## References

1. Edward Wilson Kimbark, "Power System Stability", Vol 1-3.

2. William D. Stevenson, "Power System Analysis".

3. P.M. De Russo, "State Vanables for Engineers".

4. Brennan, R.D. and R.N. Linebarger, "A survey of digital simulation" Simulation 1964.

5. Royce Beckett, James Hurt, "Numerical Calculation and Algorithms".

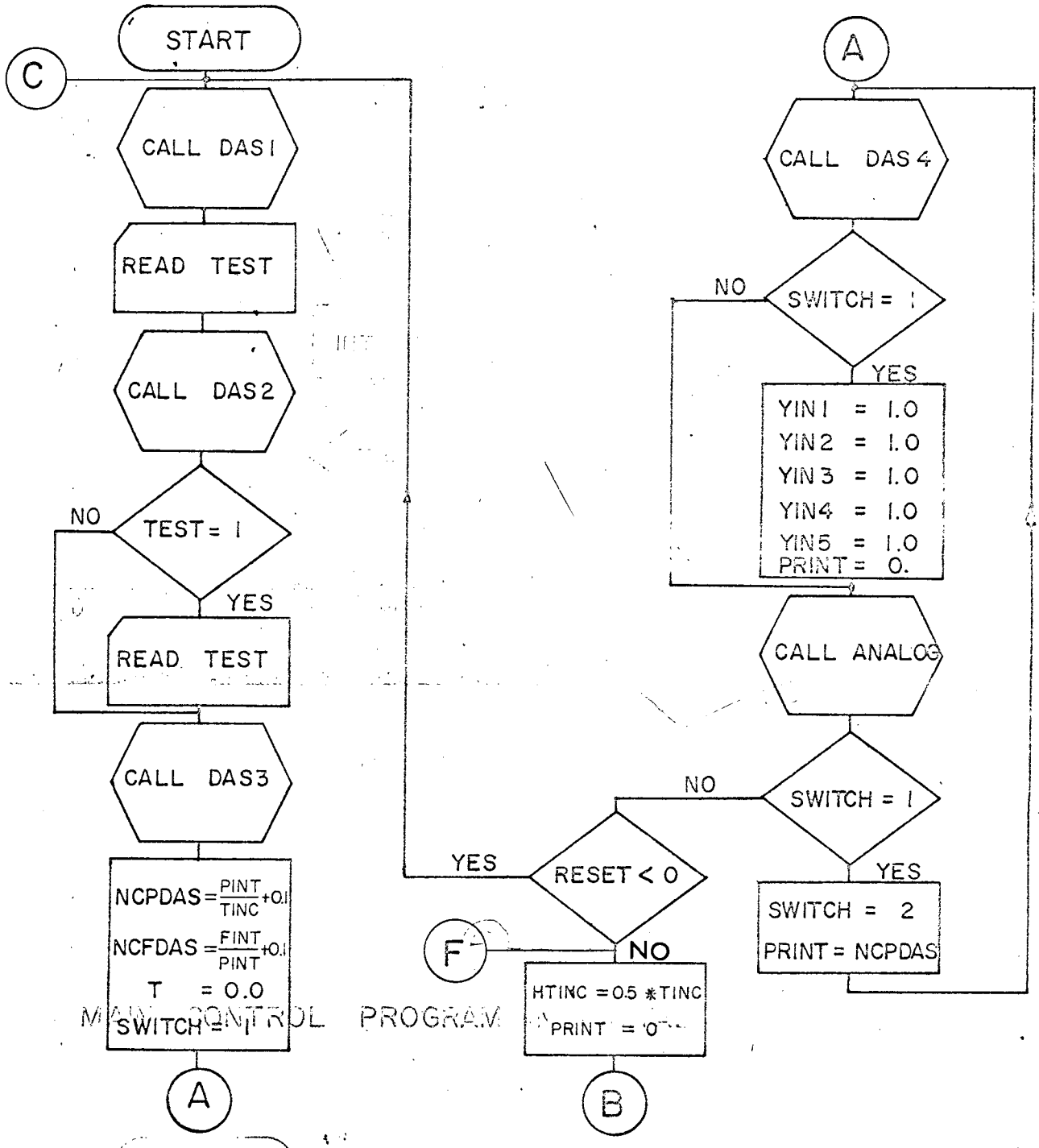
6. M.G. Savadori, M.L. Baron, "Numerical Methods in Engineering".

7. Jacob Millman, H. Taub, "Pulse, Digital, and Switching waveforms".

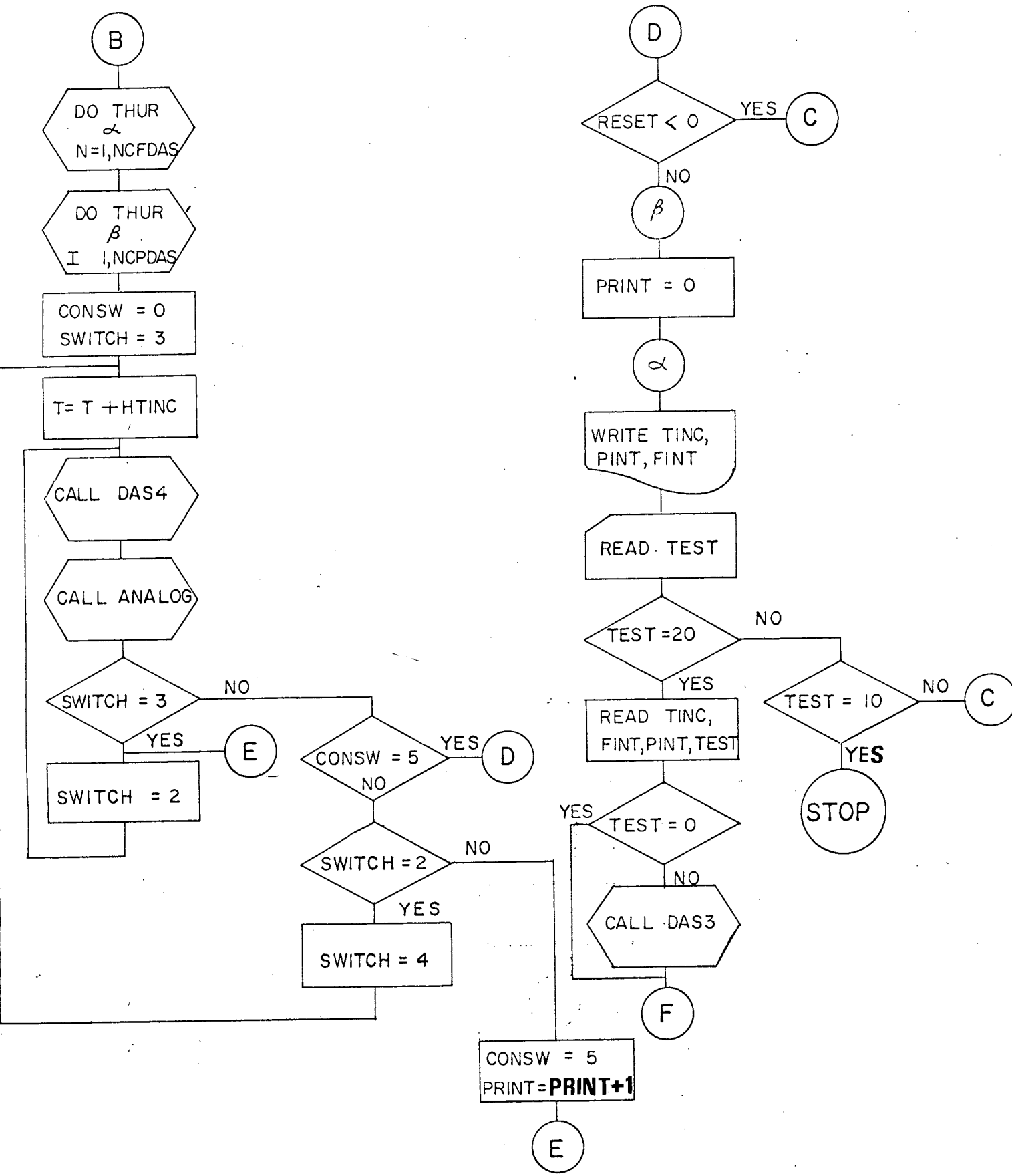
# Appendix A

## Flow Chart

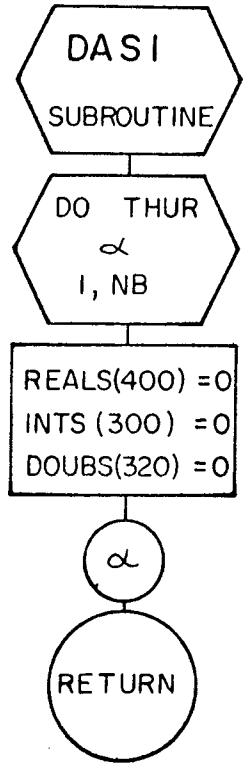
# MAIN CONTROL PROGRAM IN DIGITAL ANALOG SIMULATION



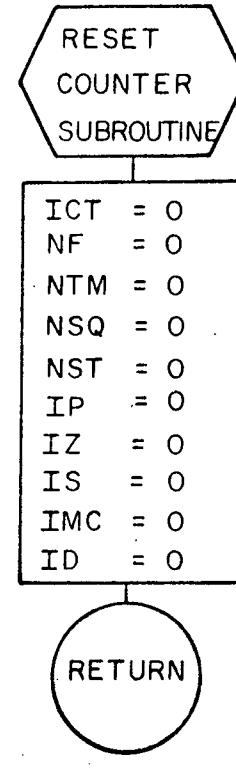

MAIN CONTROL PROGRAM


B
DO THUR α N=1,NCFDAS
DO THUR β I 1,NCPDAS
CONSW = 0
SWITCH = 3
T= T +HTINC
CALL DAS4
CALL ANALOG
SWITCH = 3
NO
YES
E
SWITCH = 2
CONSW = 5
YES
D
NO
SWITCH = 2
NO
YES
SWITCH = 4
CONSW = 5
PRINT=PRINT+1
E
D
RESET < 0
YES
C
NO
β
PRINT = 0
α
WRITE TINC, PINT, FINT
READ TEST
TEST =20
NO
YES
READ TINC, FINT, PINT, TEST
TEST = 10
NO
C
YES
STOP
YES
TEST = 0
NO
CALL DAS3
F

