1}^{k} (y_i \delta x + \frac{1}{2} \delta y_i \cdot \delta x_i) + \sum_{i=1}^{k} \frac{1}{12}\ \delta y_i^{(II)} \cdot \delta x \qquad (2\text{-}48)$$

The approximated interpolated function of integral s (x) for k interval will be:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i^{*}\ s \qquad (2\text{-}49)$$

$$= \sum_{i=1}^{k} (y_i \cdot \delta_i x + \frac{1}{2} \delta_i y \cdot \delta_i x) \qquad (2\text{-}50)$$

and the error of method is:

$$\Gamma(x) = \sum_{i=1}^{k} \Gamma_i (x) \tag{2-51}$$

$$= \sum_{i=1}^{k} \frac{1}{12} \delta_i^{II} (y) \cdot \delta_i x \tag{2-52}$$

by using

$$\delta^{II}(y) = y'' \cdot (\delta x)^2$$

we will have

$$\Gamma(x) = \sum_{i=1}^{k} \frac{1}{12} (\delta x)^3 \cdot y'' (x) \qquad x \in (x_i, x_{i+1}) \tag{2-53}$$

In the case of unitary increment $\delta x = \Delta x$, $\delta y = \Delta y$ and

$$\Delta x = \frac{x_k - x_o}{k},$$

assuming:

$$y''(\xi) = \frac{1}{k} \sum_{i=1}^{k} y'' (x) \qquad \xi \in (x_o, x_k) \tag{2-54}$$

then the error of method $\Gamma(x)$ is:

$$\Gamma(x) = \sum_{i=1}^{k} \frac{1}{12} (\Delta x)_i^3 \cdot y'' (x) \tag{2-55}$$

$$= \sum_{i=1}^{k} \frac{1}{12} \cdot \frac{(x_k - x_o)^3}{k^3} \cdot y'' (x) \tag{2-56}$$

$$= \frac{1}{12} \cdot \frac{(x_k - x_o)^3}{k^2} y'' (\xi) \quad \xi \in (x_o, x_k) \tag{2-57}$$

or:

$$\Gamma\ (x) = \frac{1}{12} \cdot \frac{(x_k - x_o)^3}{k^2}\, y''(\xi_1) \quad \xi_1 \in (x_o, x_k) \tag{2-58}$$

It can be shown that $y''(\xi) < \dfrac{\left|\sum_{i=1}^{b} \Delta y\right|_{max}}{(\Delta x)^2}$ as b = 7, so the error of method $\Gamma(x)$ for unitary incremental computer will be:

$$\Gamma\ (x) < \frac{7}{12}\, \Delta y\ (x_k - x_o) \tag{2-59}$$

By comparing the error of rectangular method (eq. 2-26), and the error of trapezoidal method (2-58), it is clear that by increasing k, the error in trapezoidal formula is decreasing more rapide than the rectangular method, in others words, the error in the trapezoidal method is decreasing k time more rapide than the rectangular method.

2.1.3. Integration by the three points formula in unitary or multiple increment computation.

The interpolation function f_{iy} (x) which is replaced to y (x) in interval $x \in (x_i, x_{i+1})$, is the first three terms of Newton interpolation, in other words, f_{iy} (x) use the information of three points, (x_{i+1}, y_{i+1}), (x_i, y_i) and (x_{i-1}, y_{i-1}), figure (2.5).

$$f_{iy}\ (x) = y_i - \xi \circ \delta_i^{I}\, y - \frac{\xi(\xi+1)}{2!}\ \delta_i^{II}\, y \tag{2-60}$$

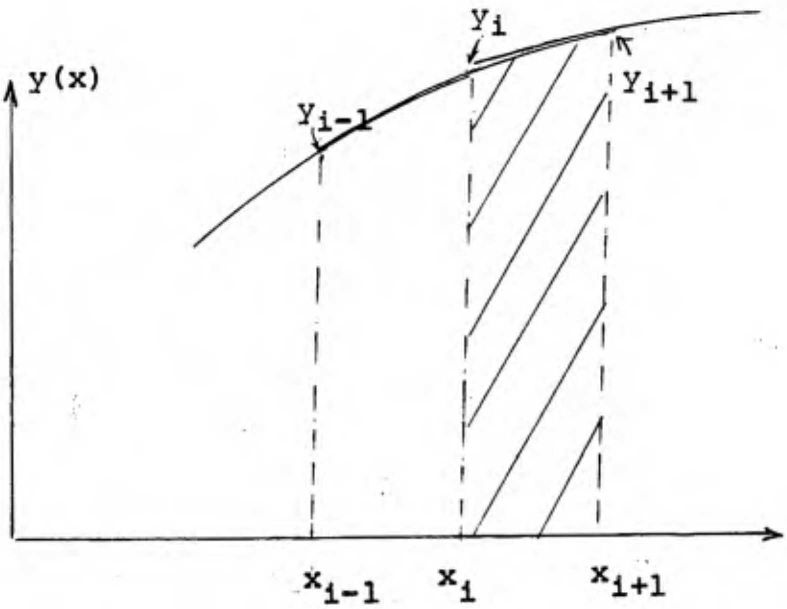


fig. 2.5.

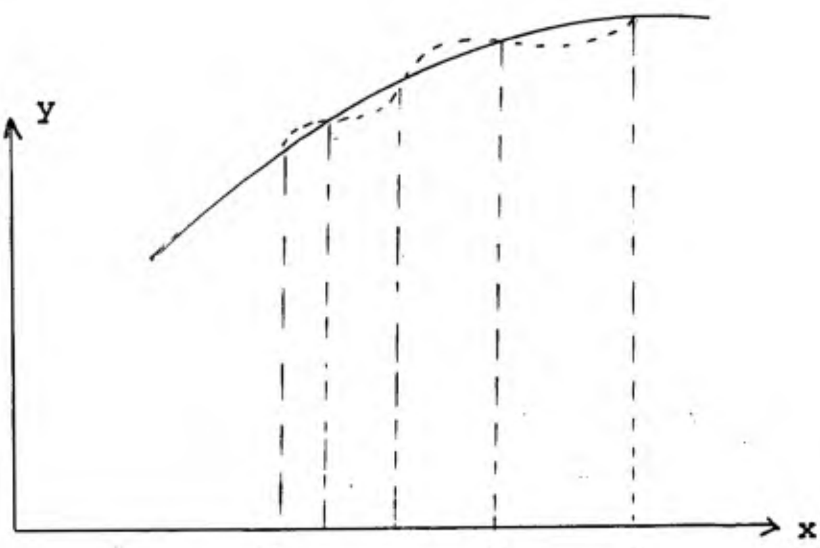


fig. 2.6.

$$\xi \in (-1,0) \text{ or } x \in (x_i, x_{i+1})$$

where

$$f_i y \ (x) = f\,[(x_{i+1}, y_{i+1}), (x_i, y_i), (x_{i-1}, y_{i-1})$$

There is an error ε_{iy} between the actual function y (x) and the interpolated function f_{iy} (x) which can be find by the expression:

$$\varepsilon_{iy} = f_{iy}\ (x) - y\ (x) \tag{2-61}$$

The formula of integration in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_i\ s = \int_{x_i}^{x_{i+1}} y\ (x)\ dx \tag{2-62}$$

If we put the value of y (x) from equation (2-61) in equation (2-62), we will have:

$$\delta_i\ s = \int_{x_i}^{x_{i+1}} [f_{iy}\ (x) - \varepsilon_{iy}\ (x)]dx \tag{2-63}$$

$$x \in (x_i, x_{i+1})$$

for k interval the integral formula s (x) will be:

$$s(x) = \sum_{i=1}^{k} \delta_i\ s(x) \tag{2-64}$$

$$= \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} [f_{iy}(x) - \varepsilon_{iy}(x)]\, dx \tag{2-65}$$

In order to calculate the equation (2-65), we should have the $y\ (x)$, ε_{iy} and $f_{iy}\ (x)$.

The value of $y\ (x)$ can be found with infinit points of

$$[\ x_i, y_i\ (i=1,\ 2, \ldots\ldots\infty)\ ]$$

from Newton interpolation formula (eq.2-15) for $i=\infty$. From equation (2-60),(2-61) and (2-15), the ε_{iy} can be found as following.

$$\varepsilon_{iy} = f_{iy}\ (x) - y\ (x) = + \frac{\xi(\xi+1)(\xi+2)}{3!}\, \delta_i^{III} y + $$

$$+ \frac{\xi(\xi+1)(\xi+2)\ldots(\xi+n-1)}{n!}\, \delta_i^{(n)} y + \ldots. \tag{2-66}$$

if we neglect the second and other terms of ε_{iy} with respect to the first one, we will have:

$$\varepsilon_{iy} = + \frac{\xi(\xi+1)(\xi+2)}{3!}\, \delta_i^{III}\ y \tag{2-67}$$

By putting the ε_{iy} from (2-67) and f_{iy} from (2-60) in equation (2-65) we will have:

$$\delta_i\ s = -\ \delta x \int_0^{-1} \Big\{ y_i - \xi\ \delta_i^{I}\ (y) - \frac{\xi(\xi+1)}{2!}\, \delta_i^{II}(y) - $$

$$- \frac{\xi(\xi+1)(\xi+2)}{3!} \delta_y^{III} \Big) \, d\xi$$

$$= \delta x \, [y_i + \frac{1}{2} \delta_i y + \frac{1}{12} \delta_i^{II} y + \frac{1}{24} \delta_i^{III} y] \qquad (2\text{-}69)$$

$$= y_i \; \delta x + \frac{1}{2} \delta y_i \; \delta x_i + \frac{1}{12} \delta_i^{II} y \; \delta x + \frac{1}{24} \delta_i y^{III} \; \delta x \qquad (2\text{-}70)$$

by putting $\delta_i^{II} y = \delta_i y - \delta_{i-1} y$ in equation (2-70), we will have:

$$\delta_i \; s \; (x) = y_i \; \delta_i x + \frac{1}{2} \delta_i y \cdot \delta_i x + \frac{1}{12} \delta_i x \; (\delta_i y - \delta_{i-1} y) +$$

$$+ \frac{1}{24} \delta_i y^{III} \cdot \delta_i x \qquad (2\text{-}71)$$

as the approximated interpolated function in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_i \; s^{*} = y_i \cdot \delta x_i + \frac{1}{2} \delta x_i \cdot \delta y_i + \frac{1}{12} (\delta x_i \cdot \delta y_i - \delta_{i-1} y \cdot \delta x_i) \qquad (2\text{-}72)$$

then equation (2-71) can be written as following:

$$\delta_i \; s(x) = \delta_i s^{*} \; (x) + \Gamma_i \; (x) \qquad (2\text{-}73)$$

where $\Gamma_i(x)$ is the error of method in interval $x \in (x_i, x_{i+1})$

which is equal to: $$\Gamma_i \; (x) = + \frac{1}{24} \delta_i x \cdot \delta_i^{III} \; y \qquad (2\text{-}74)$$

The integral formula s (x) for k interval can be found from equation (2-73) as following:

$$s\ (x) = \sum_{i=1}^{k} \delta_i\ s\ (x) \qquad (2\text{-}75)$$

$$= \sum_{i=1}^{k} \delta_i\ s^*(x) + \sum_{i=1}^{k} \Gamma_i\ (x) \qquad (2\text{-}76)$$

$$= \sum_{i=1}^{k} (y_i\ \underset{i}{\delta x} + \frac{1}{2}\delta y_i\ \underset{i}{\delta x} + \frac{1}{12}\ \delta y_i\ \underset{i}{\delta x} -$$

$$- \frac{1}{12}\ \delta y_{i-1} \cdot \underset{i}{\delta x}) + \sum_{i=1}^{k} \frac{1}{24}\ \delta x \cdot \delta_i^{III} y \qquad (2\text{-}77)$$

$$= s^*(x) + \Gamma(x) \qquad (2\text{-}78)^*$$

where $s^*(x)$ is the approximated interpolated integration function for k interval as:

$$s^*(x) = \sum_{i=1}^{k} (y_i\ \underset{i}{\delta x} + \frac{1}{2}\delta y_i\ \underset{i}{\delta x} + \frac{1}{12}\ (\delta y_i \cdot \underset{i}{\delta x} -$$

$$- \delta y_{i-1}\ \underset{i}{\delta x})\ ) \qquad (2\text{-}78)$$

and the error of method in k interval is:

$$\Gamma(x) = \sum_{i=1}^{k} + \frac{1}{24}\ \underset{i}{\delta x} \cdot \delta_i^{III} y \qquad (2\text{-}79)$$

$$x \in (x_i, x_{i+1})$$

by using

$$\delta_i y^{III} = y'''(x)(\delta_i x)^3$$

the equation (2-79) will be:

$$\Gamma(x) = \sum_{i=1}^{k} + \frac{1}{24} (\delta_i x)^4 \cdot y'''(x) \tag{2-80}$$

Therefore the algorithms and errors of integration are:

$$\begin{cases} s(x) = s^*(x) + \Gamma(x) \\ s^*(x) = \sum_{i=1}^{k} y_{eq} \cdot \delta_i x \\ y_{eq} = y_i + \frac{1}{2}\delta y_i + \frac{1}{12}(\delta y_i - \delta y_{i-1}) \\ \Gamma(x) = \sum_{i=1}^{k} + \frac{1}{24}(\delta_i x)^4 \, y'''(x) \quad x \in (x_i, x_{i+1}) \end{cases} \tag{2-81}$$

In the case of unitary increment computation $\delta x = \Delta x = \frac{x_k - x_o}{k}$

so:

$$\Gamma(x) = \frac{1}{24} \sum_{i=1}^{k} \frac{(x_k - x_o)^4}{k^4} y'''(x)$$

$$x \in (x_i, x_{i+1}) \tag{2-82}$$

assuming

$$y'''(\xi_1) = \frac{1}{k}\sum_{i=1}^{k} y'''(x) \qquad \xi_1 \in (x_o, x_k) \qquad (2\text{-}83)$$

then the equation (2-82) can be written as:

$$\Gamma(x) = + \frac{1}{24} \cdot \frac{(x_k - x_o)^4}{k^3} y'''(\xi_1) \quad \xi_1 \in (x_o, x_k) \qquad (2\text{-}84)$$

if the maximum or $y'''(\xi_1)$ in interval $\xi_1 \in (x_o, x_k)$ is M:

$$[y'''(\xi_1)]_{max} = M$$

then the equation (2-84) is written as:

$$\Gamma(x) = + \frac{1}{24} \cdot \frac{(x_k, x_o)^4}{k^3} M \qquad (2\text{-}85)$$

By comparing the error of rectangular method in equation (2-26) and the error of second order method, (eq.(2-84) it is clear that, by increasing k, the error of second order formula is decreasing more rapide than rectangular method to zero. In other words, in second order interpolation the error decreasing k^2 time more rapide than rectangular method.

2.2. The method of integration in incremental computation when the independent variable of integration x is a function of the independent variable t.

We have discused the algorithms of integration in the incremental computation, when the independent variable of integral x is the independent variable t of machine. So we have interpolated the y (x) function by f_{iy} (x) as following:

$$y\ (x) = f_{iy}\ (x) \qquad x = t$$

and the integral formula

$$s\ (x) = \int_{x_o}^{x_k} y\ dx \tag{2-101}$$

was replaced by the approximated integration formula $s^*(x)$ as:

$$s^*(x) = \int_{x_o}^{x_k} f_{iy}\ (x)\ dx \tag{2-102}$$

if the input dx of integrator is not time, but a function of time, like x (t) then the y (x) function will be a function of time t and the equation (2-101) can be written as:

$$s\ (t) = \int_{t_o}^{t_k} y\ (t) \cdot d\ \frac{x(t)}{dt}\ dt \tag{2-103}$$

The integral formula for interval $x \in (x_i, x_{i+1})$ is:

$$\delta s\ (t) = \int_{t_i}^{t_{i+1}} y\ (t) \cdot d\frac{x(t)}{dt} \cdot dt \quad t \in (t_i, t_{i+1}) \qquad (2\text{-}104)$$

for instance for generating $e^{-At} \sin \omega t$

as it is shcun in figure (2-8), the variable of integra'ors (2) and (3) are:

$$X\ (t) = e^{-At} \qquad Y\ (t) = e^{-At} \sin \omega t$$

Therefore each integrator in general can be represented as in figure (2-9).

The function X (t) and Y (t) are replaced with their approximated interpolated value f_{ix} and f_{iy}.

The interpolated functions f_{ix} (t) and f_{iy} (t) in interval $x \in (x_i, x_{i+1})$, as it was discussed before should use the backward information in the form of increments.

As it is seen from figures (2-10) and (2-11) the interpolated functions f_{ix} (t) and f_{iy} (t), in interval $x \in (x_i, x_{i+1})$, use the p backword points information, as following:

$$f_{ix}\ (t) = f_x\ [x_i,\ \delta x_i,\ \delta x_{i-1},\ \delta x_{i-2}, \ldots \delta x_{i-p},$$

$$,\ t_i,\ \delta t_i,\ \delta t_{i-1}, \ldots,\ \delta t_{i-p}] \qquad (2\text{-}105)$$

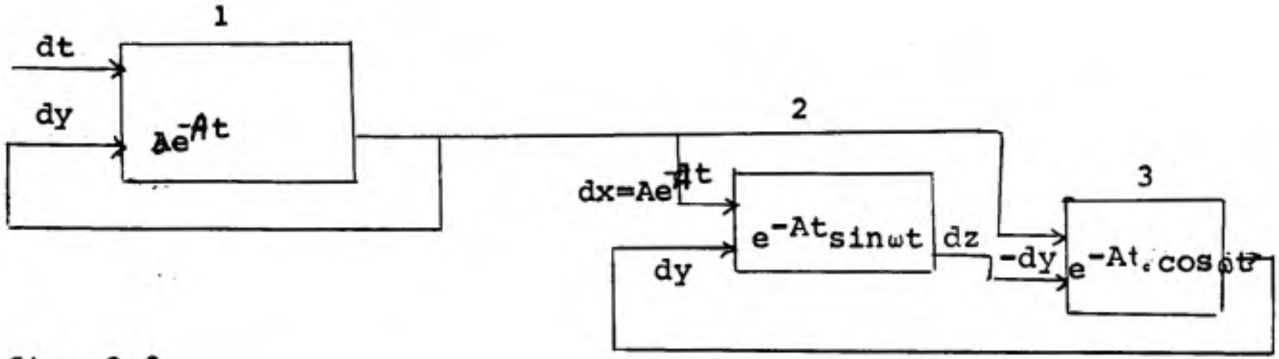

1
dt
dy
Ae-At
2
dx=Ae-At
e-Atsinωt
dz
dy
-dy
3
e-At.cosωt

fig. 2.8.

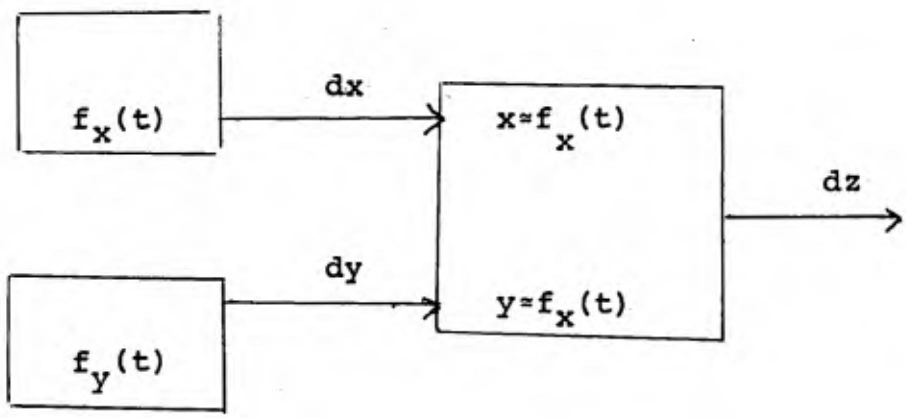

fx(t)
dx
x=fx(t)
dz
dy
y=fx(t)
fy(t)

fig. 2.9.

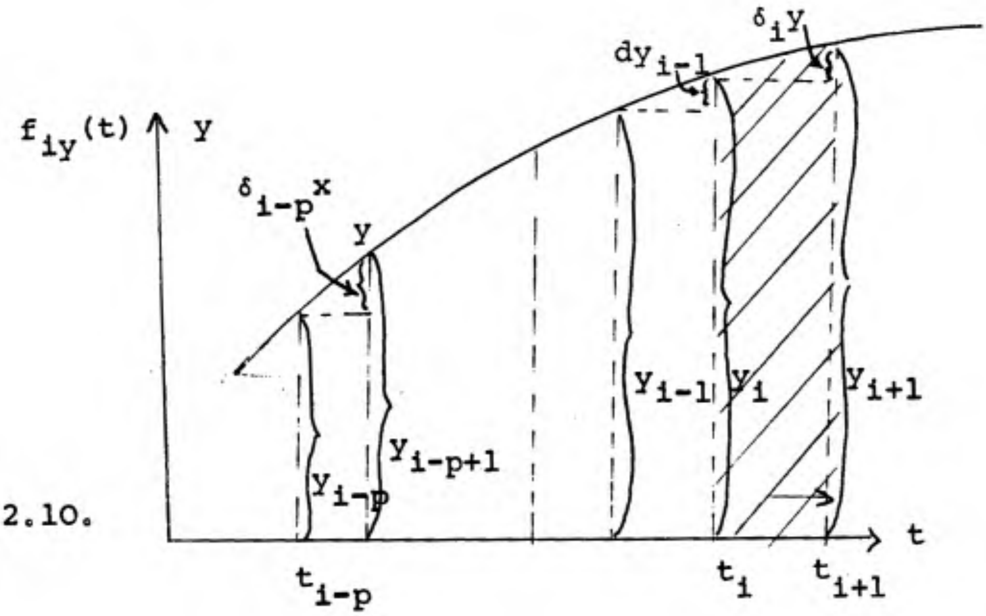


fig. 2.10.

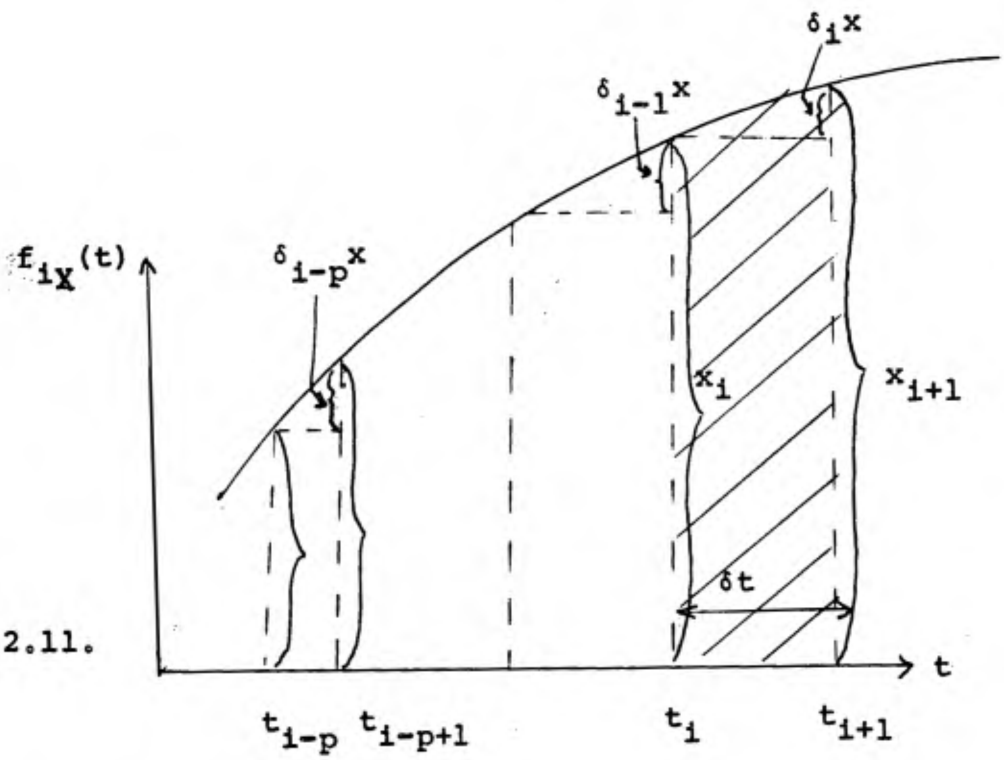


fig. 2.11.

$$f_{iy}(t) = f_y[y_i, y_i, y_{i-1}, y_{i-2}, \ldots, y_{i-p}, t_i, t_i, t_{i-1}, \ldots, t_{i-p}]$$

$$t \in (t_i, t_{i+1})$$

As it was discussed earlier, there are the error ε_{ix} and ε_{iy}, between the actual functions X (t), Y (t) and the interpolated function f_{ix} (t), f_{iy} (t) in interval $t \in (t_i, t_{i+1})$ as:

$$\begin{cases} \varepsilon_{ix}(t) = f_{ix} - X(t) \\ \varepsilon_{iy}(t) = f_{iy} - Y(t) \end{cases} \tag{2-106}$$

if we put the value of equations (2-105) and (2-106) in equation (2-104), we will have:

$$\delta_i\, s(t) = \int_{t_i}^{t_{i+1}} Y(t) \cdot d\frac{X(t)}{dt}\, dt \tag{2-107}$$

$$= \int_{t_i}^{t_{i+1}} [f_{iy} - \varepsilon_{iy}]\, d\frac{[f_{ix} - \varepsilon_{ix}]}{dt}\, dt \tag{2-108}$$

$$= \int_{t_i}^{t_{i+1}} f_{iy}\, d\frac{f_{ix}}{dt}\, dt +$$

$$+ \int_{t_i}^{t_{i+1}} [- \varepsilon_{iy}\, d\, \frac{f_{ix} - \varepsilon_{ix}}{dt}\, dt - f_{yi}\, d\, \frac{\varepsilon_{ix}}{dt}\, dt] \quad (2\text{-}109)$$

$$= \int_{t_i}^{t_{i+1}} f_{iy}\, d\frac{f_{ix}}{dt}\, dt - \int_{t_i}^{t_{i+1}} \varepsilon_{iy}\, d\frac{f_{ix}}{dt}\, dt -$$

$$- \int_{t_i}^{t_{i+1}} f_{yi}\, d\frac{\varepsilon_{ix}}{dt}\, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iy}\, d\frac{\varepsilon_{ix}}{dt}\, dt \quad (2\text{-}110)$$

The integral formula for k interval will be:

$$s\ (t) = \sum_{i=1}^{k} \delta_i\, s\ (t) \quad (2\text{-}111)$$

$$s\ (t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy} \cdot d\frac{f_{ix}}{dt}\, dt + \sum_{i=1}^{k} \left[- \int_{t_i}^{t_{i+1}} \varepsilon_{iy}\, d\frac{f_{ix}}{dt}\, dt - \right.$$

$$\left. - \int_{t_i}^{t_{i+1}} f_{iy}\, d\frac{\varepsilon_{ix}}{dt}\, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iy}\, d\frac{\varepsilon_{ix}}{dt}\, dt \right] \quad (2\text{-}112)$$

The approximated interpolated function of integral in interval

$x \in (x_i,\ x_{i+1})$, is:

$$\delta_i \; s^*(t) = \int_{t_i}^{t_{i+1}} f_{iy} \cdot d\frac{f_{ix}}{dt}\, dt \tag{2-113}$$

which instead of Y (t) and X (t) their approximated interpolated functions f_{ix} , f_{iy} are used. So the approximated integration formula for k interval will be:

$$s^*(t) = \sum_{i=1}^{k} \; s^*(t) \tag{2-114}$$

$$= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy} \cdot d\frac{f_{ix}}{dt}\, dt \tag{2-115}$$

By putting the equation (2-114) and (2-115) in equation (2-112) we will have:

$$s(t) = s^*(t) + \Gamma(t) \tag{2-116}$$

where s (t) is the exact value of integration, $s^*(t)$ is the approximated value of integration, and $\Gamma(t)$ is the error of method which has the following value:

$$\Gamma(t) = \sum_{i=1}^{k} \Big[- \int_{t_i}^{t_{i+1}} \epsilon_{iy} \; d\frac{f_{ix}}{dt}\, dt \; -$$

$$- \int_{t_i}^{t_{i+1}} f_{iy} \, d\frac{\varepsilon_{ix}}{dt} \, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iy} \, d\frac{\varepsilon_{ix}}{dt} \, dt \Bigg] \quad (2\text{-}117)$$

2.2.1. Integration by the general interpolation formula in unitary or multiple increment computation.

As it was discussed before the general formula of integration $\delta_i \, s\,(t)$ in interval $t \in (t_i, t_{i+1})$ is:

$$\delta_i \, s(t) = \int_{t_i}^{t_{i+1}} Y\,(t) \cdot d\frac{X(t)}{dt} \, dt \qquad (2\text{-}118)$$

and the approximated interpolated formula of integral $\delta_i \, s^*(t)$ which is the algorithm of machine is:

$$\delta_i \, s^*(t) = \int_{t_i}^{t_{i+1}} f_{iy} \,(t) \, d\frac{f_{ix}(t)}{dt} \, dt \qquad (2\text{-}119)$$

In order to calculate the approximated interpolated formula of integral $\delta_i \, s^*(t)$ from equation (2-119), we should have f_{iy} and ε_{ix} as following:

$$f_{ix} = x_i - \xi \, \delta x_i - \frac{\xi(\xi+1)}{2!} \, \delta x_i^{(2)} - \frac{\xi(\xi+1)(\xi+2)}{3!} \, \delta x_i^{(3)} - \dots$$

$$f_{iy} = y_i - \xi \cdot \delta y_i - \frac{\xi(\xi+1)}{2!}\delta y_i^{(2)} - \frac{\xi(\xi+1)(\xi+2)}{3!}\delta y_i^{(3)} - \ldots$$

(2-120)

By changing the variable $t \in (t_i, t_{i+1})$ to $\xi \in (-1,0)$ in equation (2-119) we will have:

$$\delta_i \; s^*(t) = \int_0^{-1} f_{iy}\;(\xi) \cdot d\frac{f_{ix}(\xi)}{d\xi}\, d\xi \qquad \xi \in (-1,0) \qquad (2\text{-}121)$$

in order to calculate the equation (2-121) we should calculate the

$d\frac{f_{ix}\;(\xi)}{d\xi}$, this value can be calculated from equation (2-120) as following:

$$d\;\frac{f_{ix}(\xi)}{d\xi} = \delta x_i - \frac{1}{2!}(2\xi+1)\delta_i^2 x - \frac{1}{3!}(3\xi^2+6\xi+2)\delta_i^3 x -$$

$$-\frac{(4\xi^3+18\xi^2+23\xi+6)}{4!}\delta_i^{(4)}x - \ldots.. \qquad (2\text{-}122)$$

by putting the value of $f_{iy}(\xi)$ from equation (2-120) and $d\frac{f_{ix}(\xi)}{d\xi}$ from equation (2-122) in equation (2-121) we will have:

$$\delta_i \; s^* = \int_0^{-1} [y_i - \xi\delta y_i - \frac{\xi(\xi+1)}{2!}\overset{2}{\delta} y_i - \frac{\xi(\xi+1)(\xi+2)}{3!}\overset{3}{\delta} y_i - \ldots].$$

$$\cdot [-\ \delta_i x - \frac{1}{2!} \overset{(2)}{\delta_i} x\ (2\xi+1) - \frac{1}{3!} \overset{(3)}{\delta_i} x\ (3\xi^2 + 6\xi + 2) -$$

$$- \frac{1}{4!} \overset{(4)}{\delta_i} x\ (4\xi^3 + 18\xi^2 + 22\xi + 6) \cdots] d\xi$$

$$= \Big| -y_i \cdot \delta_i x \cdot \xi - \frac{1}{2} y_i\ (\xi^2 + \xi)\ \overset{(2)}{\delta_i} x - \frac{1}{3} y_i\ \overset{(3)}{\delta_i} x\ (\xi^3 + 35^2 + 2\xi) +$$

$$+ \frac{\xi^2}{2} \delta y_i \cdot \delta_i x + \frac{\delta y_i}{2!} \overset{(2)}{\delta x_i} \left(\frac{2\xi^3}{3} + \frac{\xi^2}{2}\right) + \frac{\delta y_i}{3!} \overset{(3)}{\delta x_i} \left(\frac{3\xi^4}{4} + 2\xi^3 + \xi^2\right) +$$

$$+ \frac{1}{2!} \delta y_i^2 \cdot \delta x_i \left(\frac{\xi^3}{3} + \frac{\xi^2}{2}\right) + \frac{\delta x_i^2\ \delta y_i^3}{4} \left(\frac{2\xi^4}{4} + \xi^3 + \frac{\xi^2}{2}\right) +$$

$$+ \frac{1}{3!} \delta y_i^3 \cdot \delta x_i \left(\frac{\xi^4}{4} + \xi^3 + \xi^2\right) + \ldots \Big|_0^1 \qquad (2\text{-}123)$$

$$= + y_i\ \delta x_i + \frac{1}{2} \delta y_i \cdot \delta x_i + \frac{1}{12} \delta y_i^2 \cdot \delta x_i - \frac{1}{12} \delta y_i \cdot \delta x_i^2 + \ldots$$

$$+ \frac{1}{24} \delta y_i^3 \cdot \delta_i x - \frac{1}{24} \delta y_i \cdot \delta_i^3 x + \ldots\ldots \qquad (2\text{-}124)$$

The approximated interpolated formula of integral $s^*(x)$ for k interval will be:

$$s^*(x) = \sum_{i=1}^{k} [y_i\ \delta x_i + \frac{1}{2} \delta y_i \cdot \delta x_i + \frac{1}{12} (\overset{(2)}{\delta} y_i\ \delta x_i - \delta y_i\ \overset{(2)}{\delta x_i}) +$$

$$+\frac{1}{24}\left(\delta\, y_i^{(3)}\ \delta_i x - \delta y_i \cdot \delta_i^{(3)} x + \dots\ \right] \tag{2-125}$$

The equation (2-125) is the general formula of integration which is based on the Newton interpolation formula.

