CHAPTER 3

DESCRIPTION ON ANALOG COMPUTER ELEMENTS IN SIMULATION

Digital-Analog Simulation นี้ประกอบขึ้นด้วย Elements ต่าง ๆ ที่เขียนอยู่ในรูปของ subroutine subprogram subprogram เหล่านี้จะ independentซึ่งกันและกัน และจะต้องอาศัย instruction ANALOG SUBROUTINE คอยเป็นสมองที่จะนำเอา Elements เหล่านี้เข้ามา link ด้วยกัน Subprogram ต่าง ๆ ใน DAS นอกจากจะเป็น elements ที่สำคัญ ๆ ใน ANALOG COMPUTER แล้ว ยังได้ combine พวก logical circuit elements - ต่าง ๆ เพื่อสดวกในการใช้งาน ดังจะได้อธิบายถึงหน้าที่ความสำคัญและวิธีการ simulate ใน passage ต่อไป

Subroutine subprogram ต่าง ๆ ใน DAS แบ่งออกได้เป็นพวกใหญ่ ๆ ๓ พวก คือ

A. Analog computer elements

1. Operational Integrator

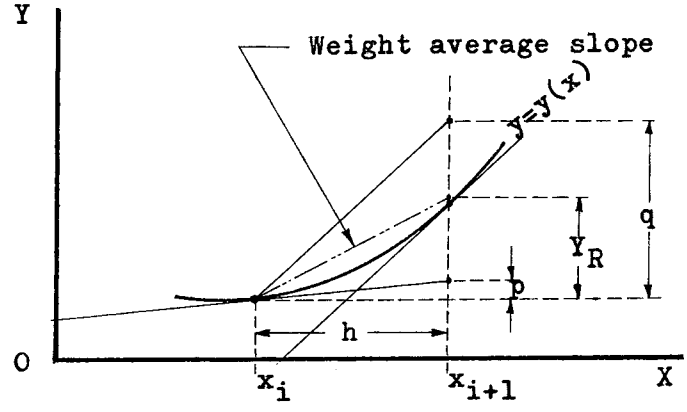
Element นี้เป็นหัวใจของ Analog computer ซึ่งเป็น low pass RC circuit ธรรมดา ๆ ถ้าพิจารณาทาง mathematical form แล้ว integration ก็คือการหา area ใต้ curve ของ input นั่นเอง และด้วยเหตุที่ว่า ANALOG COMPUTER เป็น continuous device ส่วน digital computer จะให้คำเป็นแบบ discreted value ฉะนั้นใน digital computer เราจึงต้องใช้ numerical method เข้าช่วยในการหาค่าที่เกี่ยวกับ continuous value และความละเอียดจึงขึ้นอยู่กับช่วงของ interval ที่ใช้ในการ integrate และวิธีการ integrate วิธีที่ง่ายที่สุดของการ integration ก็คือใช้วิธีของ trapezoidal rule หรือวิธีของ Simpson's rule ซึ่งวิธีทั้งสองนี้ ถ้า integration period กว้าง error ก็จะมาก และ error นี้จะสะสมไปทุกช่วงของการ integrate ฉะนั้นถ้าเราแบ่ง integration period ให้มันเล็กเข้า ๆ accuracy ก็จะสูง แต่ก็จะต้องเสียเวลาในการ integrate มากขึ้น สำหรับ method ที่ใช้กันมาก และ error

จะมีค่าน้อยที่สุดก็คงจะได้แก่ Rung-Kutta Method

Rung-Kutta Method คล้ายคลึงกับวิธีการของ Euler Method ซึ่งใช้วิธี linear approximation ที่ต่างกันก็คือว่า วิธีนี้ใช้ค่าของ slope ในรูปของ weight average ในช่วงของ x_i ถึง x_{i+1} ในการหาค่าการเปลี่ยนแปลงของค่า , y Rung-Kutta Method ที่จะใช้ต่อไปนี้เป็นแบบ Second order method ซึ่ง Solution ของมันจะมี truncation error เป็น $O(h^2)$ (๑)

การหาค่า Weighted average ของ slope มีด้วยกัน ๒ วิธี ดังนี้คือ.-

<u>วิธีที่ ๑.</u> หา Slope ที่จุดเริ่มต้นและจุดสุดท้าย แล้วหาค่า average ของค่าของ slope ทั้งสอง ส่วนการหาค่าจุดต่อไปก็คือ หาผลบวกของจุดเริ่มต้นกับผลคูณของ weighted average ของ slope และ interval ดังรูป.



<u>รูปที่ ๑</u> วิธีการหา Slope เฉลี่ย

---

(๑) Royce Beckett, James Hurt, "Numerical Calculations and Algorithms", pp 202-207

$$p = hf(x_i, y_i) \quad (1)$$

$$q = hf(x_i+h, y_i+p) \quad (2)$$

$$\therefore \quad y_{i+1} = y_i + \tfrac{1}{2}(p+q) \quad (3)$$

วิธีที่ ๒ หาจาก slope ของครึ่งหนึ่งของช่วงของ interval ในการหา solution ก็เช่นเดียวกับในการหาในวิธีที่ ๑ ซึ่งจะได้กล่าวอย่างละเอียดต่อไปข้างล่าง และเป็นวิธีที่ใช้ใน program.

ด้วยเหตุที่การทำงานของ DAS เป็นแบบ sequential ความจำเป็นของการคำนวณของแต่ละจุดในแต่ละขณะจึงต้องระมัดระวังอย่างมาก เพราะว่า integrator ตัวที่สองจะหาได้ก็ต่อเมื่อตัวที่หนึ่งหาได้แล้ว ใน Rung Kutta second order method integrator แต่ละตัวจะหาแต่ละจุดได้ต้องผ่านกรรมวิธีถึง ๔ ครั้งที่เวลาต่างกัน เช่น integrator ตัวที่หนึ่งกำลังทำการหา slope ตัวที่สองซึ่งหาในเวลาต่อมาจะทำการ integrate หาจุด ซึ่ง output ของตัวที่หนึ่งยังไม่มีค่าออกมาเลย ทำให้ตัวที่สอง integrate หาค่าออกมาไม่ได้ ฉะนั้นจึงต้องใช้ technique ในการ integration โดยแยกการ integrate และหา slope ด้วย control "SWITCH", "SWITCH" นี้จะมีอยู่ ๔ ค่าด้วยกัน หรือ ๔ ตำแหน่ง นั่นเอง integrator ทุกตัวจะทำงานเหมือนกันหมดในแต่ละครั้งของ "SWITCH" ที่อยู่ตำแหน่งหนึ่ง ๆ ดังนี้ :-

๑. SWITCH อยู่ที่ ๑ INTEGRATOR ทุกตัวจะ set ค่า input ของมันอยู่ที่ initial condition ตามที่ data ได้ควบคุมที่เวลา $t = 0^+$