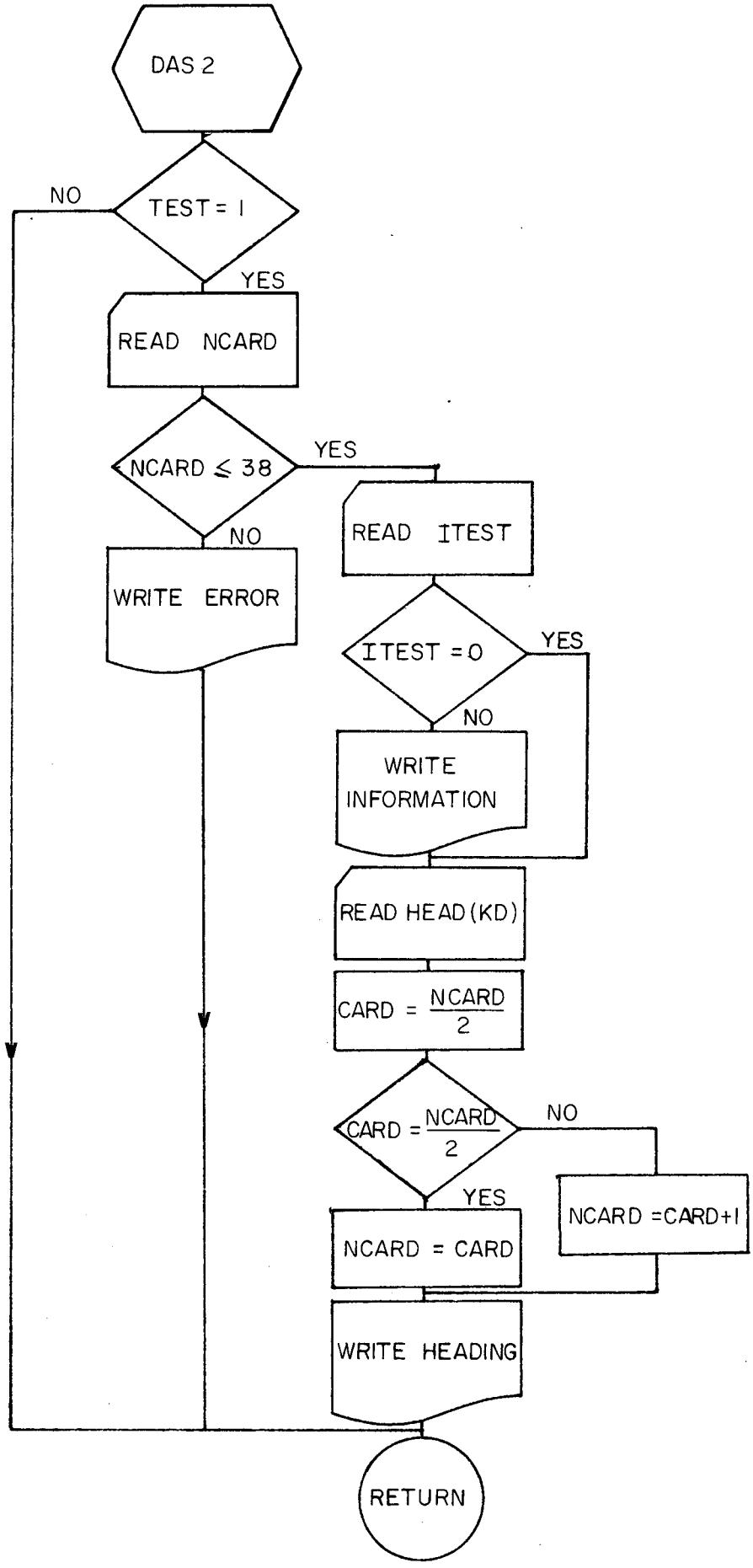
## SUBROUTINE DASI INITIALIZATION




## RESET COUNTER SUBROUTINE DAS4

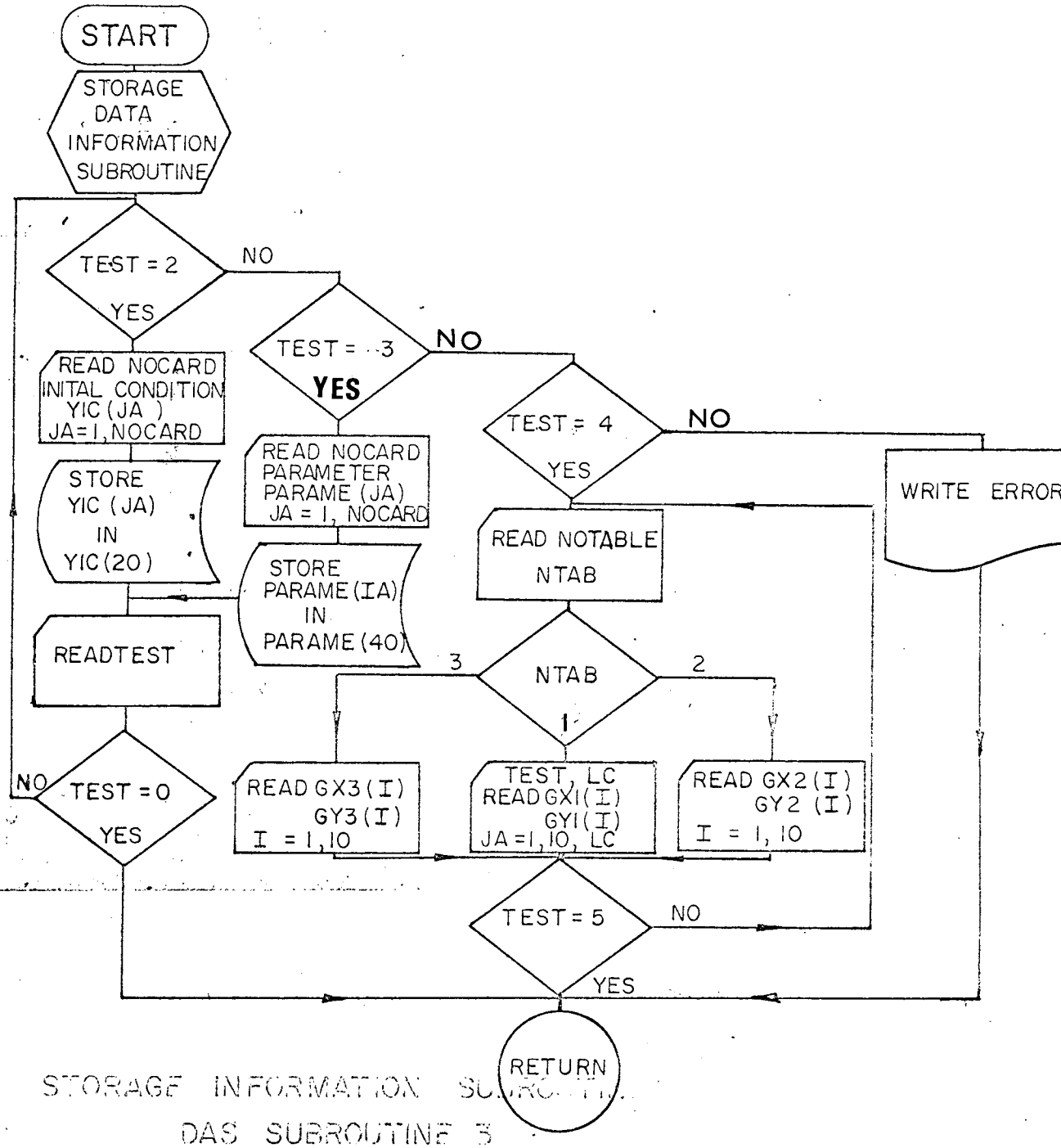


# SUBROUTINE DAS2 HEADING



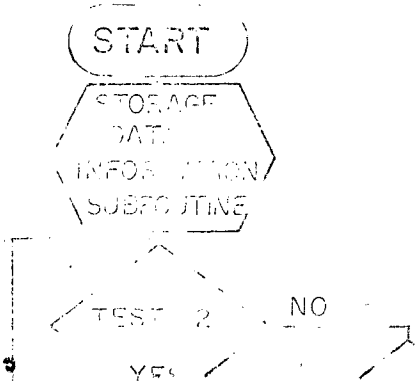
# STORAGE INFORMATION SUBROUTINE

## DAS SUBROUTINE 3

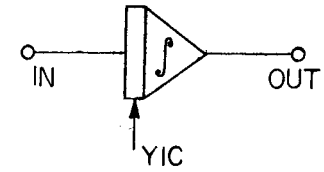


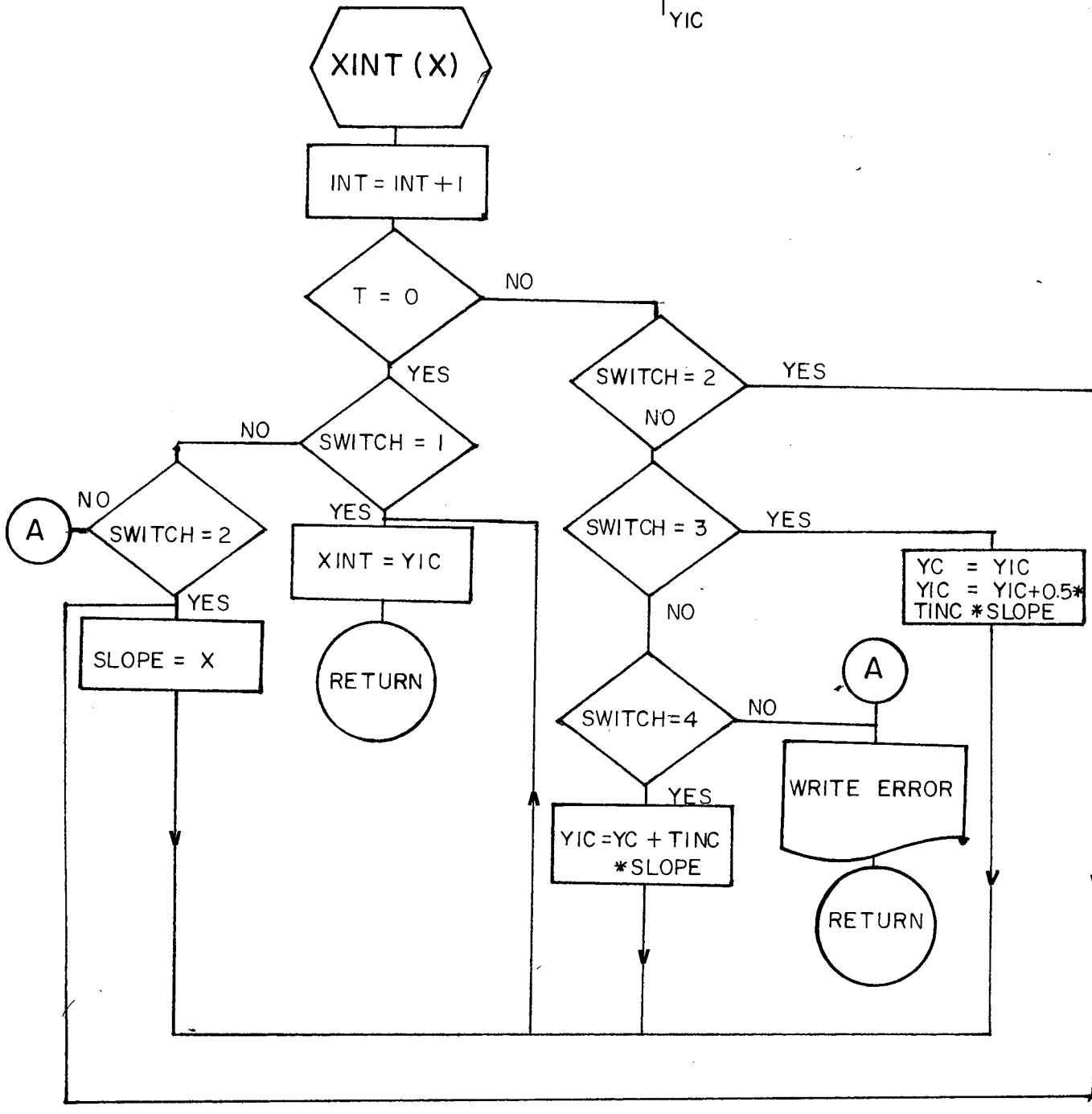

STORAGE INFORMATION SUBROUTINE

DAS SUBROUTINE 3

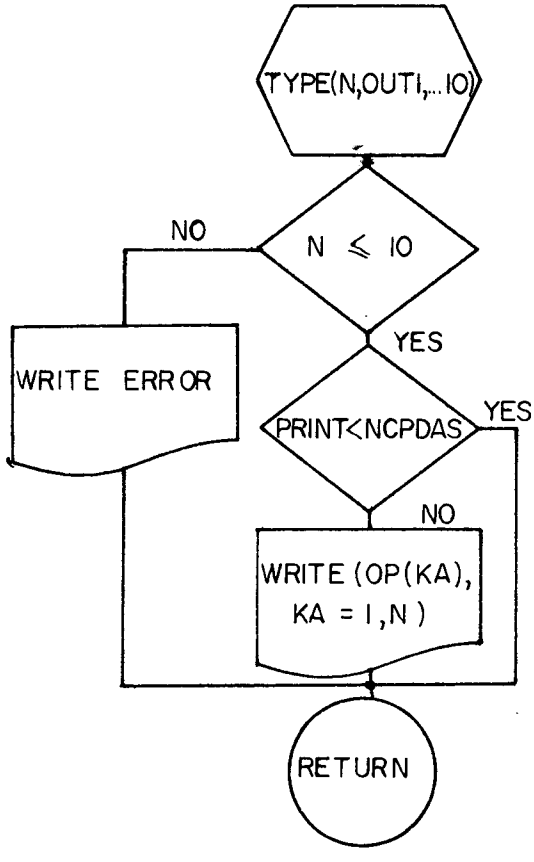


INTEGRATOR

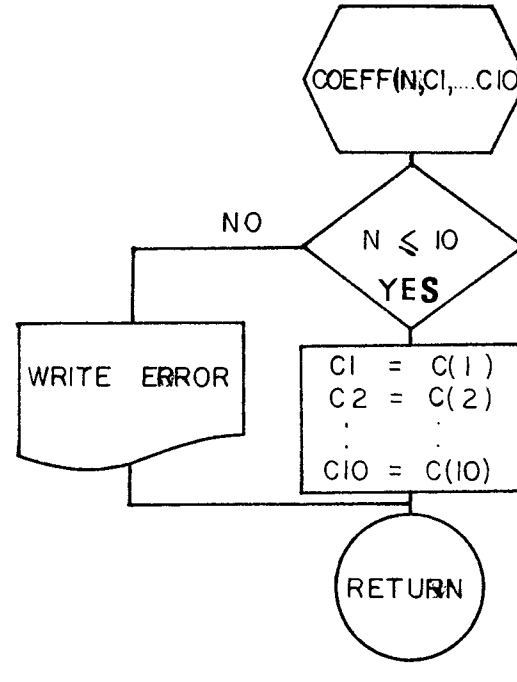

IN
OUT
YIC


XINT (X)
INT = INT + 1
T = 0
NO
YES
SWITCH = 1
NO
YES
SWITCH = 2
NO
A
YES
SLOPE = X
XINT = YIC
RETURN
SWITCH = 2
YES
NO
SWITCH = 3
YES
YC = YIC
YIC = YIC+0.5*
TINC *SLOPE
NO
A
SWITCH=4
NO
YES
YIC =YC + TINC
*SLOPE
WRITE ERROR
RETURN