2.2.2. Integration by the interpolated rectangular formula in unitary or multiple increment computation.

By choosing the first terms of equation (2-124), the algorithm of integral in interval $t \in (t_i, t_{i+1})$ will be:

$$\delta_i\ s^* = y_i\ \delta x_i \tag{2-126}$$

the $\delta_i\ s^*$ is the approximated formula of integration which use the information of point (x_i, y_i) in interval $t \in (t_i, t_{i+1})$, so the equation (2-124) can be written as:

$$\delta_i\ s(x,y) = \delta_i\ s^*(x,y) + \Gamma_i\ (x,y) \tag{2-127}$$

where $\delta_i\ s\ (x,y)$ is the actual value of integral, in interval $t \in$ $\in (t_i, t_{i+1})$, $\delta_i\ s^*(x,y)$ is the approximated value of integral which is the algorithm of machine, and $\Gamma_i\ (x,y)$ is the error of method in this interval, that can be calculated from equations (2-124), (2-126) and (2-127) as following:

$$\Gamma_i\ (x,y)\ = \frac{1}{2}\delta y_i \cdot \delta x_i + \frac{1}{12}\left(\delta^{(2)} y_i\ \delta x_i - \delta y_i\ \delta x_i^{(2)}\right) + \dots$$

$$+\frac{1}{24}\,(\delta y_i^{(3)} \cdot \delta_i x - \delta y_i \cdot \delta_i x^{(3)}) + \ldots \qquad (2\text{-}128)$$

by using the equation (2-127) we can find the integral formula for k interval as:

$$s\ (x,y) = \sum_{i=1}^{k} \delta_i\ s(x,y) \qquad (2\text{-}129)$$

$$= \sum_{i=1}^{k} \delta_i\ s^{*}(x,y) + \sum_{i=1}^{k} \Gamma_i\ (x,y) \qquad (2\text{-}130)$$

The algorithms of integration for all k interval, can be found from equation (2-126) and (2-130) as following:

$$s^{*}(x,y) = \sum_{i=1}^{k} \delta_i\ s^{*}(x,y) \qquad (2\text{-}131)$$

$$= \sum_{i=1}^{k} y_i \cdot \delta x_i$$

so the equation (2-120) can be written as:

$$s(x,y) = s^{*}(x,y) + \Gamma(x,y) \qquad (2\text{-}132)$$

where $s(x,y)$ is the actual value of integration, $s^{*}(x,y)$ is the approximated value of integration which is the algorithm of machine, and $\Gamma(x,y)$ is the error of method which is equal to:

$$\Gamma(x,y) = \sum_{i=1}^{k}\left[\frac{1}{2}\delta y_i \cdot \delta_i x + \frac{1}{12}\left(\delta y_i^{(2)} \delta_i x - \delta y_i \cdot \delta_i x^{(2)}\right) + \right.$$

$$\left. + \frac{1}{24}\left(\delta^{(3)} y_i \cdot \delta_i x - \delta y_i \cdot \delta x_i^{(3)}\right) + \ldots\ldots \right] \quad (2-133)$$

2.2.3. Integration by the interpolated trapezoidal formula in unitary or multiple increment computation.

In trapezoidal method of integration, the Y (t) and X (t) functions are interpolated lineary, in other words, it use the information of two points t_i and t_{i+1}, in interval $t \in (t_i, t_{i+1})$ as it is shcun in figure (2-12) and at the equation (2-134).

$$\begin{cases} X\ (t) = f_{ix}\ (x_i,\ x_{i+1},\ t_i,\ t_{i+1}) \\ Y\ (t) = f_{iy}\ (y_i,\ y_{i+1},\ t_i,\ t_{i+1}) \end{cases} \quad (2-134)$$

by choosing the first two terms of equation (2-124) wi will have:

$$\delta_i\ S^{*}(x) = y_i \cdot \delta x_i + \frac{1}{2}\delta y_i \cdot \delta x_i \quad (2-135)$$

The equation (2-124) can be written as:

$$\delta_i\ s(x) = \delta_i\ S^{*}(x) + \Gamma_i\ (x) \quad (2-136)$$

where δ_i s (x) is the actual value of integration, $\delta_i\ s^{*}(x)$ is the approximated value, and Γ_i (x) is the error of method in interval

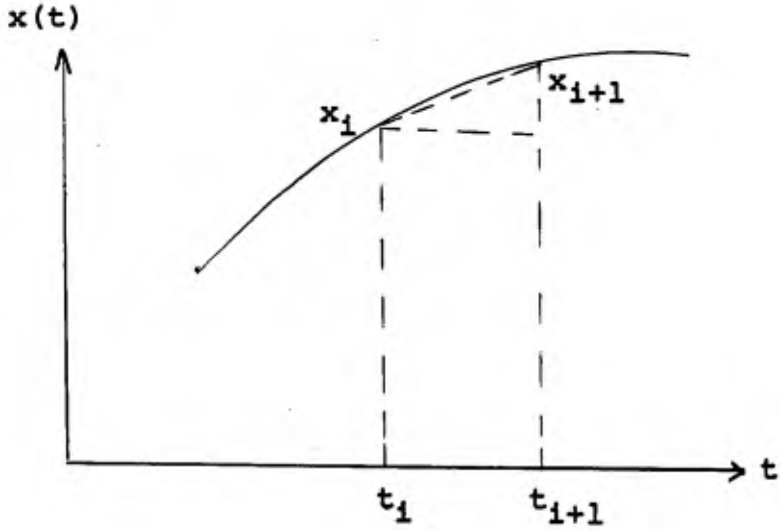
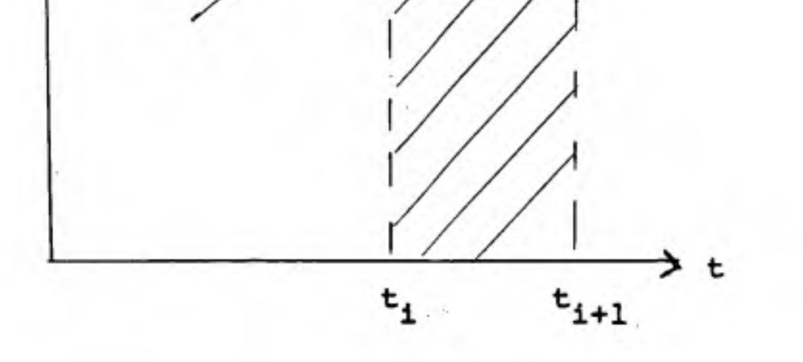


fig. 2.12.



$x \in (x_i, x_{i+1})$ which from equation (2-125),(2-135) and (2-136) can be calculated as:

$$\Gamma_i\ (x) = \frac{1}{12}\ (\delta_i^{(2)}y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(2)}x) + \frac{1}{24}\ (\delta_i^{(3)}y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(3)}x) + \ldots\ldots \tag{2-137}$$

The integral formula S (x) for k interval can be found from equation (2-136) as following:

$$S\ (x) = \sum_{i=1}^{k} \delta_i\ s\ (x) \tag{2-138}$$

$$= \sum_{i=1}^{k} \delta_i\ S^*(x) + \sum_{i=1}^{k} \Gamma_i\ (x) \tag{2-139}$$

The algorithm of integration in k interval from equation (2-135) is:

$$S^*(x) = \sum_{i=1}^{k} \delta_i\ s^* \tag{2-140}$$

$$= \sum_{i=1}^{k} (y_i \cdot \delta x_i + \frac{1}{2} \delta x_i \cdot \delta y_i) \tag{2-141}$$

The equation (2-139) can be written as:

$$S\ (x) = S^*(x) + \Gamma(x) \tag{2-142}$$

where S (x) is the actual integration function, $S^*(x)$ is the approximated integration function and $\Gamma(x)$ is the error of method in k interval which is equal to:

$$\Gamma(x) = \sum_{i=1}^{k} \Gamma_i(x) \qquad (2\text{-}143)$$

$$= \sum_{i=1}^{k} \left[\frac{1}{12} (\delta_i^{(2)}y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(2)}x) + \right.$$

$$\left. + \frac{1}{24} (\delta_i^{(3)}y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(3)}x) + \ldots\ldots \right] \qquad (2\text{-}144)$$

2.2.4. Integration by the interpolated three points formula in unitary or multiple increment computation.

In the three points interpolation method of integration, the interpolated function f_{ix} (t) and f_{iy} (t) use the information of three points (x_{i+1}, y_{i+1}), (x_i, y_i) and (x_{i-1}, y_{i-1}) in interval $x \in (x_i, x_{i+1})$ as following

$$\begin{cases} X(t) \approx f_{ix} = f(x_{i+1}, x_i, x_{i-1}, t) \\ Y(t) \approx f_{iy} = f(y_{i+1}, y_i, y_{i-1}, t) \end{cases} \qquad (2\text{-}145)$$

By choosing the first three therms of equation (2-124), we will have:

$$\delta_i\, s^* = y_i \cdot \delta x_i + \frac{1}{2}\delta x_i \cdot \delta y_i + \frac{1}{12}\left(\delta y_i^{(2)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(2)}\right) \tag{2-146}$$

by putting the value of $\delta y_i^{(2)}$, $\delta x_i^{(2)}$ by its equivalent,

$$\begin{aligned} \delta y_i^{(2)} &= \delta y_i - \delta y_{i-1} \\ \delta x_i^{(2)} &= \delta x_i - \delta x_{i-1} \end{aligned} \tag{2-147}$$

then the equation (2-146) can be changed to:

$$\delta_i\, s^* = y_i \cdot \delta x_i + \frac{1}{2}\delta x_i \cdot \delta y_i + \frac{1}{12}\left(\delta y_i \cdot \delta x_{i-1} - \delta y_{i-1} \cdot \delta x_i\right) \tag{2-148}$$

The equation (2-148) is the algorithm of integration with three points interpolation, which use the information of points (x_{i+1}, y_{i+1}), (x_i, y_i) and (x_{i-1}, y_{i-1}) in the incremental forms. As it was mentionned earlier, this algorithm has the property of smoothing effect as it is shoun in figure (2- 5). The actual value of integration s (x) is:

$$s(x) = s^*(x) + \Gamma(x) \tag{2-149}$$

where $s^*(x)$ is the algorithm of integration for k interval as:

$$s^*(x) = \sum_{i=1}^{k} \delta_i\, s^*$$

$$= \sum_{i=1}^{k} [y_i \cdot \delta x_i + \frac{1}{2}\delta x_i \cdot \delta y_i +$$

$$+ \frac{1}{12} (\delta y_i \cdot \delta x_{i-1} - \delta y_{i-1} \cdot \delta x_i)] \qquad (2\text{-}150)$$

with the error of method equal to:

$$\Gamma(x) = \sum_{i=1}^{k} \Gamma_i (x)$$

$$= \sum_{i=1}^{k} \left[\frac{1}{24} (\delta y_i^{(3)} \cdot \delta_i x - \delta y_i \cdot \delta_i x^{(3)}) + \ldots \right] \qquad (2\text{-}151)$$

2.3. The error of method in incremental computation, when the independent variable of integration is a function of the independent variable t.

2.3.1. Error of the rectangular method of integration in unitary or multiple increment computation.

As it was discussed before, the general formula of integration which is used in incremental computer is based on Newton interpolation (eq. 2-124) as following:

$$\delta_i s^* = y_i \cdot \delta x_i + \frac{1}{2}\delta y_i \cdot \delta x_i + \frac{1}{12}(\delta y_i^{(2)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(2)}) +$$

$$\frac{1}{24}(\delta y_i^{(3)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(3)}) + \ldots\ldots \qquad (2\text{-}152)$$

In the rectangular method of integration, the interpolated function f_{ix} and f_{iy} use one point information, in other words, the approximated formula of integration is the first term of equation (2-152) as:

$$\delta_i s^* = y_i \cdot \delta x_i \qquad (2\text{-}153)$$

The approximated formula of integration $S^*(x)$ has the error $r_i(x,y)$ with the actual value of integral $\delta_i s$ as:

$$\delta_i s = \delta_i s^* + \Gamma_i (x,y) \qquad t \in (t_i, t_{i+1}) \quad (2-154)$$

where

$$\Gamma_i (x,y) = + [\frac{1}{2}\delta y_i \cdot \delta x_i + \frac{1}{12}(\delta y_i^{(2)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(2)}) + \ldots]$$

$$t \in (t_i , t_{i+1}) \quad (2-155)$$

By neglecting the second and higher terms of $\Gamma_i (x,y)$ with respect to the first term, we will have:

$$\Gamma_i (x,y) = +\frac{1}{2}\delta x_i \cdot \delta y_i \quad (2-156)$$

The exact integration formula for k interval from equation (2-154) will be:

$$s (x,y) = \sum_{i=1}^{k} \delta_i s \quad (2-157)$$

$$= \sum_{i=1}^{k} \delta_i s^* + \sum_{i=1}^{k} \Gamma_i (x,y) \quad (2-158)$$

where the approximated value of integration is:

$$s^*(t) = \sum_{i=1}^{k} \delta_i s^* \quad (2-159)$$

and the error of method in k interval will be:

$$\Gamma(t) = \sum_{i=1}^{k} \Gamma_i (x,y) \quad (2-160)$$

so the equation (2-158) can be written as:

$$s\ (x,y) = S^{*}\ (x,y) + \Gamma(t) \tag{2-161}$$

in equation (2-161) the actual value of integratal is s (x,y), the approximated value of integral which is the algorithm of machine is $s^{*}(x,y)$, and the error of method $\Gamma(x)$ is as following:

$$\Gamma(t) = \sum_{i=1}^{k} \Gamma_i\ (x,y) \tag{2-162}$$

$$= +\frac{1}{2} \sum_{i=1}^{k} \delta x_i \cdot \delta y_i \qquad t \in (t_i, t_{i+1}) \tag{2-163}$$

assuming

$$\delta x_i = x'(t) \cdot \delta t$$

and

$$\delta y_i = y'(t) \cdot \delta t$$

The equation (2-163) can be written as:

$$\Gamma(t) = +\frac{1}{2} \sum_{i=1}^{k} \cdot (\delta t)^2 \cdot y'(t) \cdot x'(t)$$

$$t \in (t_i, t_{i+1}) \tag{2-164}$$

In the case of unitary increment computer

$$\delta y = \Delta y \text{ and } \delta x = \Delta x = \frac{x_k - x_o}{k}$$

so the equation (2-164) can be written as:

$$\Gamma(t) = +\frac{1}{2}\sum_{i=1}^{k} \frac{(t_k-t_o)^2}{k^2} y'(t) \cdot x'(t) \qquad (2\text{-}165)$$

$$t \in (t_i, t_{i+1})$$

assuming

$$y'(\xi_i) \cdot x'(\xi_i) = \frac{1}{k}\sum_{i=1}^{k} y'(t) \cdot x'(t) \qquad \xi_i \in (t_o, t_k)$$

The equation (2-165) can be expressed as following:

$$\Gamma(t) = +\frac{1}{2}\frac{(t_k-t_o)^2}{k} y'(\xi_i) \cdot x'(\xi_i) \quad \xi \in (t_o, t_k) \qquad (2\text{-}166)$$

As it is seen from equation (2-166), the error of method $\Gamma(t)$ in k interval depends to the interval $t \in (t_o, t_k)$ and also to the derivative of functions x (t) and y (t) which are applied in the input of integrator.

If we assume the maximum value fo $y'(\xi)$ as M_1

$$[y'(\xi)]_{max} = M_1 \qquad (2\text{-}167)$$

and the maximum value of $x'(\xi)$ equal to M_2

$$[x'(\xi)]_{max} = M_2 \qquad (2\text{-}168)$$

then the maximum error of rectangular method will be:

$$\Gamma(t) = +\frac{1}{2}\frac{(t_k-t_o)^2}{k} M_1 \cdot M_2 \qquad (2\text{-}169)$$

2.3.2. Error of the trapezoidal method of integration in unitary or multiple increment computation.

In trapezoidal method of integration, the interpolated function f_{ix} (t) and f_{iy} (t) which are replaced to x (t), y (t) are using two backwords points information, in other words the approximated value of integration is the first two terms of equation (2-124) as following:

$$\delta_i \, s^* = y_i \cdot \delta x_i + \frac{1}{2} \delta x_i \cdot \delta y_i \qquad (2\text{-}170)$$

The equation (2-152) can be written as:

$$\delta_i \, s = \delta_i \, s^* + \Gamma_i \, (t) \qquad (2\text{-}171)$$

where δ_i s is the actual value of integration, $\delta_i \, s^*$ is the approximated value and Γ_i (t) is the error of method in interval $t \in (t_i, t_{i+1})$ which is equal to:

$$\Gamma_i \, (t) = + \frac{1}{12} (\delta y_i^{(2)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(2)}) + + \frac{1}{24} (\delta y_i^{(3)} \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(3)}) + \ldots \qquad (2\text{-}172)$$

from equation (2-171) the actual integration formula for k interval will be:

$$s \, (t) = \sum_{i=1}^{k} \delta_i \, s \qquad (2\text{-}173)$$

$$= \sum_{i=1}^{k} \delta_i s^* + \sum_{i=1}^{k} \Gamma_i (t) \tag{2-174}$$

$$= s^*(t) + \Gamma(t) \tag{2-175}$$

where $s^*(t)$ is the approximated value of integral which is the algc rithm of machine and the $\Gamma(t)$ is the error of method in k interval that is equal to:

$$\Gamma(t) = \sum_{i=1}^{k} \Gamma_i (t) \tag{2-176}$$

$$= \sum_{i=1}^{k} [\frac{1}{12} (\delta_i^{(2)} y \cdot \delta x_i - \delta y_i \cdot \delta x_i^{(2)})$$

$$+ \frac{1}{24} (\delta_i^{(3)} y \cdot \delta_i x - \delta y_i \cdot \delta x_i^{(3)}) + \ldots] \tag{2-177}$$

by neglecting the second, third,...paranthesis of equation (2-177) with respect to the first one, we will have:

$$\Gamma(t) = \sum_{i=1}^{k} \frac{1}{12} (\delta_i^{(2)} y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(2)} x)$$

$$t \in (t_i, t_{i+1}) \tag{2-178}$$

by using

$$\delta_i y = y' (t) \cdot \delta t \quad , \quad \delta_i x = x' (t) \cdot \delta t$$

$$\delta_i^{(2)} y = y''(t) \cdot (\delta t)^2, \ \delta_i^{(2)} x = x''(t) \cdot (\delta t)^2$$

The equation (2-178) can be written as:

$$\Gamma(x) = \sum_{i=1}^{k} \frac{(\delta t)^3}{12} \left[y''(t) \cdot x'(t) - y'(t) \cdot x''(t)\right]$$

$$t \in (t_i, t_{i+1}) \qquad (2\text{-}179)$$

when unitary increment is used $\delta x = \Delta x$ and $\delta y = \Delta y$, by assuming:

$$y''(\underset{'}{\xi}) \cdot x'(\underset{'}{\xi}) = \frac{1}{k} \sum_{i=1}^{k} y''(t) \cdot x'(t)$$

$$y'(\underset{'}{\xi}) \cdot x''(\underset{'}{\xi}) = \frac{1}{k} \sum_{i=1}^{k} y'(t) \cdot x''(t) \qquad \underset{'}{\xi} \in (t_o, t_k)$$

and

$$\Delta t = \frac{t_k - t_o}{k}$$

the equation (2-179) can be written as following:

$$\Gamma(t) = \frac{(t_k - t_o)^3}{12 \cdot k^2} \left[y''(\underset{'}{\xi}) \cdot x'(\underset{'}{\xi}) - y'(\underset{'}{\xi}) \cdot x''(\underset{'}{\xi})\right]$$

$$\underset{'}{\xi} \in (t_o, t_k) \qquad (2\text{-}180)$$

As it is seen from equation (2-180) the error of method depends to the first and second derivative of function x (t) and y (t) which are applied to the inputs of integrator, and also to the interval of integration $t_k - t_o$.

2.3.3. Error of the three points method of integration in unitary or multiple increment computation.

In three points interpolation formula, the interpolation function f_{ix} (t) and f_{iy} (t) which are replaced to x (t) and y (t) are using three backwords information points, in other words the approximated value of integration is the first three terms of equation (2-152) as:

$$s^{*}(t) = y_i\ \delta x_i + \frac{1}{2}\delta x_i \cdot \delta y_i + \frac{1}{12}\left(\delta_i^{(2)}y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(2)}x\right) \tag{2-181}$$

the equation (2-152) can be written as following:

$$\delta_i\ s(t) = \delta_i\ s^{*}(t) + \Gamma_i\ (t) \tag{2-182}$$

where δ_i s (t) is the actual value of integration, $\delta_i\ s^{*}(t)$ is the algorithm of integration and Γ_i (t) is the error of method in interval $t \in (t_i, t_{i+1})$. The error of mehtod in interval Γ_i (t) can be find from equation (2-152), (2-181) and (2-182) as following:

$$\Gamma_i\ (t) = +\left[\frac{1}{24}\left(\delta_i^{(3)}y \cdot \delta_i x - \delta y_i \cdot \delta_i^{(3)}x\right) + \ldots\ldots\right] \tag{2-183}$$

The value of integral in k interval is:

$$s\ (t) = \sum_{i=1}^{k} \delta_i\ s\ (t) \tag{2-184}$$

$$= \sum_{i=1}^{k} \delta_i\, s^*(t) + \sum_{i=1}^{k} \Gamma_i\,(t) \qquad (2\text{-}185)$$

where s (t) is the actual value of integral in k interval, $s^*(t)$ is the algorithm of integration in k interval, and

$$\Gamma(t) = \sum_{i=1}^{k} \Gamma_i\,(t)$$

is the error of method in k interval. The value of $s^*(t)$ and $\Gamma(t)$ can be find as:

$$s^*(t) = \sum_{i=1}^{k} \delta_i\, s^*(t) \qquad (2\text{-}186)$$

$$= \sum_{i=1}^{k} [\, y_i \cdot \delta x_i + \frac{1}{2} \delta x_i \cdot \delta y_i + \frac{1}{12} (\delta_i^{(2)} y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(2)} x\,)\,] \qquad (2\text{-}187)$$

$$\Gamma(t) = \sum_{i=1}^{k} \Gamma_i(t) \qquad (2\text{-}188)$$

$$= \sum_{i=1}^{k} [\frac{1}{24} (\delta_i^{(3)} y \cdot \delta_i x - \delta_i y \cdot \delta_i^{(3)} x\,) + \ldots] \qquad (2\text{-}189)$$

by putting

$$\begin{bmatrix} \delta_i\, y = y'\,(t) \cdot \delta t \\ \delta_i^{(3)} y = y'''(t) \cdot (\delta t)^3 \end{bmatrix} \qquad \begin{bmatrix} \delta_i\, x = x'\,(t) \cdot \delta t \\ \delta_i^{(3)} x = x'''(t) \cdot (\delta t)^3 \end{bmatrix}$$

in equations (2-188) and (2-189) we can write:

$$\Gamma(t) = \sum_{i=1}^{k} \left[\frac{(\delta t)^4}{24}[y'''(t) \cdot x'(t) - y'(t) \cdot x'''(t)] + \ldots \right] \tag{2-190}$$

if we neglect the higher order terms with respect to first two terms, the error of method in k interval will be:

$$\Gamma(t) = \sum_{i=1}^{k} \frac{(\delta t)^4}{24}[y'''(t) \cdot x'(t) - y'(t) \cdot x'''(t)] \tag{2-191}$$

$$t \in (t_i, t_{i+1})$$

in unitary increment computation $\delta t = \Delta t$, $\delta y = \Delta y$ and $\delta x = \Delta x$ assuming:

$$y'''(\xi_{,}) \cdot x'(\xi_{,}) = \frac{1}{k} \sum_{i=1}^{k} y'''(t) \cdot x'(t) \qquad \xi_{,} \in (t_o, t_k)$$

$$y'(\xi_{,}) \cdot x'''(\xi_{,}) = \frac{1}{k} \sum_{i=1}^{k} y'(t) \cdot x'''(t) \qquad \xi_{,} \in (t_o, t_k)$$

and $$\Delta t = \frac{t_k - t_o}{k}$$

The equation (2-191) can be written as:

$$\Gamma(t) = \frac{(t_k - t_o)^4}{24\, k^3} [y'''(\xi_{,}) \cdot x'(\xi_{,}) - y'(\xi_{,}) \cdot x'''(\xi_{,})] \tag{2-192}$$

$$\xi_{,} \in (t_o, t_k)$$

The equation (2-192) is the error of method $\Gamma(t)$ in three points interpolation formula, which depends on the first and third derivative of functions $x(t)$ and $y(t)$.

2.4. Conclusion.

In this chapter, we calculated the error of method $\Gamma(t)$ in incremental computation, in the general case, where the increments δx, δy and δs can take any values. Therefore, these computations are valid for multiple incremental computations, ($\delta x = 2^r \cdot \Delta x$, $\delta y = 2^r \cdot \Delta y$, $\delta s = 2^r \cdot \Delta s$) as well as for unitary incremental computation ($\delta x = \Delta x$, $\delta y = \Delta y$, $\delta s = \Delta s$).

Unitary incremental computation, which is used in digital differential analyzer (D.D.A) is a special case of multiple incremental computation ($r = 0$.

We have calculated the error of method $\Gamma(t)$, for methods of integration, when the independent variable of integral X is equal to or is a function of the independent variable t of machine.

Consequently, the error of method $\Gamma(t)$ is the difference between the value of integral s (t) and the approximated interpolated value of integral $s^*(t)$:

$$\Gamma(t) = s(t) - s^*(t) \qquad (2\text{-}216)$$

The error of method $\Gamma(t)$ depends on the degree of the interpolation formula used for the algorithms of integration.

In the following table, we are comparing the different algorithms of integration and their respective error for unitary incremental computation, when the independent variable of integral X is equal to the independent variable t of machine.

Method of integration	Algorithm of integration	Error of method
Rectangular method (zero degree interpolation)	$s^{*}(t) = \sum_{i=1}^{k} y_i \circ \Delta_i x$	$\Gamma(t) < \frac{7}{2}(x_k - x_o)\ \Delta y$
Trapezoidal method (first degree interpolation)	$s^{*}(t) = \sum_{i=1}^{k} (y_i \circ \Delta_i x + \frac{1}{2}\Delta_i y \circ \Delta_i x)$	$\Gamma(t) < \frac{7}{12}(x_k - x_o)\ \Delta y$
Three points method (second degree interpolation)	$s^{*}(t) = \sum_{i=1}^{k} [y_i \circ \Delta_i x + \frac{1}{2}\Delta_i y \circ \Delta_i x + (\frac{1}{12}\Delta_i y \circ \Delta_{(i-1)} x - \frac{1}{12}\Delta_{(i-1)} y \circ \Delta_i x)]$	$\Gamma(t) < \frac{7}{12}(x_k - x_o)\ \Delta y$

As it is seen from the table, the choice of the trapezoidal method (first degree interpolation), instead of the rectangular method (zero degree interpolation), the error of method is reduced by the factor $\frac{7/12}{7/2} = 1/6$. Therefore, the trapezoidal method is much more accurate than the rectangular method.

As it is seen from the algorithm of three points method, for unitary incremental computation, the values of the paranthesis is smaller than the quantum Δy of function y (x). Threrfore, in unitary incremental computation, it is not worthwhile to use higher degree interpolation formula than the first one. So, the first degree interpolation formula, known as trapezoidal method, is good approximation for unitary incremental computation.

On the other hand, when we use the multiple incremental computation, where the step of integration is 2^r larger than unitary incremental computation ($\delta x = 2^r \cdot \Delta x$, $\delta y = 2^r \cdot \Delta y$ and $\delta s = 2^r \cdot \Delta s$), we should use more accurate integration formula.

The three points method of integration (second degree interpolation) is a more convenient one for multiple increments computation:

$$s^{**}(t) = \sum_{i=1}^{k} [y_i \cdot \delta_i x + \frac{1}{2}\delta_i y \cdot \delta_i x +$$

$$+ \frac{1}{12} (\delta_i y \cdot \delta_{(i-1)} x - \delta_{(i-1)} y \cdot \delta_i x)]$$

If the degree of the interpolation formula increased to higher than two, then the integration formula will become more complex, because it needs too much equipment and operating time to do the integration.

In the increment computer of industrial electronic laboratory of the Brussel University, which is devised by the author, the algorithms of integration in unitary incremental computation, can be chosen either rectangular or trapezoidal method, and in multiple incremental computation, it is the three points method.

CHAPTER III

THE QUANTIZATION AND ERROR IN INCREMENTAL COMPUTATION.

3.1. The quantization process in incremental computation.

In incremental computation, the results of a given mathematical operation is transmitted for use in another mathematical operation by use of quantized increments. The operation of quantization of continuous function y (x), may be done by the quantum of independent variable Δx, or by the dependent variable Δy. The more natural quantization which is done in the incremental computation, is the complete quantization with respect to the quantum Δx and Δy with inherent delay of digital system.

As we have discussed earlier, the more general integration operation in incremental computer is:

$$s\ (x) = \int_{t_o}^{t_k} Y\ (t) \cdot d\ \frac{X\ (t)}{dt}\, dt \qquad t \in (t_o, t_k) \qquad (3\text{-}1)$$

We have seen in chapter 2, that the continuous functions Y (t) and X (t) which have the information of infinits points, were approximated interpolated to the functions f_{iy} (t) and f_{ix} (t) which have the information of finits points $[(x_i, y_i), (x_{i-1}, y_{i-1}), \ldots t_i \ (i=1,2,\ldots k)]$

In this case there will be an error between the actual functions X (t), Y (t) and the approximated interpolated functions f_{ix}, f_{iy} equals to:

$$\begin{cases} \varepsilon_{ix} = f_{ix}\ (t) - X\ (t) \\ \varepsilon_{iy} = f_{iy}\ (t) - Y\ (t) \end{cases} \tag{3-2}$$

The functions f_{ix} and f_{iy} can be represented as:

$$\begin{cases} f_{ix} = f_x\ [x_i,\ x_{i-1}, \ldots\ t_i \quad (i=1,2,\ldots .k)] \\ f_{iy} = f_y\ [y_i,\ y_{i-1}, \ldots\ t_i \quad (i=1,2,\ldots ,k)] \end{cases} \tag{3-3}$$

In the quantization process of incremental computation, the ranges of magnitude f_{ix} (t) and f_{iy} (t) are divided into interval k which are not necessary equal. All the magnitude falling within each interval are quantized (equaled) to a single value within the interval of the analog inputs signal X (t) and Y (t), as it is shcwn in figures (3-1) and (3-2).

Therefore the incremental machine, instead of using the information of points $[\ (x_i,\ y_i),\ (x_{i-1},\ y_{i-1}), \ldots .\ t_i \quad (i=1,2,\ldots k)\ ]$ use the quantized points $[\ (x_{iQ},\ y_{iQ}),\ (x_{(i-1)Q},\ y_{(i-1)Q}), \ldots .\ t_{iQ}$

$(i=1,2,\ldots.k)$]

So the approximated interpolated functions f_{ix}, f_{iy} are quantized and converted to the approximated interpolated and quantized functions f_{iQx} (t), f_{iQy} (t), which have the error of quantization ε_{iQx}, ε_{iQy} in each points (x_i, y_i) with the unquantized interpolated functions f_{ix}, f_{iy} as following:

$$\begin{cases} \varepsilon_{iQx} = f_{ix}(t) - f_{ixQ}(t) \\ \\ \varepsilon_{iQy} = f_{iy}(t) - f_{iyQ}(t) \end{cases} \tag{3-4}$$

the functions f_{iQx} (t), f_{iQy} (t) can be written as:

$$\begin{cases} f_{iQx}(t) = f_{iQx}[x_{iQ}, x_{(i-1)Q}, \ldots\ldots t_{iQ} \quad (i=1,2,\ldots k)] \\ f_{iQy}(t) = f_{iQy}[y_{iQ}, y_{(i-1)Q}, \ldots\ldots t_{iQ} \quad (i=1,2,\ldots k)] \end{cases} \tag{3-5}$$

As in incremental computation the quantities are represented in increments $\delta_i x$, $\delta_i y$, the equation (3-5) can be written as:

$$\begin{cases} f_{iQx}(t) = f_{iQx}[x_{iQ}, \delta_{iQ}x, \delta_{(i-1)Q}x, \ldots\ldots t_{iQ} \ (i=1,2,\ldots k)] \\ f_{iQy}(t) = f_{iQy}[y_{iQ}, \delta_{iQ}y, \delta_{(i-1)Q}y, \ldots\ldots t_{iQ} \ (i=1,2,\ldots k)] \end{cases} \tag{3-6}$$

The quantization process in incremental computer cause the error of quantization ε_{iQx}, ε_{iQy} which is the difference between the quantized and unquantized value of function in each points (x_i, y_i) that cause the total error of quantization ε_{tQ} in the process of incremental computation. Considering the above discussion, the block diagram of incremental computer can be represented as in figure (3-3)

The quantization process has the influence in the mathematical operations of incremental computer and the choice of the algorithms of integration. The total quantization error ϵ_{tQ} should not exceed from some acceptable limit. In foregoing paragraph, we will study the quantization process, the error of quantization and also the inherent delay in process of quantization.

3.1.1. Quantization in incremental computation by the independent variable X, and algorithm of quantized points.

The quantized point is the intersection between the line $-n\Delta x, \ldots \quad -2\Delta x, -\Delta x, 0, \Delta x, 2\Delta x, \ldots \quad +n\Delta x$, and the f_y (x) function which is shown in fig. (3.4.a). But in digital iteration machine, because the time of mathematical operations, the quantized points have always the delay with respect to the original continuous fonction y (x), the maximum of the delay is equal to one quantum of Δx, as it is shown in figure (3.4.b).

The delay of quantizated function with respect to the continuous function, present an error ϵ_{iQx} in each point of quantization which is the difference between the continuous function f_y (x) and quantized function f_{yQ} in that point as:

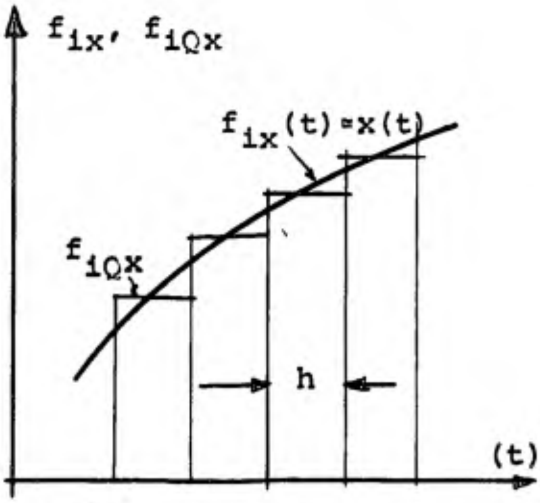


FIG.:3-1

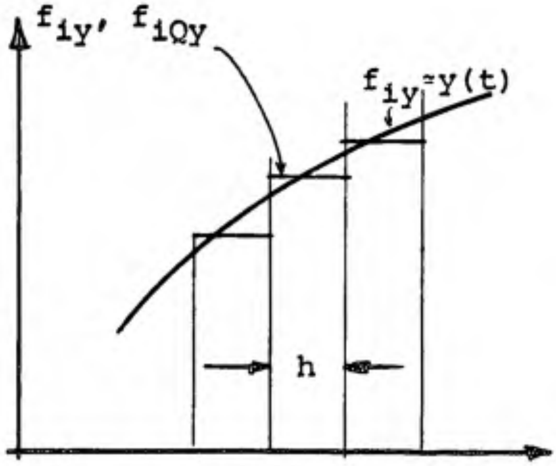


FIG.:32

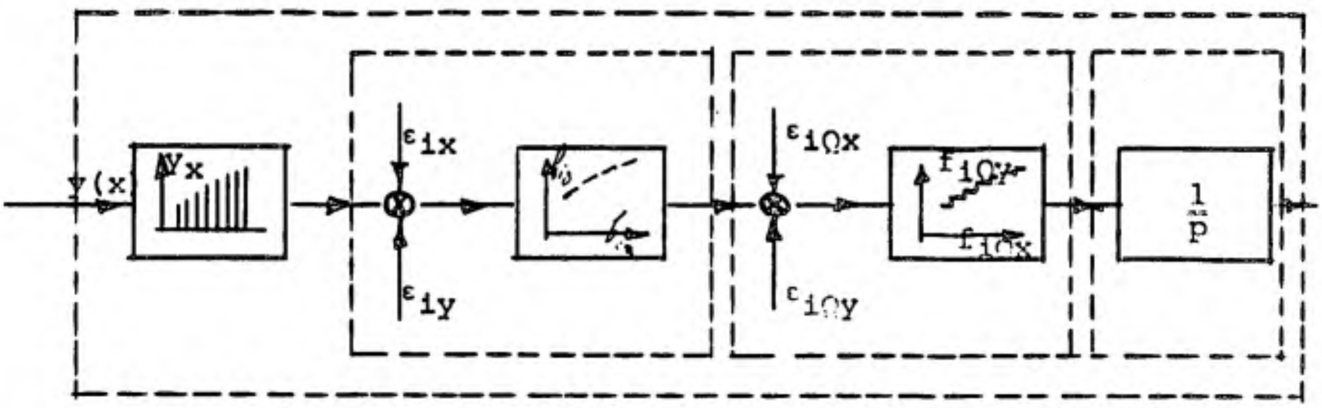


FIG.: 3-3.