๒. SWITCH อยู่ที่ ๒ integrator ทุกตัวจะหาค่าของ input value ของ curve และหา integration ของจุดนั้นที่เวลานั้น

๓. SWITCH อยู่ที่ ๓ integrator ทุกตัวจะหาค่าของ integration ที่ half-interval

๔. SWITCH อยู่ที่ ๔ integrator ทุกตัวจะหาค่า integration ที่ interval ที่กำหนดให้

SWITCH ทั้ง ๔ จะทำงานตาม sequence ดังนี้ คือ ๑-๒-๓-๒-๔-๒-๓-๒-๔ .....เช่นนี้เรื่อยไป จะเห็นว่าทุกครั้งที่ SWITCH = 2 เวลาจะต้องไม่เพิ่มขึ้น และ integrator จะทำการคำนวณ slope ที่จุดนั้น

หลักการทำงานของ integrator ในแต่ละ integration period มีดังนี้คือ :-

๑. ที่ $t = 0^+$ Define ค่า initial condition ของ integrator ทุกตัว ซึ่งเสมือนหนึ่ง charge capacitor ของ integrator ทุกตัวใน Analog computer ขณะนี้ "SWITCH" จะอยู่ที่ ๑

๒. ที่ $t = 0^+$ นี้เองหา slope ของจุด โดยที่ slope ของจุดนั้น ก็คือค่า input function ที่จุดนั้น ดังสมการ

$$SLOPE = f(x_0, y_0) \qquad (4)$$

ขณะนี้ SWITCH จะอยู่ที่ ๒

๓. Define ค่าช่วงของการ integrate

interval = ค่า abscissa ของจุดถัดไป - ค่า abscissa ของจุดที่กำลังกล่าวถึง

๔. หาค่าของ integration ที่จุด ½ interval ได้ ขณะนี้ t = ½ interval

ค่าของ integration ที่จุดนี้ = ค่าของจุดที่กล่าวใน 1 + ½ ของ interval x ค่าของ slope ที่กล่าวใน ๒ หรือก็คือ $f(x_0, y_0)$ นั่นเอง ..... (๕)

ขณะนี้ "SWITCH" อยู่ที่ตำแหน่ง ๓

๔. หาค่าของ SLOPE หรือค่าของ input function value ของจุดที่ ½ interval ซึ่งเป็นจุดเดียวกับที่กล่าวใน ๔

$$SLOPE = f(x_{½}, y_{½}) \qquad (5a)$$

ขณะนี้ SWITCH อยู่ที่ตำแหน่ง ๒

๕. หาค่าของ integration ของจุดที่ช่วง ๑ interval ได้จาก

ค่า integration ที่ช่วง ๑ interval = ค่า integration ของจุดที่กล่าวใน 1 + ค่าของ interval x ค่าของ slope หรือ $f(x_{½}, y_{½})$ ที่หาใน (5a)

..... (๖)

ขณะนี้ SWITCH อยู่ที่ตำแหน่ง ๔

การทำงานของการ integrate จะไปซ้ำกันอีกที่ข้อ ๒ เช่นนี้เรื่อยไป

เนื่องจาก Core storage ใน computer มีค่าจำกัด program นี้จึงจำกัดไว้ว่า integrator จะต้องไม่เกิน ๒๐ ตัว มิฉะนั้น computer จะไม่ทำงานและพิมพ์ error massage ออกมา

๒. Potentiometer หรือ PARAMETER SUBROUTINE ใน DAS.

ใน ANALOG COMPUTER เมื่อเราต้องการ OUTPUT VOLTAGE ที่ค่าต่าง ๆ กัน devide ที่สำคัญก็คือ potentiometer ซึ่งสามารถจะ vary resistance ได้ ในกรณีที่ต้องการ Voltage คงที่ตลอดการทำงานนั้น เราก็สามารถจะ specified ค่าลงใน program ซึ่งเป็น arithmatic statement ของ FORTRAN ทั่ว ๆ ไป เช่น

A = 2xB

๒. ก็คือค่าคงที่ที่ว่าซึ่งใน analog computer จะ set potentiometer ให้ได้ output อยู่ที่ ๒ ตลอดการทำงานนั้น ๆ

สำหรับ variable potentiometer เนื่องจากใน analog computer เป็นแบบกึ่ง automatic ซึ่งจะสามารถ vary ค่าได้ทุกขณะ แต่ว่าใน digital computer ซึ่งเป็นแบบ automatic ทุกอย่างจะต้อง control ด้วย program หรือ card data และจะเจาะจงลงใน program ไม่ได้ Variable parameter จึงไม่มีใน FORTRAN statement DAS ได้ develope program นี้ขึ้นมาให้สามารถ vary ได้ทุกขณะตามที่เราต้องการ โดยวิธี input data card ซึ่งต่างกับ analog computer ที่ใช้ manual control petentiometer แทน จะเห็นว่า program นี้มีประโยชน์อย่างมาก

เวลาใช้งานใน program ก็เพียงแต่เรียกชื่อของ parameter นั้น ๆ ส่วนค่าของมันให้ define ไว้ใน card data ซึ่งสามารถจะ control ได้ด้วย card data อีกชุด อย่างไรก็ตาม PARAMETER นี้จะต้องไม่เกิน ๑๐ ตัว ตามที่ PARAMETER SUB-ROUTINE ได้กำหนดเอาไว้

๓. DC Amplifier

ประโยชน์ของ DC amplifier ใน ANALOG COMPUTER ก็คือต้องการให้ทำหน้าที่เป็น gain และ summer, elements ทั้งสองนี้ เราสามารถที่จะ simulate ลงใน DAS ได้ โดยอาศัย APPLICATION ของ FORTRAN ARITHMETIC STATEMENT.

๔. Multiplier และ Divider

Elements ทั้งสองนี้สามารถจะ simulate ได้ด้วย FORTRAN ARITHMETIC STATEMENT.

๕. FUNCTION GENERATOR

a. function generator.