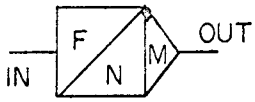
## TYPE OUTPUT

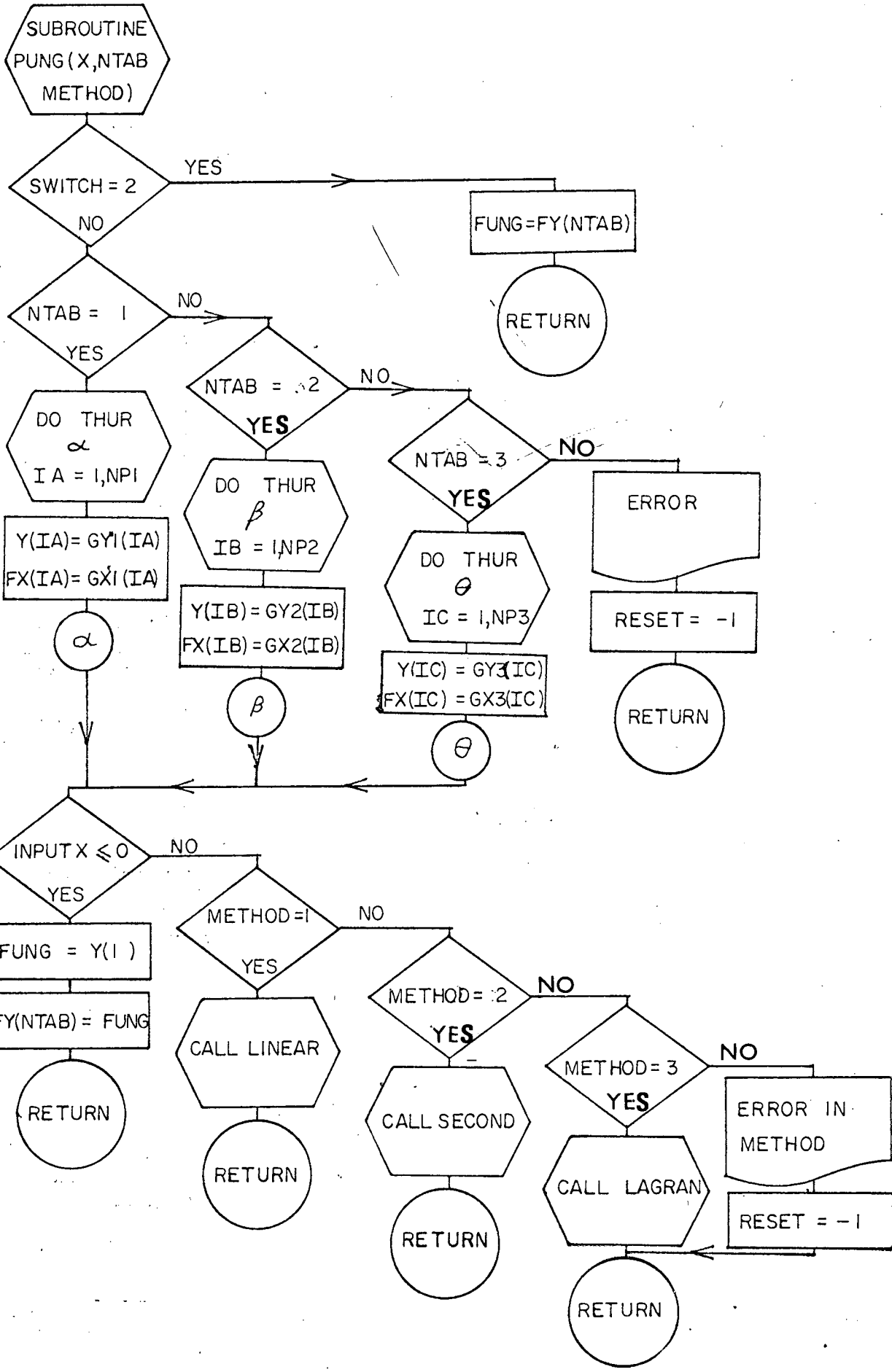



## COEFFICIENT SUBROUTINE

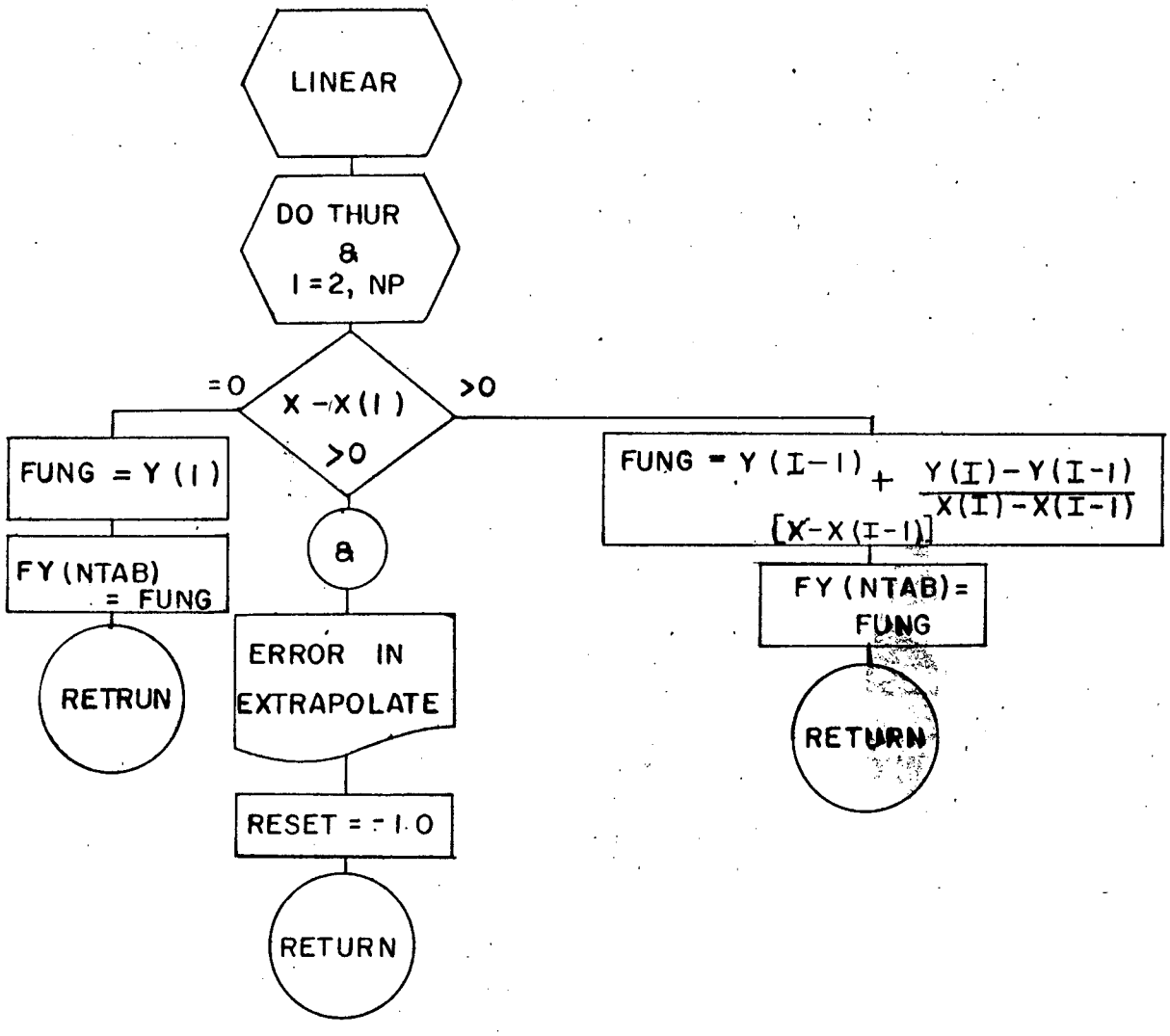


# MAIN FUNCTION GENERATOR SUBROUTINE

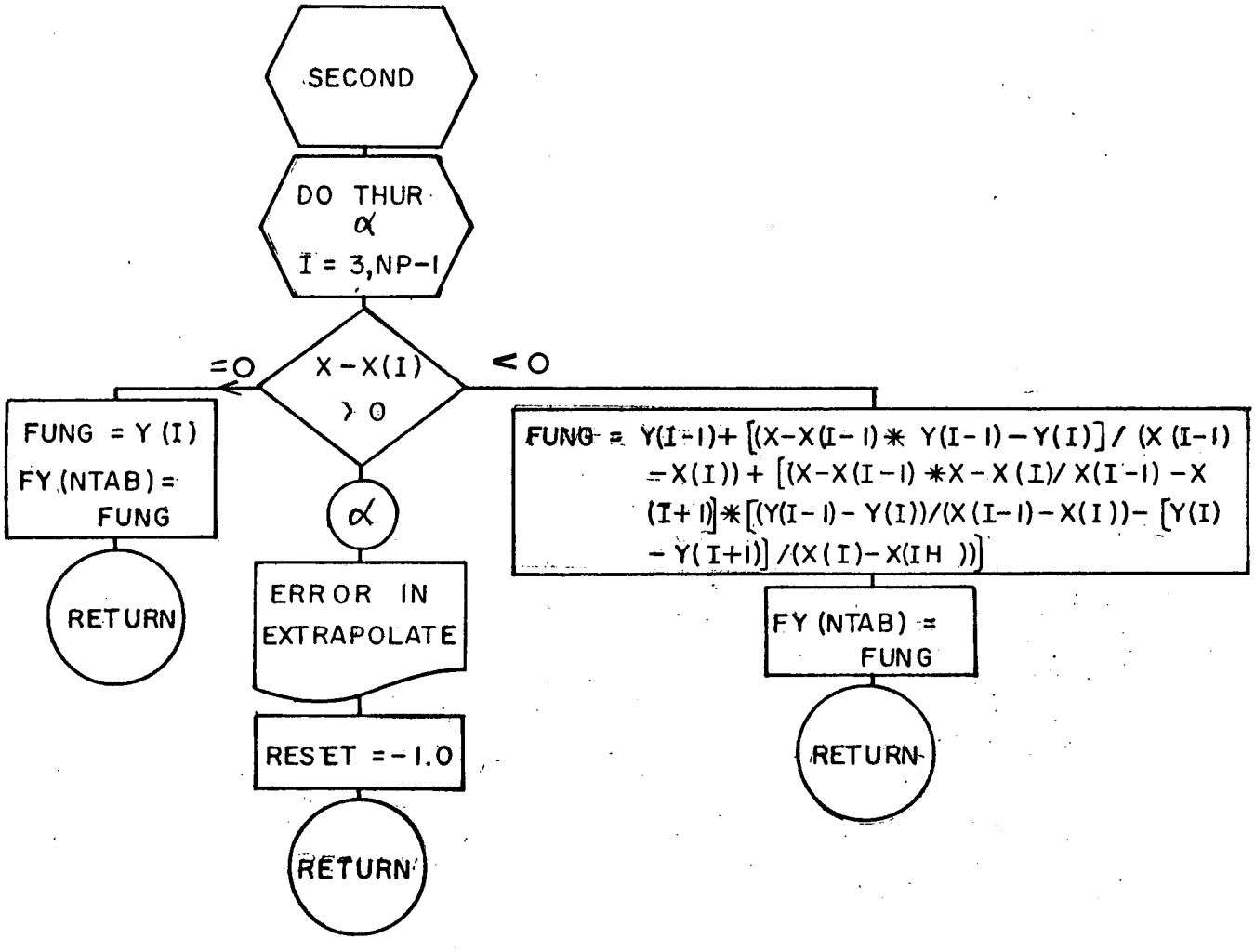


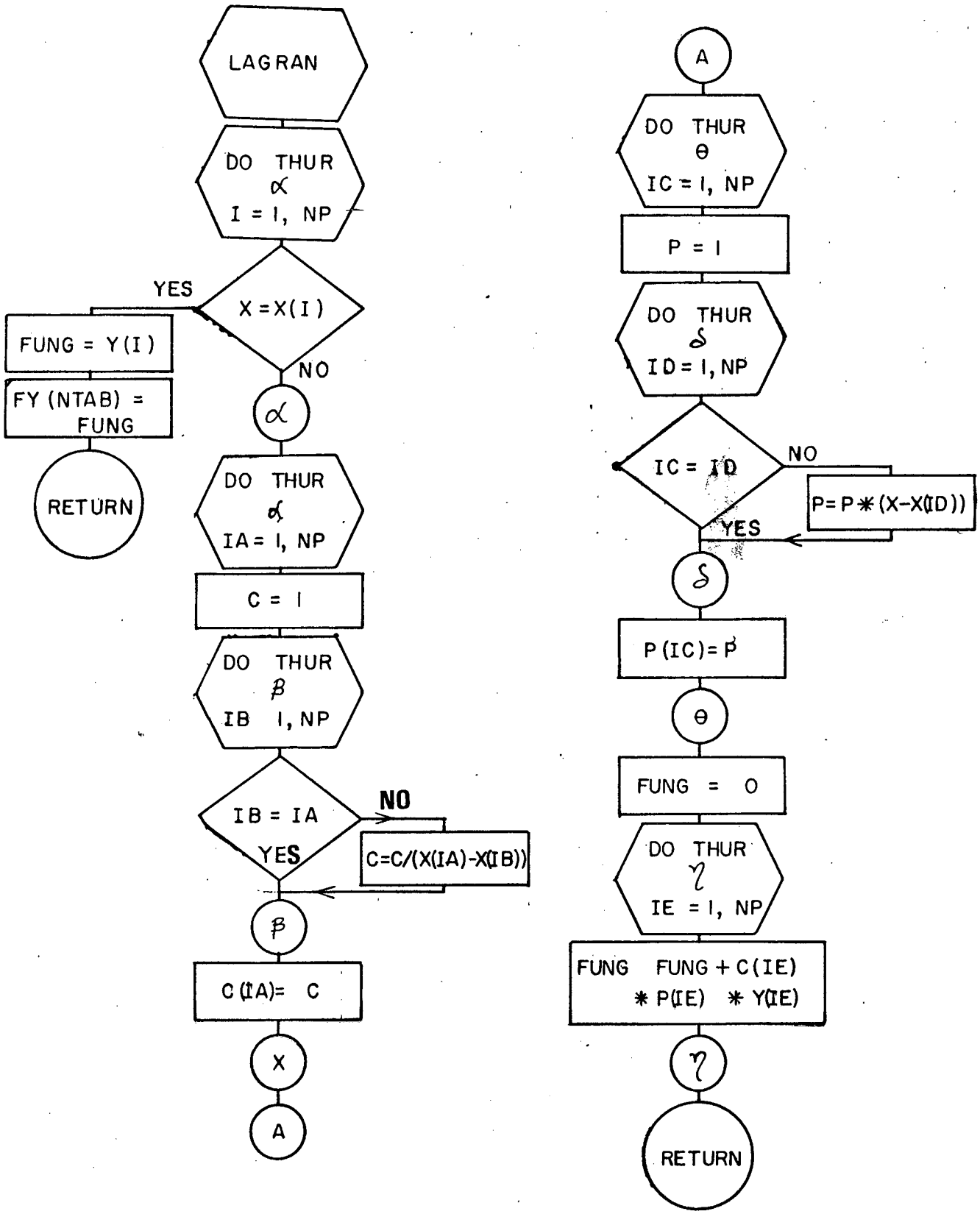
# Function Generator
# (S1)



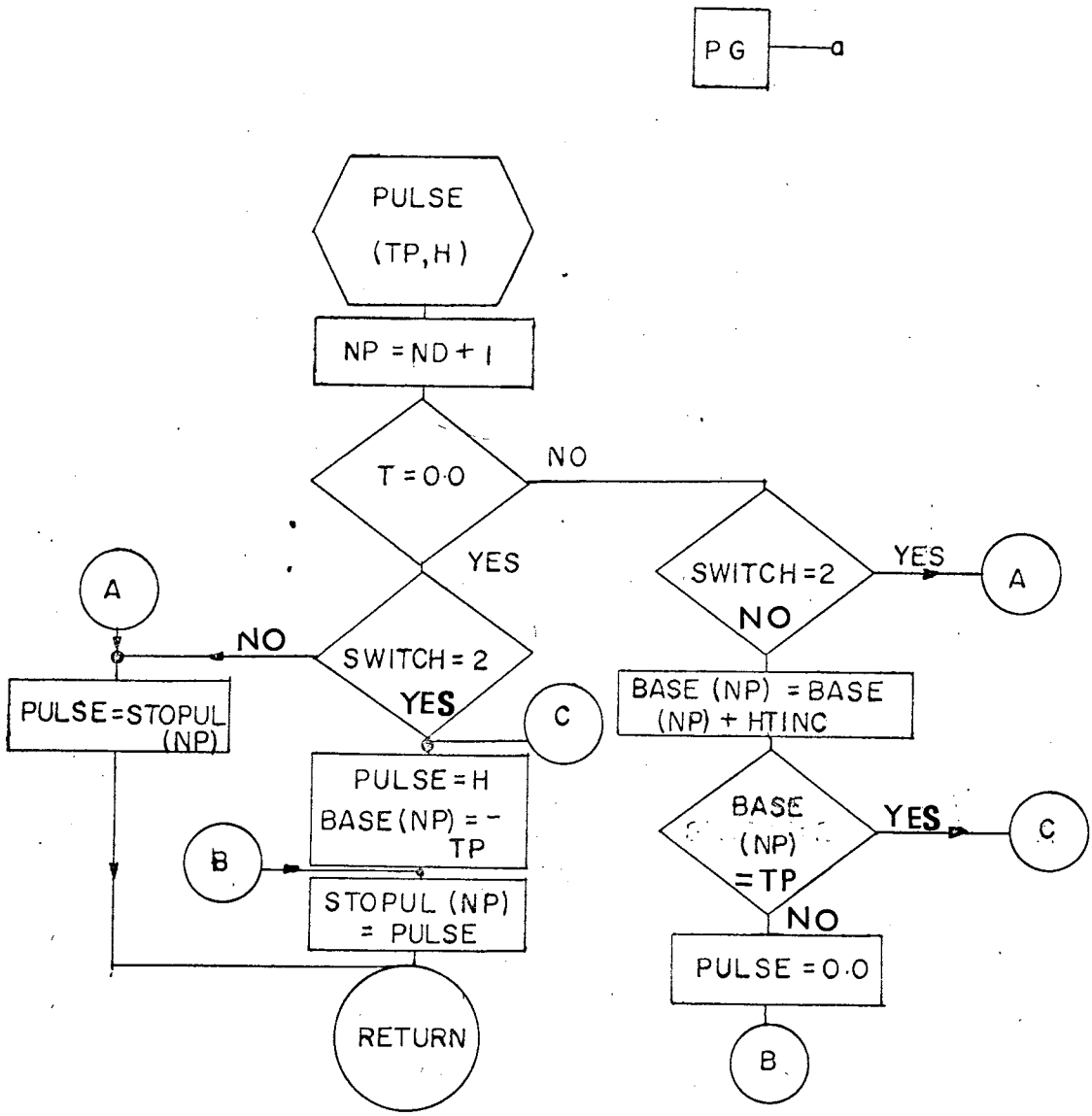
# Function Generator
# (S2)