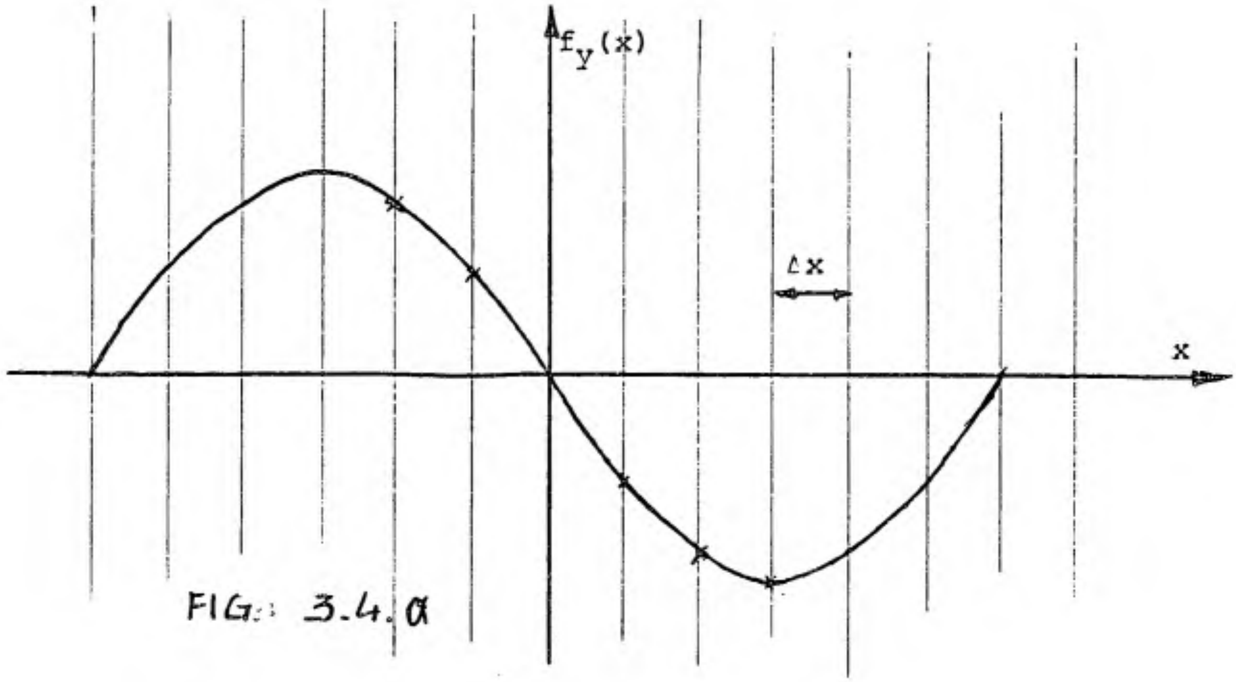


FIG. 3.4.a

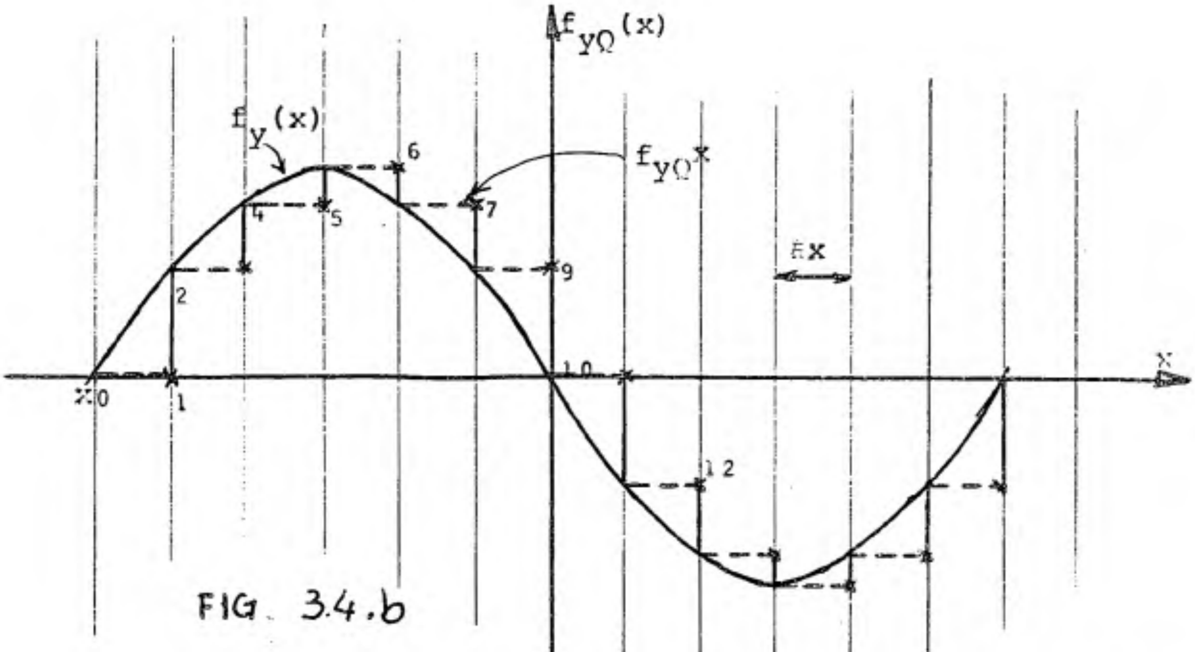


FIG. 3.4.b

The quantized function $f_{y\Omega}(x)$ which has the delay of Δx with respect to the continuous function $f_y(x)$

$$\varepsilon_{iQx} = f_{iy}\ (x) - f_{iyQ}\ (x) \tag{3-7}$$

This error may produce the phase shift φ between the continuous function f_{iy} (x) and the quantized function f_{iyQ} (x). It will be shown later on, that the quantization error ε_{iQx} depend to the quant Δx and Δy. In order to reduce the quantization error, the value of quant Δx and Δy should be decreased.

The algorithm of quantized points can be found by the equation below:

$$\begin{cases} x_{iQ} = y_{(i-1)Q} + a_{ix} \cdot \Delta_Q x \\ y_{iQ} = y_{(i-1)Q} + a_{iy} \cdot \delta_i y \\ \quad i = 1, 2, \ldots K \end{cases} \tag{3-8}$$

As it is seen from equation (3-8) and figure (3-4), each point (x_{iQ}, y_{iQ}) is calculated by the points $(x_{(i-1)Q}, y_{(i-1)Q})$, the quants $\Delta_Q x$, $\delta_i y$ and the parameters a_{ix}, a_{iy} (they can be $\pm$ 1 or 0) which determine wether the quants $\Delta_Q x$, $\delta_i y$ should be added to (+1), substracted form (-1) or inefected to the value of $(x_{(i-1)Q}, y_{(i-1)Q})$. But as it is seen from figure (3-4), the quantized points are not determined completely in this procedure, because the increment $\delta_i y$ is not quantized, and is unknown. Therefore the quantization of function by the only variable x is not sufficient to determine the quantized points (x_{iQ}, y_{iQ}).

3.1.2. Quantization in incremental computation by the dependent variable y, and algorithm of quantized points.

In this case the continuous function $f_y(x)$ is quantized with the quantum of dependent variable Δy. So for quantization process, we should choose the value of quantum Δy and the initial point (x_o, y_o). This process is shown in figure (3-5)

The quantized points are the intersection of continuous function $f_y(x)$, with the lines $-i\Delta y, -(i-1)\Delta y, \ldots, -\Delta y, 0, +\Delta y, +2\Delta y, \ldots +i\Delta y$.

As it was mentioned earlier, in digital iteration machine, because of time which is spend to calculate the mathematical operations, the quantized points have some delay with respect to the continuous function.

The inherent delay of digital quantization with respect to Δy, introduce an error in each point between the continuous function $f_{iy}(x)$ and the quantized function $f_{iyQ}(x)$ equal to:

$$\varepsilon_{iQy} = f_{iy}(x) - f_{iyQ}(x) \tag{3-9}$$

The delay which introduce the error ε_{iQy}, produce the phase shift and the amplitude deformation of quantized function $f_{iyQ}(x)$ with respect to the original continuous function $f_y(x)$, as it is shown in figure (3-6).

Each quantized point x_{iQ}, y_{iQ} can be find by its backward informations $x_{(i-1)Q}, y_{(i-1)Q}$, as following:

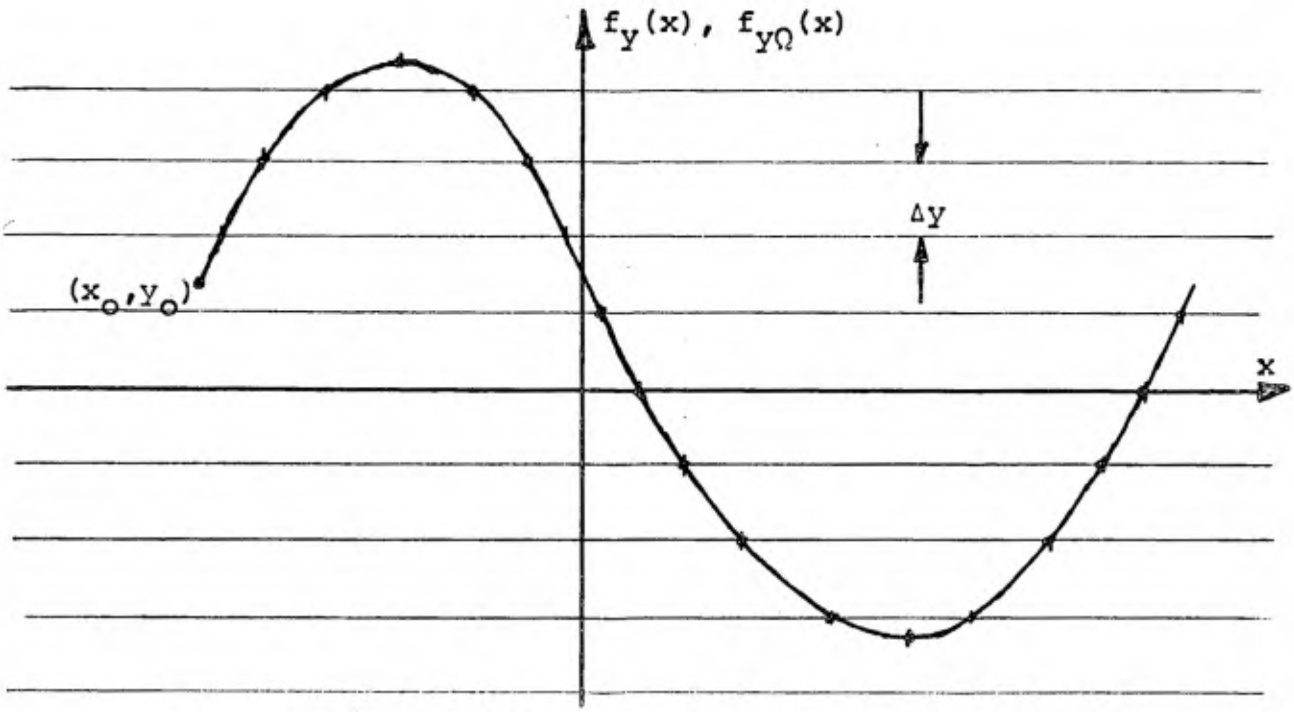

$f_y(x)$, $f_{yQ}(x)$
Δy
(x_o, y_o)
x

fig. 3.5.

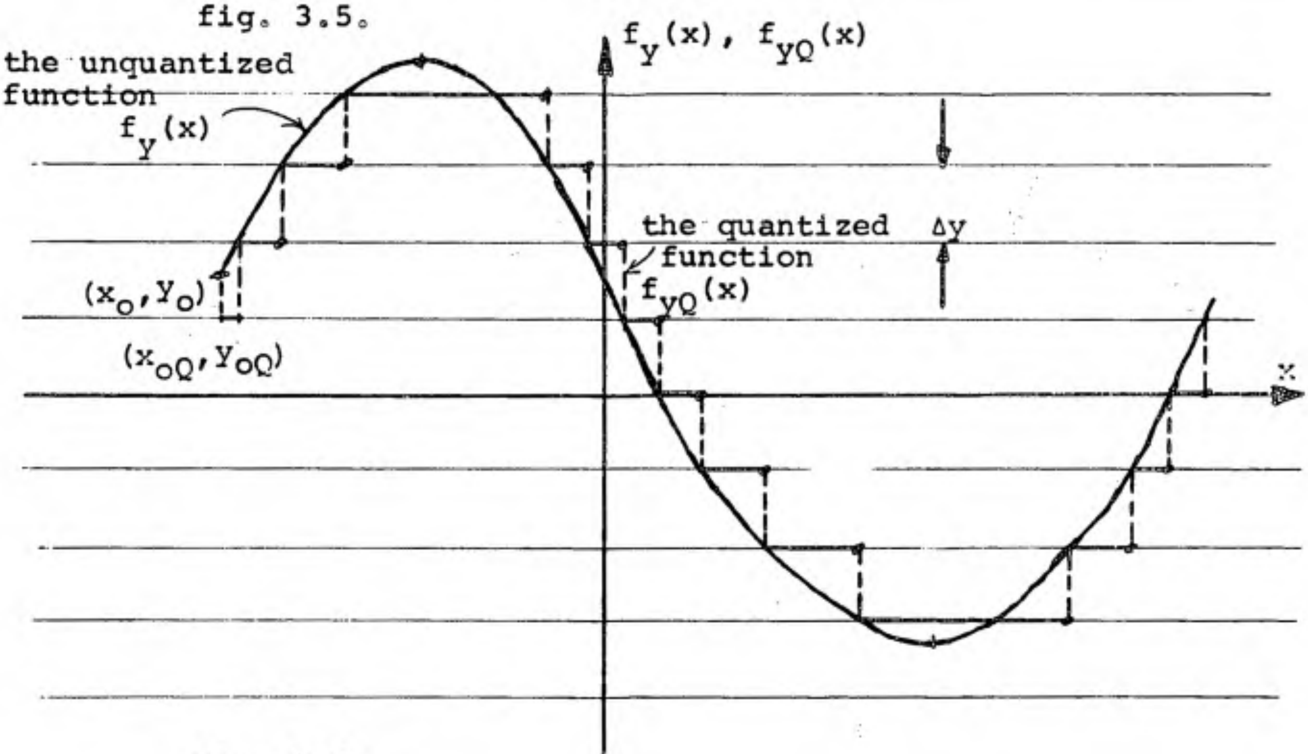

$f_y(x)$, $f_{yQ}(x)$
the unquantized function
$f_y(x)$
the quantized function
$f_{yQ}(x)$
Δy
(x_o, y_o)
(x_{oQ}, y_{oQ})
x

fig. 3.6.

$$\begin{cases} x_{iQ} = x_{(i-1)Q} + a_{ix} \circ \delta_i x \\ y_{iQ} = y_{(i-1)Q} + a_{iy} \circ \Delta_i y \\ \quad i = 1, 2, \ldots k \end{cases} \qquad (3\text{-}10)$$

In equation (3-10), the (x_{iQ}, y_{iQ}) is the quantized point, which should be determined by the point (x_{i-1}, y_{i-1}), the parameter a_{ix}, a_{iy} (equal to ± 1 or 0), the quant $\Delta_i y$ and the unquantized increment $\delta_i x$. As it is seen, the increment $\delta_i x$ is not quantized and is variable in the process of quantization. Therefore the machine can not calculate the value of $\delta_i x$ which is not known, in other words, the quantization procedures are not complet.

The equation (3-10) can be written with the information of initial point (x_o, y_o) and the increments $\delta_i x$, $\Delta_i y$, as following:

$$\begin{cases} x_{iQ} = x_{oQ} + \delta_i x \sum_{i=1}^{k} a_{ix} \\ y_{iQ} = y_{oQ} + \Delta_i y \sum_{i=1}^{k} a_{iy} \end{cases} \qquad (3\text{-}11)$$

In equation (3-11) the point (x_{iQ}, y_{iQ}) is determined by the initial quantized point (x_{oQ}, y_{oQ}) which is delaid with respect to the original point (x_o, y_o), the parameter a_{ix}, a_{iy}, and the quantums $\delta_i x$, $\Delta_i y$.

3.1.3. Quantization of the continuous function Y (X) in incremental computation by variables y and t, when the independent variable of integration X is the independent variable t of machine, and algorithm of quantized points.

When the independent variable x of integral is equal to the independent variable t of machine, then, the formula of integration

$$s\ (t) = \int_{x_o}^{x_k} y\ (x)\ dx \tag{3-12}$$

will be:

$x = t$

$$s(t) = \int_{t_o}^{t_k} y\ (t) \cdot d\ (t) \tag{3-13}$$

As it was discussed earlier the continuous function y (t) is replaced by the interpolated function $f_{iy}(t)$ which gives the approximated value of integration $s^{*}(t)$ as:

$$s^{*}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy}\ (t) \cdot d\ (t) \tag{3-14}$$

by this approximation there will be the error of method $\Gamma(t)$, which is the difference between the actual value of integration s (t) and the approximated value $s^{*}(t)$,

$$\Gamma(t) = -s^{*}(t) + s\ (t) \tag{3-15}$$

In digital machine all the quantities are discrete values or quantized, within an interval, so the continuous interpolated function $f_{iy}(t)$ is quantized with the variable of machine t, and the dependent variable y.

The actual operation of quantization is done in figure (3-7), with the quantum $\Delta t = \Delta x$, Δy. The curve (1) is the quantization of function $f_{iy}(x)$ with respect to quant Δy which has accompanied with inherent delay of digital machine.

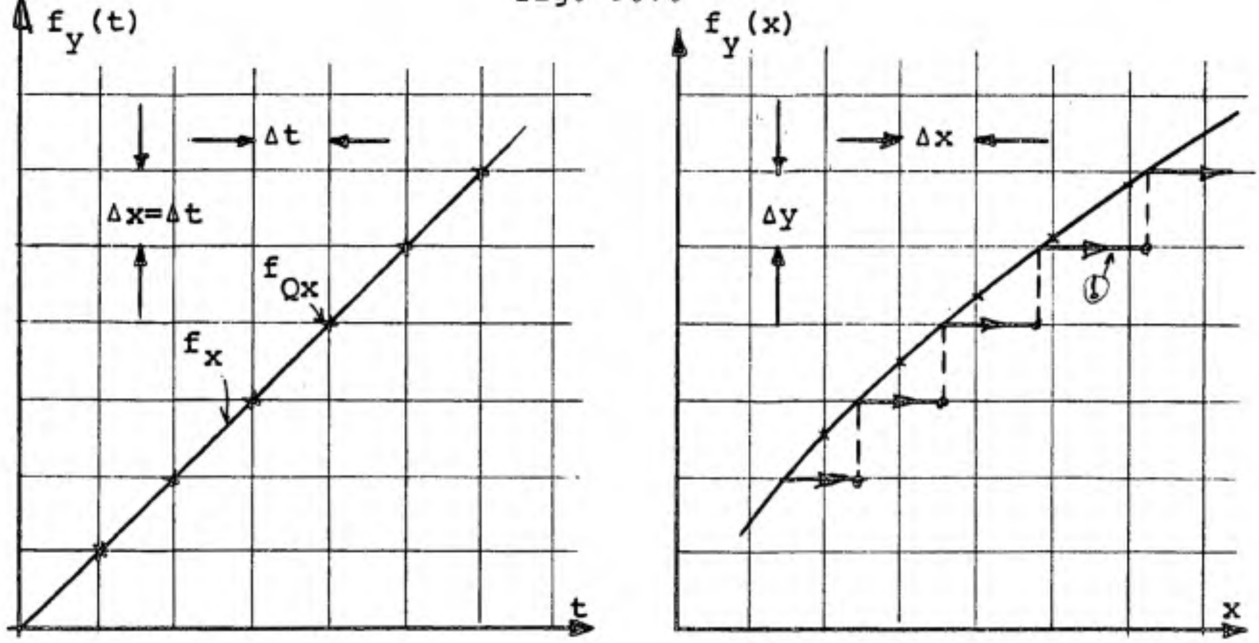
The $f_{iy}(x)$ is also quantized with respect to $\Delta x = \Delta t$, but there is no delay, ($f_{ix}(t) = x\ (t) = t$), so the quantized points are on the continuous function.

The actual points of quantization should be in curve (1), and also the lines Δx, so they are in the intersections of curve (1) and the lines Δx as it is shown figure (3-8).

The quantized function is combined with quantized points by considering the delay between the continuous function and quantized function, figure (3-9). The quantization error ε_{iQy}, in each point is defined as:

$$\begin{cases} \varepsilon_{iQy} < \Delta y \\ \varepsilon_{iQy} = f_{iy}\ (t) - f_{iyQ}\ (t) \end{cases} \tag{3-15}$$

fig. 3.7.



The quantization of function x(t) by Δt

The quantization of y(x) by Δt and Δy

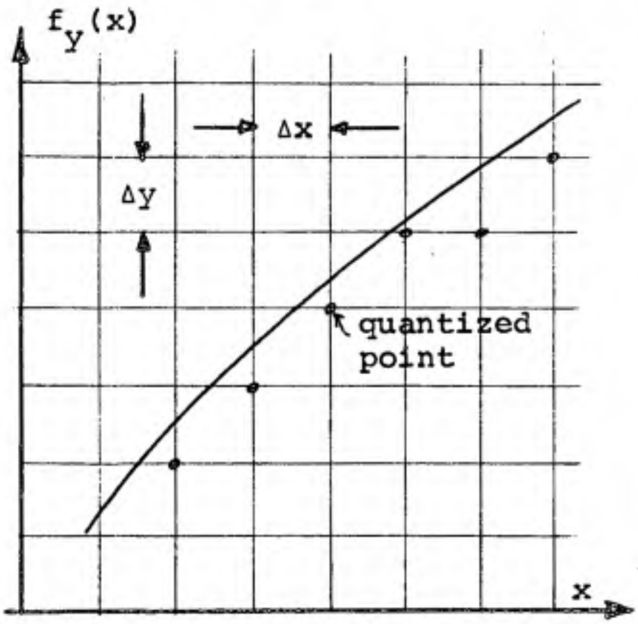


fig. 3.8.

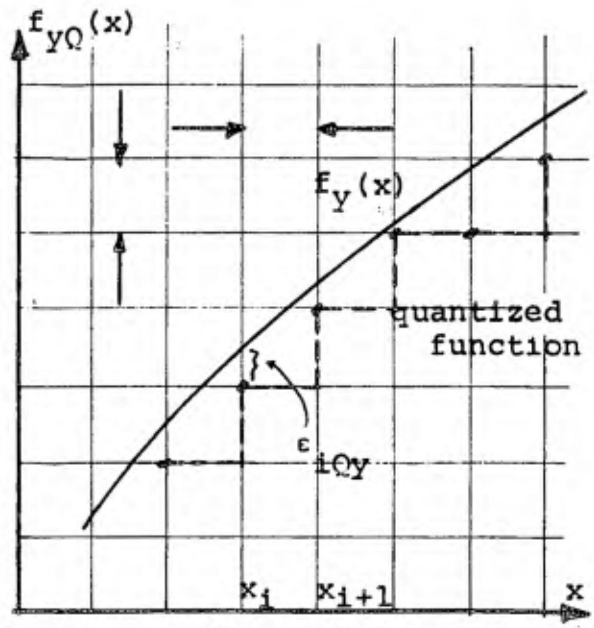


fig. 3.9.

The ε_{iQy} cause the total quantization error ε_{tQ}. It will be seen that the actual value of integral is equal to the approximated quantized value of integral s_Q^{*} (x) (algorithm of machine) plus the error of method $\Gamma(t)$ and the error of quantization ε_{tQ}, so:

$$s\ (x) = s_{Q}^{*}\ (x) + [\Gamma(t) + \varepsilon_{tQ}\ (t)] \qquad (3\text{-}22)$$

Because of the delay of quantization process, there is an error of phase, between the continuous function and discrete quantized function, it also cause the deformation of amplitude. The error of phase and quantization, depends on the quantums Δx, Δy.

By taking into account the above discussion, the block diagram of incremental computer can be drawn in figure (3.10).

As it is seen from figure (3.10), the function y (x) is first interpolated to the function f_{iy} (x) in interval $x \in (x_i, x_{i+1})$, with the error of ε_{iy}, which cause the total error of method $\Gamma(x)$. Then the approximated function is quantized by the variable of machine t and cause the error of quantization ε_{tQ} and the delay $e^{-p\tau}$ for $\tau < \Delta t$.

The algorithm of quantized points can be found from backward quantized points as it is shown in figure (3.11) and equation (3-23).

$$\begin{cases} x_{iQ} = x_{(i-1)Q} + a_{ix} \cdot \Delta_Q x \\ y_{iQ} = y_{(i-1)Q} + a_{iy} \cdot \Delta_Q y \\ \quad i = 1, 2, \ldots. k \end{cases} \qquad (3\text{-}23)$$

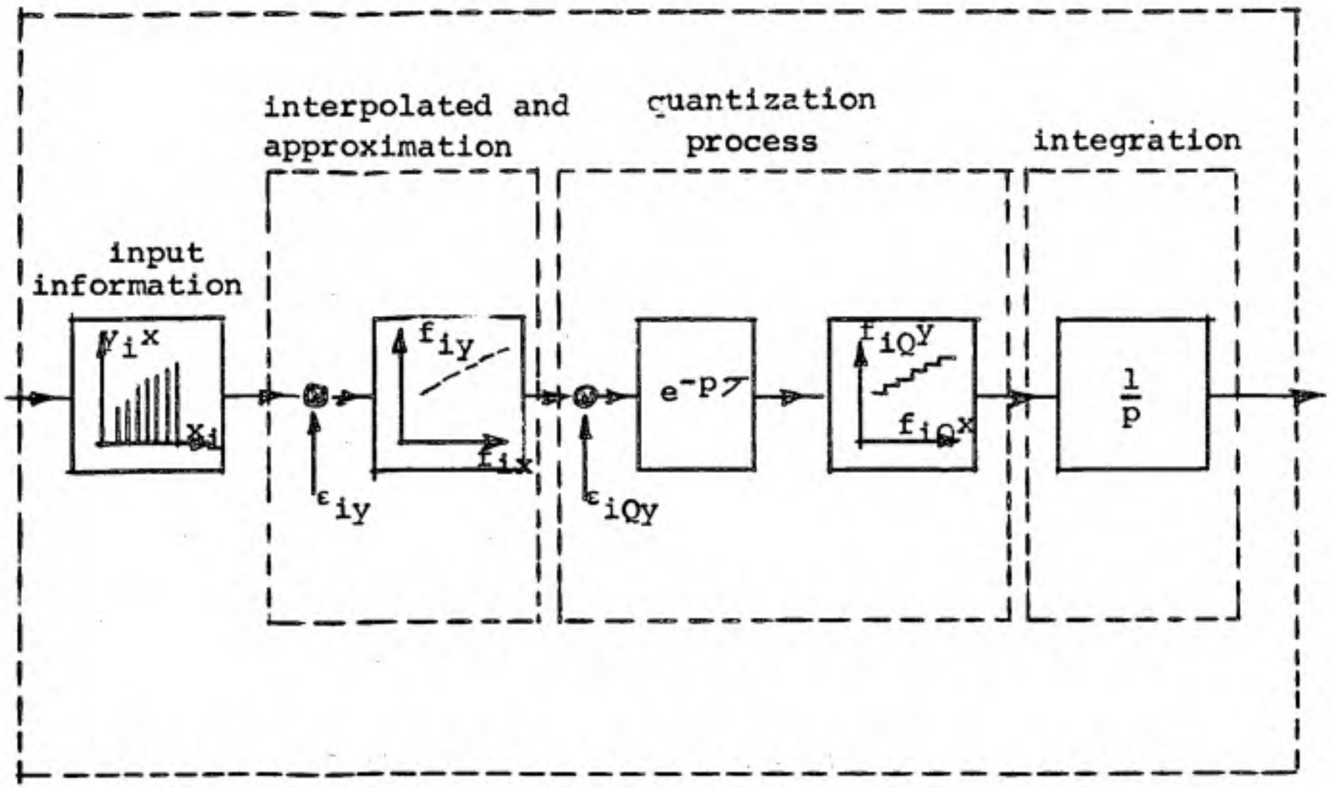


FIG.: 3-10.

The block diagram of incremental computer.

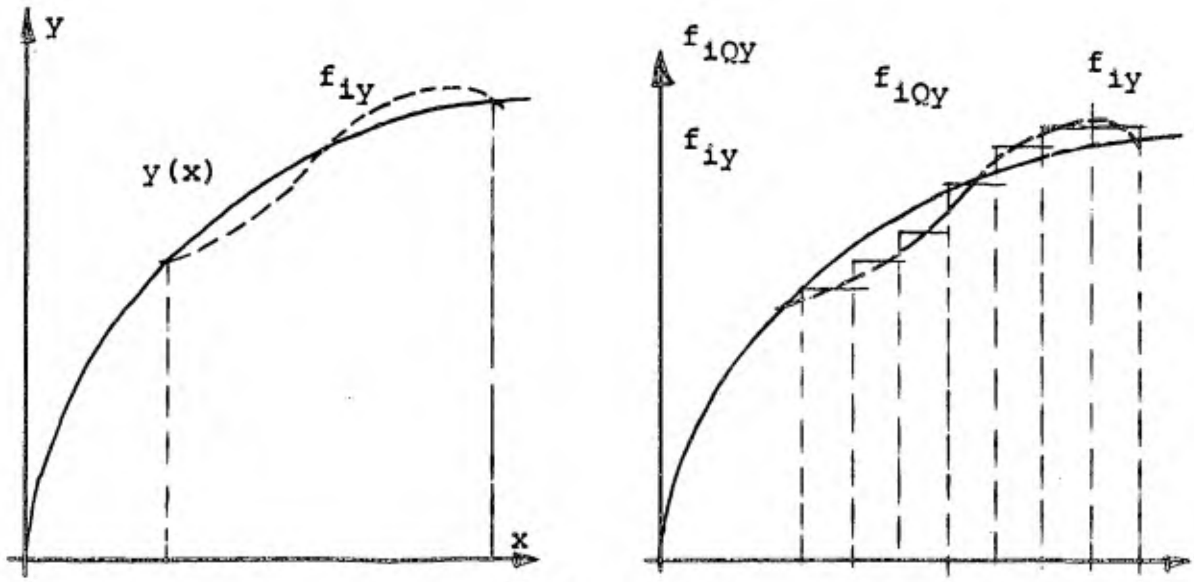


FIG. 3-12

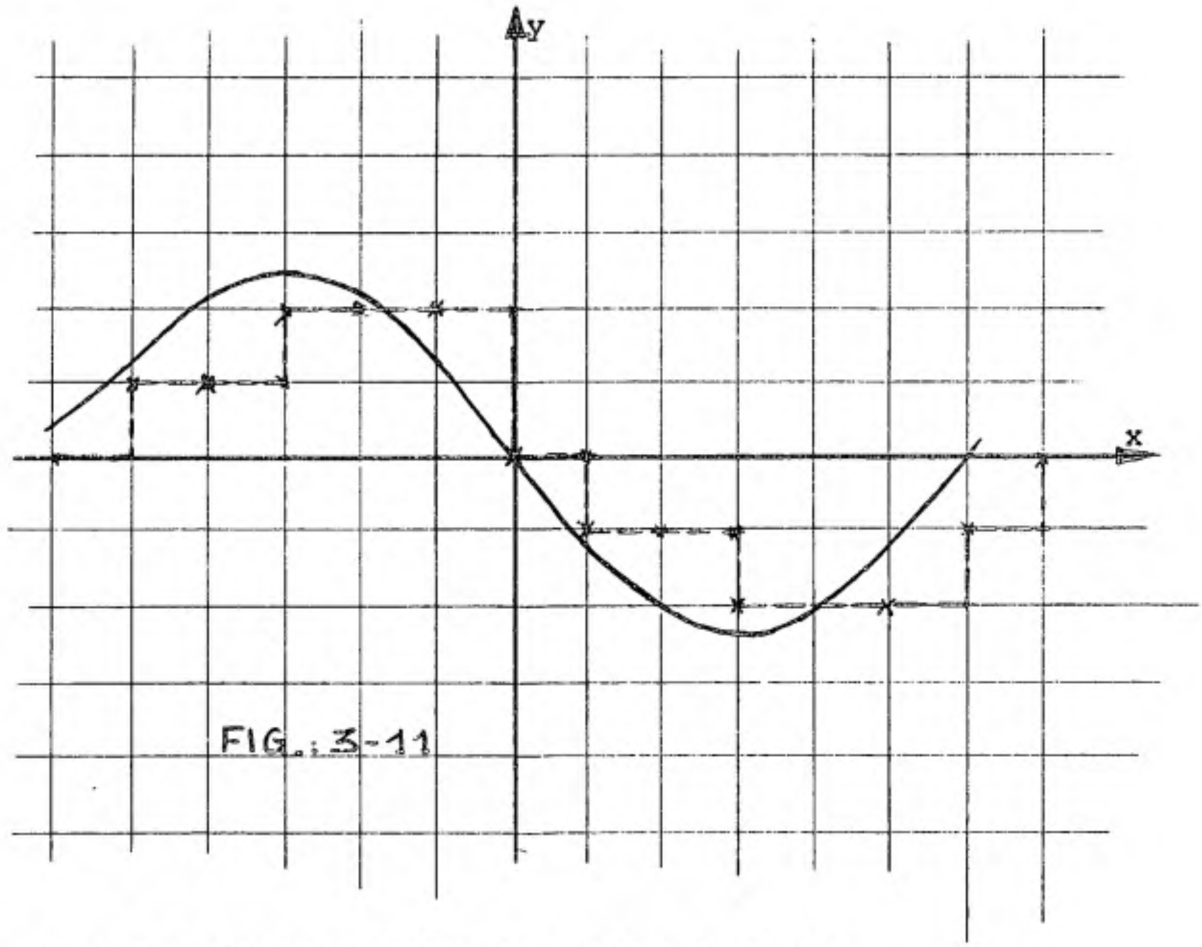


FIG.: 3-11

The quantized point of continuous function.

As it is seen from equation (3-23) and figure (3-11) each point (x_{iQ}, y_{iQ}) is calculated by the points $(x_{(i-1)Q}, y_{(i-1)Q})$, the quants $\Delta_Q x$, $\Delta_Q y$, and the parameters a_{ix}, a_{iy} (they can be ± 1 or 0) which determines wether the quant $\Delta_Q x$, $\Delta_Q y$ should be added to (+1) substracted from (-1) or ineffected to the value of $(x_{(i-1)Q}, y_{(i-1)Q})$.

If we have b input of Δy in integrator then the equation (3-23) can be written as:

$$\begin{cases} x_{iQ} = x_{(i-1)Q} + a_{ix} \cdot \Delta_Q x \\ y_{iQ} = y_{(i-1)Q} + \Delta_Q y \sum_{j=1}^{b} a_{ijy} \end{cases} \tag{3-24}$$

Normally in incremental machine, there are 4 or 10 Δy inputs of integrator depending on the construction of machine, in incremental computer of our laboratory b = 7.

The equation (3-24) can be also written in the following form:

$$\begin{cases} x_{kQ} = x_{oQ} + \Delta x_Q \sum_{i=1}^{k} a_{ix} \\ y_{kQ} = y_{oQ} + \Delta y_Q \sum_{i=1}^{k} \sum_{j=1}^{b} a_{ijy} \end{cases} \tag{3-24*}$$

As it is seen from equation (3-24)*, the quantized points are determined by the quantum Δx, Δy, the parameters a_{ix}, a_{iy} and the initial point (x_{oQ}, y_{oQ}).

3.2. The quantization error in unitary increment computation, when the independent variable of integration x is the independent variable t.