ใน ANALOG COMPUTER เราสามารถจะ generate continuous curve ที่เป็น function ที่ยากแก่การ express ในรูปของสมการที่ง่าย ๆ ทำให้เราไม่สามารถที่จะหาค่าต่าง ๆ ได้ด้วยสมการทางพีชคณิตได้ แต่อย่างไรก็ตาม เราก็สามารถที่จะอาศัย NUMERICAL METHOD ANALYSIS ในการ interpolate ค่าต่าง ๆ เมื่อจุดต่าง ๆ บน curve บางจุดเรารู้ค่าแล้ว

DAS ได้อาศัยหลักการข้างบนในการสร้าง FUNCTION GENERATOR SUBROUTINE โดยอาศัยค่า coordinate pair ใน table ที่กำหนดให้ทำการ interpolate ค่าอื่น ๆ ได้ ส่วนตัวเลขนั้นจะมีความละเอียดมากน้อยแค่ไหนย่อมขึ้นอยู่กับจำนวนจุดที่กำหนดให้ และวิธีการ interpolate แต่ว่าค่าตาม abscissa จะต้องไม่ซ้ำกันซึ่งจะเห็นได้จากสมการดังจะกล่าวต่อไปนี้ ทำให้กลายเป็นข้อเสียของ simulation นี้

Program ที่ทำการ generate function ใน DAS มีอยู่ ๓ วิธีด้วยกันคือ (๑)

METHOD 1. LINEAR INTERPOLATION

METHOD 2. SECOND ORDER INTERPOLATION

METHOD 3. LAGRANGE'S INTERPOLATION

อย่างไรก็ตาม การเลือกวิธีการ interpolate ย่อมขึ้นอยู่กับ programmer ที่จะต้องวินิจฉัยดูว่า ลักษณะของ curve เป็นอย่างไร ความละเอียดของตัวเลขต้องการน้อยมากแค่ไหน ซึ่งวิธีทั้ง ๓ นี้จะได้กล่าวอย่างละเอียดต่อไป

---

(๑) Mario G Savadori, Melvin L. Paron, "Numerical methods in Engineering," PP 87-88

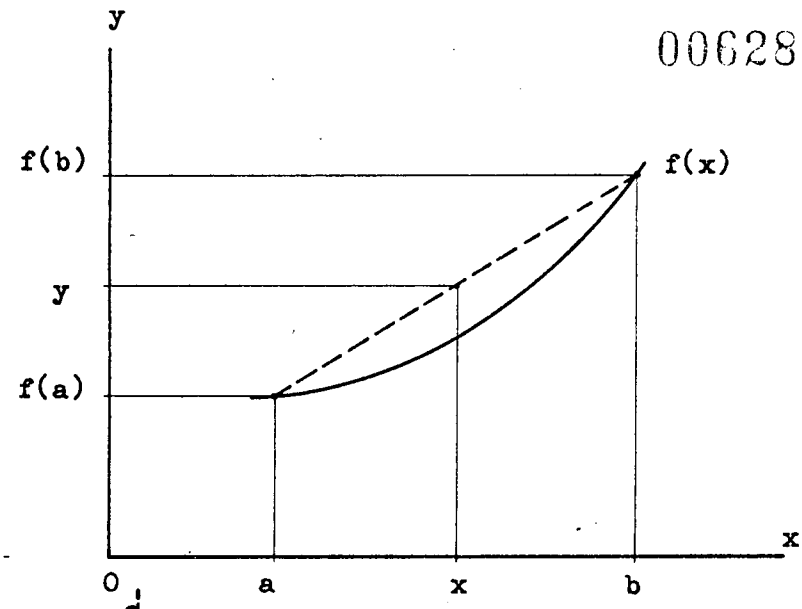
METHOD 1. LINEAR INTERPOLATION หรือเรียกว่า FIRST ORDER FINITE DIFFERENCE APPROXIMATION ของ function f (x) ที่เป็นเส้นตรงระหว่างจุดสองจุด สมการของเส้นตรงที่กล่าวนี้สามารถที่จะเขียนได้ในรูปของ Difference equation ดังนี้

$$y = f(a) + \frac{f(b) - f(a)}{(b-a)} (x-a) \qquad (7)$$

y คือ ค่าของ f(x) ที่อยู่ระหว่างจุด a และ b ๒ จุด นั่นคือ $a \leq x \leq b$

a และ b เป็นค่า abscissa

ส่วน f(a) และ f(b) เป็นค่าของ function ที่จุด a และ b ดังรูป



006285

รูปที่ ๒ Linear interpolation illustration.

จากรูปจะเห็นว่า ถ้า functionโค้งมาก การหาค่าจาก approximation นี้ไม่ได้ จะต้องหาค่าจาก polynomial equation ซึ่ง degree สูงกว่า ค่าที่ได้จะได้ใกล้เคียงกับความจริงมากยิ่งขึ้น

METHOD 2. SECOND ORDER INTERPOLATION เมื่อการ interpolationใน METHOD 1. ซึ่งได้ค่าห่างจากความจริงมาก แสดงว่า curve มีความโค้งมากซึ่งไม่สามารถจะใช้วิธี linear approximation ได้ อย่างไรก็ตาม วิธีการหาของ method นี้ก็เช่นเดียวกับ METHOD 1. ดังต่อไปนี้

ถ้ากำหนดค่าของ function f(x) ที่ $x = x_o, x_1, x_2 \ldots x_n$ เป็น $f(x_o), f(x_1), f(x_2), \ldots f(x_n)$ จาก linear interpolation ที่ได้ใน method 1.

$$f(x) = f(x_o) + (x-x_o)\frac{f(x_1) - f(x_o)}{(x_1-x_o)} \qquad (8)$$

และถ้ากำหนดให้

$$f(x_o,x_1) = \frac{f(x_o)-f(x_i)}{x_o-x_1} = \frac{f(x_1)-f(x_o)}{x_1-x_o} = f(x_1,x_o) \qquad (9)$$

นั่นคือ $f(x_i,x_j) = f(x_j,x_i)$

ทำนองเดียวกับจาก definition ที่กล่าวข้างบนนี้ สำหรับ second order interpolation ก็จะได้

$$f(x_i,x_j,x_k) = \frac{f(x_i,x_j)-f(x_j,x_k)}{x_i-x_k} \qquad (10)$$

จะเห็นว่าในการหาค่าแต่ละค่า เราต้องการคำนึงถึงค่า ๓ จุด ด้วยกัน และ x_i จะต้องไม่เท่ากับ x_k ด้วย เราสามารถที่จะหาค่าของ $f(x)$ ได้จากสมการ

$$f(x) = f(x_0) + (x-x_0)\, f(x_1,x_0) + (x-x_0)\,(x-x_1) f(x_2,x_1,x_0) \qquad (11)$$

และจาก (๕) และ (๑๐) สมการ (๑๑) จะสามารถเขียนให้ง่ายเข้าได้ดังนี้.-

$$f(x) = f(x_0) + \frac{(x-x_0)}{(x_1-x_0)}\, f(x_1)-f(x_0) + \frac{(x-x_0)\,(x-x_1)}{(x_2-x_1)(x_2-x_0)}(f(x_2)-f(x_1))$$

$$- \frac{f(x_1)-f(x_0)}{(x_1-x_0)(x_2-x_0)}(x-x_1)(x-x_0) \cdots\cdots\cdots\cdots\cdots \qquad (12)$$

จะเห็นว่าสูตรนี้ยาวมาก จำนวนจุดนอกจากจะต้องใช้มากกว่า METHOD 1 แล้วการคำนวณยังต้องใช้เวลานานกว่าอีกหลายเท่า