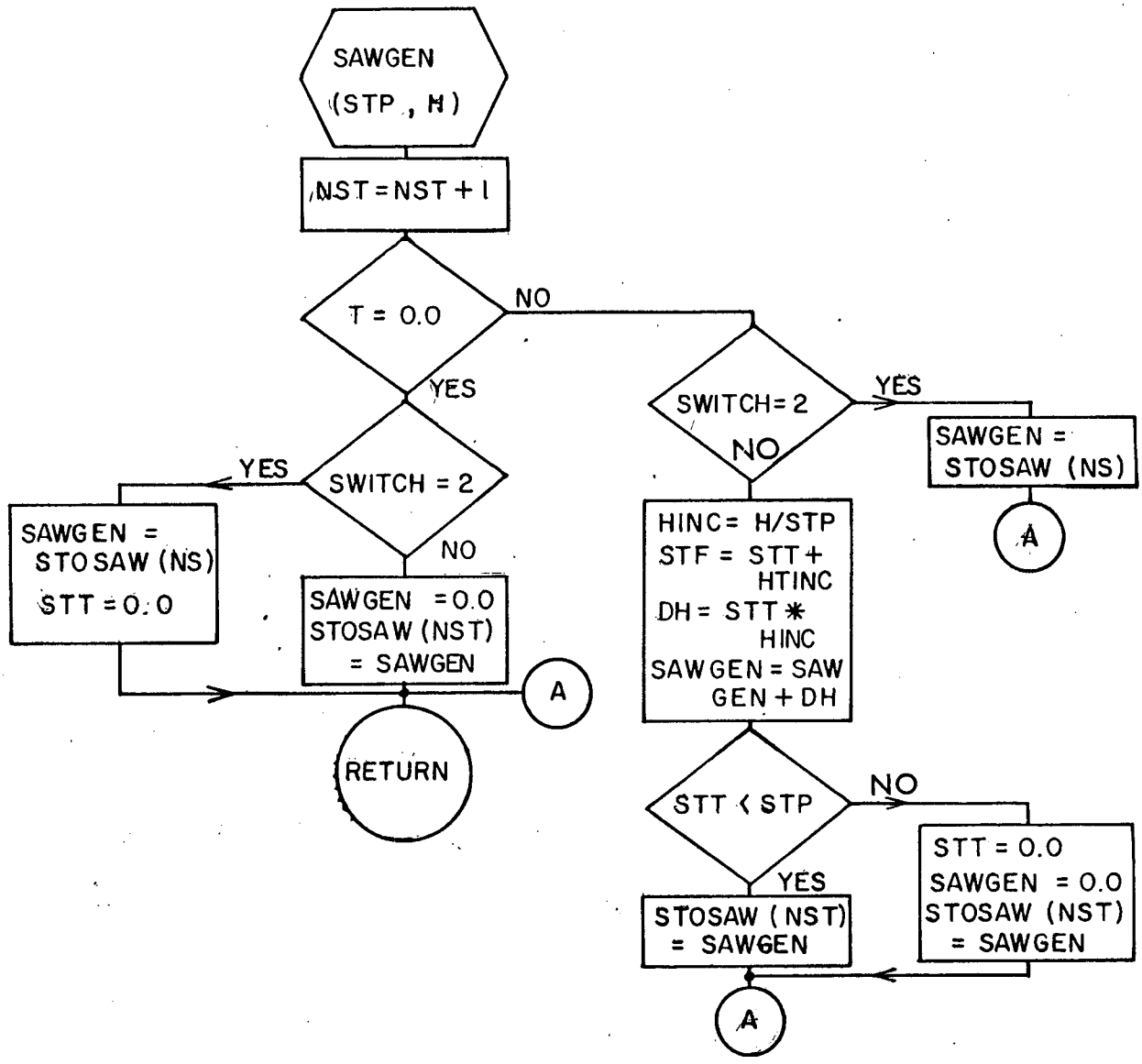
# Function Generator
# (S3)