As we have discussed earlier, the continuous function y (x) in interval $x \in (x_i, x_{i+1})$ was replaced by the approximated interpolated continuous function f_{iy}, which cause the error of method $\Gamma(x)$, between the actual value of integration s (x) and the approximated interpolated value $s^{*}(x)$ as following:

$$s(x) = s^{*}(x) + \Gamma(x) \qquad (3\text{-}25)$$

$$s^{*}(x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iy} \cdot dx \qquad (3\text{-}26)$$

$$f_{iy} = f_{iy}\ [x_1, y_1, \delta_1 x, \delta_1 y, \delta_2 x, \delta_2 y, \ldots \ldots \delta_i x, \delta_i y] \qquad (3\text{-}27)$$

But in digital machine all the quantities $[x_1, y_1, \delta_1 x, \delta_1 y, \ldots \ldots \delta_i x, \delta_i y]$ are quantized within an interval by the variable of machine t. Therefore instead of the quantities $[x_1, y_1, \delta_1 x, \delta_1 y, \ldots \ldots \delta_i x, \delta_i y]$, there will be their quantized values $[x_{1Q}, y_{1Q}, \delta_{1Q} x, \delta_{1Q} y, \ldots \ldots \delta_{iQ} x, \delta_{iQ} y]$. So the quantized function f_{iQy} which use the quantized quantities $[x_{1Q}, y_{1Q}, \delta_{1Q} x, \delta_{1Q} y, \ldots \ldots \delta_{iQ} x, \delta_{iQ} y]$ will be:

$$f_{iQy} = f_{iQy}\ [x_{1Q}, y_{1Q}, \delta_{1Q} x, \delta_{1Q} y, \ldots \ldots \delta_{iQ} x, \delta_{iQ} y]$$

$$\left\lfloor x \in (x_i,\ x_{i+1}) \right. \tag{3-28}$$

Therefore there will be an error ε_{iQy} between the approximated interpolated function f_{iy} and the approximated interpolated quantized function f_{iQy} as following:

$$\varepsilon_{iQy} = f_{iy}\ (t) - f_{iQy}\ (t) \tag{3-29}$$

by putting the equation (3-26),(3-28) and (3-29) in equation (3-25), we will have:

$$s\ (x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} (f_{iQy} + \varepsilon_{iQy})\ dx + \Gamma(x) \tag{3-30}$$

$$= \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iQy} \cdot dx + \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iQy}\ dx + \Gamma(x) \tag{3-31}$$

$$= s_Q^{*}\ (x) + \varepsilon_{tQ}\ (x) + \Gamma(x) \tag{3-32}$$

in equation (3-30),(3-31) and (3-32), the $s_Q^{*}\ (x)$ is the approximated interpolated and quantized value of integral which is the algorithm of machine equal to:

$$s_Q^{*}\ (x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} f_{iQy} \cdot dx \tag{3-33}$$

ε_{tQ} (x) is the quantization error in the process of integration equal to:

$$\varepsilon_{tQ}\ (x) = \sum_{i=1}^{k} \int_{x_i}^{x_{i+1}} \varepsilon_{iQy} \circ dx \qquad (3\text{-}34)$$

and $\Gamma(x)$ is the error of method, which we have calculated in chapter (2), for different method of integration.

As it is seen from equation (3-30), (3-31) and (3-32), the actual value of integral s (x) is equal to the approximated interpolated quantized value of integral s^{*}_{Q} (x), plus the error of method $\Gamma(x)$ and the quantization error ε_{tQ} (x). In foregoing paragraph, we will calculate the quantization error for different method of integration.

3.2.1. Quantization error in rectangular method of integration.

The interpolated function f_{iy} (x) which is replaced to the y (x) in interval $x \in (x_i, x_{i+1})$ in the rectangular method is:

$$f_{iy} = y_i \qquad (3\text{-}35)$$

and the interpolated quantized function f_{iQy} is:

$$f_{iQy} = y_{iQ} \qquad (3\text{-}36)$$

Therefore the equation (3-29) can be written as:

$$\varepsilon_{iQy} = y_i - y_{iQ} \tag{3-37}$$

The approximated formula of integration $s^*(x)$ is:

$$s^*(x) = \sum_{i=1}^{k} - \Delta_i x \int_{0}^{-1} f_{iy} \circ d\xi \tag{3-38}$$

by putting the value of f_{iy} from equation (3-35) and (3-37) in equation (3-38), we will have:

$$s^*(x) = \sum_{i=1}^{k} - \Delta x \int_{0}^{-1} (y_{iQ} + \varepsilon_{iQy})\, d\xi \tag{3-39}$$

$$= \sum_{i=1}^{k} y_{iQ} \circ \Delta_i x + \sum_{i=1}^{k} \varepsilon_{iQy} \circ \Delta_i x \tag{3-40}$$

$$= s_Q^* (x) + \varepsilon_{tQ} \tag{3-41}$$

from equation (3-39),(3-40) and (3-41) it is clear that the approximated quantized value of integratior $s_Q^*(x)$ which is the algorithm of machine, is equal to:

$$s_Q^* (x) = \sum_{i=1}^{k} y_{iQ} \cdot \Delta_i x \tag{3-42}$$

and the ε_{tQ} is the error of quantization which is the difference between the quantized and unquantized integration function as:

$$\varepsilon_{tQ} = s^{*}(x) - s^{*}_{Q}(x) \tag{3-43}$$

from equation (3-40) and (3-41), the ε_{tQ} can be find as:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQy} \cdot \Delta_i x \tag{3-44}$$

as it was discussed in the process of integration, the $\varepsilon_{iQy} < \Delta y$ so the equation (3-44) can be written as following:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} \Delta y \cdot \Delta_i x \tag{3-45}$$

or

$$\varepsilon_{tQ} < \Delta y \sum_{i=1}^{k} \Delta_i x \tag{3-46}$$

or

$$\varepsilon_{tQ} < \Delta y \, (x_{kQ} - x_{oQ}) \tag{3-47}$$

Therefore, the error of quantization ε_{tQ} depends to the quant Δy and the interval of integration $x \in (x_o, x_k)$. In order to reduce this error we should reduce the quant Δy.

3.2.2. Quantization error in the trapezoidal method of integration.

In trapezoidal method of integration, the interpolated function f_{iy}, which is replaced to y (x), is as following:

$$f_{iy} = y_i - \xi \cdot \Delta_i y \tag{3-48}$$

As the quantized point do not coincide with the unquantized points, therefore, there will be an error ε_{iQy} which is defined as:

$$\varepsilon_{iQy} = y_i - y_{iQ} \tag{3-49}$$

by putting the value of equation (3-49) in equation (3-48), we will have:

$$f_{iy} = (y_{iQ} + \varepsilon_{iQy}) - \xi \cdot \Delta_i (y_{iQ}y + \varepsilon_{iQy}) \tag{3-50}$$

$$= y_{iQ} + \varepsilon_{iQy} - \xi (\Delta_{iQ}y + \Delta\varepsilon_{iQy}) \tag{3-51}$$

because the independent variable of integral is the independent variable of machine, so

$$x_{iQ} = x_i \tag{3-52}$$

then the approximated formula of integral from equation (3-38) can be find as:

$$\delta_i s^* = - \Delta_i x \int_0^{-1} f_{iy} (\xi) \, d\xi \tag{3-53}$$

$$\xi \in (-1, 0) \quad \text{or} \quad x \in (x_i, x_{i+1})$$

by putting the equations (3-51) and (3-52) in equation (3-53), we will have:

$$\delta_i\ s^* = -\ \Delta_{iQ}x \int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi\ (\Delta_{iQ}y + \Delta\varepsilon_{iQy})]\ d\xi \tag{3-54}$$

$$= \Delta_{iQ}x \left|\ y_{iQ} + \varepsilon_{iQy} + \frac{1}{2}(\Delta_{iQ}y + \Delta\varepsilon_{iQy}) \right|_{\xi\ (-1,0)} \tag{3-55}$$

$$\delta_i\ s^* = (y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \cdot \Delta_{iQ}y) + (\varepsilon_{iQy} \cdot \Delta_{iQ}x + $$

$$+ \frac{1}{2}\Delta\varepsilon_{iQy} \cdot \Delta_{iQ}x) \qquad x \in (x_i,\ x_{i+1}) \tag{3-56}$$

The approximated interpolated formula of integral $s^*(x)$ for k interval will be:

$$s^*(x) = \sum_{i=1}^{k} \delta_i\ s^* \tag{3-57}$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \cdot \Delta_{iQ}y) + $$

$$+ \sum_{i=1}^{k} (\varepsilon_{iQy} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta\varepsilon_{iQy} \cdot \Delta_{iQ}x) \tag{3-58}$$

in the equation (3-58), the first sum is the approximated interpolated and quantized value of integral s_Q^* (x) which is the algorithm of machine as following:

$$s_Q^*(x) = \sum_{i=1}^{k} (y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \cdot \Delta_{iQ}y) \qquad (3\text{-}59)$$

and the second bracket is the error of quantization in the process of integration, as following:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (\varepsilon_{iQy} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta\varepsilon_{iQy} \cdot \Delta_{iQ}x) \qquad (3\text{-}60)$$

so the equation (3-58) can be written as:

$$s^*(x) = s_Q^*(x) + \varepsilon_{tQ} \qquad (3\text{-}61)$$

where $s^*(x)$ is the approximated interpolated continuous function, $s_Q^*(x)$ is the approximated interpolated quantized function of integral and ε_{tQ} is the error of quantization.

In equation (3-60), the second term can be neglected with respect to the first one, so the equation (3-60) can be written as:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQy} \cdot \Delta_{iQ}x \qquad (3\text{-}62)$$

as we have seen in the process of quantization $\varepsilon_{iQy} < \Delta y$, so the equation (3-62) can be written as:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} \Delta y \cdot \Delta_{iQ}x \qquad (3\text{-}63)$$

or

$$\varepsilon_{tQ} < \Delta y \cdot \sum_{i=1}^{k} \Delta_{iQ} x \tag{3-64}$$

as $\sum_{i=1}^{k} \Delta_{iQ} x = x_o - x_k$ so:

$$\varepsilon_{tQ} < \Delta y \; (x_o - x_k) \tag{3-65}$$

The equation (3-65) gives the value of quantization error that depend to the quantum Δy and the interval $(x_o - x_k)$.

3.2.3. Quantization error in the three points method of integration.

In three points method of integration, the interpolated function f_{iy} which is replaced to y (x) is:

$$f_{iy} = y_i - \xi \; \Delta_i y - \frac{\xi(\xi+1)}{2!} \Delta_i^{II} y \tag{3-66}$$

by putting the value y_i from equation (3-49) in equation (3-66), we will have:

$$\begin{aligned} f_{iy} &= (y_{iQ} + \varepsilon_{iQy}) - \xi \Delta_i \; (y_{iQ} + \varepsilon_{iQy}) - \\ &\quad - \frac{\xi(\xi+1)}{2!} \Delta_i^{II} \; (y_{iQ} + \varepsilon_{iQy}) \end{aligned} \tag{3-67}$$

$$= (y_{iQ} + \varepsilon_{iQy}) - \xi(\Delta_{iQ} y + \Delta_i \; \varepsilon_{iQy}) -$$

$$- \frac{\xi(\xi+1)}{2!} (\Delta_i^{II} y_{iQ} + \Delta^{II} \varepsilon_{iQy}) \tag{3-68}$$

As the independent variable of integral X is the independent variable t of machine so:

$$x_i = x_{iQ} \tag{3-69}$$

Then the approximated interpolated formula of integration $\delta_i s^{x}$ in interval $x \in (x_i, x_{i+1})$ can be written as:

$$\delta_i s^{x} = - \Delta_{iQ} x \int_0^{-1} f(\xi) \cdot d\xi \tag{3-70}$$

$$= - \Delta_{iQ} x \int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi (\Delta_{iQ} y + \Delta_i \varepsilon_{iQy}) -$$

$$- \frac{\xi(\xi+1)}{2!} (\Delta_i^{II} y_{iQ} + \Delta^{II} \varepsilon_{iQy})] d\xi \tag{3-71}$$

$$= \Delta_{iQ} x \left| (y_{iQ} + \varepsilon_{iQy}) + \frac{1}{2} (\Delta_{iQ} y + \Delta_i \varepsilon_{iQy}) + \right.$$

$$\left. + \frac{1}{12} (\Delta_i^{II} y_{iQ} + \Delta^{II} \varepsilon_{iQy}) \right| \tag{3-72}$$

or

$$\delta_i s^{x} = (y_{iQ} \cdot \Delta_{iQ} x + \frac{1}{2} \Delta_{iQ} y \cdot \Delta_{iQ} x + \frac{1}{12} \Delta_i^{II} y_{iQ} \cdot \Delta_{iQ} x) +$$

$$+ (\varepsilon_{iQy} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_i \varepsilon_{iQy} +$$

$$+ \frac{1}{12} \Delta_i x \circ \Delta^{II} \varepsilon_{iQy}) \tag{3-73}$$

The approximated interpolated integral formuola $s^{*}(x)$ for k interval will be:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i \, s^{*} \tag{3-74}$$

$$= \sum_{i=1}^{k} (y_{iQ} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_{iQ}y + \frac{1}{12} \Delta_{iQ}^{II}y \circ \Delta_{iQ}x) +$$

$$+ \sum_{i=1}^{k} (\varepsilon_{iQy} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_i \varepsilon_{iQy} +$$

$$+ \frac{1}{12} \Delta_i x \cdot \Delta^{II} \varepsilon_{iQy}) \tag{3-75}$$

The approximated interpolated and quantized function of integral $s_Q^{*}(x)$ which is the algorithm of machine is equal to the content of first bracket as following:

$$s_Q^{*} (x) = \sum_{i=1}^{k} (y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \cdot \Delta_{iQ}y + \frac{1}{12}\Delta_{iQ}^{II}y \cdot \Delta_{iQ}x) \tag{3-76}$$

The content of second bracket is the error of quantization which is equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (\varepsilon_{iQy} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_i \varepsilon_{iQy} + \frac{1}{12} \Delta_i x \circ \Delta^{II} \varepsilon_{iQy}) \tag{3-77}$$

we can neglect the second and third terms of equation (3-77) with respect to the first one. So the error of quantization will be:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQy} \cdot \Delta_{iQ}x \tag{3-78}$$

As it was discussed in the process of quantization the $\varepsilon_{iQy} < \Delta y$, so the equation (3-78) can be written as:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} \Delta y \circ \Delta_{iQ}x \tag{3-79}$$

as $\sum_{i=1}^{k} \Delta_{iQ}x = x_k - x_o$

so

$$\varepsilon_{tQ} < \Delta y \ (x_k - x_o) \tag{3-80}$$

The equation (3-80) gives the error of quantization in the three points method of integration. As it is seen, the ε_{tQ} depend to the value of quant Δy, and to the interval $(x_k - x_o)$

3.3. The quantization error in unitary increment computation, when the independent variable of integration X is a function of the independent variable t.

If we have the continuous function X (t) and Y (t) which are replaced by their approximated interpolated function f_{ix} (t) and $f_{iy}(t)$ then the integral in interval $t \in (t_o, t_k)$ is:

$$s\ (t) = \int_{t_o}^{t_k} Y\ (t) \cdot d\ \frac{X(t)}{dt}\ dt \tag{3-88}$$

As we have discussed earlier, this integral formula is replaced by the approximated interpolated formula of integration $s^*(x)$ as following:

$$s^*(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy}\ (t) \cdot d\frac{f_{ix}(t)}{dt}\ dt \tag{3-89}$$

where

$$\begin{cases} f_{iy} = f_y\ [y_i,\ \delta_i y,\ \delta_{i-1} y, \ldots\ldots\ t_{iQ}, \ldots (i=1,2,\ldots k)] \\ f_{ix} = f_x\ [x_i,\ \delta_i x,\ \delta_{i-1} x, \ldots\ldots\ t_{iQ}, \ldots (i=1,2,\ldots k)] \end{cases} \tag{3-90}$$

The error of this approximation is called the error of method r (t) which is equal to:

$$\Gamma(t) = s\ (t) - s^{*}(t) \tag{3-91}$$

The equation (3-90) can be written as:

$$\begin{cases} f_{iy} = f_y\ [y_i,\ y_{i-1},\ y_{i-2},\ldots\ldots y_{i-p},\quad t_{iQ}\ (i=1,2,\ldots k)\ ] \\ f_{ix} = f_x\ [x_i,\ x_{i-1},\ x_{i-2},\ldots\ldots x_{i-p},\quad t_{iQ}\ (i=1,2,\ldots k)\ ] \end{cases} \tag{3-92}$$

As it is seen from equation (3-92), the interpolated function f_{ix}, f_{iy}, have the information of $[x_i,\ y_i,\ x_{i-1},\ y_{i-1},\ldots t_{iQ}]$.

But in digital machine all the value are quantized with the variable of machine t, so instead of $[x_i,\ y_i,\ x_{(i-1)},\ y_{(i-1)},\ldots t_{iQ}\ (i=1,2,\ldots k)]$, we will have the quantized points $[x_{iQ},\ y_{iQ},\ x_{(i-1)Q},\ y_{(i-1)Q},\ldots\ldots t_{iQ}\ (i=1,2,\ldots k)]$, therefore, the approximated interpolated functions f_{ix}, f_{iy}, are replaced with the approximated interpolated and quantized function f_{iQx}, f_{iQy}, with the information of quantized points $[x_{iQ},\ y_{iQ},\ x_{(i-1)Q},\ y_{(i-1)Q},\ldots\ldots t_{iQ}\ (i=1,2,\ldots k)]$.

So, in each point, we have the quantized error ε_{iQx}, ε_{iQy} which are the difference between the quantized and unquantized value of function as following:

$$\begin{cases} \varepsilon_{iQx} = f_{ix}\ (t) - f_{iQx}\ (t) \\ \varepsilon_{iQy} = f_{iy}\ (t) - f_{iQy}\ (t) \end{cases} \tag{3-93}$$

For instant, if the $X\ (t) = e^{+t}$, and $Y\ (t) = \sin \omega t$, fig.(3.13)

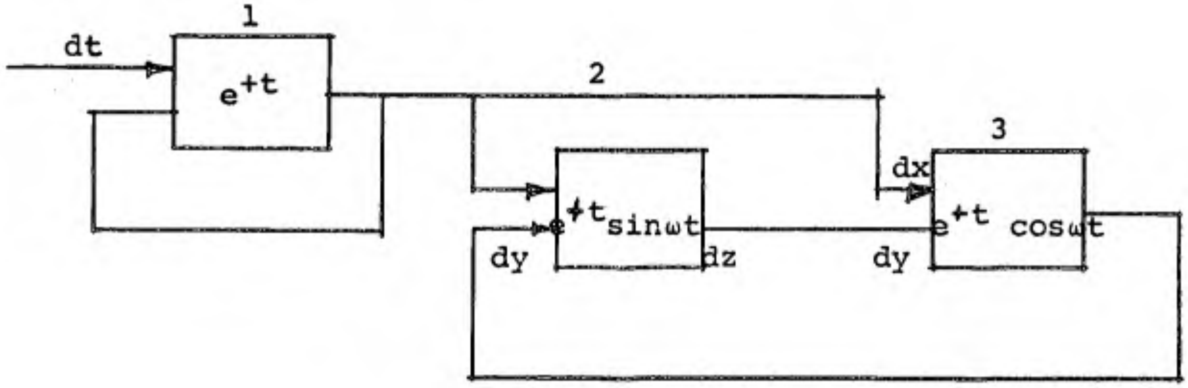


FIG. 3-13.

the f_{ix}, f_{iy}, f_{iQx}, f_{iQy} are shown in following figures (3-14) and (3-15).

As we have discussed earlier, the quantized points of function f_{ix}, are the intersection of the curve b and the line Δt = const. with the quantized error ε_{iQx} in each point.

In order to quantize the function f_{iy}, it is sufficient to find the intersection of (curve 2) with the lines Δt = const. as before. The procedure is shown in figure (3-15).

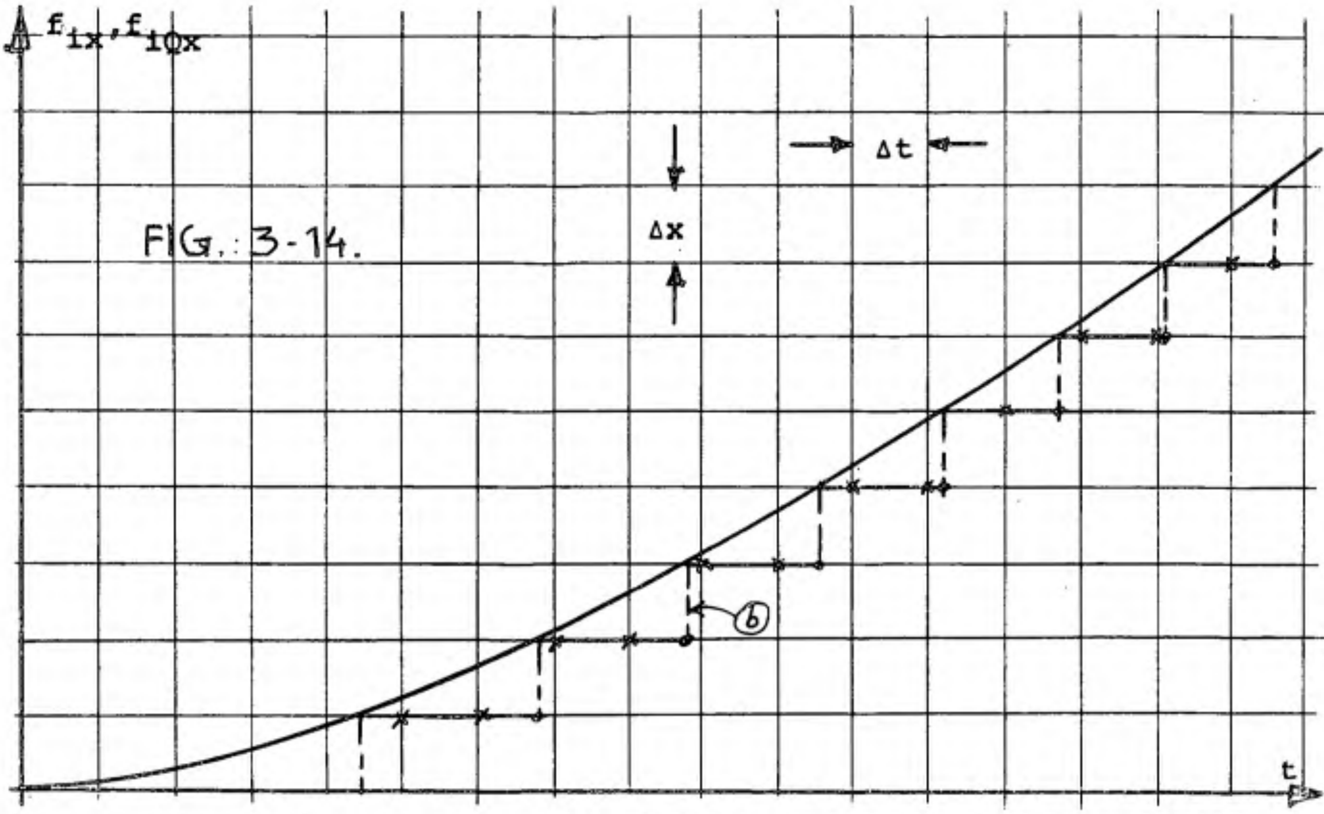
The quantization of function $e^{\neq t} \cdot \sin \omega t$ with respect to the quantums Δx and Δy, can be find by the intersections of (curve 1 and 2) as it is shown in figure (3-16)

The quantized points x_{iQ}, y_{iQ}, are delaid with the actual continuous function y (x). The difference between these quantized points and correspondent points of continuous function y (x) is the error of quantization ε_{iQx}, ε_{iQy} which should not be greater than Δx and Δy as it is shown in figure (3-17) and in equation (3-94).

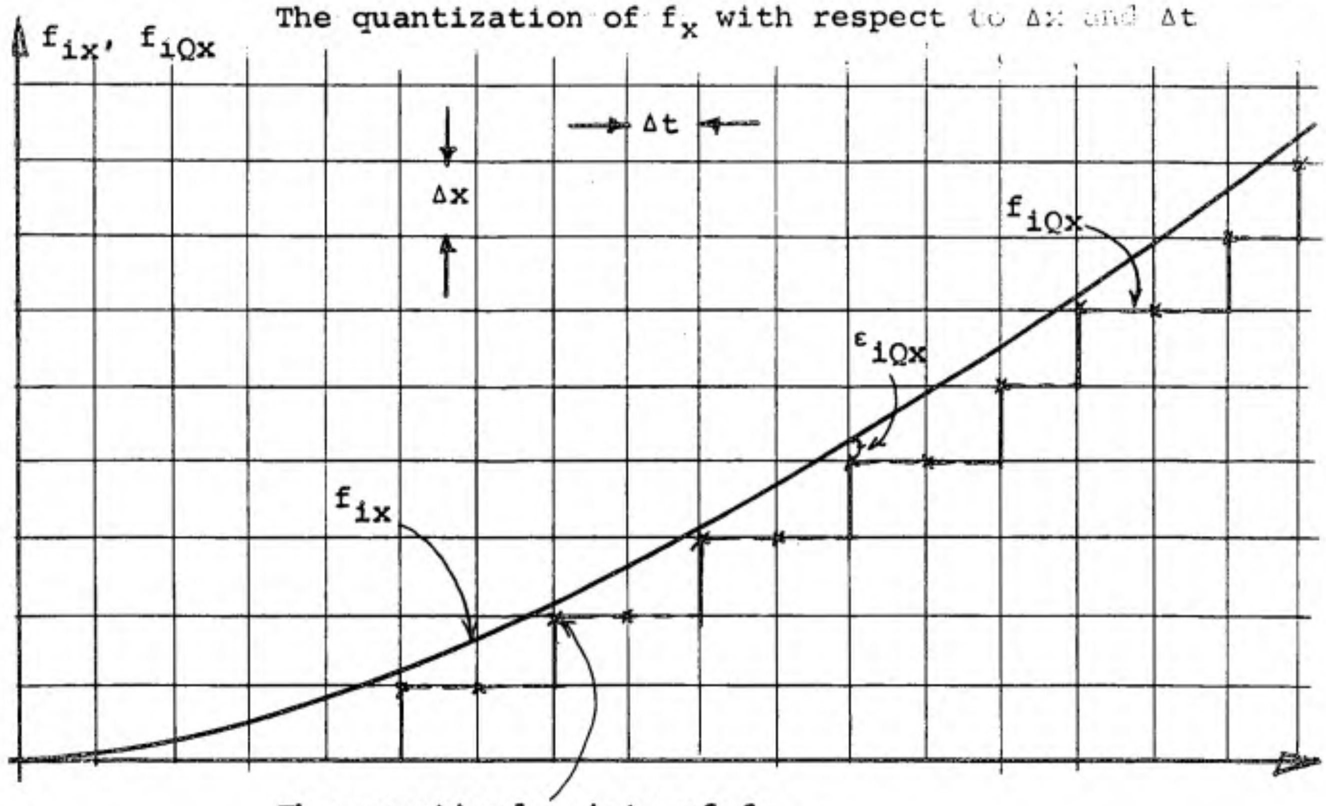
So

$$\begin{cases} \varepsilon_{iQx} = \overline{AB} < \Delta x \\ \varepsilon_{iQy} = \overline{CB} < \Delta y \end{cases} \tag{3-94}$$

The approximated interpolated integration function $\delta_i s^{*}(t)$ is:



The quantization of f_x with respect to Δx and Δt



The quantized points of f_{iQx}.

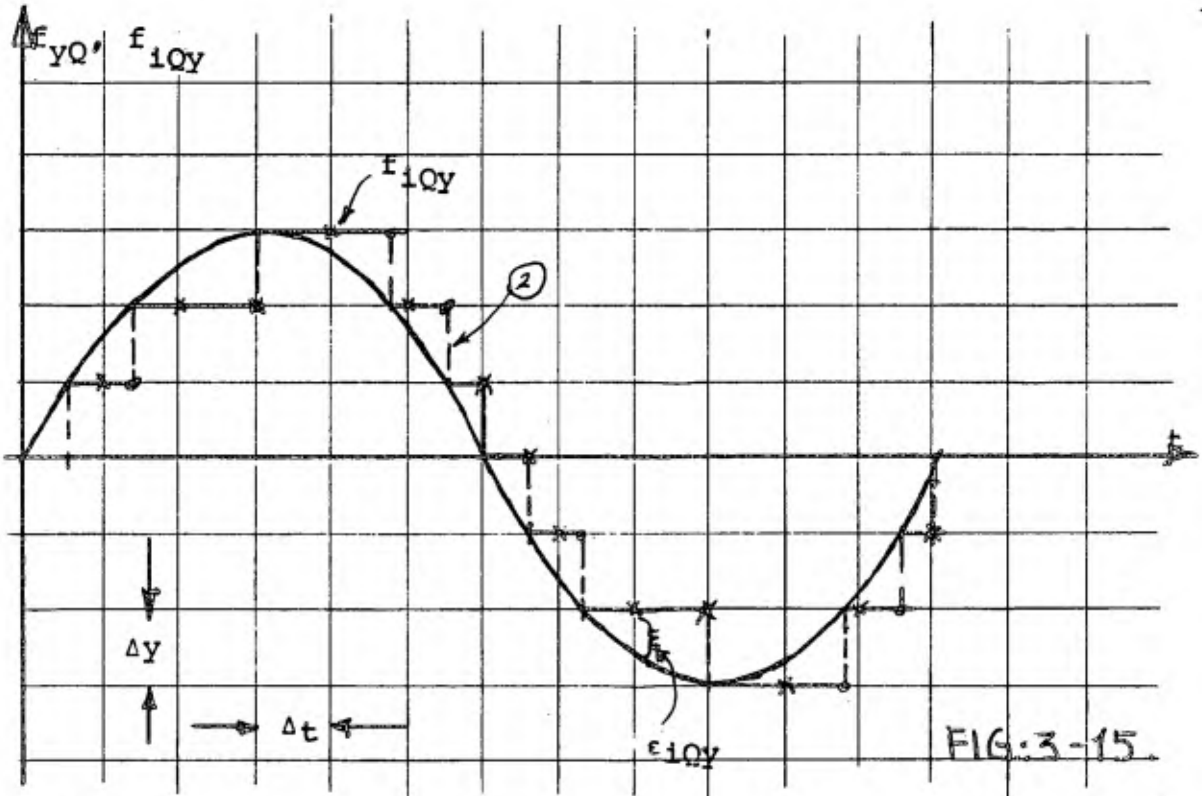


FIG. 3-15.

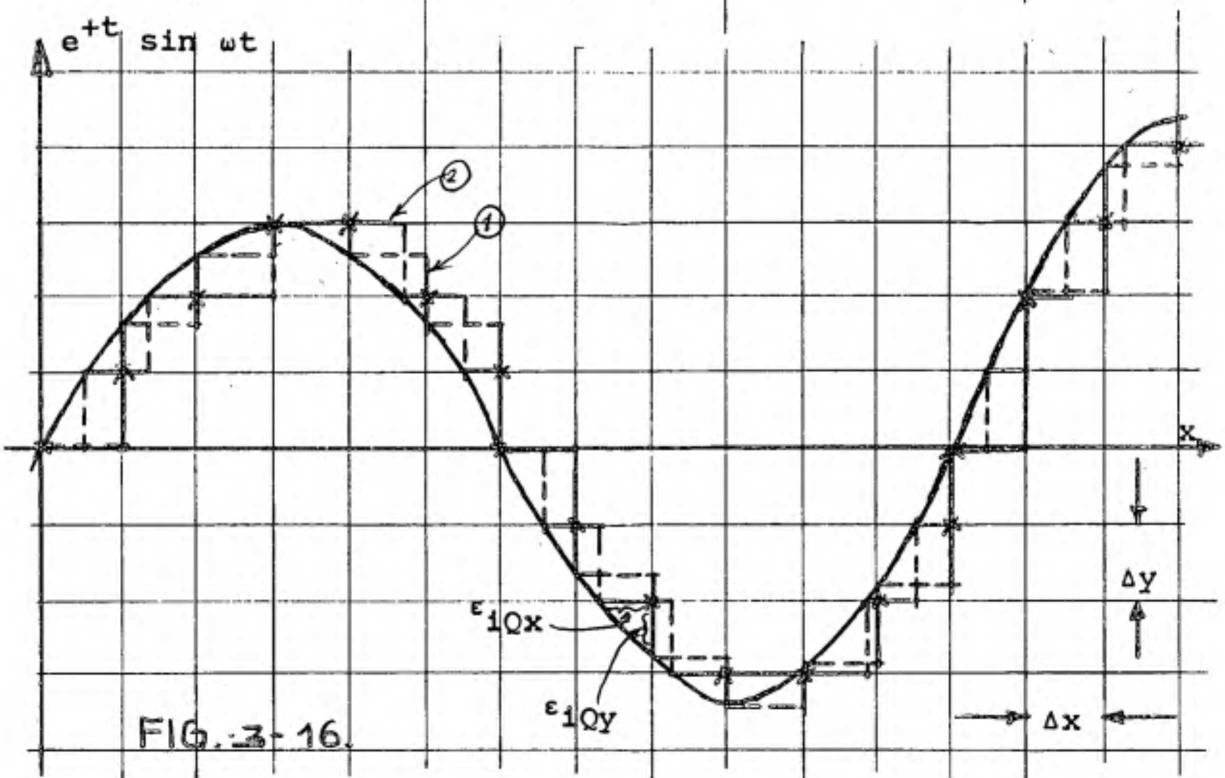


FIG. 3-16.

$$\delta_i \ s^{*}(t) = \int_{t_i}^{i+1} f_{iy} \ (t) \cdot d \frac{f_{ix}(t)}{dt} \, dt \tag{3-95}$$

if we put the value of the functions $f_{ix}(t)$ and $f_{iy}(t)$ from equation (3-93) in equation (3-95), then we will have :

$$\delta_i \ s^{*}(t) = \int_{t_i}^{t_{i+1}} (f_{iQy} + \varepsilon_{iQy}) \cdot d \frac{f_{iQx} + \varepsilon_{iQx}}{dt} \, dt \tag{3-96}$$

$$= \int_{t_i}^{t_{i+1}} f_{iQy} \cdot d \frac{f_{iQx}(t)}{dt} \, dt \left[+ \int_{t_i}^{t_{i+1}} f_{iQy} \cdot d \frac{\varepsilon_{iQx}}{dt} \, dt + \right. \tag{3-97}$$

$$\left. + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d \frac{f_{iQx}(t)}{dt} \, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d \frac{\varepsilon_{iQx}}{dt} \, dt \right]$$

The first term of equation (3-97) is the approximated interpolated quantized formula of integration $\delta_{iQ} \ s^{*}(t)$ which is the algorithm of machine as following :

$$\delta_{iQ} \ s^{*}(t) = \int_{t_i}^{t_{i+1}} f_{iQy} \cdot d \frac{f_{iQx}(t)}{dt} \, dt \tag{3-98}$$

The other terms of equation (3-97), are the error of quantization

ε_{iQ} in interval $t \in (t_i, t_{i+1})$ which is equal to:

$$\varepsilon_{iQ} = \int_{t_i}^{t_{i+1}} f_{iQy}(t) \cdot d\frac{\varepsilon_{iQx}(t)}{dt}\, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d\frac{f_{iQx}(t)}{dt}\, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy}(t) \cdot d\frac{\varepsilon_{iQx}(t)}{dt}\, dt \qquad (3\text{-}98)^{*}$$

So the equation (3-97) can be written as:

$$\delta_i\, s^{*}(t) = \delta_{iQ}\, s^{*}(t) + \varepsilon_{iQ}(t) \qquad t \in (t_i, t_{i+1}) \qquad (3\text{-}99)$$

By summing the equation (3-99) for k interval, we will have:

$$s^{*}(t) = \sum_{i=1}^{k} \delta_i\, s^{*}(t) \qquad (3\text{-}100)$$

$$= \sum_{i=1}^{k} \delta_i\, s_Q^{*}(t) + \sum_{i=1}^{k} \varepsilon_{iQ}(t) \qquad (3\text{-}101)$$

$$= s_Q^{*}(t) + \varepsilon_{tQ} \qquad (3\text{-}102)$$

where $s_Q^{*}(t)$ is the approximated interpolated and quantized formula of integration which is the algorithm of machine, equal to:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy} \cdot d\frac{f_{iQx}}{dt}\, dt \qquad (3\text{-}103)$$

and ε_{tQ} is the total quantization error in interval $t \in (t_o, t_k)$ in the process of integration as following;

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQ}\ (t) \tag{3-104}$$

$$= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy}(t) \cdot d\frac{\varepsilon_{iQx}(t)}{dt}\, dt +$$

$$+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d\frac{f_{iQx}(t)}{dt}\, dt + \tag{3-105}$$

$$+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy}(t) \cdot d\frac{\varepsilon_{iQx}(t)}{dt}\ dt$$

In equation (3-102), the $s^{*}(t)$ is the approximated interpolated continuous function of integration, s^{*}_{Q} (t) is the approximated interpolated quantized function of integration which is the Algorithm of machine, and $\varepsilon_{tQ}(t)$ is the error of quantization that is the difference between the approximated quantized and unquantized integration functions. The delay which exist in the process of quantization cause the phase shift between the continuous function and quantized function, it also cause the deformation of amplitude and the error of quantization. The error of quantization depends to the quantums of Δx, Δy, Δt which should take into consideration in the designing of incremental computer. In some cases, the error of quantization may dominate all others errors. In order to reduce the error of quantization, it is sufficient to reduce the value of quantums Δx, Δy, and Δt.