METHOD 3 LAGRANGE'S INTERPORATION FOR UNEVENLY SPACED POINTS

ถ้ากำหนดจุด (x_0,y_0), (x_1, y_1) ..... (x_n, y_n) ให้ เราจะหาค่า Polynomial equation ได้ดังนี้

$$P_x(x) = C_x p_x(x) = C_k(x-x_0)(x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})(x-x_n) \qquad \ldots.(13)$$

สมการนี้ $P_{x_i}(x)$ จะมีค่าเป็น o เมื่อ $x = x_i$ และ $P_x(x)$ จะมีค่าที่ $x = x_k$ เท่านั้น และค่าของมันตาม definition แล้วมีค่าเท่ากับ ๑

$$C_k = \frac{1}{(x_k-x_0)(x_k-x_1)(x_k-x_2)\ldots(x_k-x_n)} \qquad \cdots\cdots\cdots (14)$$

นั่นคือ จะได้ degree ที่ n ของ polynomials ดังนี้

$$P_k(x) = \sum_{k=0}^{n} y_k\, P_k(x) \qquad \cdots\cdots\cdots (15)$$

ซึ่ง $P_k(x_i) = 0$ เมื่อ $i \neq k$

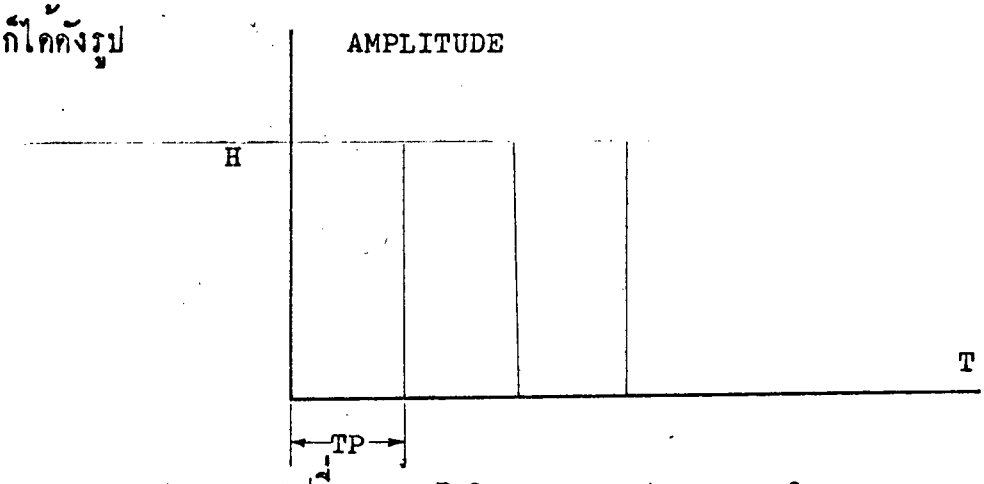
$= 1$ เมื่อ $i = k$

ในจำนวนวิธีที่กล่าวทั้ง ๓ วิธีนี้จะยากที่สุด และเสียเวลามากในการคำนวณ ซึ่งควรหลีกเลี่ยงวิธีนี้ให้มาก ในกรณีที่ curve โค้งมาก ๆ METHOD ๑ และ ๒ จะมี error มากกว่า METHOD ๓ นี้

สิ่งที่น่าสังเกตอย่างหนึ่งก็คือว่า space ของแต่ละจุดที่กำหนดให้ใน function generator ไม่จำเป็นต้องเท่ากัน และ data ก็ไม่จำเป็นต้องเรียงตาม sequence ด้วย DAS ไม่สามารถที่จะ accept จำนวน table ให้มากกว่า ๓ ได้ และจำนวนจุดของแต่ละ table จะต้องไม่เกิน ๑๐ จุด

b. Pulse generator

DAS สามารถที่จะ generate pulse ได้คล้ายกับ Block Oscillator ใน electronic circuit (๑) สิ่งที่สำคัญของ element subroutine นี้ก็คือจะต้องกำหนดค่า base time period และ amplitude ของ pulse ค่าทั้งสองอาจจะเป็นค่า constant ที่กำหนดให้ หรืออาจจะเป็น variable เปลี่ยนแปลงไปตาม program ก็ได้ดังรูป

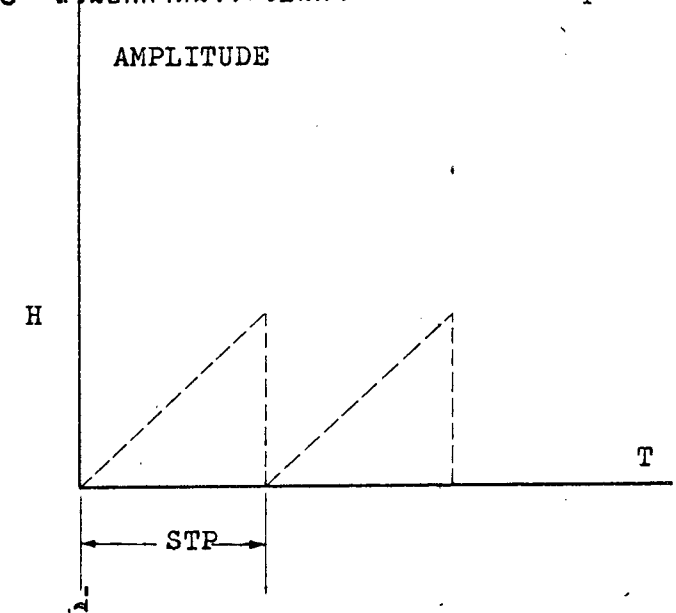


รูปที่ ๓ Pulse generator waveforms

(๑) Jacob Millman, Herbert taub, "Pulse, Digital, and Switching waveforms", PP. 597-621.

Pulse generator ใน DAS จะต้องน้อยกว่า ๕ ชุดเสมอ

c. Sawtooth generator (๑) DAS จะ generate sawtooth waveform ได้ในลักษณะที่เป็น Discreted value, Programmer จะต้องกำหนดค่าของ sawtooth time period และ amplitude ของ sawtooth ซึ่งอาจจะคงที่หรือ very ไปตาม program อย่างไรก็ตาม 5 generator จะเป็นค่า maximum ที่ DAS สามารถจะ generate ได้ วิธีการ generate sawtooth ใน DAS ก็คือว่า ให้หาค่า Slope ก่อน แล้วหาจุดต่อไปได้จากการรวมค่าที่จุดเริ่มแรกกับ slope คูณด้วย T ที่เพิ่มขึ้น เมื่อใดก็ตามที่ T = sawtooth pulse period แล้ว ค่า amplitude ของมันจะมีได้ ๒ ค่า คือค่าหนึ่งจะเป็น o ส่วนอีกค่าหนึ่งจะเป็นค่า maximum amplitude ดังรูป