# Pulse Generator



# SAWTOOTH GENERATOR

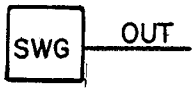

STP = SAWTOOTH TIME PERIOD

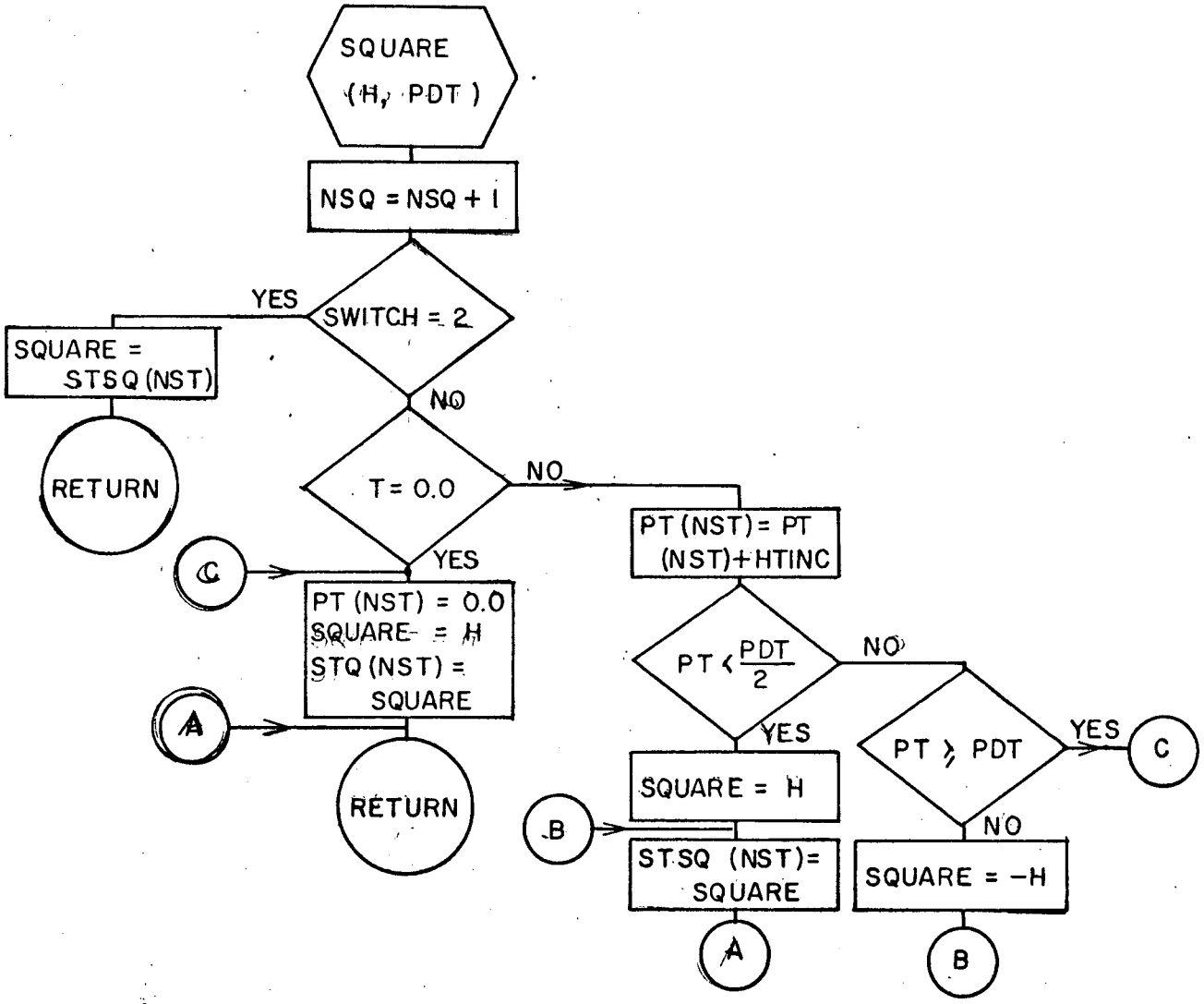
H = HEIGHT

NST = NO. OF SAWTOOTH

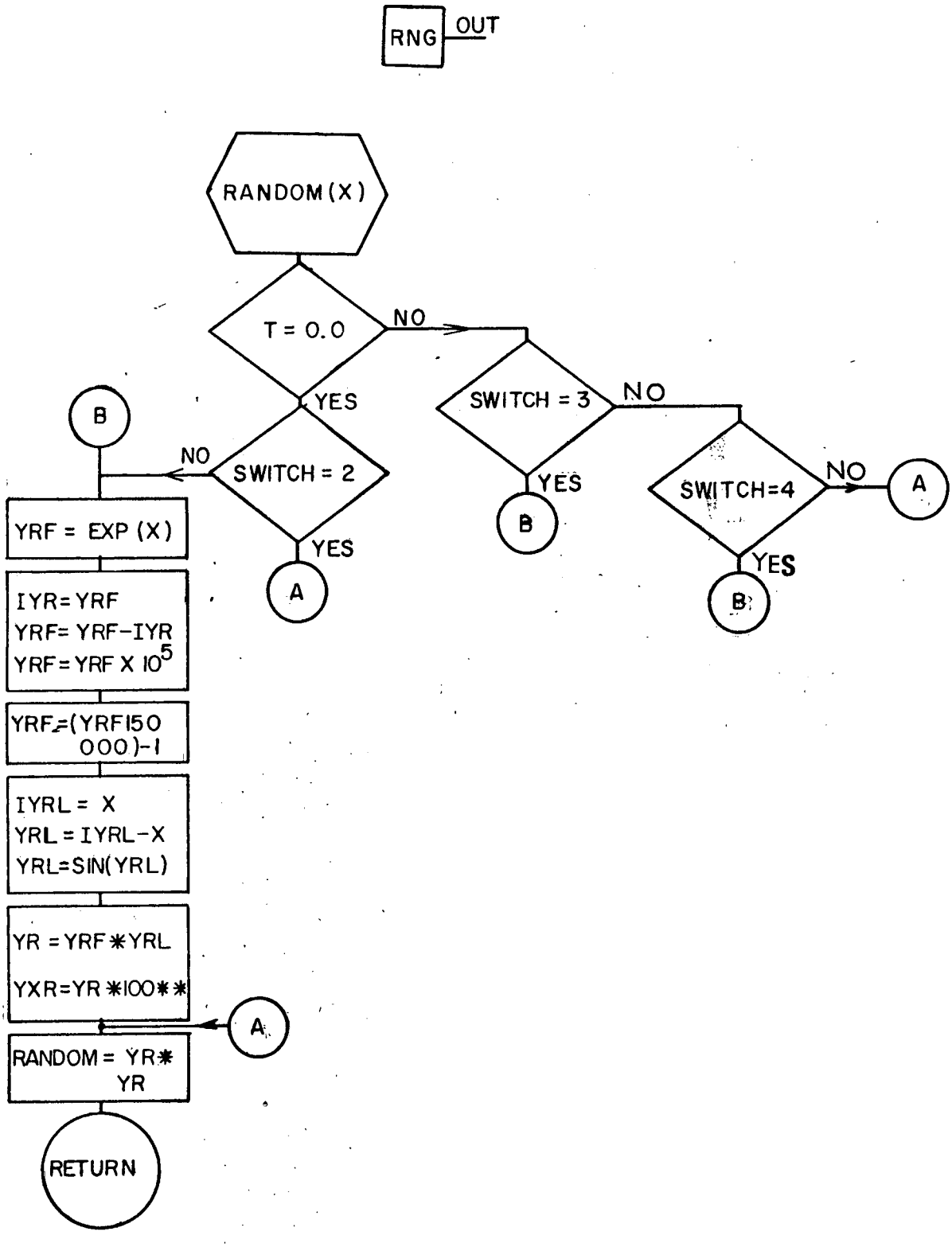
STT = SAWTOOTH TIME INCREMENT

# SQUARE WAVE GENERATOR

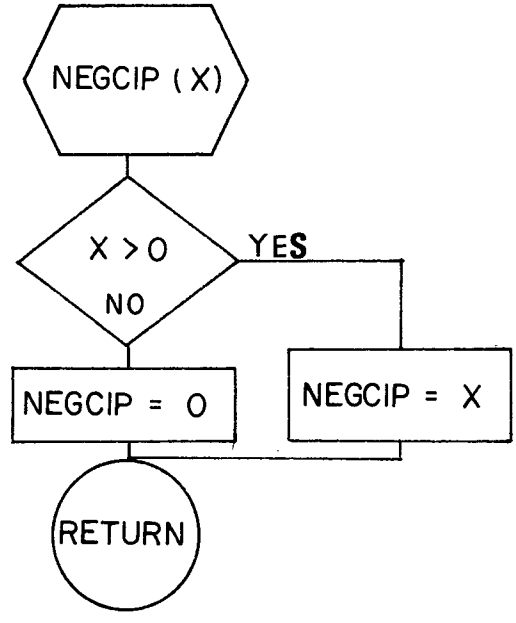


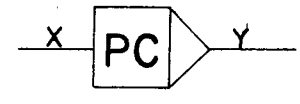
# RANDOM NOISE GENERATOR

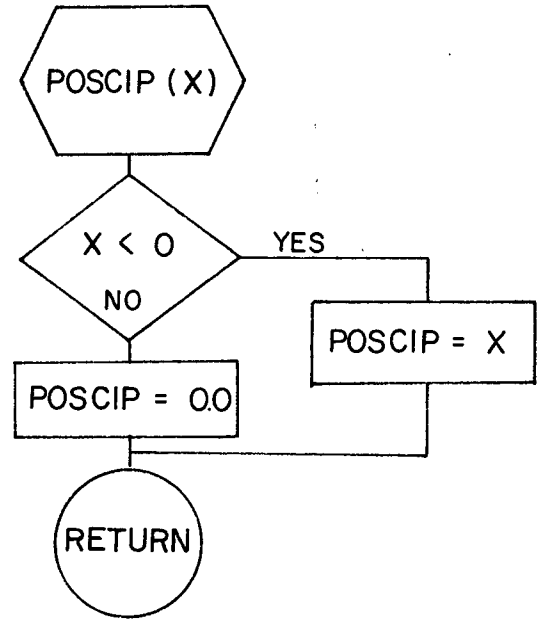


## NEGATIVE CLIPPER

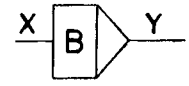



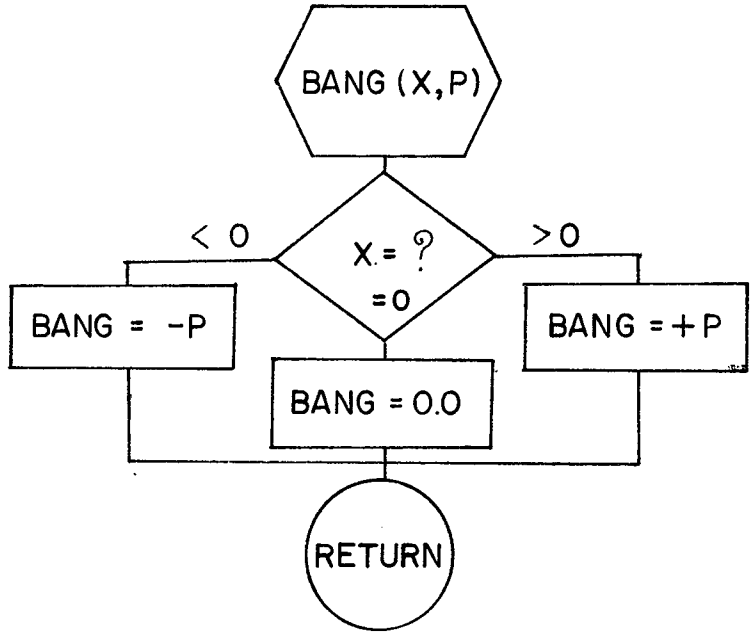

## POSITIVE CLIPPER



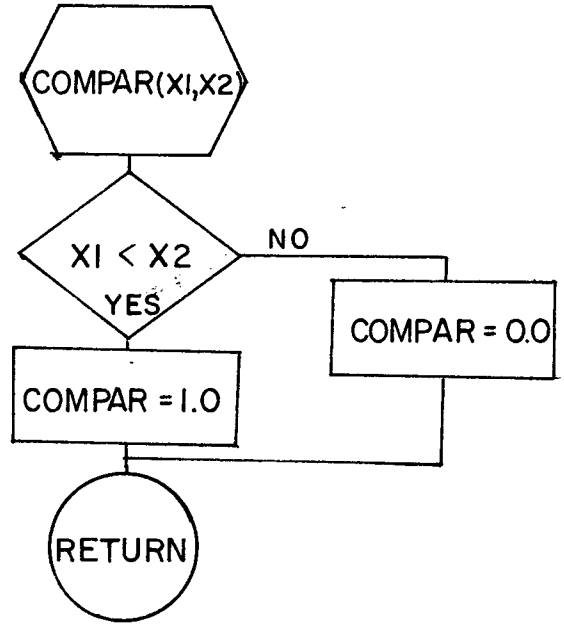



## BANG - BANG

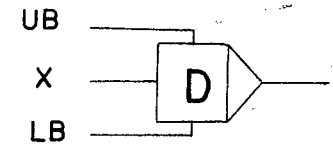


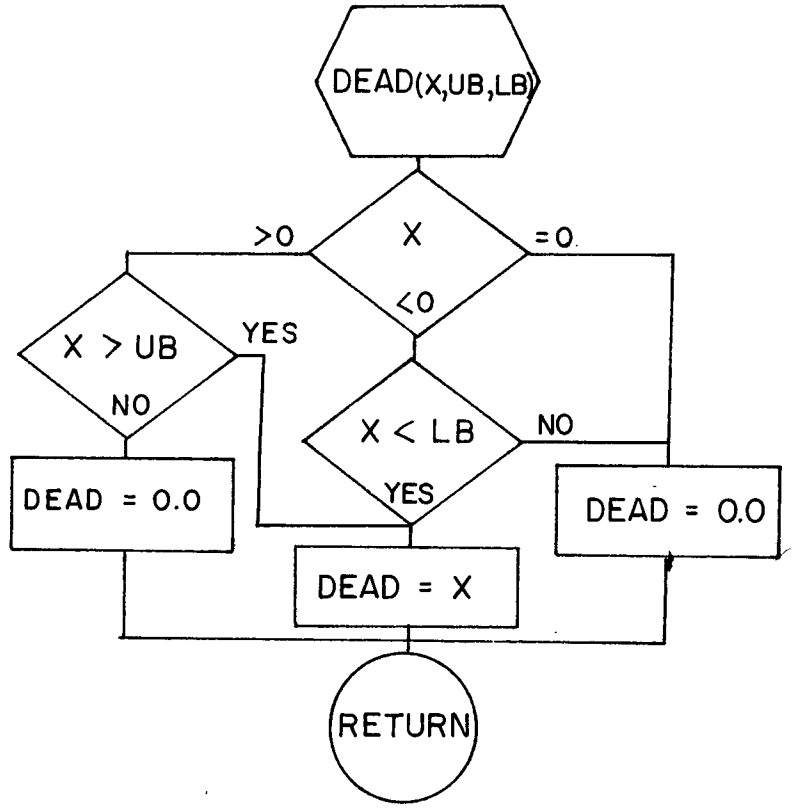




## COMPARATOR

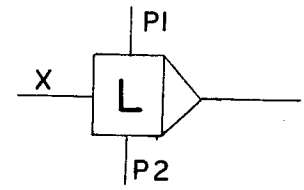



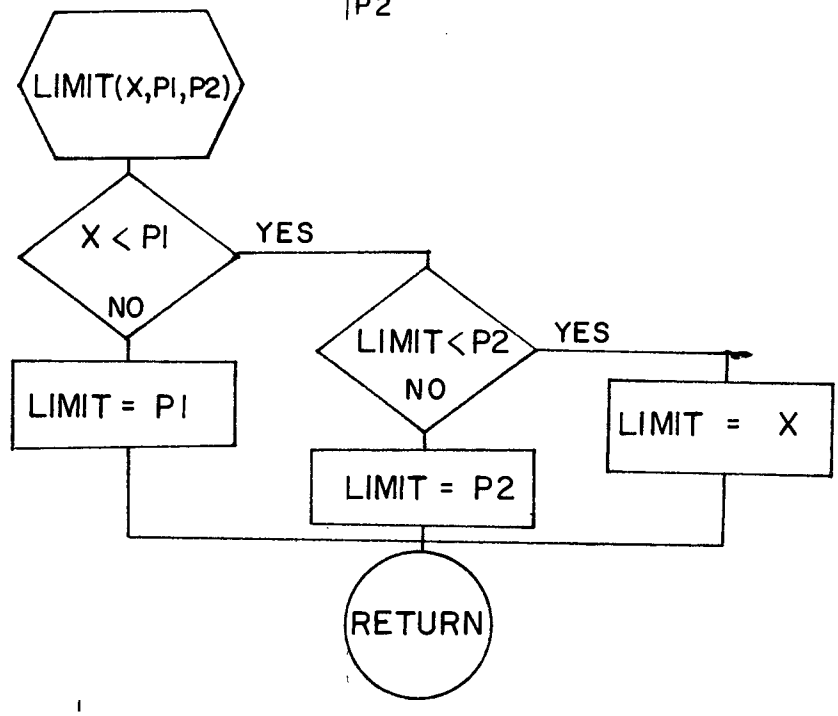
## DEAD SPACE



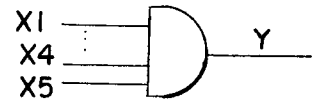



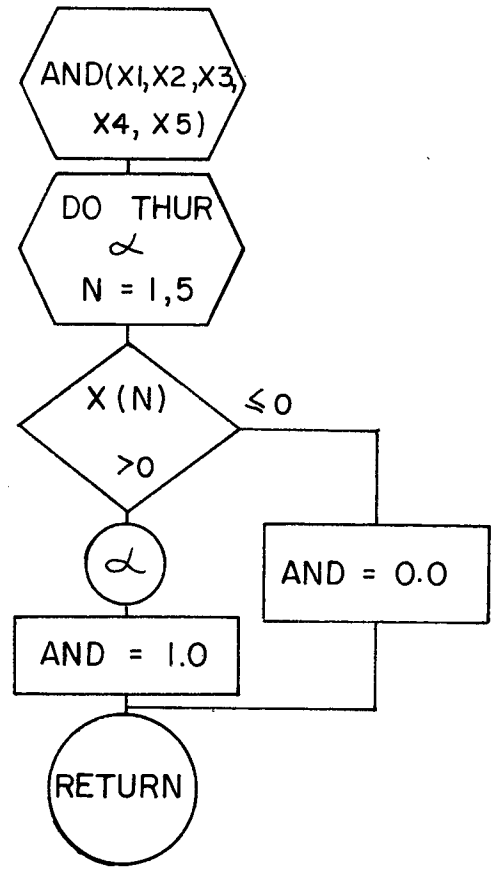

## LIMITER



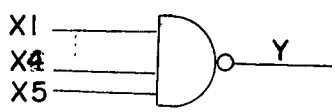



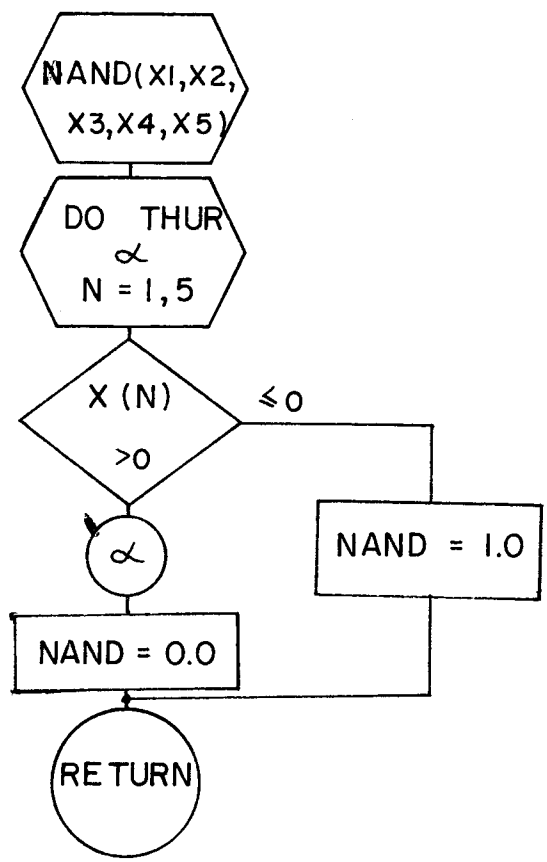
## AND GATE



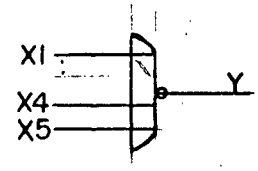



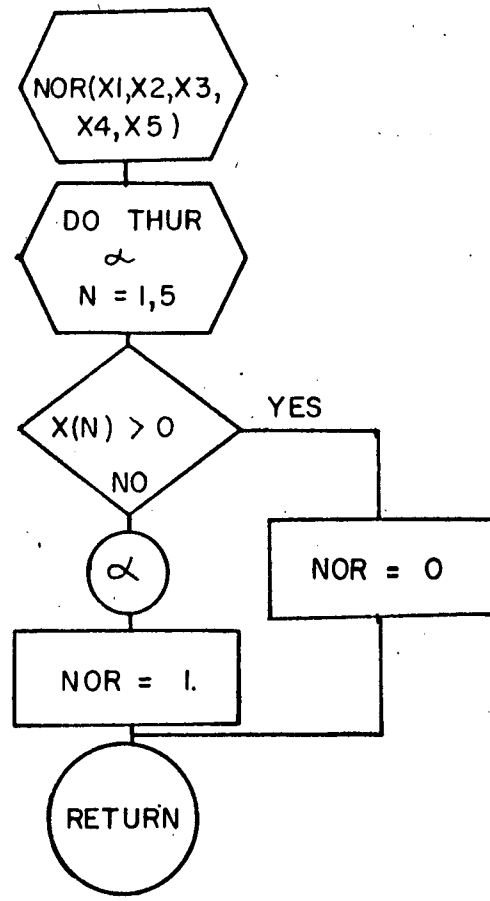

## NAND GATE



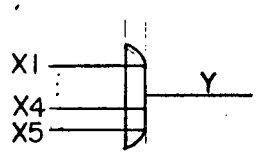



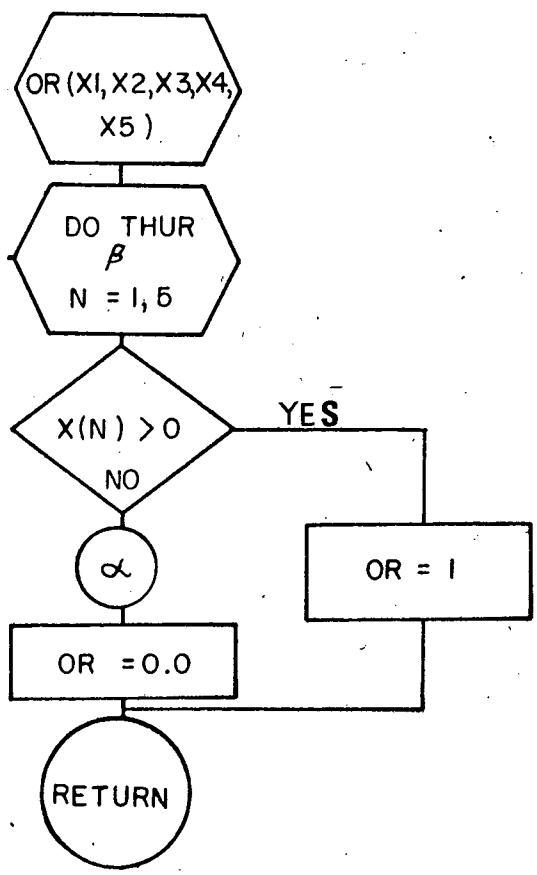
## NOR GATE



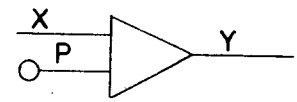



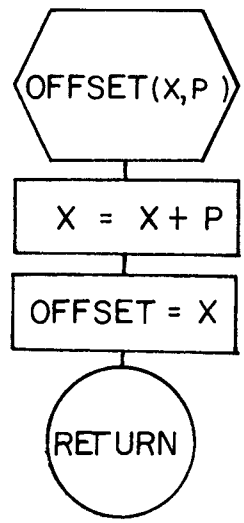

## OR GATE



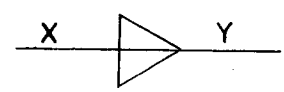



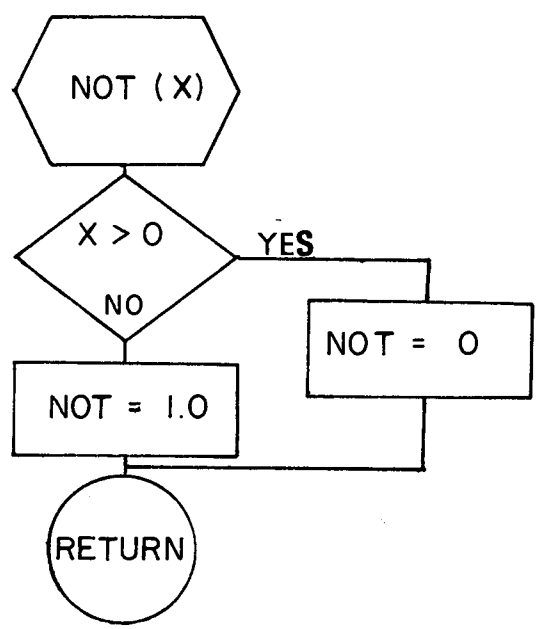
## OFF SET



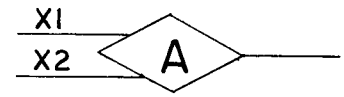



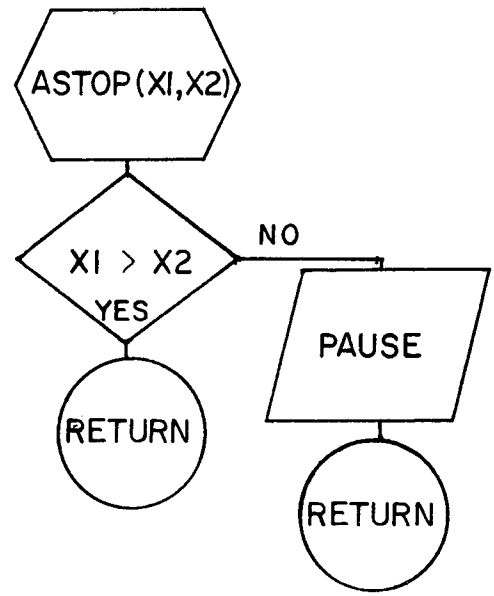

## NOT GATE



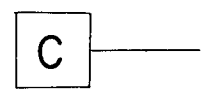



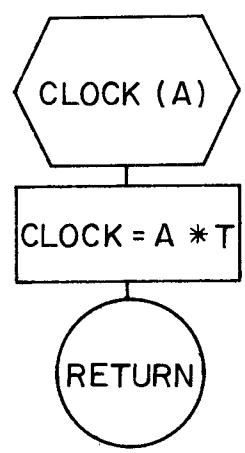
AUTOMATIC STOP



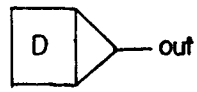



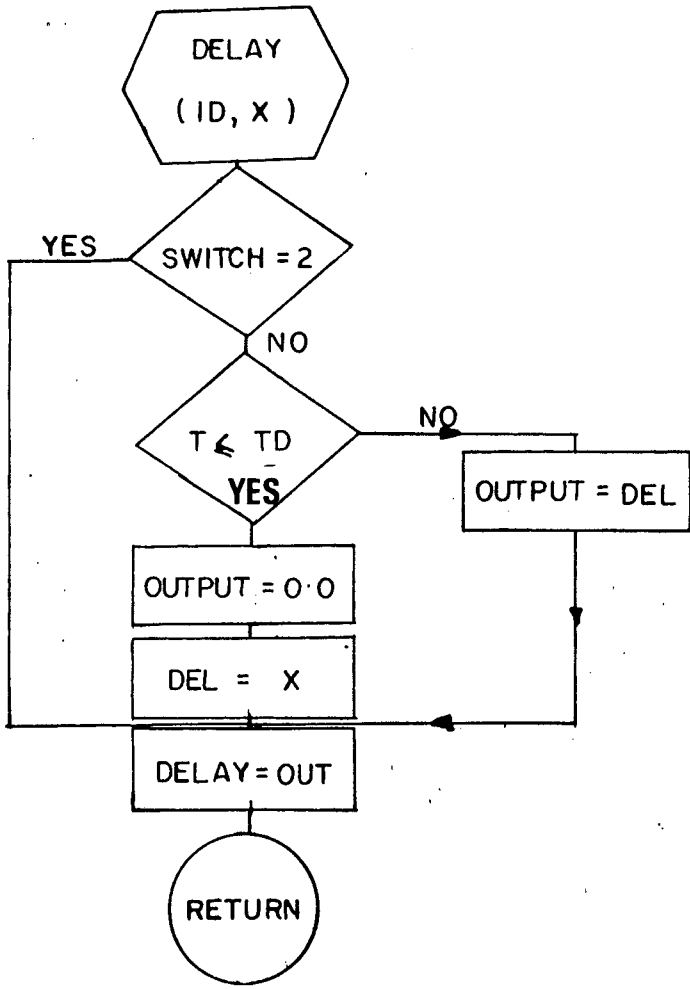

CLOCK



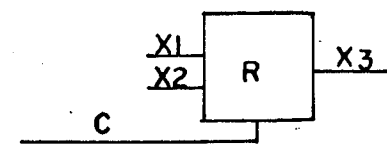



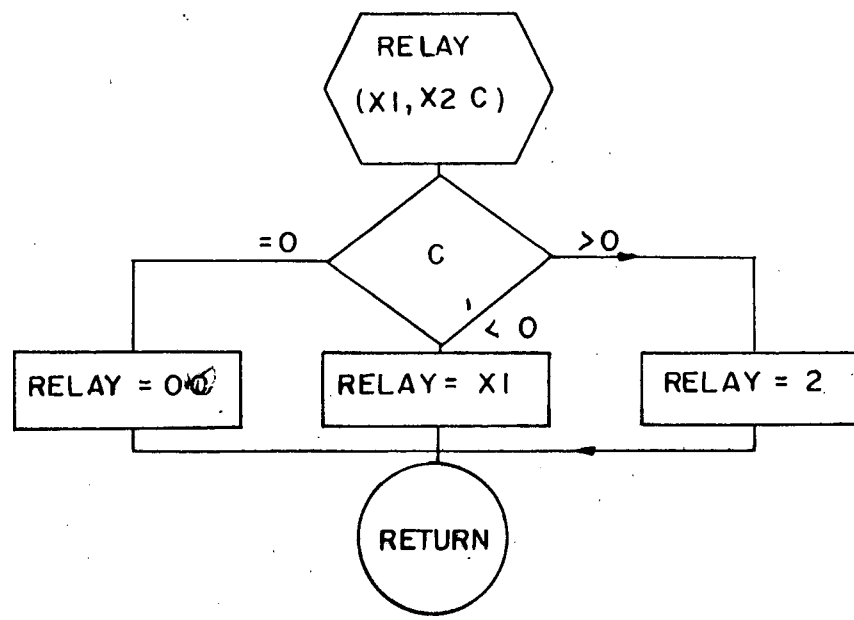
# Delay


D
out


DELAY
( ID, X )
SWITCH = 2
YES
NO
T ≤ TD
YES
NO
OUTPUT = DEL
OUTPUT = 0·0
DEL = X
DELAY = OUT
RETURN

# Relay


X1
X2
R
X3
C


RELAY
(X1, X2 C)
C
=0
>0
< 0
RELAY = 0·0
RELAY = X1
RELAY = 2
RETURN

# Appendix B

## Program

```
C MAIN CONTROL IN DIGITAL-ANALOG SIMULATION
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),TEST,SWITCH,CONSW,PRINT,RESET
      COMMON DOUBS,REALS,INTS
      DIMENSION YIN(5)
      EQUIVALENCE (REALS(1),T)
      EQUIVALENCE (REALS(62),TINC)
      EQUIVALENCE (REALS(146),PINT)
      EQUIVALENCE (REALS(147),FINT)
      EQUIVALENCE (REALS(148),YIN(1))
      EQUIVALENCE (REALS(153),HTINC)
      EQUIVALENCE (YIN(1),YIN1)
      EQUIVALENCE (YIN(2),YIN2)
      EQUIVALENCE (YIN(3),YIN3)
      EQUIVALENCE (YIN(4),YIN4)
      EQUIVALENCE (YIN(5),YIN5)
      EQUIVALENCE (INTS(2),SWITCH)
      EQUIVALENCE (INTS(3),RESET)
      EQUIVALENCE (INTS(10),TEST)
      EQUIVALENCE (INTS(11),NCPDAS)
      EQUIVALENCE (INTS(12),NCFDAS)
      EQUIVALENCE (INTS(13),CONSW)
      EQUIVALENCE (INTS(14),N)
      EQUIVALENCE (INTS(15),I)
      EQUIVALENCE (INTS(25),PRINT)
    1 CALL DAS1
      READ(1,2) TEST
    2 FORMAT(I3)
      CALL DAS2
      IF(TEST-1) 4,3,4
    3 READ(1,2) TEST
    4 CALL DAS3
      NCPDAS=PINT/TINC+0.1
      NCFDAS=FINT/PINT+0.1
      T=0.0
      SWITCH=1
    5 CALL DAS4
      IF(SWITCH-1) 8,6,8
    6 DO 7 NA=1,5
      YIN(NA)=1.0
    7 CONTINUE
      PRINT = 0
    8 CALL ANALOG(YIN1,YIN2,YIN3,YIN4,YIN5)
      IF(SWITCH-1) 10,9,10
    9 SWITCH=2
      PRINT = NCPDAS
      GO TO 5
   10 IF(RESET) 1,11,11
   11 HTINC=0.5*TINC
   50 PRINT = 0
      DO 25 N=1,NCFDAS
      DO 20 I=1,NCPDAS
      CONSW=0
```

```
07/05/69              FORTRAIN
      SWITCH=3
   12 T=T+HTINC
   13 CALL DAS4
      CALL ANALOG(YIN1,YIN2,YIN3,YIN4,YIN5)
      IF(SWITCH-3) 15,14,15
   14 SWITCH=2
      GO TO 13
   15 IF(CONSW-5) 16,19,16
   16 IF(SWITCH-2) 18,17,18
   17 SWITCH=4
      GO TO 12
   18 CONSW=5
      PRINT = PRINT+1
      GO TO 14
   19 IF(RESET) 1,20,20
   20 CONTINUE
      PRINT = 0
   25 CONTINUE
      WRITE (9,500) TINC,PINT,FINT
  500 FORMAT (10X,'THE ABOVE INTEGRATION IS BASED ON INCREMENT OF',F14.8
     1,/10X,'IT IS PRINTED EVERY',F14.8,/10X,'THE FINAL VALUE OF THIS IN
     2TEGRATION IS',F14.8)
      READ(1,2) TEST
      IF(TEST-20) 27,26,27
   26 READ(1,28) TINC,PINT,FINAL
   28 FORMAT(3F14.8)
      NCFDAS = (FINAL-FINT)/PINT +0.1
      NCPDAS = PINT/TINC + 0.1
      FINT=FINAL
      READ(1,40) TEST
   40 FORMAT(I3)
      IF(TEST) 41,42,41
   41 CALL DAS3
   42 CONTINUE
      GO TO 50
   27 IF(TEST-10) 21,23,21
   21 WRITE(9,22)
   22 FORMAT('1',10X,'ENTER NEW SOURCE PROGRAM')
      GO TO 1
   23 STOP
      END
```

```
      SUBRCUTINE DAS1
      DOUBLE PRECISICN DCUBS(320)
      REAL REALS(40C)
      INTEGER INTS(200)
      COMMCN DOUES,REALS,INTS
      DO 1 NB =1,40C
    1 REALS(NB)=C.0
      DO 2 NB =1,200
    2 INTS(NB)=0
      DO 3 NB =1,300
    3 DOUBS(NB)=0.0
      RETURN
      END
```

```
      SUBROUTINE DAS2
      DOUBLE PRECISION DOUBS(320),HEAD(312)
      REAL REALS(400)
      INTEGER INTS(200),TEST
      COMMON DOUBS,REALS,INTS
      EQUIVALENCE (DOUBS(1),HEAD(1))
      EQUIVALENCE (INTS(10),TEST)
      IF(TEST-1) 10,1,10
    1 READ(1,16) NCARD
      IF (NCARD-38) 2,2,11
    2 READ(1,16) ITEST
      IF(ITEST) 3,4,3
    3 WRITE(9,12)
    4 DO 5 KA = 1,NCARD
      KB = (KA-1)*8+1
      KC = KB+7
    5 READ(1,13) (HEAD(KD),KD=KB,KC)
      CARD =NCARD/2
      IF(CARD-(NCARD/2)) 7,6,7
    6 NCARD = CARD
      GO TO 8
    7 NCARD=CARD + 1
    8 DO 9 KE = 1,NCARD
      KF = (KE-1)*16+1
      KG = KF + 15
    9 WRITE(9,14) (HEAD(KH),KH=KF,KG)
   10 RETURN
   11 WRITE (9,15)
      RETURN
   12 FORMAT(51X,'YANHEE ELECTRICITY AUTHORITY',/43X,'SYSTEM PROTECTION
     1AND COMMUNICATION DIVISION',/52X,'SYSTEM PROTECTION SECTION',////)
   13 FORMAT(1X,8A8)
   14 FORMAT(1X,16A8)
   15 FORMAT(//10X,'NO OF CHARACTERS OF HEADING ARE GREATER THAN SPECIFI
     1ED LIMIT')
   16 FORMAT(I3)
      END
```

```
      SUBROUTINE DAS3
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),TEST
      COMMON DOUBS,REALS,INTS
      DIMENSION YIC(20),PARAME(20),GX1(10),GY1(10),GX2(10),GY2(10)
      DIMENSION GX3(10),GY3(10)
      EQUIVALENCE (REALS(2),YIC(1))
      EQUIVALENCE (REALS(62),TINC)
      EQUIVALENCE (REALS(83),GX1(1))
      EQUIVALENCE (REALS(93),GY1(1))
      EQUIVALENCE (REALS(103),GX2(1))
      EQUIVALENCE (REALS(113),GY2(1))
      EQUIVALENCE (REALS(123),GX3(1))
      EQUIVALENCE (REALS(133),GY3(1))
      EQUIVALENCE (REALS(146),PINT)
      EQUIVALENCE (REALS(147),FINT)
      EQUIVALENCE (REALS(154),PARAME(1))
      EQUIVALENCE (INTS(4),NTAB)
      EQUIVALENCE (INTS(10),TEST)
    1 IF(TEST-2) 5,2,5
    2 READ(1,17) NOCARD
      READ(1,18) (YIC(JA),JA=1,NOCARD)
      READ(1,20) TINC,PINT,FINT
    3 READ(1,17) TEST
      IF(TEST) 1,4,1
    4 CONTINUE
      RETURN
    5 IF(TEST-3) 8,6,8
    6 READ(1,17) NOCARD
    7 READ(1,18) (PARAME(JA),JA=1,NOCARD)
      GO TO 3
    8 IF(TEST-4) 15,9,15
    9 READ(1,17) NTAB
      IF(NTAB-1) 10,14,10
   10 IF(NTAB-2) 11,13,11
   11 IF(NTAB-3) 15,12,15
   12 READ(1,19) TEST,LC,(GX3(JA),GY3(JA),JA=1,LC)