By considering the above discussion, the block diagram of incremental computer can be drawn as in the figure (3-18).

As it is seen from figure (3-18), the function y (x) is first interpolated and approximated to the functions f_{ix}, and f_{iy}, in interval $x \in (x_i, x_{i+1})$, with the error of ε_{ix}, ε_{iy} which cause the total error of method $\Gamma(t)$. Then this approximated function is quantized by the variable t of machine, and cause the error of quantization ε_{iQx}, ε_{iQy}, in each point that cause the total error of quantization $\varepsilon_{tQ}(t)$, it also introduce the delay $e^{-p\tau}$ where $|\tau| < \Delta x$.

3.3.1. Quantization error in the rectangular method of integration.

As it was shown in chapter 2, the interpolation functions f_{ix}, and f_{iy} in rectangular method of integration are:

$$\begin{cases} f_{iy} = y_i \\ f_{ix} = x_i \end{cases} \tag{3-106}$$

In the quantization process, there are the errors ε_{iQx}, ε_{iQy}, between the actual unquantized point (x_i, y_i) and their correspondent

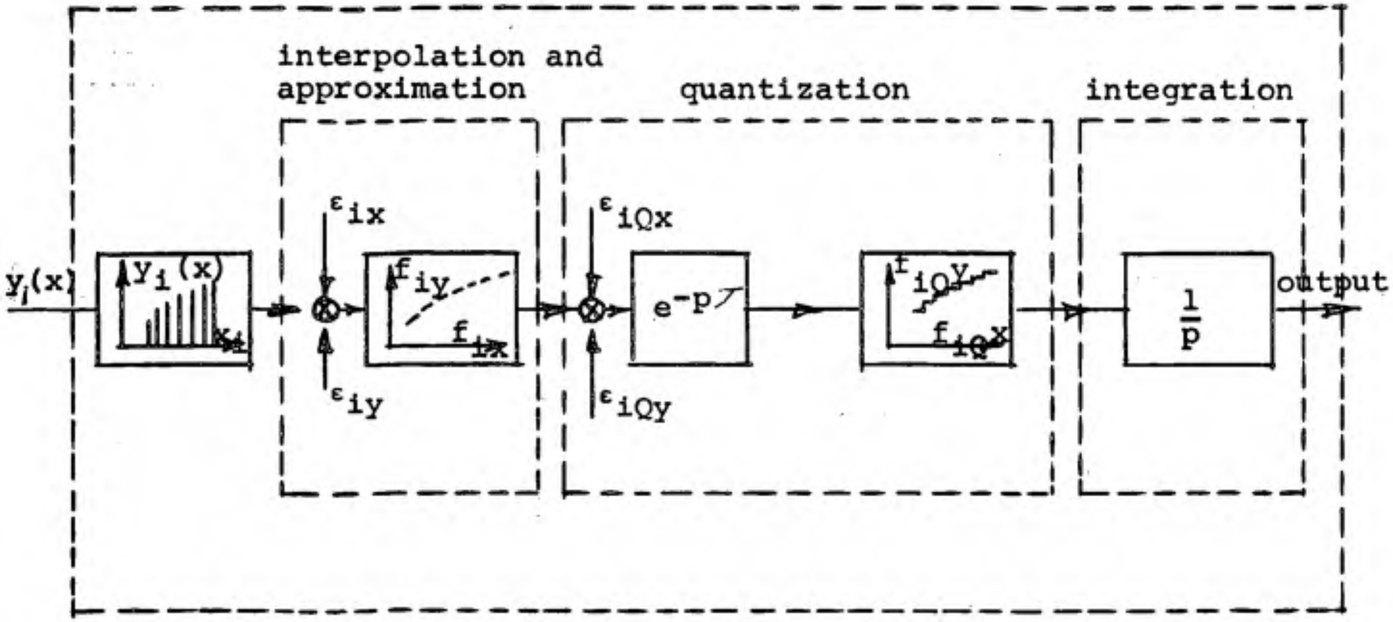


FIG.: 3 -18.

The block diagram of incremental computer.

quantized points (x_{iQ}, y_{iQ}) as following:

$$\begin{cases} \varepsilon_{iQx} = x_i - x_{iQ} \\ \varepsilon_{iQy} = y_i - y_{iQ} \end{cases} \tag{3-107}$$

by putting the equations (3-106), (3-107) in the integral equation $\delta_i s^*$:

$$\delta_i \; s^* = -\Delta x \int_0^{-1} f_{iy} \, d\xi \tag{3-108}$$

we will have:

$$\delta_i \; s^* = - (\Delta_{iQ} x + \Delta\varepsilon_{iQx}) \int_0^{-1} (y_{iQ} + \varepsilon_{iQy}) d\xi \tag{3-109}$$

$$= (\Delta_{iQ} x + \Delta\varepsilon_{iQx}) \left| \, y_{iQ} + \varepsilon_{iQy} \, \right| \tag{3-110}$$

or

$$\delta_i \; s^* = y_{iQ} \circ \Delta_{iQ} x + (y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ} x + + \Delta\varepsilon_{iQx} \circ \varepsilon_{iQy}) \tag{3-111}$$

The interpolated quantized function of integral $\delta_{iQ} \; s^*$ in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_{iQ} \; s^* = y_{iQ} \circ \Delta_{iQ} x \tag{3-112}$$

Then the equation (3-111) can be written as:

$$\delta_i \; s^{*} = \delta_{iQ} \; s^{*} + \varepsilon_{iQt} \tag{3-113}$$

where ε_{iQt} is the quantization error in interval $x \in (x_i, x_{i+1})$ as:

$$\varepsilon_{iQt} = y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x + \Delta\varepsilon_{iQx} \circ \varepsilon_{iQy} \tag{3-114}$$

The approximated interpolated formula $s^{*}(x)$ for k interval can be found from equations (3-112), (3-113) and (3-114) as following:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i \; s^{*} \tag{3-115}$$

$$= \sum_{i=1}^{k} \delta_{iQ} \; s^{*} + \sum_{i=1}^{k} \varepsilon_{iQt} \tag{3-116}$$

$$= \sum_{i=1}^{k} y_{iQ} \circ \Delta_{iQ}x + \sum_{i=1}^{k} (y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x + \Delta\varepsilon_{iQx} \circ \varepsilon_{iQy}) \tag{3-117}$$

$$= s_Q^{*} \; (x) + \varepsilon_{tQ} \tag{3-118}$$

from equations (3-115), (3-116), (3-117) and (3-118), it can be seen that the approximated interpolated quantized function of integral s_Q^{*} (x), which is the algorithm of machine, is:

$$s_Q^{*} \; (x) = \sum_{i=1}^{k} y_{iQ} \circ \Delta_{iQ}x \tag{3-119}$$

and the quantization error ε_{tQ} is:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x + \Delta\varepsilon_{iQx} \circ \varepsilon_{iQy}) \tag{3-120}$$

In equation (3-120) the third term can be neglected with respect to the two first ones. So the quantization error ε_{tQ} is equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x) \tag{3-121}$$

As $\varepsilon_{iQy} < \Delta y$ so

$$\sum_{i=1}^{k} \varepsilon_{iQy} \circ \Delta_{iQ}x < \Delta y \; (x_k - x_o) \tag{3-122}$$

The first term of equation (3-121) is:

$$\sum_{i=1}^{k} y_{iQ} \circ \Delta\varepsilon_{iQx} = \sum_{i=1}^{k} y_{iQ} \circ (\varepsilon_{iQx} - \varepsilon_{(i-1)Qx}) \tag{3-123}$$

as $\varepsilon_{iQx} < \Delta x$ and $\varepsilon_{(i-1)Qx} < \Delta x$, the equation (3-123) can be transformed to:

$$\sum_{i=1}^{k} y_{iQ} \circ \Delta\varepsilon_{iQx} < \Delta x \; (y_k - y_o) \tag{3-124}$$

if we put the value of equation (3-122) and (3-124) in equation (3-121), we will have:

$$\varepsilon_{tQ} < \Delta y \, (x_{kQ} - x_{oQ}) + \Delta x \, (y_{kQ} - y_{oQ}) \qquad (3\text{-}125)$$

The equation (3-125) gives the quantization error ε_{tQ} in the rectangular method of integration. As it is seen, the quantization error ε_{tQ} depends to the quantums Δx, Δy, and the interval of integration $x \in (x_o, x_k)$.

3.3.2. Quantization error in the trapezoidal method of integration.

In the trapezoidal method of integration, the approximated interpolated functions f_{ix} and f_{iy} are:

$$\begin{cases} f_{ix} = x_i - \xi \, \Delta_i x \\ f_{iy} = y_i - \xi \, \Delta_i y \end{cases} \qquad (3\text{-}126)$$

by putting the equation (3-107) in equation (3-126), we will have:

$$\delta_i \, s_Q^{*} = - \Delta x \int_0^{-1} f_{iy} \, d\xi = - \Delta (x_{iQ} + \varepsilon_{iQx})$$

$$\int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi \, (\Delta_Q y + \Delta \varepsilon_{iQy})] \, d\xi \qquad (3\text{-}127)$$

$$= (\Delta_{iQ}x + \Delta\varepsilon_{iQx}) \; [(y_{iQ} + \varepsilon_{iQy}) + \frac{1}{2}(\Delta_{iQ}y + \Delta\varepsilon_{iQy})] \tag{3-128}$$

$$= [y_{iQ} \circ \Delta_{iQ}x + \frac{1}{2} \Delta_{iQ}x \circ \Delta_{iQ}y] + [y_{iQ} \cdot \Delta\varepsilon_{iQx} + \tag{3-129}$$

$$+ \varepsilon_{iQy} \cdot \Delta_{iQ}x + \Delta_{iQ}x \cdot \Delta\varepsilon_{iQy} + \varepsilon_{iQx} \circ \Delta\varepsilon_{iQy} +$$

$$+ \frac{1}{2}\varepsilon_{iQy} \; (\Delta_{iQ}y + \Delta\varepsilon_{iQy})]$$

The formula of trapezoidal method of integration in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_i \; s_Q^{*} = y_{iQ} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_{iQ}y \tag{3-130}$$

from equation (3-129) and (3-130), we can write the following equation:

$$\delta_i \; s^{*} = \delta_i \; s_Q^{*} + \varepsilon_{iQt} \tag{3-131}$$

where ε_{iQt} is the quantization error in interval $x \in (x_i, x_{i+1})$, which is equal to:

$$\varepsilon_{iQt} = y_{iQ} \cdot \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x + \Big(\Delta_{iQ}x \cdot \Delta\varepsilon_{iQy} + \tag{3-132}$$

$$+ \varepsilon_{iQx} \cdot \Delta\varepsilon_{iQy} + \Delta\varepsilon_{iQx} \circ \Delta\varepsilon_{iQy} + \frac{1}{2}\varepsilon_{iQy} \circ \Delta_{iQ}y +$$

$$+ \frac{1}{2}\varepsilon_{iQy} \circ \Delta\varepsilon_{iQy}\Big)$$

In equation (3-132), the terms in the bracket are neglectable in comparing with the first two terms. Therefore, the quantization error in interval $x \in (x_i, x_{i+1})$ will be:

$$\varepsilon_{iQt} = y_{iQ} \cdot \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \Delta_{iQ}x \qquad (3\text{-}133)$$

The approximated interpolated formula of integral for k interval can be found from equation (3-131) as:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i s^{*} \qquad (3\text{-}134)$$

$$= \sum_{i=1}^{k} \delta_{iQ} s^{*} + \sum_{i=1}^{k} \varepsilon_{iQt} \qquad (3\text{-}135)$$

$$= s_Q^{*}(x) + \varepsilon_{tQ} \qquad (3\text{-}136)$$

where $s_Q^{*}(x)$ is the interpolated quantized integration formula that can be found from equation (3-130), (3-135) and (3-136) as following:

$$s_Q^{*}(x) = \sum_{i=1}^{k} \delta_{iQ} s^{*} = \sum_{i=1}^{k} \left(y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \cdot \Delta_{iQ}y\right) \qquad (3\text{-}137)$$

and the total quantization error ε_{tQ} in k interval from equations (3-133), (3-135) and (3-136) will be:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQt} \qquad (3\text{-}138)$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \Delta_{iQ}x) \qquad (3\text{-}139)$$

or

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (y_{iQ} \cdot \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \Delta_{iQ}x) \qquad (3\text{-}140)$$

It is seen from equations (3-121) and (3-140) that, the quantization error in trapezoidal method of integration is same as the rectangular method. Therefore, from equations (3-122), (3-124) and (3-140) the quantization error ε_{tQ} is:

$$\varepsilon_{tQ} < \Delta y\ (x_{kQ} - x_{oQ}) + \Delta x\ (y_{kQ} - y_{oQ}) \qquad (3\text{-}141)$$

As it is seen from equation (3-141), the quantization error $_{tQ}$ depends to the quantums Δx, Δy and the interval of integration.

3.3.3. Quantization error in the three points method of integration.

In the three points method of integration, the approximated interpolated functions f_{ix} and f_{iy} which are replaced to the functions X (t) and Y (t) are:

$$\begin{cases} f_{iy}\ (\xi) = y_i - \xi\ \Delta_i y - \dfrac{\xi(\xi+1)}{2!}\ \Delta_i^{(2)} y \\ \\ f_{ix}\ (\xi) = x_i - \xi\ \Delta_i x - \dfrac{\xi(\xi+1)}{2!}\ \Delta_i^{(2)} x \end{cases} \qquad (3\text{-}142)$$

$$d\,\frac{f(\xi)}{d\xi} = -\,\Delta_i x - \frac{2\xi+1}{2!}\,\Delta_i^2 x$$

As we have discussed, the approximated interpolated formula of integral $\delta_i s^*$ in interval $t \in (t_i, t_{i+1})$ is:

$$\delta_i\; s^* = \int_0^{-1} f_{iy}\;(\xi)\; d\frac{f_{ix}(\xi)}{d\xi}\, d\xi \tag{3-143}$$

$$= \int_0^{-1} [y_i - \xi\;\Delta_i y - \frac{\xi(\xi+1)}{2!}\Delta_i^2 y]\;[-\Delta_i x - \frac{2\xi+1}{2!}\Delta_i^{(2)} x]\; d\xi \tag{3-144}$$

By putting the equation (3-107) in equation (3-144), we will have:

$$\delta_i\; s^* = \int_0^{-1} \left[(y_{iQ} + \epsilon_{iQy}) - \xi\;\Delta_i(y_{iQ} + \epsilon_{iQy}) - \frac{\xi(\xi+1)}{2!}\,\Delta_i^{(2)}(y_{iQ} + \epsilon_{iQy}) \right] \tag{3-145}$$

$$\left[-\,\Delta_i\;(x_{iQ} + \epsilon_{iQx}) - \frac{2\xi+1}{2!}\,\Delta_i^{(2)}\;(x_{iQ} + \epsilon_{iQx}) \right]\, d\xi$$

$$= \left[\bar{y}_{iQ} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_{iQ}y + \frac{1}{12}(\Delta_{iQ}y \circ \Delta_{(i-1)Q}x - \right.$$

$$\left. - \Delta_{iQ}x \circ \Delta_{(i-1)Q}y) \right] + \left[\bar{y}_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQx} \circ \Delta_{iQ}x) + \right. \tag{3-146}$$

$$+ \frac{1}{2}\Delta\varepsilon_{iQx} \circ \Delta\varepsilon_{iQy} + \frac{1}{2}\Delta_{iQ}x \circ \Delta\varepsilon_{iQy} + \frac{1}{12}\Delta_{iQ}x \circ \varepsilon_{iQy} +$$

$$\left. + \cdots\cdots \right]$$

$$= \delta_{iQ}\, s^{*} + \varepsilon_{iQt} \tag{3-147}$$

As we have discussed before in three points method, the approximated quantized formula of integral $\delta_{iQ}\, s^{*}$ in interval $t \in (t_i, t_{i+1})$ is:

$$\delta_{iQ}\, s^{*} = y_{iQ} \circ \Delta_{iQ}x + \frac{1}{2}\Delta_{iQ}x \circ \Delta_{iQ}y + \frac{1}{12}[\Delta_{iQ}y \circ \Delta_{(i-1)Q}x - \tag{3-148}$$

$$- \Delta_{(i-1)Q}y \circ \Delta_{iQ}x]$$

and the quantization error ε_{iQt} in this interval of integration from equations (3-146) and (3-147) is:

$$\varepsilon_{iQt} = \left[(y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x) + \frac{1}{2}\Delta\varepsilon_{iQx} \cdot \Delta\varepsilon_{iQy} + \right. \tag{3-149}$$

$$+ \frac{1}{2}\Delta_{iQ}x \cdot \Delta\varepsilon_{iQy} + \frac{1}{12}\Delta_{iQ}x \circ \Delta\varepsilon_{iQy} + \frac{1}{12}\Delta_{iQ}x \circ \Delta\varepsilon_{(i-1)Qy} +$$

$$+ \ldots\ldots \Big]$$

In equation (3-149), all terms can be neglected with respect to the two first one. Therefore the quantization error ε_{iQt} in interval $t \in (t_i, t_{i+1})$ will be:

$$\varepsilon_{iQt} = y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x \qquad (3\text{-}150)$$

The approximated interpolated formula of integration $s^*(x)$ for k interval will be:

$$s^*(x) = \sum_{i=1}^{k} \delta_i \, s^*(x) \qquad (3\text{-}151)$$

$$= \sum_{i=1}^{k} \delta_{iQ} s^* + \sum_{i=1}^{k} \varepsilon_{iQt} \qquad (3\text{-}152)$$

$$= s_Q^* \,(x) + \varepsilon_{tQ} \qquad (3\text{-}153)$$

The approximated interpolated and quantized formula of integration s_Q^* (x) which is the algorithm of machine is:

$$s_Q^* \,(x) = \sum_{i=1}^{k} \Big[y_{iQ} \cdot \Delta_{iQ}x + \frac{1}{2} \Delta_{iQ}x \cdot \Delta_{iQ}y + \frac{1}{12} (\Delta_{iQ}y \cdot \Delta_{(i-1)Q}x - \Delta_{(i-1)Q}y \cdot \Delta_{iQ}x) \Big] \qquad (3\text{-}154)$$

and the quantization error ε_{tQ} can be found from equations (3-150), (3-152) and (3-153) as following:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (y_{iQ} \circ \Delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \Delta_{iQ}x) \qquad (3\text{-}15\overset{*}{4})$$

As we have seen before, the equation (3-15$\overset{*}{4}$) can be transformed to the equation (3-155).

$$\varepsilon_{tQ} < \Delta y\,(x_{kQ} - x_{oQ}) + \Delta x\,(y_{kQ} - y_{oQ}) \qquad (3\text{-}155)$$

It is seen from equation (3-155), the quantization error ε_{tQ} depends on the quantums Δx, Δy, and the interval of integration.

3.4. The quantization error in multiple increment computation, when the independent variable of integration X is the independent variable t.

When the independent variable x of integral is equal to the independent variable t of machine, then the formula of integration is:

$$s(x) = \int_{x_o}^{x} y(x)\,dx \tag{3-166}$$

or its approximated interpolated formula for interval $t \in (t_i, t_{i+1})$ is:

$$\delta_i\, s^*(t) = \int_{t_i}^{t_{i+1}} f_{iy}(t)\,dt \tag{3-167}$$

As it was discussed earlier y (t) is replaced by the interpolated function $f_{iy}(t)$ which gives the approximated value of integral $s^*(t)$ so:

$$\begin{cases} f_{iy}(t) = f_y\,[t_o, y_o, \delta y_1, \delta y_2, \ldots\ldots] \\ s^*(t) = \sum\limits_{i=1}^{k} \displaystyle\int_{t_i}^{t_{i+1}} f_{iy}(t)\,dt \end{cases} \tag{3-168}$$

By this approximation, there will be an error of method $\Gamma(t)$, which is the difference between the actual value of integration s (t) and the approximated interpolated value $s^{*}(t)$ like:

$$\begin{cases} \Gamma(t) = - s^{*}(t) + s(t) \\ \Gamma(t) = \Gamma[x_o, y_o, \delta_i x, \delta_i y] \end{cases} \qquad (3\text{-}169)$$

In digital machine all the quantities are in discret value or are quantized within an interval, so the continuous interpolated function f_{iy} (t) is quantized with the quantum of variable of the machine Δt and the dependent variable Δy.

An example is the generation of a sine wave as figure (3-19).

As we have seen the continuous function f_{iy} (x) is quantized with respect to the quantums Δt, Δy with the inherent delay of digital machine (is shown in figure 3.20). The quantized points of continuous function f_{iy} (x) are the intersection of curve (1) with the line Δt = const. (it is shown in figure 3.20).

As we work with multiple increments, in this case we assume $\delta x = 3\ \Delta t$ and $2^{-2} < \delta y < 2^{2}$, so the only quantized points which are available in the machine are the points in interval $\Delta x = 3\ \Delta t$, that are shown in figure (3-20) by the points ⊗.

As the independent variable of integral is the independent variable of machine, f_{ix} (x) = x (t) = t, so there is no quantification error for function f_{ix} in each point, i.e : $\epsilon_{Qtx} = 0$.

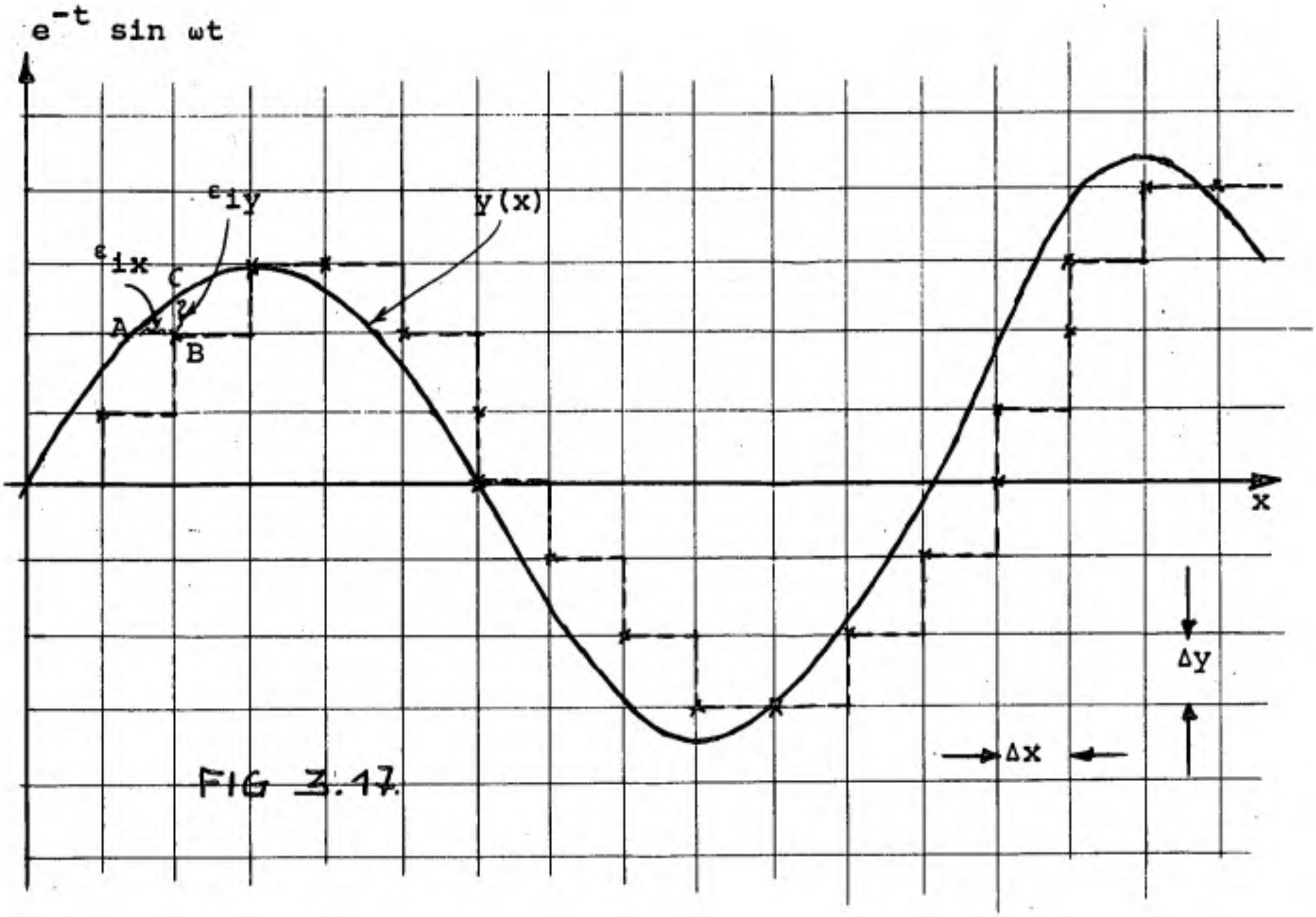


FIG 3.17.

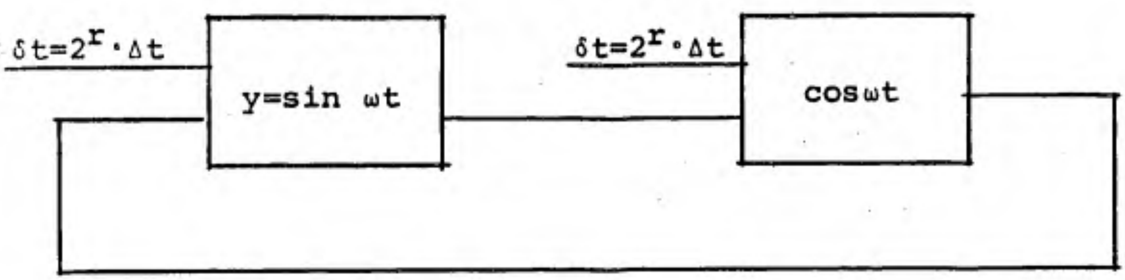
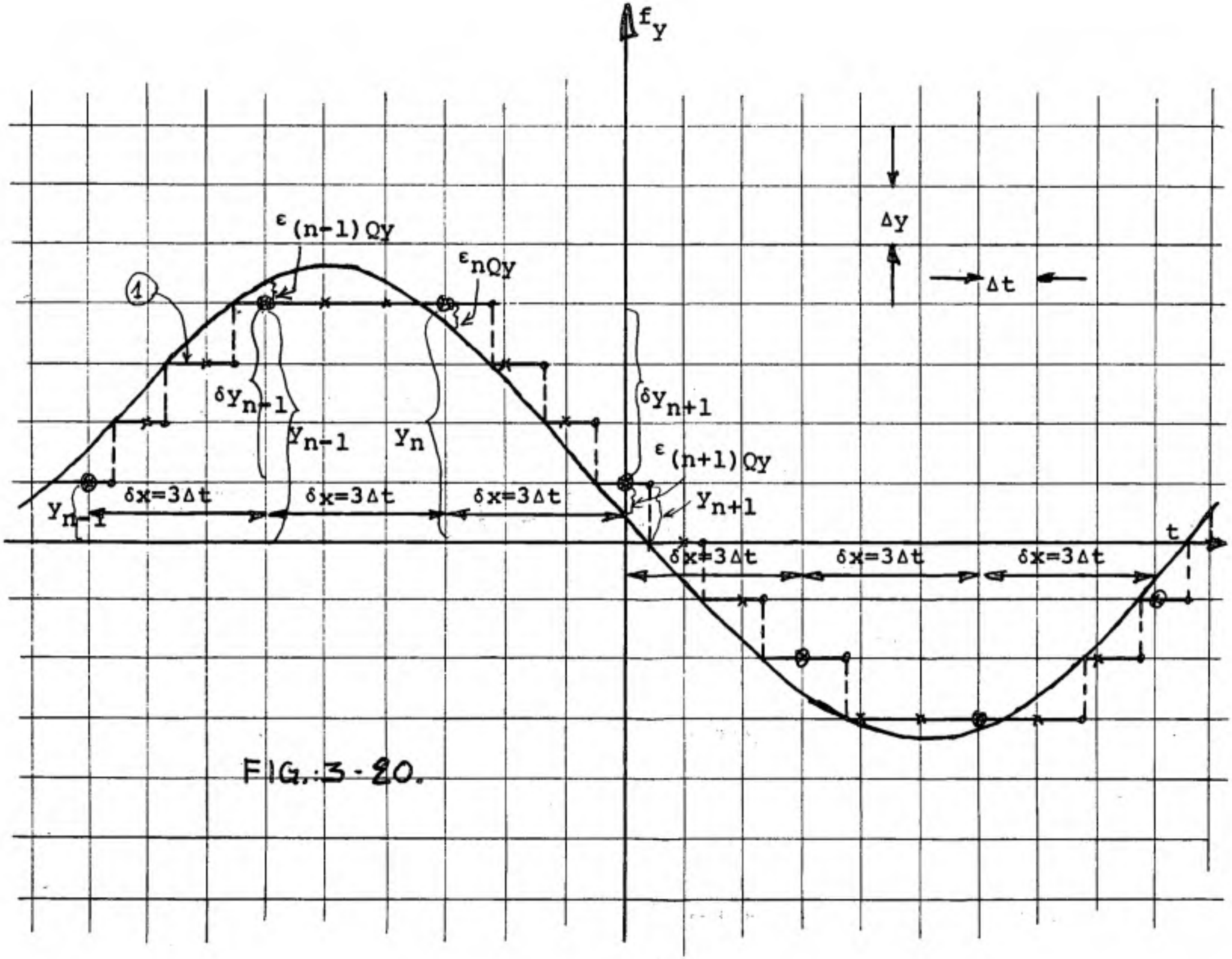


FIG.: 3-19.



f_y
Δy
Δt
$\varepsilon_{(n-1)Qy}$
ε_{nQy}
1
δy_{n-1}
y_{n-1}
y_n
δy_{n+1}
$\varepsilon_{(n+1)Qy}$
y_{n+1}
y_{n-1}
$\delta x=3\Delta t$
$\delta x=3\Delta t$
$\delta x=3\Delta t$
$\delta x=3\Delta t$
$\delta x=3\Delta t$
$\delta x=3\Delta t$
t

FIG. 3-20.

But as it is seen from figure (3-20) and (3-21), in each quantized point there is an error of quantization ε_{iQy} which is the difference between the continuous function f_{iy} and the quantized function f_{iQy} as following:

$$\varepsilon_{Qty} = f_{iy}(x) - f_{iQy}(x) \tag{3-170}$$

the quantization error ε_{iQy} in points (x_i, y_i) is smaller than the quantum Δy so:

$$\left|\varepsilon_{iQy}\right| < \Delta y \tag{3-171}$$

Therefore the quantization error in each point (x_i, y_i) will be:

$$\begin{cases} \varepsilon_{iQx} = 0 \\ \varepsilon_{iQy} = f_{iy}(x) - f_{iQy}(x) \qquad x \subseteq (x_i, x_{i+1}) \end{cases} \tag{3-172}$$

by putting the value of f_{iy} from equation (3-172) into equation (3-167), we will have:

$$\delta_i \, s^{*} = \int_{t_i}^{t_{i+1}} [f_{iQy}(t) + \varepsilon_{iQy}] \, dt \qquad x \in (x_i, x_{i+1}) \tag{3-173}$$

The approximated formula of integral in k interval will be:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i \, s^{*} \tag{3-174}$$

$$= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} [f_{iQy}(t) + \varepsilon_{iQy}] \; dt \tag{3-175}$$

$$= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy}(t) \; dt + \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot dt \tag{3-176}$$

$$= s_Q^{*}(t) + \varepsilon_{tQ} \tag{3-177}$$

where $s_Q^{*}(t)$ is the approximated interpolated and quantized function of integral which is the algorithm of machine and equal to:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy} \cdot dt \tag{3-178}$$

and the ε_{tQ} is the error of quantization in k interval which is equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot dt \tag{3-179}$$

from equations (3-169) and (3-177), it can be seen that the actual value of integration is the sum of the approximated interpolated quantized function of integral $s_Q^*(x)$, plus the error of method $\Gamma(x)$ and the error of quantization ε_{tQ} as following:

$$s(x) = s_Q^*(x) + [\Gamma(x) + \varepsilon_{Qt}(x)] \qquad (3\text{-}180)$$

As we have discussed already, because of delay of quantization process, there is an error of phase betxeen the continuous function and the discret quantized function. The error of phase and quantization depends to the value of quantums Δx, Δy, Δs, and also to the multiple increments $\delta x = 2^r \cdot \Delta x$, $\delta y = 2^r \cdot \Delta y$ and $\delta s = 2^r \cdot \Delta s$. ($0 < r < h$). By reducing the value of quantums Δx, Δy, Δs, and the increments δx, δy, δs, the quantization error will reduce too.

By taking into account the errors and delay of incremental system, the block diagram of integration will be as figure (3-10).

It is seen from figure (3-10) that the function $y(x)$ is first interpolated to the function $f_{iy}(x)$ in interval $x \in (x_i, x_{i+1})$, with the error ε_{iy} which cause the total error of method $\Gamma(x)$, then the approximated function is quantized by the variable of machine t, and cause the error ε_{iQy} in each point which cause the total error of quantization $\varepsilon_{tQ}(t)$, and also it introduce the delay of $e^{-p\tau}$ for $\tau < \Delta t$.