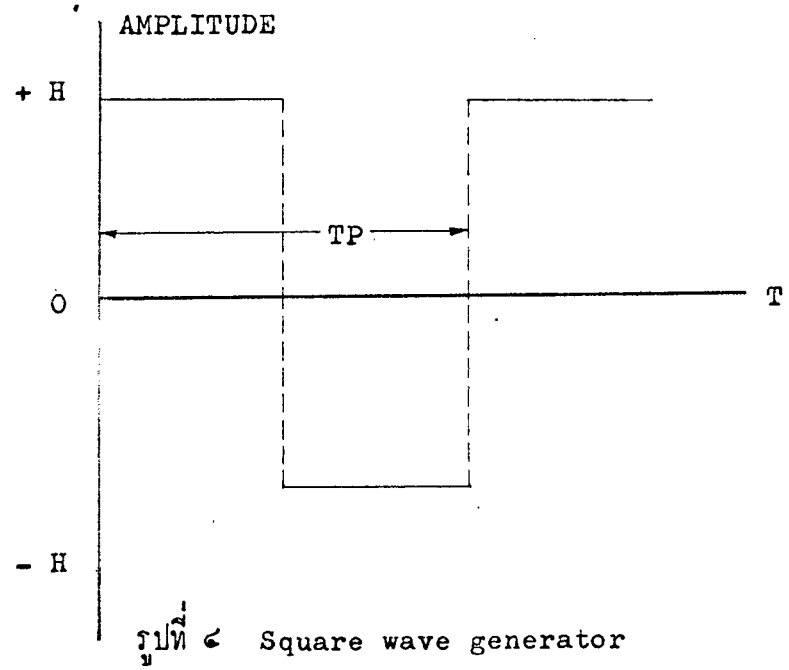


รูปที่ ๔๐ Sawtooth generator waveforms

---

(๑) Jacob Millman, Herbert Taub, "Pulse, Digital, and Switching waveforms", PP. 597-621.

d. Square wave generator เช่นเดียวกับ sawtooth generator. programmer จะต้องกำหนดค่า amplitude และ period time ของ waveform ให้ โดยเริ่มต้น initial condition ค่าของ square wave generator จะอยู่ที่ maximum amplitude เมื่อใดก็ตามที่ time เพิ่มขึ้นจนเท่ากับ period time ของ generator แล้ว ทางทฤษฎีจะมีได้ ๒ ค่า คือ ที่ + H และ - H แต่Compu... จะบอกค่าเพียงแค่ที่ค่าเดียวเท่านั้น จำนวน square wave generator จะต้องไม่เกิน ๔ ... เพราะประหยัดในเรื่อง core storage



รูปที่ ๔ Square wave generator

e. Random noise generator. Subroutine subprogram ชุดนี้ เป็นการ simulate random noise ที่เกิดขึ้นมาใน electronic circuit ซึ่งไม่สามารถจะรู้ค่าของมันล่วงหน้าได้ ค่าของ random นี้จะขึ้นอยู่กับค่า input waveform สำหรับวิธีการ generation ทำได้ดังนี้คือ

๑. นำค่า input waveform value นี้มายกกำลัง

ถ้า x คือ ค่า input

ให้ YRF = e^x

๒. ลบค่าตัวเลขหน้าจุดทศนิยมของ YRF ให้หมด เพื่อที่จะให้ YRF มีค่าเฉพาะไม่ถึง ๑

๓. ทำค่าตัวเลขหลังจุดทศนิยมของ YRF ให้เป็น integer value โดยย้ายจุดทศนิยมไป ๕ ตำแหน่ง ทั้งนี้ ก็เพื่อที่จะไม่ให้ค่า constant นี้เกินค่า maximan value ของ integer constant ใน FORTRAN STATEMENT

๔. นำค่าที่ได้ใน ๓ หารด้วย ๕๐,๐๐๐ ค่าที่ได้อาจจะมากกว่า ๑ หรือน้อยกว่า ๑ ก็ได้ ทำให้ค่านี้มีค่าเป็นจุดทศนิยม หลังจากนี้เอา ๑ ไปลบออกอีกครั้ง ทำให้ค่านี้อาจจะเป็น - หรือ + ก็ได้

๕. เอาเฉพาะค่าจำนวนเต็มของ input value มาหาค่า sine ค่านี้จะต้องมีค่าไม่เกิน ๑

๖. นำค่าที่ได้จาก (๔) และ (๕) มาคูณกัน ค่าที่ได้นี้จะมีค่าน้อยกว่า ๑ เสมอ

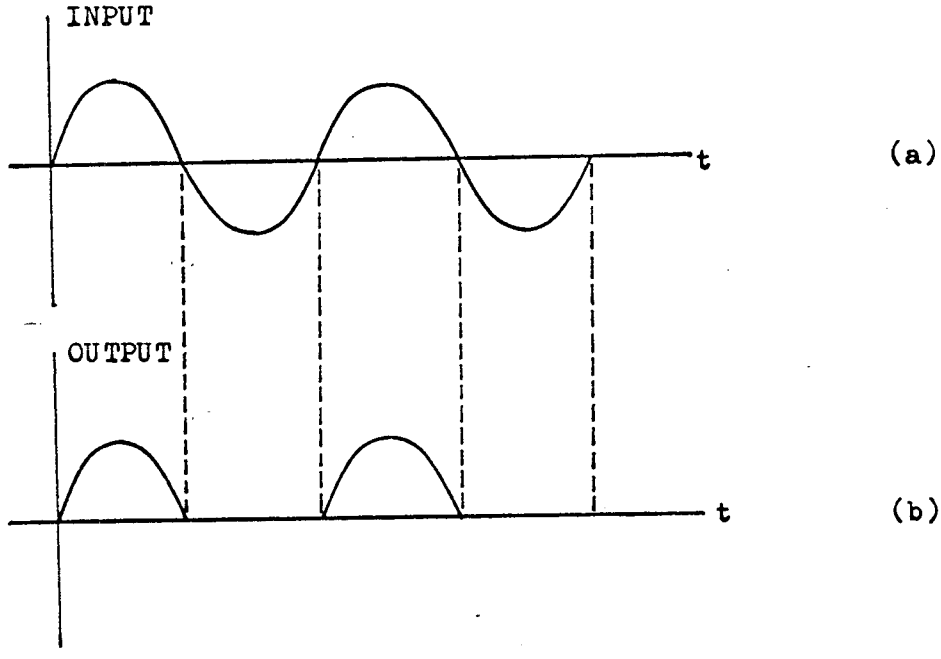
๗. คูณค่าที่ได้ใน (๖) ด้วย ๑๐ ยกกำลัง x แล้วเอา ๒ ค่านี้มาลบออกจากกัน ค่าที่ได้ก็คือ ค่า ranrom value

f. Sine, cosine, exponential, square root และ arctangent function, function generator ทั้งหมดที่กล่าวถึงนี้ มีอยู่ใน FORTRAN MATHEMATICAL FUNCTION แล้ว DAS จึงนำมาใช้เฉย ๆ

B. Clipping circuit และ limiting comparator.

Clipping circuit เป็นวงจรที่ต้องการให้ส่วนหนึ่งส่วนใดของ Waveform ผ่านไปได้ และส่วนอื่นก็จะผ่านไปไม่ได้ ส่วนของ Waveform ที่ผ่านไปได้นั้น อาจจะอยู่บนหรือข้างล่างของ Reference level นั้น ๆ

๑. Negative clipper subprogram นี้ จะสามารถ simulate วงจรที่รับ input signal มาแล้ว, ทำให้ output waveform ที่สามารถจะผ่านไปได้เพียงส่วนที่เป็น positive เท่านั้น ส่วนที่เป็น negative จะถูก clip ออก ดังรูป.