      IF(TEST-5) 9,4,9
   13 READ(1,19) TEST,LC,(GX2(JA),GY2(JA),JA=1,LC)
      IF(TEST-5) 9,4,9
   14 READ(1,19) TEST,LC,(GX1(JA),GY1(JA),JA=1,LC)
      IF(TEST-5) 9,4,9
   15 WRITE(9,16)
   16 FORMAT(//10X,'NUMBER OF TABLE MUST BE CHECKED OR THE ERROR IS DUE
     1TO THE SWITCH OF TEST IN SUBROUTINE IS WRONG')
   17 FORMAT(I3)
   18 FORMAT(5F14.8)
   19 FORMAT(2I3,/(5F14.8))
   20 FORMAT(F14.8,F14.8,F14.8)
      RETURN
      END
```

```
      SUBROUTINE DAS4
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      EQUIVALENCE (INTS(1),INT)
      EQUIVALENCE (INTS(16),ICT)
      EQUIVALENCE (INTS(17),NF)
      EQUIVALENCE (INTS(18),NTM)
      EQUIVALENCE (INTS(19),NSQ)
      EQUIVALENCE (INTS(20),NST)
      EQUIVALENCE (INTS(21),IP)
      EQUIVALENCE (INTS(22),IZ)
      EQUIVALENCE (INTS(23),IS)
      EQUIVALENCE (INTS(24),ID)
      INT=0
      ICT=0
      NF=0
      NSQ=0
      NTM=0
      NST=0
      IP=0
      IZ=0
      IS=0
      ID=0
      RETURN
      END
```

```
      SUBROUTINE COEFF(N,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10)
      DOUBLE PRECISION DOUBS(320)
      INTEGER INTS(200)
      REAL REALS(400)
      COMMON DOUBS,REALS,INTS
      DIMENSION PARAME(20)
      EQUIVALENCE(REALS(154),PARAME(1))
      IF(N-10) 1,1,2
    1 C1 = PARAME(1)
      C2 = PARAME(2)
      C3 = PARAME(3)
      C4 = PARAME(4)
      C5 = PARAME(5)
      C6 = PARAME(6)
      C7 = PARAME(7)
      C8 = PARAME(8)
      C9 = PARAME(9)
      C10= PARAME(10)
      RETURN
    2 WRITE(9,3)
    3 FORMAT(//1CX,'NUMBER OF COEFFICIENTS MUST BE LESS THAN 10')
      RETURN
      END
```

```
      FUNCTION XINT(X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),RESET,SWITCH
      COMMON DOUBS,REALS,INTS
      DIMENSION YIC(20),SLOPE(20),YC(20)
      EQUIVALENCE (REALS(1),T)
      EQUIVALENCE (REALS(2),YIC(1))
      EQUIVALENCE (REALS(22),SLOPE(1))
      EQUIVALENCE (REALS(42),YC(1))
      EQUIVALENCE (REALS(62),TINC)
      EQUIVALENCE (INTS(1),INT)
      EQUIVALENCE (INTS(2),SWITCH)
      EQUIVALENCE (INTS(3),RESET)
      INT=INT+1
      IF(INT-20) 1,1,11
    1 IF(T) 6,2,6
    2 IF(SWITCH-1) 4,3,4
    3 XINT=YIC(INT)
      RETURN
    4 IF(SWITCH-2) 11,5,11
    5 SLOPE(INT)=X
      GO TO 3
    6 IF(SWITCH-2) 7,5,7
    7 IF(SWITCH-3) 8,10,8
    8 IF(SWITCH-4) 11,9,11
    9 YIC(INT)=YC(INT)+TINC*SLOPE(INT)
      GO TO 3
   10 YC(INT) = YIC(INT)
      YIC(INT) = YIC(INT)+0.5*TINC*SLOPE(INT)
      GO TO 3
   11 RESET=-1
      WRITE(9,12)
   12 FORMAT(//10X,'SWITCH IN SUBRCUTINE MUST BE CHECKED OR NUMBER OF IN
     1TEGRATORS ARE GREATER THAN 20')
      RETURN
      END
```

```
FUNCTION CLOCK(A)
DOUBLE PRECISION DOUBS(320)
REAL REALS(400)
INTEGER INTS(200)
COMMON DOUES,REALS,INTS
EQUIVALENCE (REALS(1),T)
CLOCK=A*T
RETURN
END
```

```
      SUBROUTINE TYPE(N,CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),PRINT
      COMMON DOUBS,REALS,INTS
      DIMENSION CP(10)
      EQUIVALENCE (INTS(11),NCPDAS)
      EQUIVALENCE (INTS(13),CONSW)
      EQUIVALENCE (INTS(25),PRINT)
      IF(N-10) 1,1,5
    1 IF(PRINT-NCPDAS) 2,4,4
    2 RETURN
    4 OP(1) = CP1
      OP(2) =CP2
      OP(3) =CP3
      OP(4) =CP4
      OP(5) =CP5
      OP(6) =CP6
      OP(7) =CP7
      OP(8) =CP8
      OP(9) =CP9
      OP(10) = CP10
      WRITE(9,8) (OP(J),J=1,N)
    8 FORMAT(10F13.8)
      RETURN
    5 WRITE(9,6)
    6 FORMAT(//10X,'PRINT OUTPUT MUST BE LESS THAN 10')
      RETURN
      END
```

```
      SUBROUTINE FUNG(X,NTAB,METHOD)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),RESET,SWITCH
      COMMON DOUBS,REALS,INTS
      DIMENSION Y(10),FX(10),GX1(10),GX2(10),GX3(10),GY1(10),GY2(10)
      DIMENSION GY3(10),FY(3)
      EQUIVALENCE (REALS(63),FX(1))
      EQUIVALENCE (REALS(73),Y(1))
      EQUIVALENCE (REALS(83),GX1(1))
      EQUIVALENCE (REALS(93),GY1(1))
      EQUIVALENCE (REALS(103),GX2(1))
      EQUIVALENCE (REALS(113),GY2(1))
      EQUIVALENCE (REALS(123),GX3(1))
      EQUIVALENCE (REALS(133),GY3(1))
      EQUIVALENCE (REALS(143),FY(1))
      EQUIVALENCE (INTS(2),SWITCH)
      EQUIVALENCE (INTS(3),RESET)
      EQUIVALENCE (INTS(6),NP)
      EQUIVALENCE (INTS(7),NP1)
      EQUIVALENCE (INTS(8),NP2)
      EQUIVALENCE (INTS(9),NP3)
      IF(SWITCH-2) 1,12,1
    1 IF(NTAB-1) 2,4,2
    2 IF(NTAB-2) 3,6,3
    3 IF(NTAB-3) 19,8,19
    4 NP=NP1
      DO 5 IA=1,NP
      Y(IA)=GY1(IA)
    5 FX(IA)=GX1(IA)
      GO TO 10
    6 NP=NP2
      DO 7 IB=1,NP
      Y(IB)=GY2(IB)
    7 FX(IB)=GX2(IB)
      GO TO 10
    8 NP=NP3
      DO 9 IC =1,NP
      Y(IC)=GYS(IC)
    9 FX(IC)=GX3(IC)
   10 IF(X) 11,11,13
   11 FUNG=Y(1)
      FY(NTAB)=FUNG
      RETURN
   12 FUNG=FY(NTAB)
      RETURN
   13 IF(METHOD-1) 15,14,15
   14 CALL LINEAR
      RETURN
   15 IF(METHOD-2) 17,16,17
   16 CALL SECOND
      RETURN
   17 IF(METHOD-3) 19,18,19
   18 CALL LAGRAN
```

```
      RETURN
   19 WRITE(3,20)
   20 FORMAT(//10X,'METHOD IN SUBROUTINE IS IN ERROR OR NUMBER OF TABLE
     1IS GREATER THAN 3')
      RESET=-1
      RETURN
      END
```

```
      SUBROUTINE LINEAR(X,NTAB,METHOD)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),RESET
      COMMON DOUBS,REALS,INTS
      DIMENSION Y(10),FX(10),GX1(10),GX2(10),GX3(10),GY1(10),GY2(10)
      DIMENSION GY3(10),FY(3)
      EQUIVALENCE (REALS(63),FX(1))
      EQUIVALENCE (REALS(73),Y(1))
      EQUIVALENCE (REALS(83),GX1(1))
      EQUIVALENCE (REALS(93),GY1(1))
      EQUIVALENCE (REALS(103),GX2(1))
      EQUIVALENCE (REALS(113),GY2(1))
      EQUIVALENCE (REALS(123),GX3(1))
      EQUIVALENCE (REALS(133),GY3(1))
      EQUIVALENCE (REALS(143),FY(1))
      EQUIVALENCE (INTS(3),RESET)
      EQUIVALENCE (INTS(6),NP)
      EQUIVALENCE (INTS(7),NP1)
      EQUIVALENCE (INTS(8),NP2)
      EQUIVALENCE (INTS(9),NP3)
      DO 1 KA=2,NP
      IF(X-FX(KA)) 4,3,1
    1 CONTINUE
      WRITE(3,2)
    2 FORMAT(//10X,'ERROR IS DUE TO THIS METHOD CANNOT BE APPLIED TO EXT
     1RAPOLATED')
      RESET=-1
      RETURN
    3 FUNG=Y(KA)
      FY(NTAB)=FUNG
      RETURN
    4 FUNG=Y(KA-1)+((Y(KA)-Y(KA-1))*(X-FX(KA-1)))/(FX(KA)-FX(KA-1))
      FY(NTAB)=FUNG
      RETURN
      END
```

```
      SUBROUTINE SECOND(X,NTAB,METHOD)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),RESET
      COMMON DOUBS,REALS,INTS
      DIMENSION Y(10),FX(10),GX1(10),GX2(10),GX3(10),GY1(10),GY2(10)
      DIMENSION GY3(10),FY(3)
      EQUIVALENCE (REALS(63),FX(1))
      EQUIVALENCE (REALS(73),Y(1))
      EQUIVALENCE (REALS(83),GX1(1))
      EQUIVALENCE (REALS(93),GY1(1))
      EQUIVALENCE (REALS(103),GX2(1))
      EQUIVALENCE (REALS(113),GY2(1))
      EQUIVALENCE (REALS(123),GX3(1))
      EQUIVALENCE (REALS(133),GY3(1))
      EQUIVALENCE (REALS(143),FY(1))
      EQUIVALENCE (INTS(3),RESET)
      EQUIVALENCE (INTS(6),NP)
      EQUIVALENCE (INTS(7),NP1)
      EQUIVALENCE (INTS(8),NP2)
      EQUIVALENCE (INTS(9),NP3)
      JA=NP-1
      DO 1 NK=3,JA
      IF(X-FX(NK)) 4,3,1
    1 CONTINUE
      WRITE(3,2)
    2 FORMAT(//10X,'ERROR IS DUE TO THIS METHOD CANNOT BE APPLIED TO EXT
     1RAPOLATED')
      RESET=-1
      RETURN