3.4.1. Quantization error in the rectangular method of integration.

The approximated interpolated function f_{iy} which is replaced to the function y (x) in interval $x \in (x_i, x_{i+1})$ in the rectangular method of integration is:

$$f_{iy} = y_i \qquad\qquad x \in (x_i, x_{i+1}) \tag{3-181}$$

and the approximated interpolated formula of integral $\delta_i s^*$ in interval $x \in (x_i, x_{i+1})$ is:

$$\delta_i s^* = - \delta_i x \int_0^{-1} f_{iy} \cdot d\xi \tag{3-182}$$

In the quantization process, there is the error ε_{iQy} between the actual unquantized points (x_i, y_i) and their correspondent quantized points (x_{iQ}, y_{iQ}) as following:

$$\varepsilon_{iQy} = y_i - y_{iQ} \tag{3-183}$$

By putting the equations (3-181) and (3-183) in equation (3-182) we will have:

$$\delta_i s^* = - \delta_i x \int_0^{-1} (y_{iQ} + \varepsilon_{iQy}) \ d\xi \tag{3-184}$$

$$= \delta_i x \circ y_{iQ} + \delta_i x \circ \varepsilon_{iQy} \tag{3-185}$$

$$= \delta_i \; s_Q^{*} + \delta_i x \circ \varepsilon_{iQy} \tag{3-186}$$

The approximated formula of integral for k interval will be:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i \; s^{*}(x) \tag{3-187}$$

$$= \sum_{i=1}^{k} y_{iQ} \circ \delta_{iQ}x + \sum_{i=1}^{k} \varepsilon_{iQy} \circ \delta_{iQ}x \tag{3-188}$$

$$= s_Q^{*} \; (x) + \varepsilon_{tQ} \tag{3-189}$$

In equations (3-187), (3-188) and (3-189), the approximated quantized function of integral s_Q^{*} (x) which is the algorithm of machine is:

$$s_Q^{*} \; (x) = \sum_{i=1}^{k} y_{iQ} \circ \delta_{iQ}x \tag{3-190}$$

where

$$\delta_{iQ}x = 2^r \circ \Delta x$$

and the quantization error ε_{tQ} is equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQy} \circ \delta_{iQ}x \tag{3-191}$$

as the increment $\delta_{iQ}x$ is the multiple increment $\delta_{iQ}x = 2^r \circ \Delta x$, and the ε_{iQy} is smaller than the quantum Δy so:

$$\begin{cases} \delta_{iQ}x = 2^r \cdot \Delta x \\ \left|\varepsilon_{iQy}\right| < \Delta y \end{cases} \tag{3-192}$$

by putting the values of equation (3-192) in equation (3-191), we will have:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} \Delta y \cdot 2^r \cdot \Delta_{iQ}x$$
$$h > r > 0 \tag{3-193}$$

as the maximum value of $r_{max} = + h$ so the equation (3-193) can be written as:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} \Delta y \cdot 2^h \cdot \Delta_{iQ}x \tag{3-194}$$

or

$$\varepsilon_{tQ} < \Delta y \cdot 2^h \sum_{i=1}^{k} \Delta_{iQ}x \tag{3-195}$$

or

$$\varepsilon_{tQ} < \Delta y \cdot 2^h (x_{kQ} - x_{oQ}) \tag{3-196}$$

It is seen from equation (3-196), that the quantization error ε_{tQ} depends to the quantum Δy to the number of bit (h) which is chosen for multiple increment $\delta_i x = 2^h \cdot \Delta x$, and also to the duration of integral $(x_k - x_o)$.

3.4.2. Quantization error in the trapezoidal method of integration.

In the trapezoidal method of integration, the approximated interpolated function f_{iy}, which is replaced to the function y (x) is as following:

$$f_{iy} = y_i - \xi \circ \delta_i y \qquad (3\text{-}197)$$

where $\delta_i y = 2^r \cdot \Delta_i y$

By putting the value of y_i from equation (3-183) into the equation (3-197), we will have:

$$f_{iy} = (y_{iQ} + \varepsilon_{iQy}) - \xi\ \delta_i\ (y_{iQ} + \varepsilon_{iQy}) \qquad (3\text{-}198)$$

Therefore the approximated interpolated formula of integral $\delta_i\ s^*(x)$ from equation (3-182) can be written as:

$$\delta_i\ s^*(x) = -\delta_i x \int_0^{-1} f_{iy}\ d\xi \qquad (3\text{-}199)$$

$$= -\delta_i x \int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi\ \delta_i\ (y_{iQ} + \varepsilon_{iQy})]\ d\xi \qquad (3\text{-}200)$$

$$= y_{iQ} \circ \delta_{iQ}x + \frac{1}{2}\delta_{iQ}x \circ \delta_{iQ}y + (\delta_{iQ}x \circ \varepsilon_{iQy} + \frac{1}{2}\delta_{iQ}x \circ \delta\varepsilon_{iQy}) \tag{3-201}$$

$$= \delta_{iQ}\, s^{*} + \varepsilon_{iQt} \tag{3-202}$$

The approximated interpolated formula of integral $s^{*}(x)$ for k interval is:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i\, s^{*}(x) \tag{3-203}$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}x \cdot \delta_{iQ}y) + \sum_{i=1}^{k} \delta_{iQ}x\, (\varepsilon_{iQy} + \frac{1}{2}\delta\varepsilon_{iQy}) \tag{3-204}$$

$$= s_Q^{*}\,(x) + \varepsilon_{tQ} \tag{3-205}$$

from equations (3-203), (3-204) and (3-205), it can be seen that, the approximated interpolated quantized formula of integration s_Q^{*} (x) which is algorithm of machine, is equal to:

$$s_Q^{*}\,(x) = \sum_{i=1}^{k} (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}x \cdot \delta_{iQ}y) \tag{3-206}$$

$$\delta_{iQ}x = \delta_i x = 2^{r} \cdot \Delta t$$

$$\delta_{iQ}y = 2^r \cdot \Delta y$$

and the total quantization error ε_{tQ} is:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \delta_{iQ}x \; (\varepsilon_{iQy} + \frac{1}{2}\delta\varepsilon_{iQy}) \tag{3-207}$$

in equation (3-207), the second term can be neglected with respect to the first one, so ε_{tQ} will be:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \delta_{iQ}x \cdot \varepsilon_{iQy} \tag{3-208}$$

as $\qquad \delta_{iQ}x = 2^r \cdot \Delta_i x$

and $\qquad \left|\varepsilon_{iQy}\right| < \Delta y$

so the equation (3-208) can be written as:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} 2^r \cdot \delta_i x \cdot \Delta y \tag{3-209}$$

$$h > r > 0$$

The maximum value of r is equal to h, ie: $r_{max} = h$, so the equation (3-209) is expressed as following:

$$\varepsilon_{tQ} < 2^h \cdot \Delta y \sum_{i=1}^{k} \Delta_i x \tag{3-210}$$

or

$$\varepsilon_{tQ} < 2^h \cdot \Delta y \ (x_{kQ} - x_{oQ}) \tag{3-211}$$

The equation (3-211) gives the value of quantization error in multiple increment integration, as it is seen, the quantization error ε_{tQ} in trapezoidal method of integration is same as the quantization error of rectangular integration (equation 3-196).

3.4.3. Quantization error in the three points method of integration.

In the three points formula of integration, the y (x) function is replaced with the interpolated function f_{iy} as:

$$f_{iy} = y_i - \xi \ \delta_i y - \frac{\xi(\xi+1)}{2!} \delta_i^{(II)} y \tag{3-212}$$

by putting the value of y_i from equation (3-183) into the equation (3-212), we will have:

$$f(\xi) = (y_{iQ} + \varepsilon_{iQy}) - \xi \ \delta_i \ (y_{iQ} + \varepsilon_{iQy}) - \frac{\xi(\xi+1)}{2!} \delta_i^{II} \ (y_{iQ} + \varepsilon_{iQy}) \tag{3-213}$$

$$= (y_{iQ} + \varepsilon_{iQy}) - \xi \ (\delta_{iQ} y + \delta \varepsilon_{iQy}) - \frac{\xi(\xi+1)}{2!} (\delta_{iQ}^{II} y + \delta^{II} \varepsilon_{iQy}) \tag{3-214}$$

Therefore, the approximated interpolated formula of integration in interval $x \in (x_i, x_{i+1})$ will be:

$$\delta_i s^{*} = -\delta_{iQ}x \int_0^{-1} f_{iy} \cdot d\xi \tag{3-215}$$

$$= -\delta_{iQ}x \int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi(\delta_{iQ}y + \delta\varepsilon_{iQy}) - \frac{\xi(\xi+1)}{2!}(\delta^{II}_{iQ}y + \delta^{II}\varepsilon_{iQy})] \; d\xi \tag{3-216}$$

$$= [y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}x \cdot \delta_{iQ}y + \frac{1}{12}\delta_{iQ}x \cdot \delta^{II}_{iQ}y] + + [\delta_{iQ}x \cdot \varepsilon_{iQy} + \frac{1}{2}\delta_{iQ}x \cdot \delta\varepsilon_{iQy} + \frac{1}{12}\delta_{iQ}x \cdot \delta^{II}\varepsilon_{iQy}] \tag{3-217}$$

The approximated interpolated formula of integral $s^{*}(x)$ for k interval will be:

$$s^{*}(x) = \sum_{i=1}^{k} \delta_i s^{*} \tag{3-218}$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \cdot \delta_{iQ}x + \frac{1}{12}\delta^{II}_{iQ}y \cdot \delta_{iQ}x) +$$

$$+\sum_{i=1}^{k}\left(\delta_{iQ}x\circ\varepsilon_{iQy}+\frac{1}{2}\delta_{iQ}x\circ\delta\varepsilon_{iQy}+\right.$$

$$\left.+\frac{1}{12}\,\delta_{iQ}x\circ\delta^{II}\varepsilon_{iQy}\right) \tag{3-219}$$

$$= s_Q^{*}\,(x) + \varepsilon_{tQ} \tag{3-220}$$

The approximated interpolated quantized function of integral s_Q^{*} (x) which is algorithm of machine from equations (3-218), (3-219) and (3-220), can be expressed as:

$$s_Q^{*}\,(x) = \sum_{i=1}^{k}\left(y_{iQ}\circ\delta_{iQ}x+\frac{1}{2}\delta_{iQ}y\circ\delta_{iQ}x+\frac{1}{12}\,\delta_{iQ}^{II}y\circ\delta_{iQ}x\right) \tag{3-221}$$

$$= \sum_{i=1}^{k}\left[y_{iQ}\cdot\delta_{iQ}x+\frac{1}{2}\delta_{iQ}y\circ\delta_{iQ}x+\frac{1}{12}\,\delta_{iQ}x\,(\delta_{iQ}y-\right.$$

$$\left.-\,\delta_{(i-1)Q}y)\right] \tag{3-222}$$

$$\begin{cases}\delta_i x = 2^r\cdot\Delta t\\ \delta_i y = 2^r\cdot\Delta y\end{cases}$$

and the error of quantization ε_{tQ} is equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k}\left(\delta_{iQ}x\cdot\varepsilon_{iQy}+\frac{1}{2}\delta_{iQ}x\circ\delta\varepsilon_{iQy}+\right.$$

$$\left.+\frac{1}{12}\,\delta_{iQ}x\cdot\delta^{II}\varepsilon_{iQy}\right) \tag{3-223}$$

in equation (3-223), the second and third terms can be neglected with respect to the first one, therefore, the equation (3-223) can be written as:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \delta_{iQ}x \cdot \varepsilon_{iQy} \tag{3-224}$$

As it was discussed earlier, $|\varepsilon_{iQy}| < \Delta y$, and $\delta_{iQ}x = 2^r \cdot \Delta_i x$, the equation (3-224) can be expressed as following:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} 2^r \cdot \Delta_i x \cdot \Delta y \tag{3-225}$$

$$h > r > 0$$

as the maximum value of r equal to h i, e $r_{max} = h$, so the equation (3-225) can be written as:

$$\varepsilon_{tQ} < 2^h \cdot \Delta Y \sum_{i=1}^{k} \Delta_i x \tag{3-226}$$

or

$$\varepsilon_{tQ} < 2^h \cdot \Delta y \ (x_k - x_o) \tag{3-227}$$

The equation (3-227) gives the quantization error ε_{tQ} in the process of integration. As it is seen, the quantization error ε_{tQ} depends to the quantum Δy, the number of increment bits h, and the duration of integral $(x_k - x_o)$.

3.5. The quantization error in multiple increment computation, when the independent variable of integral X is a function of the independent variable t.

As we have seen in chapter 2, the continuous functions X (t), and Y (t) are replaced with the approximated interpolated functions f_{ix} and f_{iy}. Then the integral formula s (t):

$$s\ (t) = \int_{t_o}^{t} Y\ (t) \cdot d\frac{X\ (t)}{dt}\, dt \tag{3-239}$$

is replaced by its approximated interpolated value of integral $s^{*}(t)$ with the error of method $\Gamma(t)$ as following:

$$\begin{cases} s^{*}(t) = \sum\limits_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iy} \cdot d\frac{f_{ix}}{dt}\, dt \\ s(t) = s^{*}(t) + \Gamma(t) \end{cases} \tag{3-240}$$

But in digital machine, the approximated interpolated function f_{iy} (t) and f_{ix} (t) are quantized with the variable of machine t.

As it was mentioned earlier, because of the time of mathematical operation in digital machine, there is an inherent delay in quantized functions f_{iyQ} (t) and f_{iQx} (t), with respect to the continuous functions f_{iy} (t) and f_{ix} (t), which cause the error of quantization. This effect is shown in figures (3-22), (3-23) and (3-24).

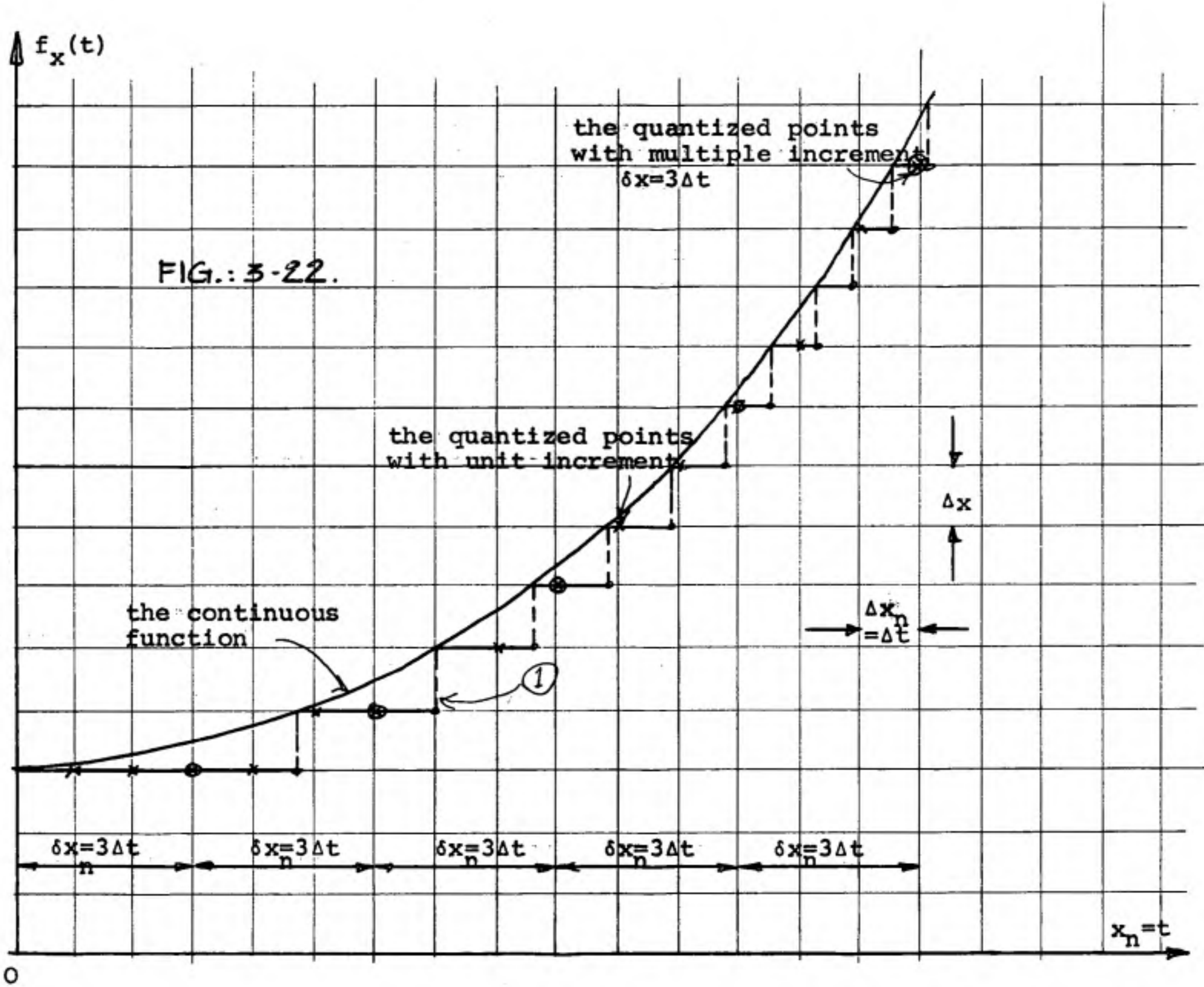
The quantization errors ε_{iQx} and ε_{iQy} which are the differences between the quantized and unquantized functions in point (x_i, y_i), are defined as following:

$$\begin{cases} \varepsilon_{iQx} = f_{ix}(t) - f_{ixQ}(t) \\ \varepsilon_{iQy} = f_{iy}(t) - f_{iyQ}(t) \end{cases} \tag{3-241}$$

As it is shown in figure (3-22), the quantized point is the intersection between the quantized function with respect to Δy (curve 1) and the line $\delta x = 3\ \delta t$, (they are shown in figures (2) and (3) by the signe $\otimes$).

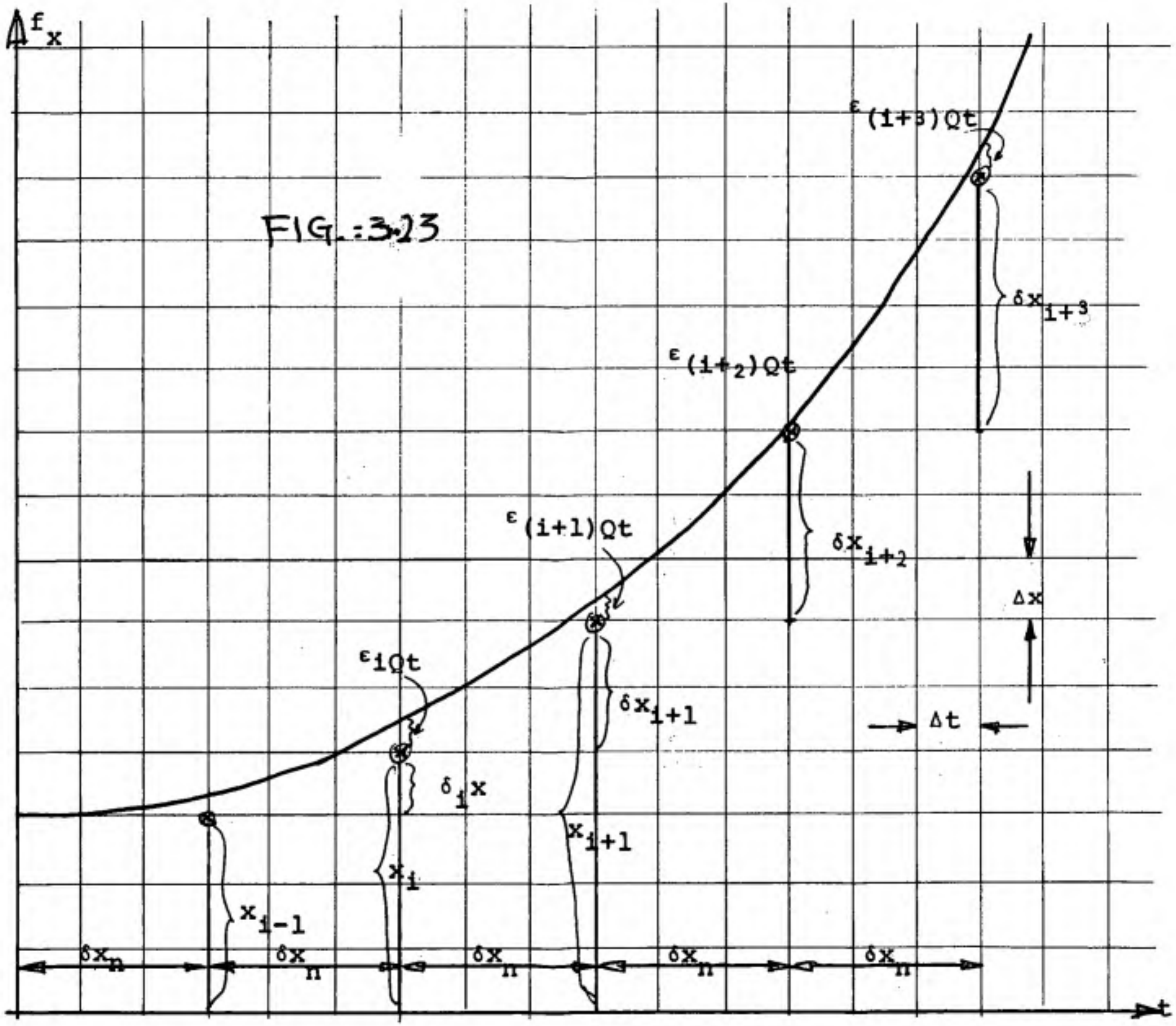
The quantized points of f_{iy} (t) $\circ$ f_{ix} (t) $= e^{-t}\sin \omega t$, are determined in figure (3-24). As it is seen, the quantized points are found by the intersection of curve (1) and (2) which are quantized function with respect to the quantums Δx and Δy. But the only intersection points are the real quantized points of the system which have the distance of $\delta_i x = 2^r \cdot \Delta x$, in our case it is supposed that, $\delta_i x = \Delta x$, $\delta_{i+1} x = 2\Delta x$, $\delta_{i+2} x = 3\Delta x$, $\delta_{i+3} x = 4\Delta x$, $\delta_{i+4} x = 5\Delta x$.

The unquantized interpolated approximated formula of integration in interval $t \in (t_i, t_{i+1})$, as it was mentionned before, is:



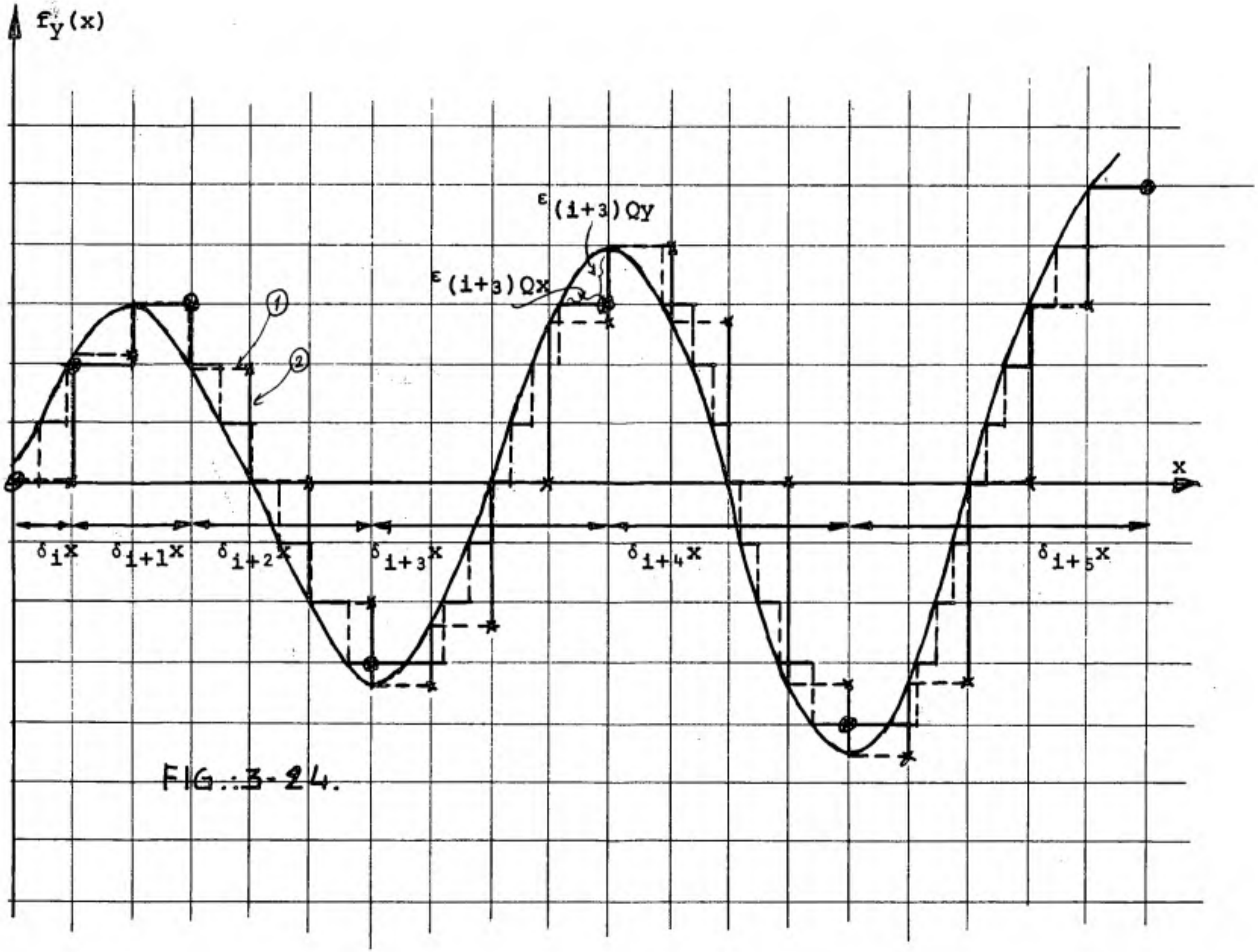
$f_x(t)$
the quantized points with multiple increment $\delta x=3\Delta t$
the quantized points with unit increment
Δx
$\Delta x_n = \Delta t$
the continuous function
1
$\delta x_n=3\Delta t$
$\delta x_n=3\Delta t$
$\delta x_n=3\Delta t$
$\delta x_n=3\Delta t$
$\delta x_n=3\Delta t$
$x_n=t$
0

FIG.: 3-22.



f_x
t
$\varepsilon_{(i+3)Qt}$
δx_{i+3}
$\varepsilon_{(i+2)Qt}$
δx_{i+2}
$\varepsilon_{(i+1)Qt}$
δx_{i+1}
ε_{iQt}
$\delta_i x$
x_{i+1}
x_i
x_{i-1}
Δx
Δt
δx_n
δx_n
δx_n
δx_n
δx_n

FIG.:3.23



$f_y(x)$
x
$\varepsilon_{(i+3)Qy}$
$\varepsilon_{(i+3)Qx}$
1
2
$\delta_i x$
$\delta_{i+1} x$
$\delta_{i+2} x$
$\delta_{i+3} x$
$\delta_{i+4} x$
$\delta_{i+5} x$

FIG.: 3-24.

$$\delta_i \ s^{*}(t) = \int_{t_i}^{t_{i+1}} f_{iy}\ (t) \circ d\frac{f_{ix}(t)}{dt} dt \tag{3-242}$$

if we put the value of the functions f_{ix} (t) and f_{iy} (t) in interval $t \in (t_i,\ t_{i+1})$ from equation (3-241) in equation (3-242), then it can be expressed as:

$$\delta_i \ s^{*}(t) = \int_{t_i}^{t_{i+1}} (f_{iQy} + \varepsilon_{iQy}) \circ d\frac{f_{iQx} + \varepsilon_{iQx}}{dt} dt \tag{3-243}$$

$$= \int_{t_i}^{t_{i+1}} f_{iQy}\ (t) \circ d\frac{f_{iQx}(t)}{dt} dt +$$

$$+ \left[\int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d\frac{f_{iQx}}{dt} dt + \right. \tag{3-244}$$

$$\left. + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \cdot d\frac{\varepsilon_{iQx}}{dt} dt + \int_{t_i}^{t_{i+1}} f_{iQy} \cdot d\frac{\varepsilon_{iQx}}{dt} dt \right]$$

The first term of equation (3-244) is the interpolated quantized formula of integration which is the algorithm of machine in interval $t \in (t_i,\ t_{i+1})$ as:

$$\delta_{iQ}\, s^{*}(t) = \int_{t_i}^{t_{i+1}} f_{iQy}\,(t) \circ d\frac{f_{iQx}(t)}{dt}\; dt \tag{3-245}$$

The others terms of equation (3-244) are equal to the quantization error ε_{itQ} of integral in the interval $t \in (t_i, t_{i+1})$ equal to:

$$\varepsilon_{iQt} = \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \circ d\frac{f_{iQx}(t)}{dt}\, dt + \int_{t_i}^{t_{i+1}} f_{iQy} \circ d\frac{\varepsilon_{iQx}}{dt}\, dt + \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \circ d\frac{\varepsilon_{iQx}}{dt}\, dt \tag{3-246}$$

so the equations (3-243) and (3-244) can be written as:

$$\delta_i\, s^{*}(t) = \delta_i\, s^{*}_{Q}\,(t) + \varepsilon_{iQt} \qquad t \in (t_i, t_{i+1}) \tag{3-247}$$

By summing the equation (3-247) in k interval, we will have the approximated interpolated integral formula in interval $t \in (t_o, t_k)$ as:

$$s^{*}(t) = \sum_{i=1}^{k} \delta_i\, s^{*}(t) \tag{3-248}$$

$$= \sum_{i=1}^{k} \delta_i\, s^{*}_{Q}\,(t) + \sum_{i=1}^{k} \varepsilon_{iQt} \tag{3-249}$$

$$= s_Q^{*}(t) + \varepsilon_{tQ} \tag{3-250}$$

where $s_Q^{*}(t)$:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \delta_i \, s_Q^{*}(t) \tag{3-251}$$

$$= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy}(t) \circ d\frac{f_{iQx}(t)}{dt} \, dt \tag{3-252}$$

is the approximated interpolated and quantized value of integral which is the algorithm of machine, and the total quantization error ε_{tQ} in interval $t \in (t_o, t_k)$ is expressed as following:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \varepsilon_{iQt} \tag{3-253}$$

$$\begin{aligned} &= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy}(t) \circ d\frac{\varepsilon_{iQt}(t)}{dt} \, dt \; + \\ &+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy} \circ d\frac{f_{iQx}(t)}{dt} \, dt \; + \\ &+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{iQy}(t) \circ d\frac{\varepsilon_{iQx}(t)}{dt} \, dt \end{aligned} \tag{3-254}$$

The equations (3-253) and (3-254) give the total quantization error ε_{tQ}, which is the difference between the unquantized and quantized function of integral in the process of digital integration.

The delay which exist in the process of quantization operation cause the phase shift between the continuous approximated function, and the quantized approximated function. It also cause the error of quantization ε_{tQ} which depends on the quantums Δx, Δy, and on the number of bit h of multiple increment.

From the above discussion, the block diagram of the incremental computer can be represented as in figure (3.18). So, we can see from the figure (3.18), that the functions X (t) and Y (t) are first interpolated approximated to the functions f_{ix} and f_{iy} in interval $x \in (x_i, x_{i+1})$, with the errors of ε_{ix} and ε_{iy}, which cause the total error of method $\Gamma(t)$. Then the approximated interpolated functions are quantized by the variable of machine (t) and cause the quantization errors ε_{iQx}, ε_{iQy}, in each point, which cause the total error of quantization ε_{tQ} and also introduce the delay $e^{-p\tau}$ with $|\tau| < \Delta x$.

3.5.1. Quantization error in the rectangular method of integration.

The interpolation functions f_{ix} and f_{iy}, in the rectangular method of integration, are:

$$f_{ix} = x_i \qquad\qquad f_{iy} = y_i$$

We have discussed earlier, that each point (x_i, y_i) is quantized by the variable t of machine to the quantized point (x_{iQ}, y_{iQ}) with the error of quantization in that point ε_{iQx}, and ε_{iQy} as:

$$\begin{cases} \varepsilon_{iQx} = x_i - x_{iQ} \\ \varepsilon_{iQy} = y_i - y_{iQ} \end{cases} \tag{3-259}$$

so if we put the values (x_i, y_i) of equation (3-259) in equation (3-108), we will have:

$$\delta_i \, s^{*}(t) = \int_0^{-1} - (y_{iQ} + \varepsilon_{iQy}) \cdot \delta \, (x_{iQ} + \varepsilon_{iQx}) \, d\xi \tag{3-260}$$

$$= - y_{iQ} \cdot \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \delta_{iQ}x + \\ + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx} \tag{3-261}$$

$$= y_{iQ} \cdot \delta_{iQ}x + (\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \\ + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx}) \tag{3-262}$$

$$= \delta_i s_Q^{*} \, (t) + \varepsilon_{itQ} \qquad t \in (t_i, t_{i+1}) \tag{3-263}$$

The approximated interpolated integral formula $s^{*}(t)$ for k interval can be find from equations (3-260), (3-261), (3-262) and (3-263) as:

$$\delta_i \, s^*(t) = \sum_{i=1}^{k} \delta_i \, s^*(t) \tag{3-264}$$

$$= \sum_{i=1}^{k} \delta_i \, s_Q^* \,(t) + \sum_{i=1}^{k} \varepsilon_{itQ} \tag{3-265}$$

$$= \sum_{i=1}^{k} y_{iQ} \circ \delta_{iQ}x + \sum_{i=1}^{k} (\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx}) \tag{3-266}$$

in the equation (3-266), the first term is the approximated interpolated quantized function of integration s_Q^* (t) which is the algorithm of machine, equal to:

$$s_Q^* \,(t) = \sum_{i=1}^{k} y_{iQ} \circ \delta_{iQ}x \qquad t \in (t_o, t_k) \tag{3-267}$$

and the other terms of equation (3-266) are the total error of quantization ε_{tQ} equal to:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx}) \tag{3-268}$$

in equation (3-268), the third term is small with respect to the two first one, therefore it can be neglected, so the total quantization error ε_{tQ} will be:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} (\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx}) \tag{3-269}$$

As we have seen already:

$$\begin{cases} |\varepsilon_{iQy}| < \Delta y \\ |\varepsilon_{iQx}| < \Delta x \\ \delta_{iQ}x = 2^r \cdot \Delta x \end{cases} \tag{3-270}$$

$$\sum_{i=1}^{k} y_{iQ} \cdot \delta\varepsilon_{iQx} < \Delta x \ (y_{kQ} - y_{oQ}) \tag{3-271}$$

then, from equations (3-269), (3-270) and (3-271) we will have:

$$\varepsilon_{tQ} < \sum_{i=1}^{k} 2^r \cdot \varepsilon_{iQy} \cdot \Delta_i x + \Delta x \ (y_{kQ} - y_{oQ}) \tag{3-272}$$

as

$$\sum_{i=1}^{k} 2^r \cdot \varepsilon_{iQy} \cdot \Delta_i x < 2^h \sum_{i=1}^{k} \varepsilon_{iQy} \cdot \Delta_i x \tag{3-273}$$

$$h > r > 0$$

and $|\varepsilon_{iQy}| < \Delta y$

so

$$\sum_{i=1}^{k} 2^r \cdot \varepsilon_{iQy} \cdot \Delta_i x < 2^h \cdot \Delta y \cdot \sum_{i=1}^{k} \Delta_i x \tag{3-274}$$

or

$$\sum_{i=1}^{k} 2^r \cdot \varepsilon_{iQy} \cdot \Delta_i x < 2^h \cdot \Delta_Q y \ (x_{kQ} - x_{oQ}) \tag{3-275}$$

From equations (3-275) and (3-2?2 , the quantization error ε_{tQ} will be:

$$\varepsilon_{tQ} < 2^h \circ \Delta_Q y \, (x_{kQ} - x_{oQ}) + \Delta_Q x \, (y_{kQ} - y_{oQ}) \qquad (3\text{-}276)$$

The equation (3-276) gives the quantization error ε_{tQ} of integration with multiple increment in the rectangular method of integration. As it is seen, the ε_{tQ} depends to the quantums Δx, Δy, to the duration of integral $(x_k - x_o)$ and $(y_k - y_o)$, and also to the number of bits h for the multiple increments.