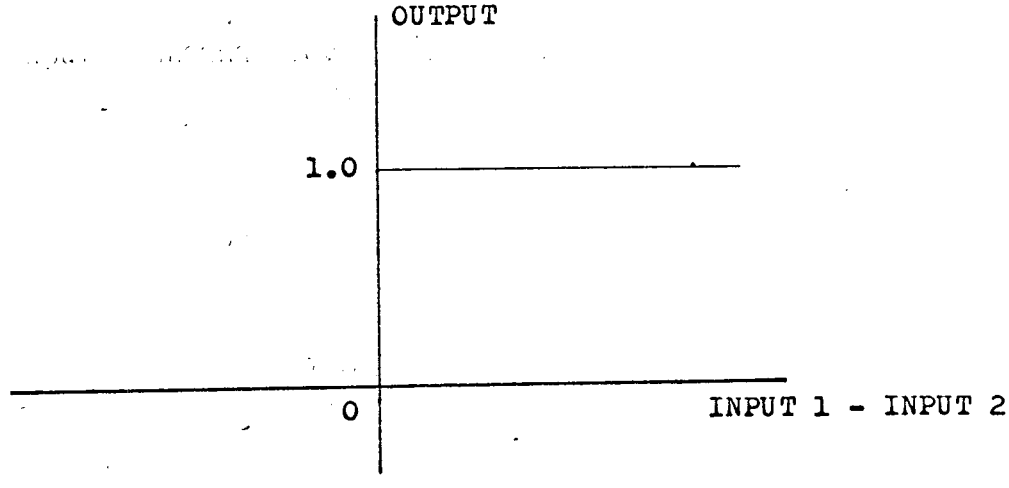
รูปที่ ๕ Negative clipper a. input waveform, b. output waveform

---

(๑) Jacob Millman, Herbert Taub, "Pulse, Digital, and Switching waveforms."

๒. Positive clipper subprogram ส่วนนี้จะทำหน้าที่เป็นส่วนกลับกับ Negative clipper

๓. Comparator subprogram นี้ จะทำหน้าที่ mark ที่เวลาหนึ่งเวลาใดว่า input waveform ถึงค่า reference level แล้วหรือยัง ซึ่งต่างกับ clipping circuit ที่ว่า comparator จะไม่มีการ produce wave form ออกมาเลย เมื่อใดก็ตามที่ค่าของ input เกินค่า level หนึ่ง ๆ แล้ว ค่าของ output อาจจะอยู่ที่ level หนึ่งตลอดไม่ว่า input จะมีค่าเป็นเท่าไรก็ตาม และจะอยู่อีก level หนึ่งเมื่อค่าของ input ต่ำกว่าค่า level นั้น ๆ ไม่ว่า input จะมีค่าเป็นเท่าไรก็ตาม ดังรูป.

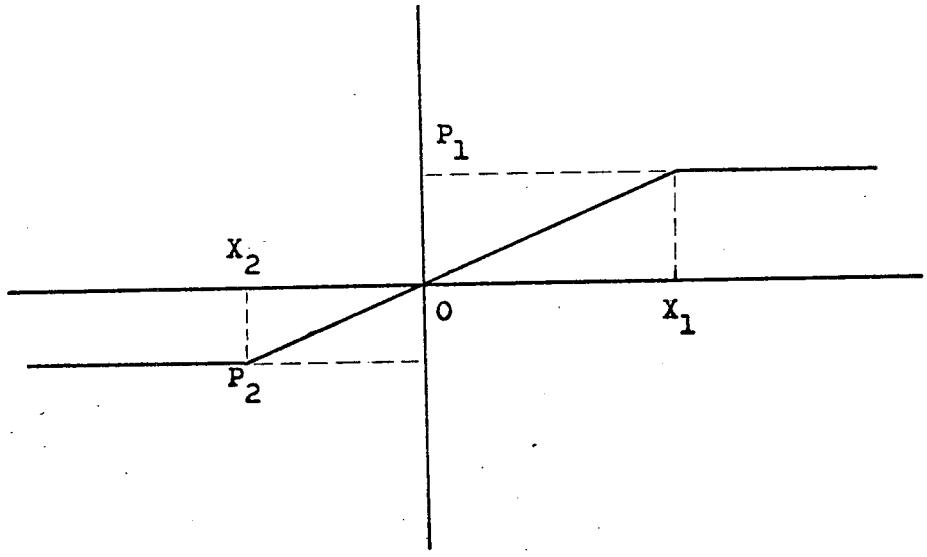


รูปที่ ๖ Comparator.

จากรูป จะเห็นว่าเมื่อไรที่ INPUT 1 มากกว่า INPUT 2 แล้ว output จะอยู่ที่ 1.0 แต่ถ้า INPUT 1 น้อยกว่า INPUT 2 แล้ว output จะมีค่าเป็น 0 เป็นต้น

Subprogram นี้ มีประโยชน์มากใน logical system ซึ่งมีหน้าที่ในการเปรียบเทียบ 2 input signals ว่า output ควรจะมีค่าเป็น 1 หรือ 0

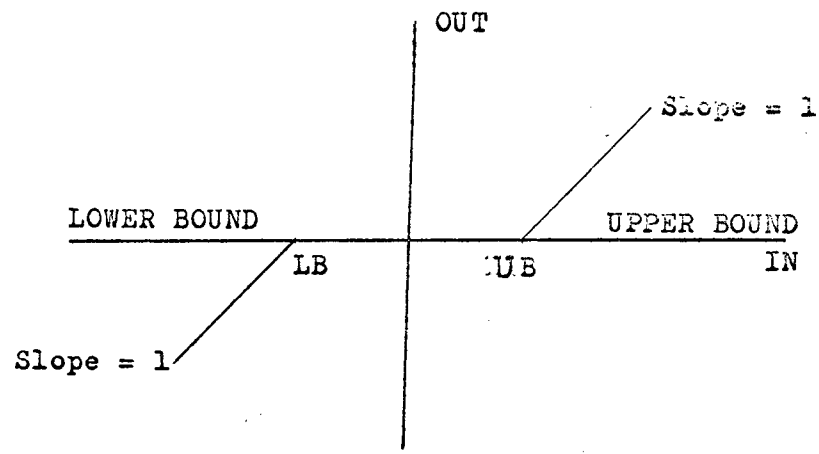
๔. Limiter หน้าที่ของ Subprogram นี้ก็คือ ต้องการ simulate ลักษณะ characteristic ของ amplifiers และ motors ที่ใช้ ทั้งนี้เพราะว่า Characterictic เหล่านี้มีลักษณะเป็น Saturation nonlinearty, programmer สามารถที่จะ vary slope และ amplitude ได้ดังรูป



รูปที่ ๙ Limiter.