    3 FUNG=Y(NK)
      FY(NTAB)=FUNG
      RETURN
    4 FUNG=Y(NK-1)+((X-FX(NK-1))*(Y(NK-1)-Y(NK)))/(FX(NK-1)+FX(NK))+(((X
     1-FX(NK-1))*(X-FX(NK)))/(FX(NK+1)-FX(NK-1)))*((Y(NK+1)-Y(NK))/(FX(N
     2K+1)-FX(NK))-(Y(NK)-Y(NK-1))/(FX(NK)-FX(NK-1)))
      FY(NTAB)=FUNG
      RETURN
      END
```

```
      SUBROUTINE LAGRAN(X,NTAB,METHOD)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),RESET
      COMMON DOUBS,REALS,INTS
      DIMENSION Y(10),FX(10),GX1(10),GX2(10),GX3(10),GY1(10),GY2(10)
      DIMENSION GY3(10),FY(3),B(10),R(10)
      EQUIVALENCE (REALS(63),FX(1))
      EQUIVALENCE (REALS(73),Y(1))
      EQUIVALENCE (REALS(83),GX1(1))
      EQUIVALENCE (REALS(93),GY1(1))
      EQUIVALENCE (REALS(103),GX2(1))
      EQUIVALENCE (REALS(113),GY2(1))
      EQUIVALENCE (REALS(123),GX3(1))
      EQUIVALENCE (REALS(133),GY3(1))
      EQUIVALENCE (REALS(143),FY(1))
      EQUIVALENCE (REALS(200),B(1))
      EQUIVALENCE (REALS(211),R(1))
      EQUIVALENCE (INTS(3),RESET)
      EQUIVALENCE (INTS(6),NP)
      EQUIVALENCE (INTS(7),NP1)
      EQUIVALENCE (INTS(8),NP2)
      EQUIVALENCE (INTS(9),NP3)
      DO 1 JA=1,NP
      IF(X-FX(JA)) 1,9,1
    1 CONTINUE
      DO 4 JB=1,NP
      C=1
      DO 3 JC=1,NP
      IF(JC-JB) 2,3,2
    2 C=C/(FX(JB)-FX(JC))
    3 CONTINUE
      B(JB)=C
    4 CONTINUE
      DO 7 JD=1,NP
      P=1
      DO 6 JE=1,NP
      IF(JD-JE) 5,6,5
    5 P=P*(X-FX(JE))
    6 CONTINUE
    7 R(JD)=P
      FUNG=0
      DO 8 JE=1,NP
    8 FUNG=FUNG+B(JE)*R(JE)*Y(JE)
      RETURN
    9 FUNG=Y(JA)
      FY(NTAB)=FUNG
      RETURN
      END
```

```
      FUNCTION SAWGEN(STP,H)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),SWITCH
      COMMON DOUBS,REALS,INTS
      DIMENSION STOSAW(5)
      EQUIVALENCE (REALS(184),STOSAW(1))
      EQUIVALENCE (REALS(1),T)
      EQUIVALENCE (INTS(2),SWITCH)
      EQUIVALENCE (INTS(20),NST)
      NST=NST+1
      IF(NST-5) 1,1,10
    1 IF(T) 2,2,5
    2 IF(SWITCH-2) 3,4,3
    3 SAWGEN=0.0
      STOSAW(NST)=SAWGEN
      RETURN
    4 SAWGEN=STOSAW(NST)
      STT=0.0
      RETURN
    5 IF(SWITCH-2) 7,6,7
    6 SAWGEN=STOSAW(NST)
      RETURN
    7 HINC=H/STP
      STT=STT+HTINC
      DH=STT*HINC
      IF(STT-STP) 8,9,9
    8 SAWGEN=SAWGEN+DH
      STOSAW(NST)=SAWGEN
      RETURN
    9 STT=0.0
      SAWGEN=0.0
      STOSAW(NST)=SAWGEN
      RETURN
   10 WRITE(3,11)
   11 FORMAT(//10X,'NUMBER OF SAWTOOTH GENERATORS ARE GREATER THAN 5')
      END
```

```
      FUNCTION SQUARE(H,PDT)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      COMMON DOUBS,REALS,INTS
      INTEGER INTS(200),SWITCH
      DIMENSION PT(5),STSQ(5)
      EQUIVALENCE (REALS(174),PT(1))
      EQUIVALENCE (REALS(1),T)
      EQUIVALENCE (INTS(2),SWITCH)
      EQUIVALENCE (INTS(19),NSQ)
      EQUIVALENCE (REALS(179),STSQ(1))
      NSQ=NSQ+1
      IF(NSQ-5) 1,1,10
    1 IF(SWITCH-2) 3,2,3
    2 SQUARE=STSQ(NSQ)
      RETURN
    3 IF(T) 4,4,5
    4 PT(NSQ)=0.0
      SQUARE=H
      STSQ(NSQ)=SQUARE
      RETURN
    5 PT(NSQ)=PT(NSQ)+HTINC
      IF(PT(NSQ)-PDT/2) 6,8,8
    6 SQUARE=H
    7 STSQ(NSQ)=SQUARE
      RETURN
    8 IF(PT(NSQ)-PDT/2) 9,4,4
    9 SQUARE=-H
      GO TO 7
   10 WRITE(3,11)
   11 FORMAT(//10X,'THE NUMBER OF SQUARE WAVES ARE GREATER THAN 5')
      RESET=-1
      RETURN
      END
```

```
      FUNCTION RANDOM(X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200),SWITCH
      COMMON DOUBS,REALS,INTS
      EQUIVALENCE (REALS(1),T)
      EQUIVALENCE (INTS(2),SWITCH)
      IF(T) 1,1,4
    1 IF(SWICH-2) 2,3,2
    2 YRF=EXP(X)
      IYR=YRF
      YRF=YRF-IYR
      YRF=YRF*(10**5)
      YRF=(YRF/50000)-1.0
      IYRL=X
      YRL=IYRL-X
      YRL=SIN(YRL)
      YR=YRF*YRL
      YR=YR*100
      IYR=YR
    3 RANDOM=YR-IYR
      RETURN
    4 IF(SWITCH-3) 5,2,5
    5 IF(SWITCH-4) 3,2,3
      END
```

```
      FUNCTION NEGCIP(X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X) 1,1,2
    1 NEGCIP=0
      RETURN
    2 NEGCIP=X
      RETURN
      END
```

```
      FUNCTION POSCIP(X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X) 1,2,2
    1 POSCIP=0.0
      RETURN
    2 POSCIP=X
      RETURN
      END
```

```
      FUNCTION LIMIT(X,P1,P2)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X-P1) 1,2,2
    1 LIMIT=P1
      RETURN
    2 IF(X-P2) 3,4,4
    3 LIMIT=P2
      RETURN
    4 LIMIT=X
      RETURN
      END
```

```
      FUNCTION DEAD(X,UB,LB)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X) 1,3,4
    1 IF(X-LB) 2,3,3
    2 DEAD=X
      RETURN
    3 DEAD=0.0
      RETURN
    4 IF(X-UB) 5,2,2
    5 DEAD=0.0
      RETURN
      END
```

```
      FUNCTION AND(M,X1,X2,X3,X4,X5)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      DIMENSION X(5)
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      DO 1 JK=1,M
      IF(X(JK)) 2,2,1
    1 CONTINUE
      AND=1.0
      RETURN
    2 AND=0.0
      RETURN
      END
```

```
      FUNCTION NAND(M,X1,X2,X3,X4,X5)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      DIMENSION X(5)
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      DO 1 JK =1,M
      IF(X(JK)) 2,2,1
    1 CONTINUE
      NAND=0.0
      RETURN
    2 NAND=1.0
      RETURN
      END
```

```
      FUNCTION OR(M,X1,X2,X3,X4,X5)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      DIMENSION X(5)
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      DO 1 JK=1,M
      IF(X(JK)) 1,1,2
    1 CONTINUE
      OR=0.0
      RETURN
    2 OR=1.0
      RETURN
      END
```

```
      FUNCTION NOR(M,X1,X2,X3,X4,X5)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      DIMENSION X(5)
      X(1)=X1
      X(2)=X2
      X(3)=X3
      X(4)=X4
      X(5)=X5
      DO 1 JK=1,M
      IF(X(JK)) 1,1,2
    1 CONTINUE
      NOR=1
      RETURN
    2 NOR=0
      RETURN
      END
```

```
      FUNCTION NOT(X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X) 1,1,2
    1 NOT=1
      RETURN
    2 NOT=0
      RETURN
      END
```

```
      SUBROUTINE ASTOP(X1,X2)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X1-X2) 1,1,2
    1 PAUSE
    2 RETURN
      END
```

```
      FUNCTION OFFSET(X,P)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      X=X+P
      OFFSET=X
      RETURN
      END
```

```
      FUNCTION BANG(X,P)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(X) 1,2,3
    1 BANG=-P
      RETURN
    2 BANG=0.0
      RETURN
    3 BANG=P
      RETURN
      END
```

```
FUNCTION CLOCK%A□
DOUBLE PRECISION DOUBS%300□
REAL REALS%40C□
INTEGER INTS%200□
COMMON DOUBS,REALS,INTS
EQUIVALENCE %REALS%1□,T□
CLOCK#A*T
RETURN
END
```

```
      FUNCTION DELAY(TD,X)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(SWITCH-2) 1,3,1
    1 IF(T-TD) 2,2,4
    2 OUTPUT=0.0
      DEL=X
    3 DELAY=OUT
      RETURN
    4 OUTPUT=DEL
      GO TO 3
      END
```

```
      FUNCTION RELAY(X1,X2,C)
      DOUBLE PRECISION DOUBS(320)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DOUBS,REALS,INTS
      IF(C) 1,2,3
    1 RELAY = X1
      RETURN
    2 RELAY=0.0
      RETURN
    3 RELAY=X2
      RETURN
      END
```