3.5.2. Quantization error in the trapezoidal method of integration.

In the trapezoidal method of integration, the approximated interpolated functions f_{ix} and f_{iy}, which are replaced to the functions X (t) and Y (t) are:

$$\begin{cases} f_{iy} = y_i - \xi \, \delta_i y \\ f_{ix} = x_i - \xi \, \delta_i x \end{cases} \qquad (3\text{-}277)$$

and the approximated interpolated integral function $\delta_i s^*$ for interval $t \in (t_i, t_{i+1})$ will be:

$$\delta_i s^* = \int_0^{-1} (y_i - \xi \, \delta_i y) \circ \delta_i x \circ d\xi \qquad \xi \in (0, -1) \qquad (3\text{-}278)$$

by putting the value of (x_i, y_i) from equation (3-259) in equation

(3-278), we will have:

$$\delta_i s^* = \int_0^{-1} [(y_{iQ} + \varepsilon_{iQy}) - \xi\, \delta(y_{iQ} + \varepsilon_{iQy})] \cdot \delta(x_{iQ} + \varepsilon_{iQy}) d\xi \tag{3-279}$$

$$= (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \cdot \delta_{iQ}x) + [y_{iQ} \cdot \delta\varepsilon_{iQx} + \tag{3-280}$$

$$+ \varepsilon_{iQy} \cdot \delta_{iQ}x + \varepsilon_{iQy} \cdot \delta\varepsilon_{iQx} + \frac{1}{2}\delta_{iQ}y \cdot \delta\varepsilon_{iQx} +$$

$$+ \frac{1}{2}\delta\varepsilon_{iQy} \cdot \delta_{iQ}x + \frac{1}{2}\delta\varepsilon_{iQy} \cdot \delta\varepsilon_{iQx}] \qquad t \in (t_i, t_{i+1})$$

The approximated interpolated formula of integration $s^*(x)$ in k interval will be:

$$s^*(x) = \sum_{i=1}^{k} \delta_i s^* \tag{3-281}$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \cdot \delta_{iQ}x) + \tag{3-282}$$

$$+ \sum_{i=1}^{k} [y_{iQ} \cdot \delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \delta_{iQ}x + \varepsilon_{iQy} \cdot \delta\varepsilon_{iQx} +$$

$$+ \frac{1}{2}\delta_{iQ}y \cdot \delta\varepsilon_{iQx} + \frac{1}{2}\delta\varepsilon_{iQy} \cdot \delta_{iQ}x + \frac{1}{2}\delta\varepsilon_{iQy} \cdot \delta\varepsilon_{iQx}]$$

The equations (3-281) and (3-282) can be written as:

$$s^{*}(x) = s^{*}_{Q}(x) + \varepsilon_{tQ} \tag{3-283}$$

where $s^{*}(x)$ is the approximated interpolated formula of integration, $s^{*}_{Q}(x)$ is the approximated interpolated quantized formula of integration which is the algorithm of machine, equal to:

$$s^{*}_{Q}(x) = \sum_{i=1}^{k} (y_{iQ} \circ \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \circ \delta_{iQ}x) \tag{3-284}$$

The quantization error ε_{tQ} in the process of integration can be find from equations (3-282) and (3-283) as:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \left[y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta_{iQ}x + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx} + \frac{1}{2}\delta_{iQ}y \circ \delta\varepsilon_{iQx} + \frac{1}{2}\delta\varepsilon_{iQy} \circ \delta_{iQ}x + \frac{1}{2}\delta\varepsilon_{iQy} \circ \delta\varepsilon_{iQx} \right] \tag{3-285}$$

In equation (3-285) the third, fourth and other terms, are very small with respect to the first two terms, therefore, they can be neglected. So the ε_{tQ} can be written as:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} [y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta_{iQ}x] \tag{3-286}$$

as it is seen from equation (3-286), the ε_{tQ} in trapezoidal method of integration is practically equal to the ε_{tQ} of rectangular method of integration (equation 3-271). So with the same reason which was discussed in rectangular method, the equation (3-286) can be transformed to the following equation.

$$\varepsilon_{tQ} < 2^h \cdot \Delta y\ (x_{kQ} - x_{oQ}) + \Delta_Q x\ (y_{kQ} - y_{oQ}) \qquad (3\text{-}287)$$

In order to reduce the quantization error ε_{tQ}, we should decrease the value of quantums Δx, Δy, and the number of increment bits h.

3.5.3. Quantization error in the three points method of integration.

The approximated interpolated functions f_{ix} and f_{iy} which are replaced to the functions X (t) and Y (t) are:

$$\begin{cases} f_{ix} = x_i - \xi\ \delta_i x - \dfrac{\xi(\xi+1)}{2!}\ \delta_i^2 x \\ \\ f_{iy} = y_i - \xi\ \delta_i y - \dfrac{\xi(\xi+1)}{2!}\ \delta_i^2 y \end{cases} \qquad (3\text{-}288)$$

Therefore the apprixomated interpolated integral function $\delta_i\ s^*(x)$ for interval $t \in (t_i, t_{i+1})$ will be:

$$\delta_i\ s^* = \int_0^{-1} f_{iy} \cdot d\frac{f_{ix}}{d\xi}\ d\ \xi \qquad (3\text{-}289)$$

$$\delta_i\ s^{*} = \int_{0}^{-1} \left[y_i - \xi\ \delta_i y - \frac{\xi(\xi+1)}{2!}\ \delta_i^{(2)} y \right] \left[\delta_i x - \right.$$
$$\left. - \frac{2\xi+1}{2!}\ \delta_i^{(2)} x \right] d\xi \qquad (3\text{-}289)^{*}$$

The equation $(3\text{-}289)^{*}$ gives the value of the approximated, interpolated formula of integral in interval $t \in (t_i,\ t_{i+1})$.

In order to find the approximated interpolated quantized value of integral $\delta_i\ s_Q^{*}$, we should replace the value x_i, y_i with its quantized value x_{iQ}, y_{iQ} from equation (3-259) in equation $(3\text{-}289)^{*}$. So the equation $(3\text{-}289)^{*}$ can be written as:

$$\delta_i\ s = \int_{0}^{-1} \left[(y_{iQ} + \varepsilon_{iQy}) + \xi\delta_i(y_{iQ} + \varepsilon_{iQy}) + \right. \qquad (3\text{-}290)$$
$$\left. + \frac{\xi(\xi+1)}{2!}\ \delta_i^{2}(y_{iQ} + \varepsilon_{iQy}) \right] \left[\delta_i\ (x_{iQ} + \varepsilon_{iQx}) + \right.$$
$$\left. + \frac{2\xi+1}{2!} \delta_i^{(2)}\ (x_{iQ} + \varepsilon_{iQx}) \right] d\xi$$

$$= \left[y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \cdot \delta_{iQ}x + \frac{1}{12}\ (\delta_{iQ}y \cdot \delta_{i-1}x - \right. \qquad (3\text{-}291)$$
$$\left. - \delta_{iQ}x \circ \delta_{i-1}y) \right] +$$

$$+ \left[\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx} + \right.$$

$$\left. + \frac{1}{2}\delta_{iQ}y \circ \delta\varepsilon_{iQx} + \frac{1}{2}\delta\varepsilon_{iQy} \circ \delta_{iQ}x + \ldots\ldots \right]$$

The approximated interpolated formula of integral $s^*(x)$ for k interval will be:

$$s^*(x) = \sum_{i=1}^{k} \delta_i \; s^* \tag{3-292}$$

$$= \sum_{i=1}^{k} \left[y_{iQ} \circ \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \circ \delta_{iQ}x + \frac{1}{12} (\delta_{iQ}y \circ \delta_{(i-1)Q}x - \right.$$

$$\left. - \delta_{iQ}x \circ \delta_{(i-1)Q}y) \right] +$$

$$+ \sum_{i=1}^{k} \left[\varepsilon_{iQy} \circ \delta_{iQ}x + y_{iQ} \circ \delta\varepsilon_{iQx} + \varepsilon_{iQy} \circ \delta\varepsilon_{iQx} + \right.$$

$$\left. + \frac{1}{2}\delta_{iQ}y \circ \delta\varepsilon_{iQx} + \frac{1}{2}\delta\varepsilon_{iQy} \circ \delta_{iQ}x + \ldots\ldots \right] \tag{3-293}$$

$$= s_Q^* \; (x) + \varepsilon_{tQ} \tag{3-294}$$

from equations (3-292), (3-293) and (3-294), it can be seen that the approximated interpolated quantized value of integral s_Q^* (x), which is the algorithm of machine, is equal to:

$$s_Q^{*}(x) = \sum_{i=1}^{k} \left[y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}y \cdot \delta_{iQ}x + \frac{1}{12}(\delta_{iQ}y \cdot \delta_{(i-1)Q}x - \delta_{iQ}x \cdot \delta_{(i-1)Q}y) \right] \quad (3\text{-}295)$$

and from the same equation, we can find the value of the total quantization error ε_{tQ} in interval $t \in (t_o, t_k)$ as following:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} \left[\varepsilon_{iQy} \cdot \delta_{iQ}x + y_{iQ} \cdot \delta\varepsilon_{iQx} + \varepsilon_{iQy} \cdot \delta\varepsilon_{iQx} + \frac{1}{2}\delta_{iQ}y \cdot \delta\varepsilon_{iQx} + \frac{1}{2}\delta\varepsilon_{iQy} \cdot \delta_{iQ}x + \ldots\ldots \right] \quad (3\text{-}296)$$

in equation (3-296), the third, fourth and other terms are very small with respect to the first two terms, so they can be neglected. Therefore, the total quantization error ε_{tQ} in k interval will be:

$$\varepsilon_{tQ} = \sum_{i=1}^{k} [\varepsilon_{iQy} \cdot \delta_{iQ}x + y_{iQ} \cdot \delta\varepsilon_{iQx}] \quad (3\text{-}297)$$

As we have already seen in the rectangular and trapezoidal method of integration, the equation (3-297) can be transformed to the following equation:

$$\varepsilon_{tQ} < 2^{h} \cdot \Delta y\ (x_{kQ} - x_{oQ}) + \Delta x\ (y_{kQ} - y_{oQ}) \quad (3\text{-}298)$$

It can be seen from equation (3-269), (3-286) and (3-297), that the quantization error in the process of integration does not depend on the method of integration, but it depends on the quantums Δx, Δy, on the duration of integral $(x_k - x_o)$, $(y_k - y_o)$ and on the number of bits h of multiple increment.

3.6. Conclusion.

In this chapter, we have calculated, for different methods of integration, the quantization error ε_{tQ} for unitary and multiple incremental computation, when the independent variable of integral X is equal to, or is a function of the independent variable t of machine.

The quantization error ε_{tQ} is the difference between the approximated interpolated integration function $s^{*}(t)$ and the approximated interpolated quantized function $s_{Q}^{*}(t)$:

$$\varepsilon_{tQ} = s^{*}(t) - s_{Q}^{*}(t)$$

The values of quantization errors for different methods of integration, in the case of unitary and multiple increment computation, are shown in the belowing table (3.1).

TABLE 3.1

Method of integration	Quantization error ε_{tQ} in unitary incremental computation	Quantization error ε_{tQ} in multiple incremental computation
Rectangular, trapezoidal and three points method when $Y = f(X)$, $X = t$	$\varepsilon_{tQ} < \Delta y\ (x_{kQ}-x_{oQ})$	$\varepsilon_{tQ} < 2^{h} \cdot \Delta y\ (x_{kQ}-x_{oQ})$
Rectangular, trapezoidal and three points method, when $X = X(t)$, $Y = Y(t)$	$\varepsilon_{tQ} < \Delta y\ (x_{kQ}-x_{oQ}) + \Delta x\ (y_{kQ}-y_{oQ})$	$\varepsilon_{tQ} < 2^{h} \cdot \Delta y\ (x_{kQ}-x_{oQ}) + \Delta x\ (y_{kQ}-y_{oQ})$

As it is seen from this table, the quantization error is the same in the rectangular, trapezoidal, and three points method. In the case of multiple incremental computation, this error depends on the quantums Δx, Δy and on the number of bits h in δX register.

In incremental computation, by choosing the more accurate interpolation formula, we can increase the step of integration from

Δx to $\delta x = 2^h \cdot \Delta x$; this increases the speed of integration by 2^h. But as it is seen in table (3.1), we also increase the quantization error ε_{tQ} by the same factor.

In the chapter (6), we shall see what is the greatest admissible h, for different methods of integration, so that, the errors don't exceeding a certain limit.

In order to reduce the quantization error ε_{tQ}, we should decrease the values of quantums $\Delta x = \Delta y = 2^{-n}$, where n is the number of bits in the y register. By increasing n, we can decrease the quantums Δx and Δy, but it is not interesting to increase n too much, because the machine speed will decrease, and the amount of equipment will increase. So, there is a compromise between the choice of bits n in the Y register, and the quantization error ε_{tQ}. Usually, the value of n is between ten and twenty, so $\Delta x = \Delta y = (2^{-1o}$ to $2^{-2o})$.

In incremental computer of industrial electronics of the Brussel University, which is designed by the author, n can be chosen as ten or sixteen.

CHAPTER IV

THE ROUND OFF, TRANSMISSION ERROR AND NONLINEARITY IN THE INCREMENTAL COMPUTER.

4.1. The round off error in the process of integration (unitary or multiple increment) in incremental computer.

As we have seen before, the continuous functions X (t) and Y (t) are replaced by the approximated interpolated functions f_{ix}, f_{iy} which have the errors ε_{ix} and ε_{iy} between the actual functions X (t) Y (t), and the approximated interpolated functions f_{ix}, f_{iy} as:

$$\begin{cases} \varepsilon_{ix} = f_{ix}\ (t) - X\ (t) \\ \varepsilon_{iy} = f_{iy}\ (t) - Y\ (t) \end{cases} \qquad (4\text{-}1)$$

The errors ε_{ix}, ε_{iy}, cause the total error of the method of integration $\Gamma(t)$, which is the difference between the actual integration function s (t), and the approximated interpolated integration function $s^{*}(t)$ as:

$$s\ (t) = s^{*}(t) + \Gamma(t) \tag{4-2}$$

In the quantization process, each point $[(x_i,\ y_i),\ (x_{i-1},\ y_{i-1}),\ \ldots]$ is replaced by the quantized points $[(x_{iQ},\ y_{iQ}),\ (x_{(i-1)Q},\ y_{(i-1)Q}, \ldots]$ which have the error of quantization ε_{iQx}, ε_{iQy}, that is the difference between the quantized functions f_{ixQ}, f_{iyQ}, and the unquantized functions f_{ix} (t), f_{iy} (t), as following:

$$\begin{cases} \varepsilon_{iQx} = f_{ix}\ (t) - f_{ixQ}\ (t) \\ \varepsilon_{iQy} = f_{iy}\ (t) - f_{iyQ}\ (t) \end{cases} \tag{4-3}$$

The ε_{iQx}, ε_{iQy} cause the total quantization error ε_{tQ} in the process of integration, that is the diffence between the approximated interpolated function of integration $s^{*}(t)$ and the approximated interpolated quantized function of integration s^{*}_{Q} (t).

$$s^{*}(t) = s^{*}_{Q}\ (t) + \varepsilon_{tQ} \tag{4-4}$$

The s^{*}_{Q} (t) is the approximated interpolated quantized function of integral which is the algorithm of the incremental machine. Therefore the relation between the actual function of integral s (t) and the approximated interpolated quantized function of integral s^{*}_{Q} (t) is:

$$s\ (t) = s^{*}_{Q}\ (t) + [\Gamma(t) + \varepsilon_{tQ}] \tag{4-5}$$

As we have seen the block diagram of incremental computer is shown in figure (4.1).

In incremental computer, the results of a given mathematical operation is transmitted for use in another mathematical operation by the use of quantized increments.

If the number of bits of y_{eq} register is n, and the number of bits of δx register is h, then the number of bits of s_Q^* (t) register is (h+n) as:

$$s_Q^* (t) = y_{eq} \cdot \delta x \qquad (4-6)$$

By the convention the absolute value of y register is arranged by scale factor in such a way that it is always less than one, so the less significant bit of y register has the weight of 2^{-n} which is equal to the quantum Δy so $\Delta y = 2^{-n}$. The weight of S register has exactly the same weight as y register, as it is shown in figure (4.2).

Therefore, S register has one fractional part S_o with n less significant bits whose content is less than one, and the other most significant parts δs which have h bits, and the content is greater or equal to one. As it is seen from figure (4.2), the most significant bit of S register has the weight of 2^h, and the maximum value of s_{Qt}^* is:

$$[s_{Qt}^*]_{max} = 2^h \qquad (4-7)$$

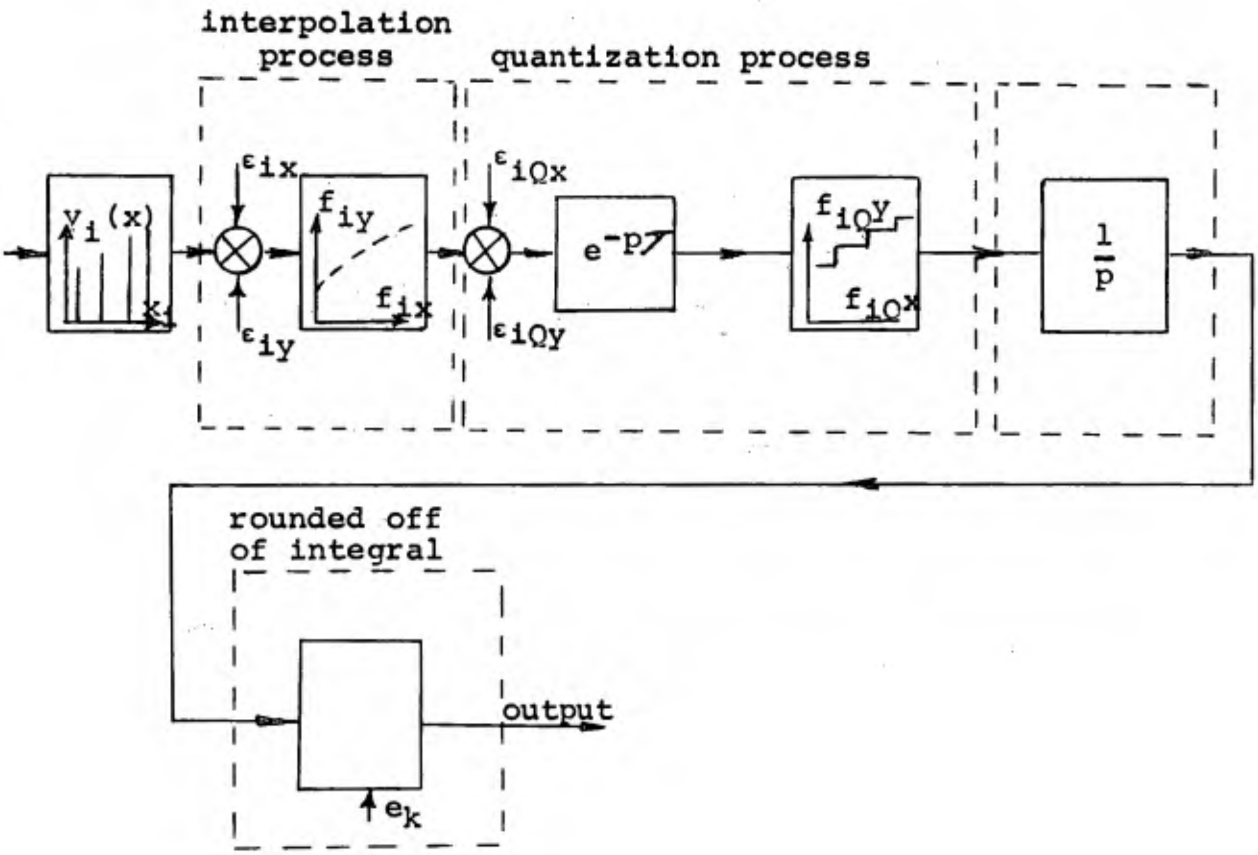


fig. 4.1.

Block diagram of incremental computer.

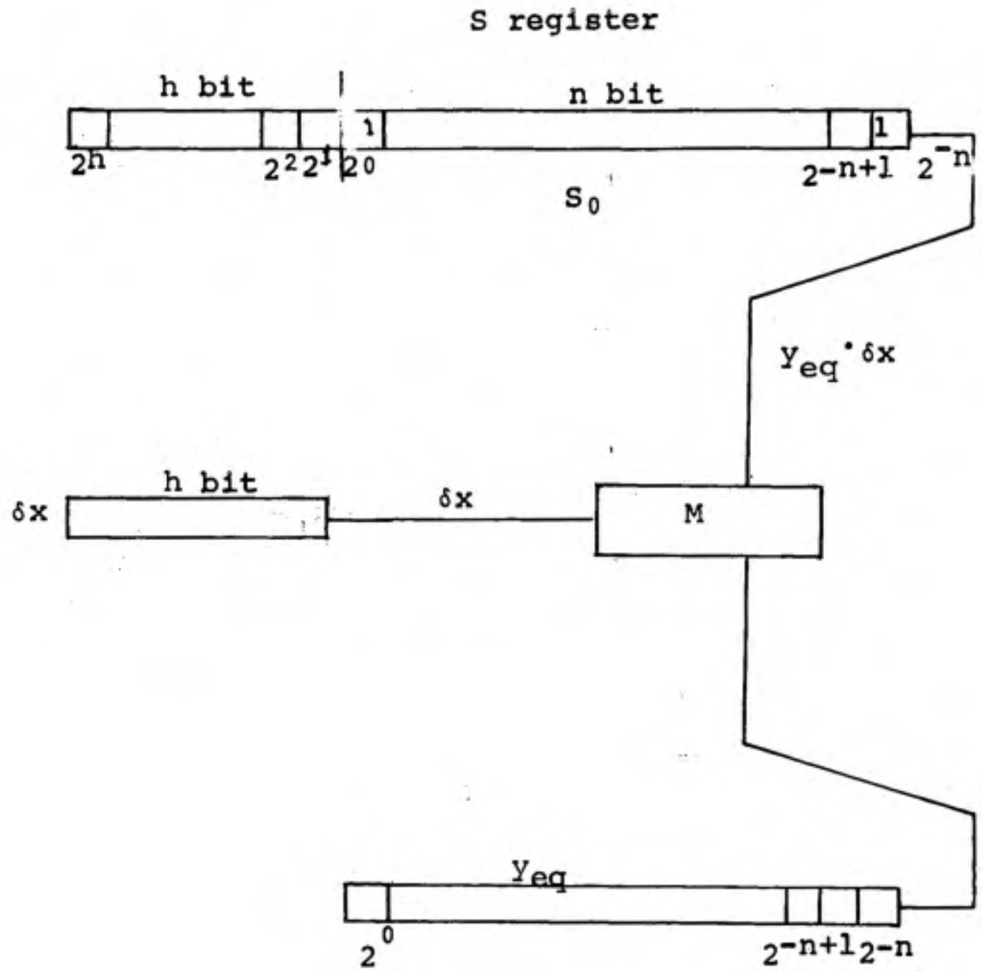

S register
h bit
n bit
1
1
2^h
2^2 2^1 2^0
2^{-n+1} 2^{-n}
S_0
$y_{eq} \cdot \delta x$
δx
h bit
δx
M
y_{eq}
2^0
2^{-n+1} 2^{-n}

fig. 4.2.

the less significant bit of S register has the weight of 2^{-n} which is the quantum of Δs, so:

$$\Delta s = 2^{-n}$$

in incremental computer, using the unitary increment $h = 1$, the maximum value of s_{Qt} is equal to one, in other words, when the content of S register becomes greater than one, there will be an overflow, equal to one in δ_{iQ} s, and the rest of integral is accumulated in S_o part of S register. In incremental computer, using multiple increments, the maximum value of s_{Qt} can be equal to $2^o = 1$ or $2^1, 2^2, \ldots 2^h$. So there will be an overflow when the value of s_{Qt} becomes greater than one, and the rest of integral will be in S_o register. Therefore, we can write in any iteration the following relation.

$$s_Q^{*}(t) = s_{QM}^{*}(t) + s_{ok} \tag{4-8}$$

where $s_Q^{*}(t)$ is the approximated interpolated quantized value of integral, for instance in the trapezoidal formula equal to:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \delta_i s_Q^{*} \tag{4-9}$$

$$= \sum_{i=1}^{k} (y_{iQ} \cdot \delta_{iQ}x + \frac{1}{2}\delta_{iQ}x \cdot \delta_{iQ}x) \tag{4-10}$$

and $s_{QM}^{*}(t)$ is the sum of increment of integral at the h most significant bit of S register, which is the output of incremental machine.

$$s^{*}_{QM}\ (t) = \sum_{i=1}^{k} \delta_i\ s^{*}_{QM} \tag{4-11}$$

In equation (4-11), $\delta_i\ s^{*}_{QM}$ is the quantized increment of integral (the h more significant bits of S register), S_o is the n less significant bit of S register or is the rest of integral s^{*}_{Q} (t), which is neglected in that iteration for output information, but is accumulated in the memory for adding to the value of integral in the next iteration.

Therefore the value of integral which is output of incremental machine s^{*}_{QM} (t) is equal to:

$$s^{*}_{QM}\ (t) = \sum_{i=1}^{k} \delta_i\ s^{*}_{QM} \tag{4-12}$$

$$= s^{*}_{Q}\ (t) - S_o \tag{4-13}$$

in equations (4-12) and (4-13), the $\delta_i\ s^{*}_{QM}$ is the increment of the approximated interpolated quantized and rounded off of integral, and S_o is the round off error of the process of integration.

The actual value of integral S (t) can be found from equations (4-5), (4-12) and (4-13) as:

$$S\ (t) = s^{*}_{QM}\ (t) + [\Gamma\ (t) + \varepsilon_{tQ}\ (t) + S_o\ (t)] \tag{4-14}$$

where Γ(t) and ε_{tQ} (t) are the errors of method and quantization, S_o(t) is the round off error and s^{*}_{QM} (t) is the approximated interpolated quantized rounded off value of the integration. From equation (4-13) we have:

$$s_Q^{*}\ (t) = s_{QM}^{*}\ (t) + S_o \tag{4-15}$$

The algorithm of machine which is equal to s_Q^{*} (t) in the rectangular, trapezoidal and three points interpolation formula of integration is:

in rectangular method:

$$s_{Qk}^{*}\ (t) = \sum_{i=1}^{k} y_i \cdot \delta_i x \tag{4-16}$$

in trapezoidal method:

$$s_{Qk}^{*}\ (t) = \sum_{i=1}^{k} (y_i \cdot \delta_i x + \frac{1}{2} \delta_i x \cdot \delta_i y) \tag{4-17}$$

in three points method:

$$s_{Qk}^{*}\ (t) = \sum_{i=1}^{k} [y_i \cdot \delta_i x + \frac{1}{2} \delta_i x \cdot \delta_i y + \frac{1}{12} (\delta_i y \cdot \delta_{i-1} x - \delta_{i-1} y \cdot \delta_i x)] \tag{4-18}$$

and the approximated interpolated quantized, round off value of integral s_{QMk}^{*} (t) which is output of incremental machine, can be found from equations (4-15), (4-16), (4-17) and (4-18) in rectangular method:

$$s^{*}_{QMk}\ (t) = \sum_{i=1}^{k} \delta_{iQM}\ (t)$$

$$= \sum_{i=1}^{k} y_i \cdot \delta_i x - S_{ok} \tag{4-19}$$

in trapezoidal method:

$$s^{*}_{QMk}\ (t) = \sum_{i=1}^{k} \delta_{iQM}\ (t)$$

$$= \sum_{i=1}^{k} (y_i \cdot \delta_i x + \frac{1}{2}\delta_i x \cdot \delta_i y) - S_{ok} \tag{4-20}$$

in three points method:

$$s^{*}_{QMk}\ (t) = \sum_{i=1}^{k} \delta_{iQM}\ (t)$$

$$= \sum_{i=1}^{k} [y_i \cdot \delta_i x + \frac{1}{2}\delta_i x \cdot \delta_i y + \frac{1}{12} (\delta_i y \cdot \delta_{i-1} x - \delta_{i-1} y \cdot \delta_i x)] - S_{ok} \tag{4-21}$$

if we find the s^{*}_{QMk} for iteration k by the rectangular method, we will have:

$$s^{*}_{QMk}\ (t) = \sum_{i=1}^{k} y_i \cdot \delta_i x - S_{ok} \tag{4-22}$$

and the $s^{*}_{QM(k-1)}$ (t) for iteration (k-1) by the same method will be:

$$s^{*}_{QM(k-1)}\ (t) = \sum_{i=1}^{k-1} y_i \cdot \delta_i x - S_{o(k-1)} \qquad (4-23)$$

Therefore, the increment of integral $\delta_i\ s^{*}_{QMk}$ (t) will be the difference of s^{*}_{QMk} and $s^{*}_{QM(k-1)}$ from equations (4-22) and (4-23), it can be expressed as:

$$\delta_i\ s^{*}_{QMk}\ (t) = s^{*}_{QMk}\ (t) - s^{*}_{QM(k-1)}\ (t) \qquad (4-24)$$

$$= y_k \cdot \delta_k x + S_{o(k-1)} - S_{ok} \qquad (4-25)$$

The equation (4-25) can be written as:

$$\delta_i\ s^{*}_{QMk}(t) + S_{ok}(t) = y_k \cdot \delta_k x + S_{o(k-1)}\ (t) \qquad (4-26)$$

The expression (4-26) gives the exact operation of integration in incremental machine. That means in each iteration, the value of $y_k \cdot \delta_k x$ is calculated and added to the rest of integral of the previous iteration $S_{o(k-1)}$, so it gives the output $\delta_i\ s_{QMk}$ (t) which is the approximated interpolated quantized rounded off of increment at the output of machine, and it also gives the new value of the rest of integral S_{ok} (t), In the n less significant bit of S register which goes to memory for memorization in order to use for the next interval.

The value of $\delta_i\ s^{*}_{QMk}$ (t) can be found easily with the same method for trapezoidal and three points formula as following:

in the trapezoidal method:

$$\delta_i \, s^{*}_{QMk}\ (t) + S_{ok}\ (t) = (y_k \cdot \delta_k x + \frac{1}{2}\delta_k x \cdot \delta_k y) + S_{o(k-1)}\ (t) \tag{4-27}$$

in three points method:

$$\delta_i \, s^{*}_{QMk}\ (t) + S_{ok}\ (t) = [y_k \cdot \delta_k x + \frac{1}{2}\delta_k x \cdot \delta_k y + \frac{1}{12}\,(\delta_k y \cdot \delta_{(k-1)} x - \delta_{(k-1)} y \cdot \delta_k x)] + S_{o(k-1)}\ (t) \tag{4-28}$$

The same conclusion of equation (4-26) can be taken for the equations (4-27) and (4-28). For instance, in the trapezoidal method of integration, the value of $(y_k \cdot \delta_k x + \frac{1}{2}\delta_k x \cdot \delta_k y)$ is added to the rest of integral from former iteration $S_{o(k-1)}$, and there will be an increment output $\delta_i \, s^{*}_{QMk}\ (t)$, and also a new value of the rest of integral $S_{ok}\ (t)$ which will go to the memory for the next iteration. The same operation is done for three points method, in this case, the value of $[y_k \cdot \delta_k x + \frac{1}{2}\delta_k x \cdot \delta_k y + \frac{1}{12}\,(\delta_k y \cdot \delta_{(k-1)} x - \delta_{(k-1)} y \cdot \delta_k x)]$ is calculated and added to the rest of integral of the preceeding iteration $S_{o(k-1)}$. The result will be the output $\delta_i \, s^{*}_{QMk}\ (t)$ and the new value of the rest of integral S_{ok} which will go to the memory for using the next iteration.

In general the round off error e_k in each iteration is a function of S_{oi}; $e_k = f\ [S_{ok}, S_{o(k-1)}, \dots]$ and in our case, the round off error e_k in each iteration is $e_k = S_{o(k-1)} - S_{ok}$.

4.1.1. Upper bound of round off error of integration in unitary incremental computer.

As we have seen in the former paragraph, the relation between the output increment of machine $\delta_i\, s^{*}_{QMk}\,(t)$ and the algorithm of machine $y_{eqk} \cdot \delta x$ is as following:

$$S_{ok} + \delta_i\, s^{*}_{QMk}\,(t) = y_{eq.k} \cdot \delta_k x + S_{o(k-1)} \tag{4-29}$$

or

$$\delta_i\, s^{*}_{QMk}\,(t) = y_{eq.k} \cdot \delta_k x + (S_{o(k-1)} - S_{ok}) \tag{4-30}$$

where $y_{eq} = y_i$ in the rectangular method, and $y_{eq} = y_i + \frac{1}{2}\delta_i y$ in the trapezoidal method.

In incremental computer, with unitary increment; the increment of integral, the dependent variable and the independent variable are equal to the quantums $\Delta s, \Delta y$ and Δx. So the equation (4-29) can be written as:

$$\Delta\, s^{*}_{QMk}\,(t) = y_{eq.k} \cdot \Delta_k x + (S_{o(k-1)} - S_{ok}) \tag{4-31}$$

if we consider the value of $\Delta s^{*}_{QMk}\,(t)$ and $\Delta_k x$, equal to the logical $\pm$ 1 or O, then we should introduce the factor 2^{+n} in the value of $\Delta\, s^{*}_{QMk}\,(t)$. In other words, the significant of $\Delta\, s^{*}_{QM}\,(t)$ is 2^n time greater than the logical $\pm$ 1. So the equation (4-30) in the coded form can be expressed as:

$$2^n \cdot \Delta\, s^{*}_{QMk}\,(t) = y_{eq.k} \cdot \Delta_k x + (S_{o(k-1)} - S_{ok}) \tag{4-32}$$

in equation (4-32), if we neglect the round off error $(S_{o(k-1)} - S_{ok})$, we will have the familiar equation of incremental machine as:

$$\Delta s^{*}_{QMk}(t) = \frac{1}{2^n} \cdot y_{eq(k)} \cdot \Delta x \tag{4-33}$$

if we use the equation (4-32), for first, second .... and k iterations we will have the following equation:

$$2^n \Delta s^{*}_{QM(1)}(t) = y_{eq(1)} \cdot \Delta x + (S_{c(o)} - S_{c(1)}) \; 1^{st} \text{ iteration} \tag{4-34}$$

$$2^n \Delta s^{*}_{QM(2)}(t) = y_{eQ(2)} \cdot \Delta x + (S_{o(1)} - S_{o(2)}) \; 2^{nd} \text{ iteration} \tag{4-35}$$

by putting $S_{o(1)}$ from equation (4-34)in equation (4-35), we will have:

$$2^n \Delta s^{*}_{QM(1)}(t) + 2^n \Delta s^{*}_{QM(2)}(t) = y_{eq(1)} \cdot \Delta x + y_{eq(2)} \cdot \Delta x + + S_{o(o)} - S_{o(2)} \tag{4-36}$$

if we find the equation (4-36) for k iteration, we will have:

$$\sum_{i=1}^{k} 2^n \Delta s^{*}_{QM(i)}(t) = \sum_{i=1}^{k} \cdot y_{eq(i)} \cdot \Delta x + S_{o(o)} - S_{o(k)} \tag{4-37}$$

The equation (4-37) can be written as following:

$$\sum_{i=1}^{k} \Delta s^{*}_{QM(k)}(t) = \sum_{i=1}^{k} \frac{1}{2^n} y_{eq(k)} \cdot \Delta x + \frac{1}{2^n} (S_{o(o)} - S_{o(k)}) \tag{4-38}$$

The value of $\frac{1}{2n}(S_{o(o)} - S_{o(k)})$ is the round off error e_k of the process of integration. By neglecting e_k, we will have the normal equation of incremental computer with unitary increment as following:

$$\sum_{i=1}^{k} \Delta\, s^{*}_{QM(k)}\ (t) = \sum_{i=1}^{k} \frac{1}{2^n}\, Y_{eq(k)} \cdot \Delta x \tag{4-39}$$

so the round off error e_k is equal to:

$$e_k = \frac{1}{2^n}(S_{o(o)} - S_{o(k)}) \tag{4-40}$$

as the number of bits of S register is one bit greater than Y register therefore, the number of bits of S register is (n+1). So the logical weight of S register for (n+1) bit is equal to $2^{n+1} - 1$ which we call N so:

$$N = 2^{n+1} - 1 \tag{4-41}$$

$$\simeq 2^{n+1} \tag{4-42}$$

by putting the value of N from equation (4-42) in equation (4-40), we will have:

$$e_k = \frac{2}{N}\ (S_{o(o)} - S_{o(k)}) \tag{4-43}$$

in order, to determine the upper bound of round off error, we consider its absolute value $|e_k|$ so:

$$\left|e_k\right| = \frac{2}{N}\ \left|S_{o(o)} - S_{o(k)}\right| \tag{4-44}$$

since $$\left|S_{o(o)} - S_{o(k)}\right| < S_{o(max)} = N \tag{4-45}$$

so $$\left|e_k\right| < 2 \tag{4-46}$$

if we put the initial condition of S register to $\frac{N}{2}$ (the most significant bit),

$$S_{o(o)} = \frac{N}{2} \tag{4-47}$$

then, by putting the value of equation (4-47) in equation (4-44), we will have:

$$\left|e_k\right| = \frac{2}{N}\left|\frac{N}{2} - S_{o(k)}\right| \tag{4-48}$$

since $$N > \left|S_{o(k)}\right| > 0 \tag{4-49}$$

then $$\left|\frac{N}{2} - S_{o(k)}\right| < \left|\frac{N}{2}\right| \tag{4-50}$$

by putting the value of (4-50) in equation (4-48), we will have:

$$\left|e_k\right| < 1 \tag{4-51}$$

Therefore, by choosing the appropriated initial condition $S_{o(o)} = \frac{N}{2}$, we will have the round off error e_k which is smaller than one, i, e, or smaller than the less significant bit of S register.