Application ของ element นี้ จะเห็นว่าสามารถ simulate characteristic curve ของ transistor ได้ โดย vary ค่าของ slope และ amplitude แล้วผ่าน Negative clipper ก็จะได้ curve ตามที่ต้องการ

๕. Dead Space. subprogram นี้ก็ทำหน้าที่เช่นเดียวกับ Limiter ใช้ในการ simulate characteristic ของ nonlinear element เช่น Diode เป็นต้น Program จะกำหนดค่า dead zone ที่มี upper bound และ lower bound มาให้ ดังรูป.



รูปที่ ๔ Dead Space

จากรูปจะเห็นว่า output ของ limiter จะเท่ากับ input ได้ก็ต่อเมื่อค่าของ input มากกว่า upper bound หรือว่าน้อยกว่า lower bound

C. Logical Subprogram.

๑. Automatic Stop

Subprogram นี้ จะทำหน้าที่ simulate การทำงานของ breaker หลักการทำงานก็คือว่ามันจะรับเอา input signal X_1 และ X_2 สองอันมา COMPARE กัน เมื่อไรก็ตามที่ variables ทั้งสองใน program เกิดผิดปกติขึ้นมา คือ อันที่มีค่ามากกลับมีค่าน้อยกว่าอีกอันหนึ่งขึ้นมา subroutine อันนี้ก็จะทำหน้าที่สั่งให้เครื่องหยุดทำงานชั่วคราว เพื่อรอการแก้ไขจาก programmer.

๒. AND gate

การทำงานของ circuit นี้คล้ายกับว่านำเอา switch มาต่อเข้าเป็นแบบ series กัน ในลักษณะนี้เราสามารถจะ express switch ว่าอยู่ในลักษณะ ๒ State คือ ปิดและเปิด ถ้าขณะที่ switch ปิดถือว่าอยู่ที่ State 1 แต่ถ้า Switch เปิดให้ถือว่าอยู่ที่ State 0 จะเห็นว่า ถ้า Switch ทุกอันอยู่ที่ State 1 จะมี voltage ปรากฏอยู่ที่ output ซึ่งเรา express ว่าอยู่ที่ state 1 แต่ถ้าหากว่า switch อันหนึ่งอันใดอยู่ที่ state 0 แล้ว กระแสจะไม่สามารถผ่านไปถึง output ได้ เรา express ว่าอยู่ที่ state 0 จะเห็นว่าประโยชน์ของ AND gate ก็คือเป็น boolean binary ที่ทำหน้าที่เป็น logical addition ได้

สำหรับ Simulated program ใน DAS นี้ กำหนด switch ได้อย่างมาก ๕ อัน DAS จะสร้าง truth table ขึ้นตามหลักการที่กล่าวไว้ข้างบน สิ่งที่น่าสังเกตอย่างหนึ่งก็คือว่า ถ้าหากค่าของ switch ใดเป็น positive เราถือได้ว่า switch นั้นอยู่ที่ state 1 ตรงกันข้ามให้ถือว่าอยู่ที่ state 0

๓. NAND gate circuit นี้จะทำหน้าที่ตรงกันข้ามกับ AND gate ฉะนั้น ใน DAS, program นี้จะทำหน้าที่สร้าง truth table ที่มีค่าตรงกันข้ามกับ AND gate

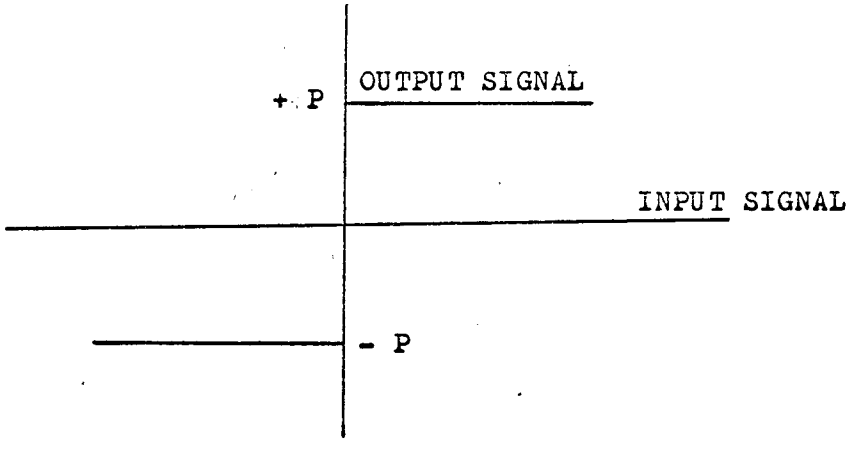
๔. OR gate input A และB ถ้าเรา express ในรูปของ Boolean Algebra แล้วจะเขียนได้เป็น A + B เพื่อความสะดวกและง่ายแก่การพิจารณา ให้พิจารณา switch เช่นเดียวกับใน AND gate แต่ circuit นี้ switch ต่อเป็นแบบ paralled state ที่เราจะให้ switch express นั้นรวมทั้ง output ก็เช่นเดียวกับที่กล่าวใน AND gate จะเห็นว่ากรณีนี้จะคล้ายกับ NAND GATE มากในเรื่องการสร้าง truth table แม้ว่า switch จะอยู่ใน state 0 คือ switch จะเปิดก็ตาม ถ้ามีเพียง switch เดียวที่อยู่ state 1 กระแสก็จะไหลผ่านไปยัง output ได้ output จะอยู่ที่ state 1 ตลอดจนกระทั่ง switch ทุกตัวอยู่ที่ state 0 เช่นเดียวกับใน simulated program กำหนดค่าของ switch ได้เพียง ๕ ชุดเท่านั้น หลักการสร้าง truth table ใน program คล้ายคลึงกับ AND gate

๕. NOR gate มีคุณสมบัติตรงข้ามกับ OR gate เพราะฉะนั้น truth table ที่สร้างขึ้นใน DAS จะมีค่าตรงกันข้ามกับ OR gate ทุก state

๖. NOT gate citcuit นี้มีความหมายว่า ถ้า proposition ที่กำหนดให้มีจริง proposition ของมันก็จะไม่จริง นั่นคือมันจะตรงกันข้ามกับ DAS จะทำหน้าที่ simulated characteristic อันนี้โดยที่รับ argument มาอันหนึ่ง ถ้า argument นั้นมากกว่า ๐ ค่าของมันจะมีค่าเป็น ๐ แต่ถ้า argument นั้นมีค่าเท่ากันหรือน้อยกว่า ๐ ค่าของมันก็จะมีค่าเป็น ๑.๐ เป็นต้น

๗. Offset subprogram นี้จะทำหน้าที่ในการ shiff signal ซึ่งกำหนดให้ใน ANALOG SUBROUTINE INSTRUCTION ประโยชน์ของมันก็คือ ช่วยในการ Plot ของ Plotter

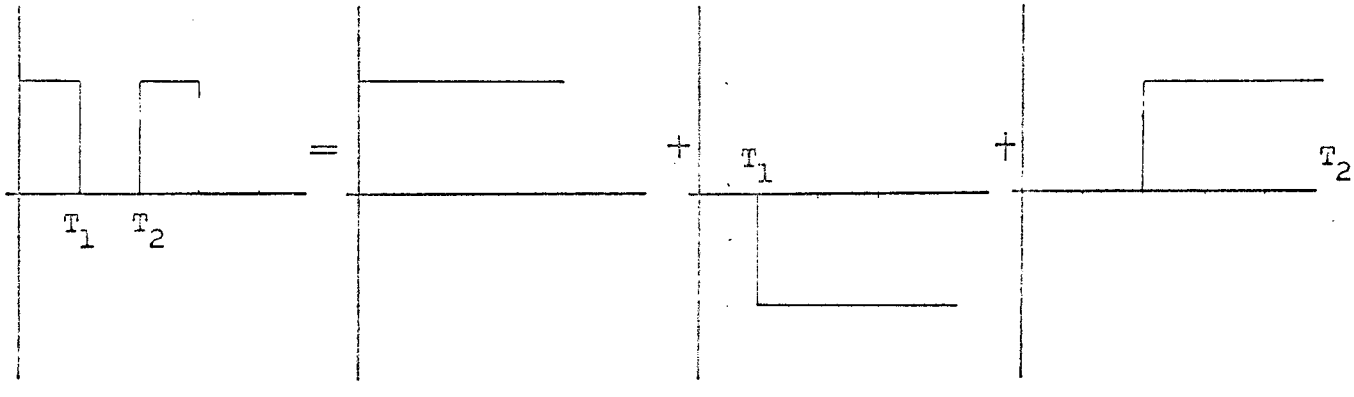
๔. Bang - Bang เนื่องจาก Digital computer เป็น automatic control device การ control ทุกอย่างจะต้องอาศัย Card data หรือ program ทั้งสิ้น ส่วนใน analog computer เป็นแบบกึ่ง automatic ฉะนั้น การปรับ voltage ให้ไปอยู่ที่ + หรือ - ย่อมขึ้นอยู่กับ switch ที่เราสามารถจะ control, subprogram นี้จะทำหน้าที่ simulate analog switch อันนี้ แทนที่จะใช้คนคอยควบคุมให้ switch ไปอยู่ที่ไหนตรงกันข้ามเราใช้ input signal คอยควบคุม เมื่อ input signal ของ element นี้ เป็น + ค่า Output signal ก็จะมีค่าเป็น + P ตรงกันข้าม ถ้า input signal ของ element นี้ เป็น - ค่า Output signal ก็จะมีค่าเป็น - P และถ้า input signal นี้ มีค่าเป็น 0, Output signal ก็มีค่าเป็น 0 ด้วย



รูปที่ ๔ Bang - Bang

๙. Clock Subprogram นี้ เหมือน circuit ทั้งใน analog computer และใน logical circuit, program นี้ได้ develope ไว้ เพื่อจุดประสงค์ที่จะให้เป็น timing device ซึ่งถ้าพิจารณาจากการ plot บน graph แล้วมันก็คือ scale factor ตามแกน x ถ้าค่านี้มีค่าเป็น ๑ ก็คือ scale ของ time increment ที่กำหนด แต่เมื่อไรก็ตามที่เราต้องการในค่าที่มี interval กว้างกว่า scale นี้ก็จะต้องขยาย ฉะนั้น subprogram นี้ จึงมีความสำคัญมากที่สุดใน ANALOG SUBRORTINE ที่เขียนขึ้น

๑๐. Delay Element นี้ มีไว้เพื่อที่จะให้ signal ที่ส่งมาช้าลงกว่าที่กำหนด สำหรับใน circuit จริง ๆ นั้น เวลาที่ delay ไปนั้น ย่อมขึ้นอยู่กับ cutoff frequency ที่ design ไว้ สำหรับใน program ของเรา signal ที่มานี้จะช้ากว่ากันมากน้อยเท่าไรย่อมขึ้นอยู่กับความต้องการของ programmer ในการกำหนดให้ program นี้มีประโยชน์มาก แม้กระทั่งในการ generate square wave หรือ pulse ก็ตาม เราก็สามารถที่จะใช้ program นี้ generate ได้ ดังตัวอย่างต่อไปนี้.-



รูปที่ ๑๒ หลักการที่ใช้ relay ในการ generate square wave.

๑๑. Relay ทั่ว ๆ ไปแล้ว relay จะทำหน้าที่เป็นตัว control switch ตาม signal ที่ส่งให้กับ relay เมื่อไรก็ตามที่ signal ที่ส่งมานี้มีค่าเกิน limit ที่ relay set ไว้แล้ว relay ก็จะทำงานทันที แต่ถ้า signal ที่ส่งมามีความแรงไม่พอ relay ก็จะไม่ทำงาน ใน DAS ของเรานี้ relay จะทำงานด้วย input signal ๒ อัน ซึ่งมี control signal เป็นตัว control ให้ signal ไหนผ่านไปได้ หรือว่าจะไม่ให้ signal ทั้งสองผ่านไปเลย เมื่อ control signal นั้นเป็น ๐

๑๒. Type output subprogram อันนี้จะทำหน้าที่ควบคุมให้ printer พิมพ์ค่าที่ต้องการออกมาในรูปที่ถูกต้องตามต้องการ format ที่ใช้ในการพิมพ์ค่าเหล่านี้ ได้แก่ F 13.8 ซึ่งเป็นค่า Floating point เท่านั้น และจะสามารถพิมพ์ค่า variable ต่าง ๆ ใน ๑ แถวได้อย่างมาก ๑๐ ตัว ใน Type output subroutine นี้ ค่า argument ตัวแรกในวงเล็บก็คือค่าที่บ่งไว้ว่าจะให้พิมพ์กี่ตัว ส่วน argument ตัวต่อ ๆ ไปนั้นเป็นชื่อของ output variable นั่นเอง.