On the other hand, the less significant bit of S register has the weight of 2^{-n} which is equal to the quantums Δx or Δs so:

$$|e_k| < \Delta s \text{ or } |e_k| < \Delta x.$$

4.1.2. Upper bound of round off error of integration with multiple incremental computer.

In multiple incremental computer, the increments Δx, Δy, and Δs are:

$$\begin{cases} \delta x = 2^r \cdot \Delta x \\ \delta y = 2^r \cdot \Delta y \qquad h > r > 0 \\ \delta s = 2^r \cdot \Delta s \end{cases} \tag{4-52}$$

We can use the general equation (4-29) for the quantized increment $\delta_i\, s^{*}_{QM(k)}$ (t) as :

$$\delta_i s^{*}_{QM(k)} (t) = y_{eq(k)} \cdot \delta_k x + (S_{o(k-1)} - S_{o(k)}) \tag{4-53}$$

The coded equation (4-53) can be find easily with the same reason that the equation (4-32) in the form:

$$2^{n-h} \cdot \delta_i s^{*}_{QM(k)} (t) = y_{eq(k)} \cdot \delta x + (S_{o(k-1)} - S_{o(k)}) \tag{4-54}$$

For k interval we can find the following equation:

$$\sum_{i=1}^{k} 2^{n-h} \cdot \delta_i \, s^{*}_{QM(k)} \, (t) = \sum_{i=1}^{k} Y_{eq(k)} \cdot \delta x + (S_{o(o)} - S_{o(k)} \qquad (4\text{-}55)$$

By dividing the equation (4-55) to 2^{n-h} we will have:

$$\sum_{i=1}^{k} \delta_i \, s^{*}_{QM(k)} \, (t) = \frac{2^h}{2^n} \sum_{i=1}^{k} Y_{eq(k)} \cdot \delta x + \frac{2^h}{2^n} (S_{o(o)} - S_{o(k)}) \qquad (4\text{-}56)$$

the second term of equation (4-56) is the round off error e_k of the process of integration with multiple increment which is equal to:

$$e_k = \frac{2^h}{2^n} (S_{o(o)} - S_{o(k)}) \qquad (4\text{-}57)$$

By neglecting the round off error e_k in equation (4-56), we will have the operation equation of incremental computer with multiple increment as following:

$$\sum_{i=1}^{k} \delta_i \, s^{*}_{QM(k)} \, (t) = \frac{2^h}{2^n} \sum_{i=1}^{k} Y_{eq(k)} \cdot \delta_k x \qquad (4\text{-}58)$$

or

$$\delta_i \, s^{*}_{QM(k)} \, (t) = \frac{2^h}{2^n} Y_{eq(k)} \cdot \delta_k x \qquad (4\text{-}59)$$

As we have seen in the preceding paragraph, $N = 2^{n+1}$, so we can write the equation (4-57) as following:

$$e_k = \frac{2^{h+1}}{N} (S_{o(o)} - S_{o(k)}) \qquad (4\text{-}60)$$

as $$\left| S_{o(o)} - S_{o(k)} \right| < N \tag{4-61}$$

so the round off error e_k will be:

$$e_k < 2^{h+1} \tag{4-62}$$

By putting the initial condition in S register, like $S_o = \frac{N}{2}$, the round off error e_k reduce to half of its upper bound as following:

$$e_k = \frac{2^{h+1}}{N} \left(\frac{N}{2} - S_{o(k)} \right) \tag{4-63}$$

since $$N > S_{o(k)} > o$$

so $$|e_k| < 2^h \tag{4-64}$$

The equation (4-64) gives the upper bound of round off error e_k in multiple incremental computer, by using the appropriate initial condition $S_o = \frac{N}{2}$.

As the maximum bits of increments δx and δs are h ($(\delta s)_{max} = (\delta x)_{max} = 2^h$), so the equation (4-64) can be written as:

$$|e_k| < |\delta s| \tag{4-65}$$

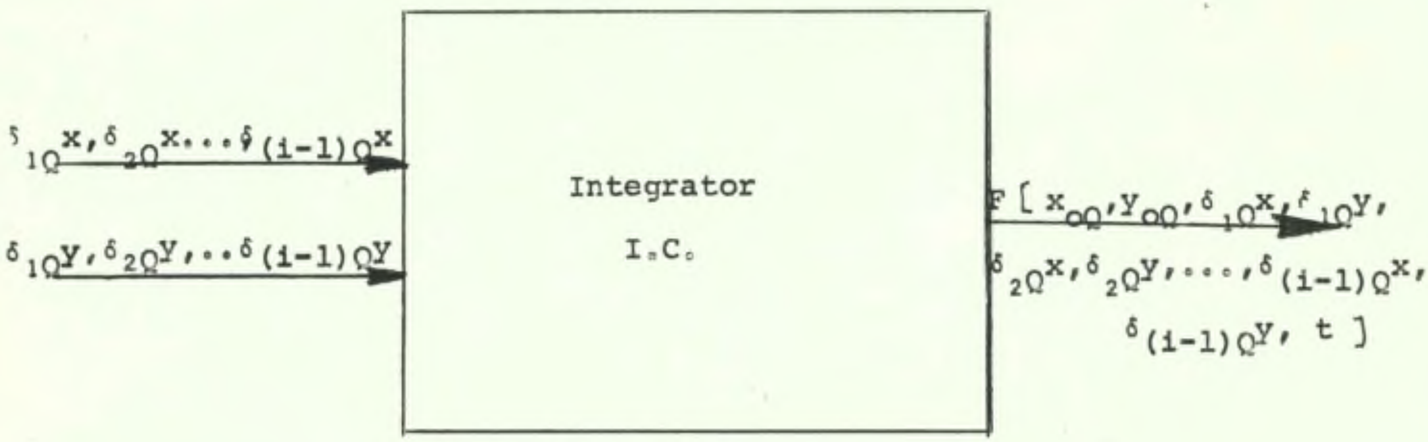
from equation (4-65), it is seen that the absolute value of round off error is smaller than the increment δs in multiple incremental computer.

4.2. The transmission errors in unitary or multiple incremental computers.

In the integral operation of incremental computers, the informations which are needed in interval $x \in (x_i, x_{i+1})$ can be expressed as following:

$$\begin{cases} f_{iQx} = f_{iQx}\,(x_{oQ},\ \delta_{1Q}x,\ \delta_{2Q}x\ldots\ldots\ \delta_{iQ}x,\ t_{iQ}) \\ f_{iQy} = f_{iQy}\,(y_{oQ},\ \delta_{1Q}y,\ \delta_{2Q}y\ldots\ldots\ \delta_{iQ}y,\ t_{iQ}) \\ \delta_i\ s_Q^{*} = \displaystyle\int_{t_i}^{t_{i+1}} f_{iQy}\,(t) \cdot d\frac{f_{iQx}\,(t)}{dt}\,dt \end{cases} \qquad (4\text{-}66)$$

As it is seen in chapter (1), because the iterative nature of incremental computer, the only informations which exist are the informations of former iterations, 1, 2, 3,.... , (i-1), which we find in the memory. Therefore, the data has a delay of one machine cycle T with respect to the quantized value of information. The delay T is produced in the input data of incremental computer, which are the output of the other integrators in the former iterations. This effect can be shown by figure (4.3). The delay cause the error of transmission ε_{Tx}, ε_{Ty} in each interval $x \in (x_i, x_{i+1})$ that is the difference between the approximated interpolated quantized functions f_{iQx}, f_{iQy}, and the approximated interpolated quantized delayed functions f_{iQDx}, f_{iQDy},



$\delta_{1\Omega}x, \delta_{2\Omega}x \ldots \delta_{(i-1)\Omega}x$
$\delta_{1\Omega}y, \delta_{2\Omega}y, \ldots \delta_{(i-1)\Omega}y$
Integrator
I.C.
$F[x_{o\Omega}, y_{o\Omega}, \delta_{1\Omega}x, \delta_{1\Omega}y,$
$\delta_{2\Omega}x, \delta_{2\Omega}y, \ldots, \delta_{(i-1)\Omega}x,$
$\delta_{(i-1)\Omega}y, t]$

fig. 4.3.

as following:

$$\begin{cases} \varepsilon_{Tx} = f_{iQx}\,[x_{oQ},\ \delta_{1Q}x,\ \delta_{2Q}x, \ldots\ \delta_{iQ}x,\ t] - \\ \qquad - f_{iQDx}\,[x_{oQ},\ \delta_{1Q}x,\ \delta_{2Q}x, \ldots,\ \delta_{(i-1)Q}x,\ t] \\ \varepsilon_{Ty} = f_{iQy}\,[y_{oQ},\ \delta_{1Q}y,\ \delta_{2Q}y, \ldots\ \delta_{iQ}y,\ t] - \\ \qquad - f_{iQDy}\,[y_{oQ},\ \delta_{1Q}y,\ \delta_{2Q}y, \ldots,\ \delta_{(i-1)Q}y,\ t] \end{cases} \qquad (4\text{-}67)$$

$$x \in (x_i,\ x_{i+1})$$

The ε_{Tx} and ε_{Ty} cause the total transmission error ε_{Tr} .

As it was discussed earlier, the approximated interpolated quantized value of integral $\delta_i\ s_Q^{*}\ (x)$ is equal to:

$$\delta_i\ s_Q^{*}\ (t) = \int_{t_i}^{t_{i+1}} f_{iQy}\ (t) \cdot d\frac{f_{iQx}\ (t)}{dt}\ dt \qquad (4\text{-}68)$$

$$t \in (t_i,\ t_{i+1})$$

by putting the value f_{iQx}, f_{iQy} from equation (4-67) in equation (4-68), we will have:

$$\delta_i\ s_Q^{*}\ (t) = \int_{t_i}^{t_{i+1}} [f_{iQDy}\ (t) + \varepsilon_{Ty}] \cdot d\frac{f_{iQD}\ (t) + \varepsilon_{Tx}}{dt}\ dt \qquad (4\text{-}69)$$

$$= \int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{f_{iQDx}(t)}{dt}\,dt + \left[\int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{\varepsilon_{Tx}}{dt}\,dt + \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{f_{iQDx}(t)}{dt}\,dt + \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{\varepsilon_{Tx}}{dt}\,dt \right] \qquad (4\text{-}70)$$

The equations (4-69) and (4-70), can be written as following:

$$\delta_i\, s_Q^{*} = \delta_i\, s_{QD}^{*} + \varepsilon_{Tir} \qquad (4\text{-}71)$$

where $\delta_i\, s_{QD}^{*}$ is the approximated interpolated quantized, rounded off and delayed which is claculated by the incremental computer as following:

$$\left[\begin{aligned} &\delta_i\, s_{QD}^{*} = \int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{f_{iQDx}(t)}{dt}\,dt \\ &f_{iQDx}(t) = f_{iQDx}[x_{oQ}, \delta_{1Q}x, \delta_{2Q}x, \ldots\ldots \delta_{(i-1)Q}x, t] \\ &f_{iQDy}(t) = f_{iQDy}[y_{oQ}, \delta_{1Q}y, \delta_{2Q}y, \ldots\ldots \delta_{(i-1)Q}y, t] \end{aligned}\right. \qquad (4\text{-}72)$$

$$t \in (t_i,\ t_{i+1})$$

and the ε_{Tir} is the total transmission error in interval $t \in (t_i, t_{i+1})$ which is equal to:

$$\varepsilon_{Tir} = \int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{\varepsilon_{Tix}}{dt}\,dt + \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{f_{iQDx}(t)}{dt}\,dt + \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{\varepsilon_{Tx}}{dt}\,dt \tag{4-73}$$

from equations (4-71), (4-72) and (4-73), the integration formula $s_Q^{*}(t)$ in interval $t \in (t_o, t_k)$ will be:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \delta_i\, s_Q^{*}(t) \tag{4-74}$$

$$= \sum_{i=1}^{k} \delta_i\, s_{QD}^{*}(t) + \sum_{i=1}^{k} \varepsilon_{Tir} \qquad t \in (t_o, t_k) \tag{4-75}$$

$$= s_{QD}^{*}(t) + \varepsilon_{Tr} \tag{4-76}$$

in the equations (4-74), (4-75) and (4-76), the $s_Q^{*}(t)$ is the approximated interpolated quantized formula of integration, which is equal to:

$$s_Q^{*}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQy}(t) \cdot d\frac{f_{iQx}(t)}{dt}\,dt \tag{4-77}$$

The s^{*}_{QD} (t) is the approximated interpolated quantized delayed formula of integration in interval $t \in (t_o, t_k)$ which is calculated by the incremental computer and is expressed as:

$$s^{*}_{QD}(t) = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{f_{iQDX}(t)}{dt}\, dt \qquad (4-78)$$

$$t \in (t_o, t_k)$$

and the ε_{Tr} is the total transmission error in interval $t \in (t_o, t_k)$ which is equal to:

$$\begin{aligned}
\varepsilon_{Tr} &= \sum_{i=1}^{k} \varepsilon_{Tir} \\
&= \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} f_{iQDy}(t) \cdot d\frac{\varepsilon_{Tx}}{dt}\, dt + \\
&+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{f_{iQDx}(t)}{dt}\, dt + \qquad (4-79) \\
&+ \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot d\frac{\varepsilon_{Tx}}{dt}\, dt \qquad t \in (t_o, t_k)
\end{aligned}$$

if the input dx of incremental computer is dt, then:

$$f_{iQDx}(t) = t$$

$$\varepsilon_{Tx} = 0 \qquad (4\text{-}80)$$

the total transmission error, when the independent variable of integral is equal to the independent variable of machine t, can be find from equation (4-79) by putting $\varepsilon_{Tx} = 0$ as following:

$$\varepsilon_{Tr} = \sum_{i=1}^{k} \int_{t_i}^{t_{i+1}} \varepsilon_{Ty} \cdot dt \qquad (4\text{-}81)$$

From the above discussion, the block diagram of incremental computer which was shown in figure (4.2) can be developed as in figure (4.4).

It is seen from figure (4.3), in transmitting the data between the integrators in incremental computers, it is introduced the delay T which cause the error of transmission ε_{Tr}.

Example: the solution of second order differential equation,

$$\frac{d^2Y}{dt^2} + Y = 0 \qquad (4\text{-}82)$$

is $\quad Y = \cos t$

This problem is programmed in incremental computer as figure (4.5).

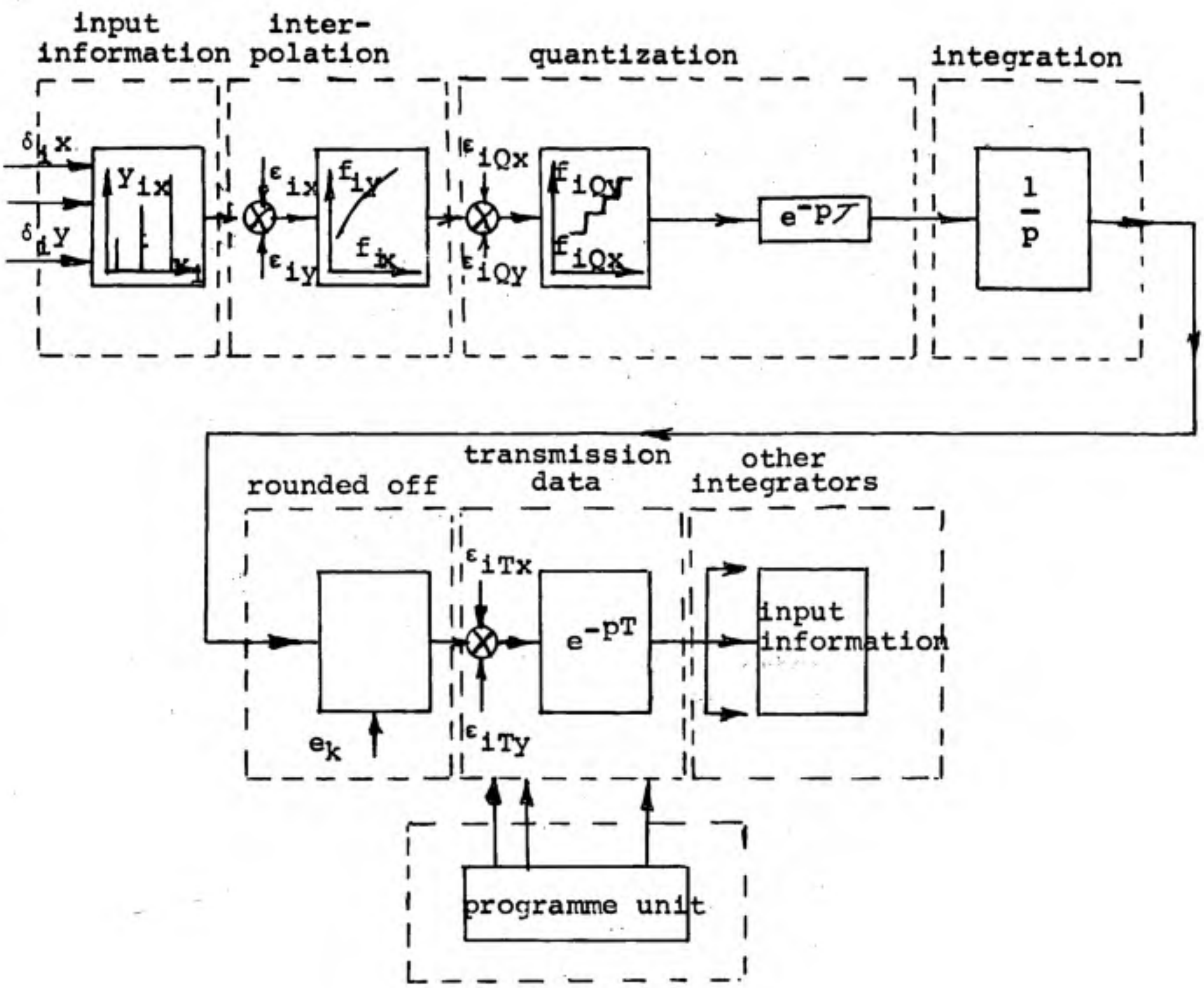

input information
inter-polation
quantization
integration
$\delta_i x$
$\delta_i y$
y_{ix}
x_i
ε_{ix}
ε_{iy}
f_{iy}
f_{ix}
ε_{iQx}
ε_{iQy}
f_{iQy}
f_{iQx}
$e^{-p\tau}$
$\frac{1}{p}$
rounded off
transmission data
other integrators
ε_{iTx}
ε_{iTy}
e^{-pT}
e_k
input information
programme unit

fig. 4.4.

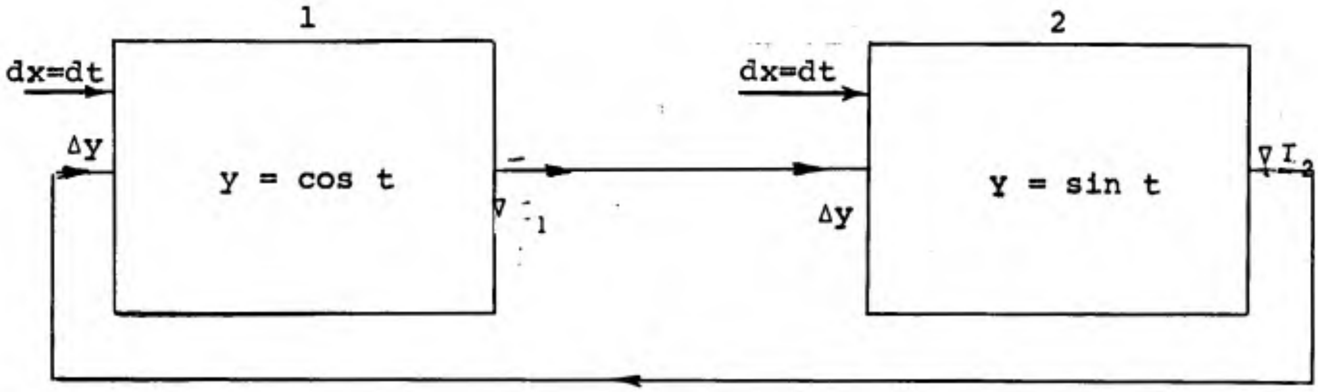


fig. 4.5.

In iteration n, the following difference equation can be written in each integrator:

in integrator (1)
$$\left\{\begin{aligned} (\nabla I_1)_n &= y_n \cdot dt \\ y_n &= y_{n-1} + \nabla y_n \end{aligned}\right. \tag{4-84}$$

in integrator (2)
$$\left\{\begin{aligned} (\nabla I_2)_n &= Y_n \cdot dt \\ Y_n &= Y_{n-1} + \nabla Y_n \end{aligned}\right. \tag{4-85}$$

as the incremental computer is the parallel type, the increments ∇I_1 and ∇I_2 which are available in the n^{th} iteration, in the input of each integrator, are from former iteration n-1 so:

$$\left\{\begin{aligned} (\nabla y)_n &= (\nabla I_2)_{n-1} \\ (\nabla Y)_n &= -(\nabla I_1)_{n-1} \end{aligned}\right. \tag{4-86}$$

by putting the value from equation (4-86) in equations (4-84) and (4-85), we will have:

$$\left\{\begin{aligned} (\nabla I_1)_n &= y_n \cdot dt \\ f_{nQDy} = y_n &= y_{n-1} + (\nabla I_2)_{n-1} \end{aligned}\right. \tag{4-87}$$

$$\begin{cases} (\nabla I_2)_n = Y_n \cdot dt \\ f_{nQDy} = Y_n = Y_{n-1} - (\nabla I_1)_{n-1} \end{cases} \tag{4-88}$$

But the right expression of equations (4-87) and (4-88) should have the information of n^{th} iteration as following:

$$\begin{cases} (\nabla I_1)_n = y_n \cdot dt \\ f_{iQy} = y_n = y_{n-1} + (\nabla I_2)_n \end{cases} \tag{4-89}$$

$$\begin{cases} (\nabla I_2)_n = Y_n \cdot dt \\ f_{iQY} = Y_n = Y_{n-1} - (\nabla I_1)_n \end{cases} \tag{4-90}$$

As it was discussed earlier in this case, the transmission error in interval $t \in (t_n, t_{n+1})$ is:

$$\begin{cases} \varepsilon_{Tx} = 0 \\ \varepsilon_{TY} = f_{xQY}(t) - f_{xQDY}(t) \qquad t \in (t_I, t_{n+1}) \end{cases} \tag{4-91}$$

from equations (4-88), (4-90) and (4-91), the ε_{Tx} and ε_{Ty} in interval $t \in (t_n, t_{n+1})$ will be:

$$\begin{cases} \varepsilon_{Tx} = 0 \\ \varepsilon_{Ty} = (\nabla I_1)_{n-1} - (\nabla I_1)_n \end{cases} \tag{4-92}$$

which will cause the total error ε_{Tr} in the process of integration.

By taking the z transformed from equations (4-89) and (4-90), we will have:

$$Y\ (z) = \frac{y_o(1-z^{-1})}{z^{-2}(1+T^2) - 2z^{-1} + 1} \tag{4-93}$$

The inverse Z transforme of the equation (4-93), can be calculated from the contour integration around the unit circle.

$$Y\ (nT) = \frac{1}{2\pi j} \int_{\Gamma} z^{n-1} \cdot Y\ (z)\ dz \tag{4-94}$$

$$= \frac{1}{2\pi j} \int_{\Gamma} z^{n-1} \frac{y_o(1-z^{-1})}{z^{-2}(1+T^2) - 2z^{-1} + 1} \tag{4-95}$$

The solution of the equation (4-95) will be:

$$Y\ (nT) = y_o \cdot e^{n \log \sqrt{1+T^2}} \cdot \cos\ (n \text{ arc Tan } T) \tag{4-96}$$

The solution of diffential equation (4-82) is the equation (4-96), it means that the transmission error ε_{Tr} has accumulated in each iteration and caused the exponential terms $e^{n \log\sqrt{1+T^2}}$ in equation (4-96).

4.3. The nonlinearity at the input of incremental computers, and choice of scale factors.

Any computation machine has a limitation in the magnitude of the numbers which it can handle. A desk calculator, for example, has an accumulator of fixed size. An electronic analogue computer operates over some limited voltage range and a digital machine has a maximum capacity of its register. In order to assure that the intermediate results stay within specified linear range during running of a problem, in incremental computer, the problem should be scaled. This means that the capacity of register must not exceed of its maximum capacity, otherwise, it will be saturated and the system becomes nonlinear. Therefore, the incremental computer has two zones, linear, and nonlinear part.

As it was discussed earlier all the quantities in incremental computer, are in the form of incremental, and any function is obtained by summing of its increment as following:

$$\begin{cases} x\ (t) = \sum\limits_{i=1}^{k} a_{ix} \cdot \delta_i x\ (t) \\ y\ (t) = \sum\limits_{i=1}^{k} a_{iy} \cdot \delta_i y\ (t) \\ \cdots\cdots\cdots\cdots\cdots \\ w\ (t) = \sum\limits_{i=1}^{k} a_{iw} \cdot \delta_i w\ (t) \end{cases} \qquad (4\cdot 97)$$

$$\left\lfloor z\ (t) = \sum_{i=1}^{k} a_{iz} \cdot \delta_i z\ (t) \right.$$

When the incremental computer works as an integrator, the dependent variable of integral y (t) is found by summing its increments $\delta_i y$ in the summator $\sum$ as it is shown in figure (4.6) and equation (4-98).

$$\left\{\begin{aligned} &\delta_i\ s_Q^{*} = \int_{t_i}^{t_{i+1}} y\ (t) \cdot dx\ (t) \\ &y\ (t) = \sum_{i=1}^{k} a_{iy} \cdot \delta_i y\ (t) \end{aligned}\right. \qquad (4\text{-}98)$$

The value of y (t) is stored in the memory of the machine, but the length of memory register is finite, there are a maximum number of increments which it can accumulate, and so the value of y (t) is limited by the capacity of Y register of memory and arithmatic unit, if the sum of increments passes the capacity of Y register of incremental computer, the Y register will be saturated.

Therefore, the input block of incremental computer can be determined as figure (4.6).

A primary purpose of scaling in incremental computer is to assure that the intermediate results of y function, stay within the specified linear range (±A) of incremental computer. The problem of

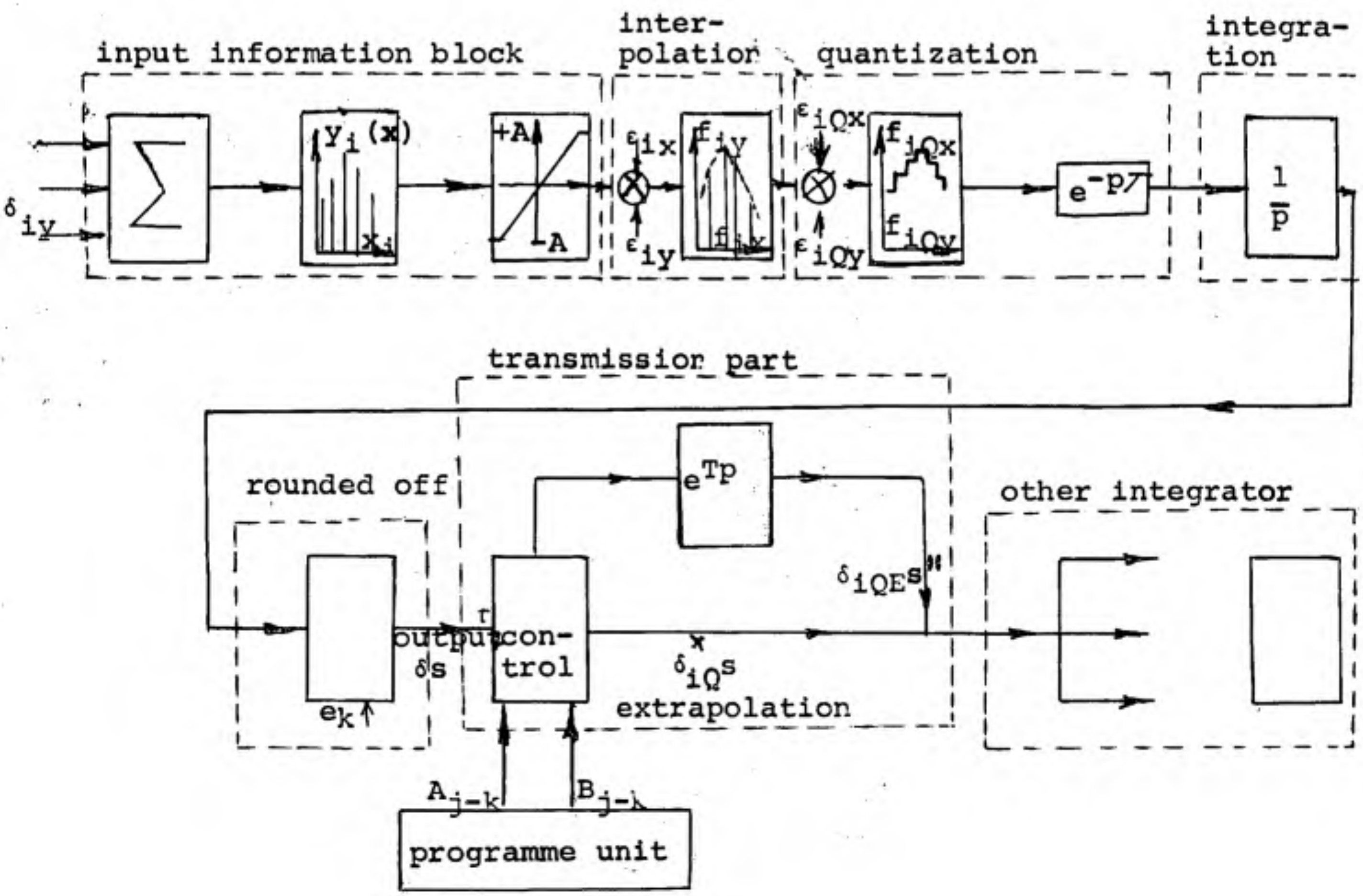


fig. 4.6.

Block diagram of serial I.C.

scaling in incremental computer, is similar to that of scaling in analogue computer. The control of scale may be achieved in a number of ways. By providing a facility that allows a choice of the number of significant digits employed in any integrator, the use of constant multiplier, and digital servos with gain etc. The first step up in scaling a problem is to estimate the maximum values of each variable which is likly to attain during the course of computation. The more accurate is this estimation, the better is the solution.
If the estimation is too low, the integrators will overflow, and the problem will have to be rescaled. If the estimation is too high, more significant places will be used than required. and it will take longer than necessary to attain a solution. Of course, it is desirable to have all scales as great as possible for the maximum capacity of register.

Although a scale factor can be any number within a machine range, restricting scale factors to integral power of the machines radix allows the product of scale factors, to be obtained by summing the exponents. So to each quantity in the machine, there corresponds a certain scale,

$$M = 2^m$$

where 2 is the radix of bineary numbers, and m is the power to which the radix 2 must be raised in order to equal M. The scale M indicates the number by which one unit of the quantity is represented in the machine. For example, if a quantity B is inserted into the Y register of the machine with a scale $M = 2^3$, this signifies that one unit of quantity B is represented in the machine in the form of 8 pulses.

Now we explain the appropriate choice of scale factor which permits the operation of incremental computer in its linear part, with the maximum accuracy.

We assume that the physical quantities are represented in the same notation as the mathematical numbers. For the unitary incremental machine we have:

$$ds = 2^{-n} \cdot y \cdot dx \qquad 4\text{-}99$$

where n is the number of bits in Y register of the integrator. Assuming ξ, ν, Γ, are physical quantities, represented by the mathematical numbers x, y, s respectively. Then, the equation (4-99) can be written as:

$$2^{S_\Gamma} \cdot d\Gamma = 2^{-n} \cdot 2^{S_\nu} \cdot 2^{S_\xi} \cdot \nu \cdot d\xi \qquad (4\text{-}100$$

where 2^{S_Γ}, 2^{S_ν}, 2^{S_ξ} are the scale factors of Γ, ν, ξ.

As the integrators have to simulate the relation between physical quantities of the form

$$d\Gamma = \xi \cdot d\nu \qquad (4\text{-}101)$$

Then the condition must be satisfied in (4-100) is:

$$S_\nu + S_\xi - n - S_\Gamma = 0 \qquad (4\text{-}102)$$

expression (4-102) is the scale relation between the foundemental quantities in unitary incremental computation.

In multiple incremental computation, we have:

$$\delta s = \frac{2^h}{2^{-n}} \; y \cdot \Delta x \tag{4-103}$$

with the same reasoning, the scale relation between the quantities of multiple incremental computation will be:

$$S_\nu + S_\xi - n + h - S_\Gamma = 0 \tag{4-104}$$

A further consideration is taken into account in choosing the value of scale factors. If in the course of variation, some physical quantity ν attains some maximum value, the quantity which is represented in the machine by the convention is:

$$\left| \frac{|\nu|}{2^{m_{max}}} \right| < 1 \tag{4-105}$$

where m_{max} is the exponent of 2 in such a way that, the value which is represented in the machine becomes smaller than one. So the quantity which is represented in the machine is $|\nu| \cdot 2^{m_{max}}$, and taking into account the scale factor 2^{S_ν}, this value is represented in the machine $|\nu| \cdot 2^{m_{max}} \cdot 2^{S_\nu}$. The maximum capacity of Y register of the integrator is 2^n, for avoiding the overflow of Y register, the following relation should be satisfied:

$$2^{m_{max}} \cdot 2^{S_\nu} < 2^n \tag{4-106}$$

or

$$S_\nu + m_{max} < n \tag{4-107}$$

The equations (4-102), (4-104) and (4-107) determine the scale factor of each integrator in unitary and multiple incremental computer.

In order to increase the accuracy of the problem, the scale factors should be chosen in such a way, to use the full capacity of the Y register provided that the machine works in linear zone and does not saturate.

4.4. Conclusion.

We have seen in this chapter that, by using the appropriate initial condition at S register, in unitary increment computation, the round off error becomes smaller than one, and in multiple incremental computation, becomes smaller than 2^h.

We also calculated, the transmission error in increment computers, and in the next chapter, we will study the way of minimizing this error.

As it is shown, the nonlinearity at the input of incremental computers, can be avoided by a good choice of scaling.