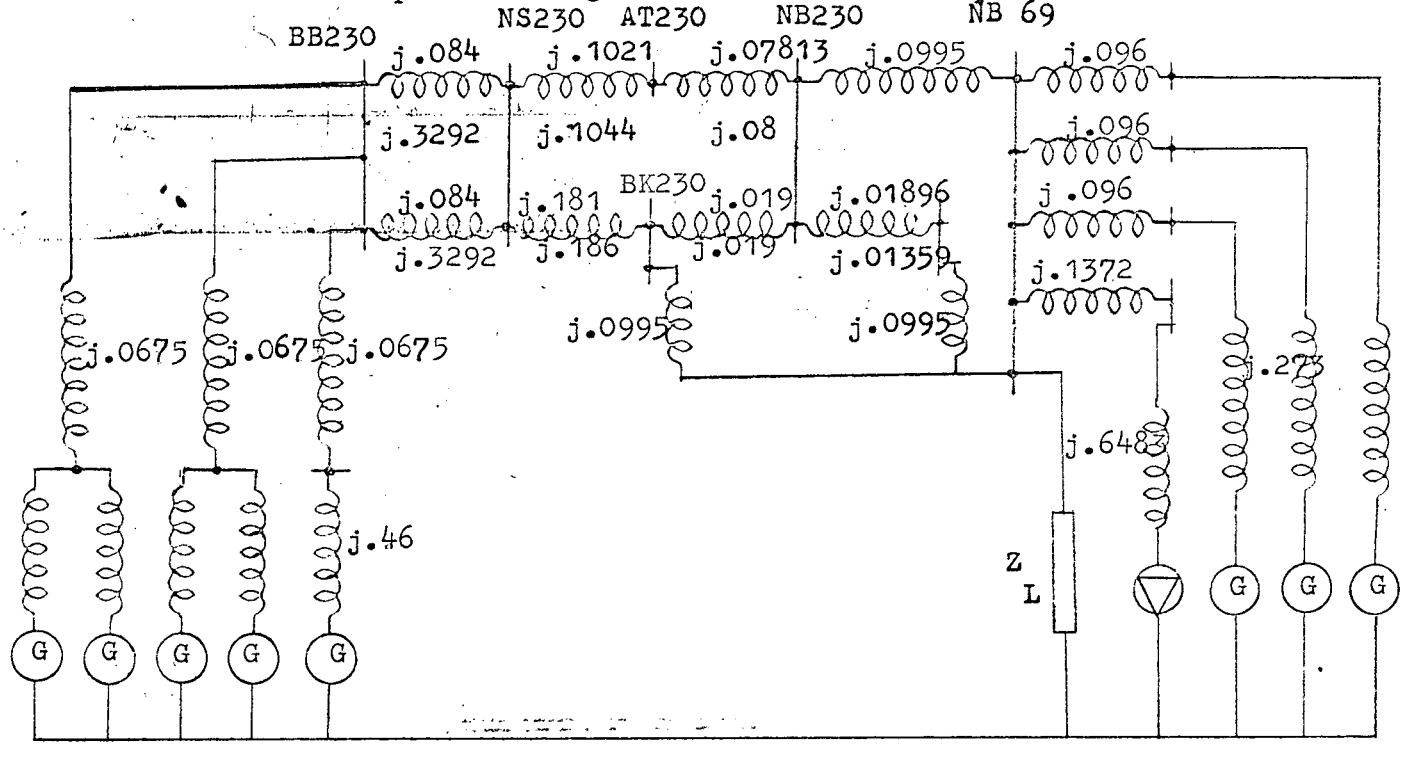
# APPENDIX C

## DETAILS OF THE NETWORK CALCULATION

### PREFAULT CONDITION

Assumption :-

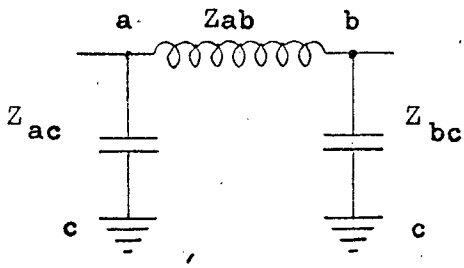
1. Load ทุกแห่ง นอกจาก MEA น้อยมากเมื่อเทียบกันแล้ว ให้ neglect
2. Neglect line resistance
3. BASE MVA = 100
4. BASE KV = 230/69/13.2
5. Load of MEA at B.K. and N.B. รวมเป็นload เดียวกันที่ NB 69
6. Impedance diagram of YEA system ดังรูป




PREFAULT CONDITION

7. Line ทุกเส้น ถือว่าต่อแบบ $\pi$

8. การคำนวณ Network ทุกครั้งอาจจะต้องมีการเปลี่ยนจาก T เป็น $\pi$ และจาก $\pi$ เป็น T




$$Zac = \frac{1}{Y_{ac}}$$

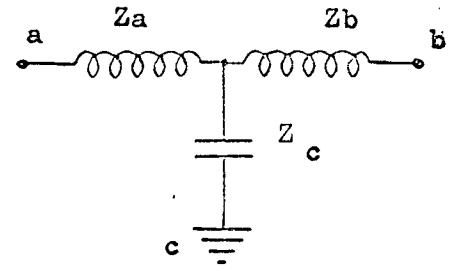
$$Zbc = \frac{1}{Y_{bc}}$$

การเปลี่ยนจาก T เป็น $\pi$ ดังสมการ

$$Zab = \frac{ZaZb + ZbZc + ZcZa}{Zc}$$

$$Z_{bc} = \frac{Za}{ZaZb + ZbZc + ZaZc}$$

$$Z_{ac} = \frac{Zb}{ZaZb + ZbZc + ZcZa}$$




$$Zc = \frac{1}{Y_c}$$

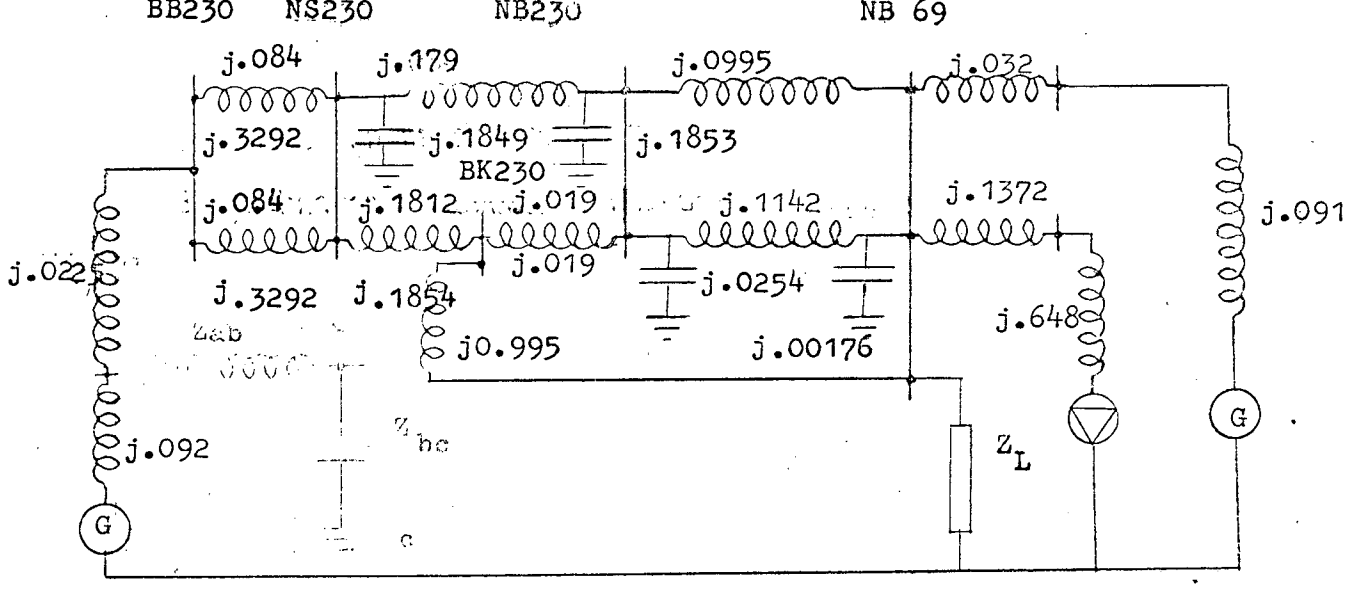
การเปลี่ยนจาก $\pi$ เป็น T

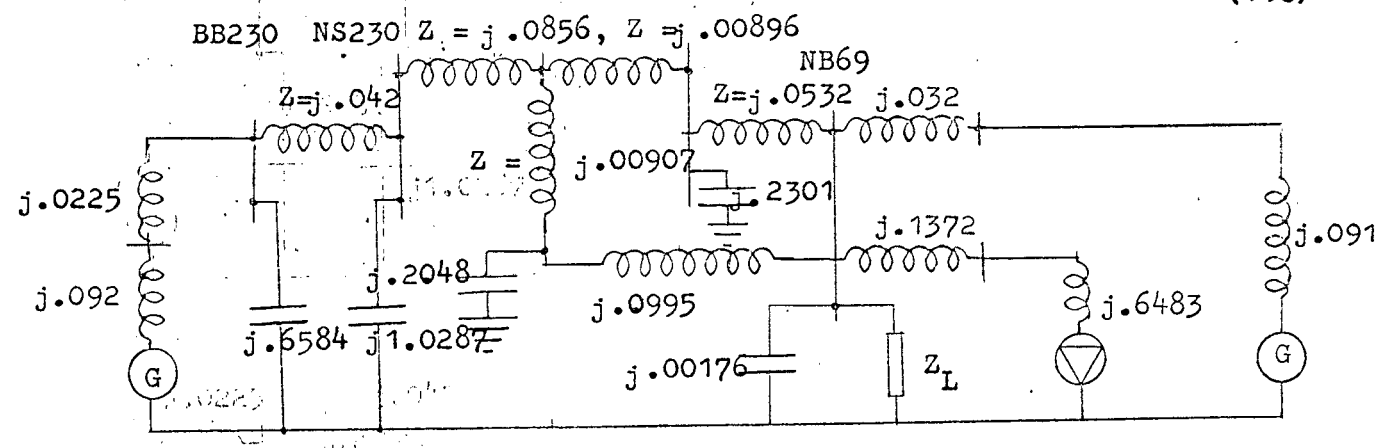
$$Za = \frac{ZabZac}{Zab + Zbc + Zca}$$

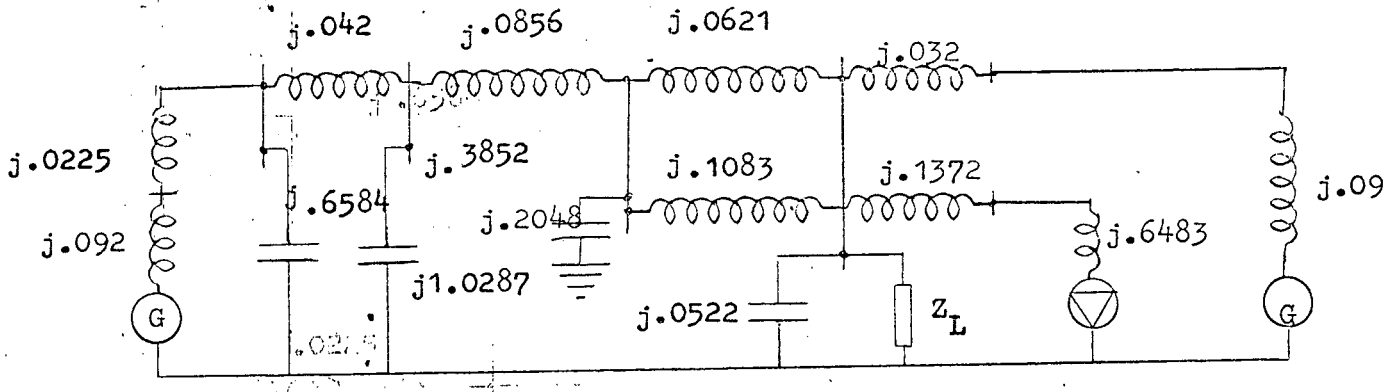
$$Zb = \frac{ZbcZab}{Zab + Abc + Zac}$$

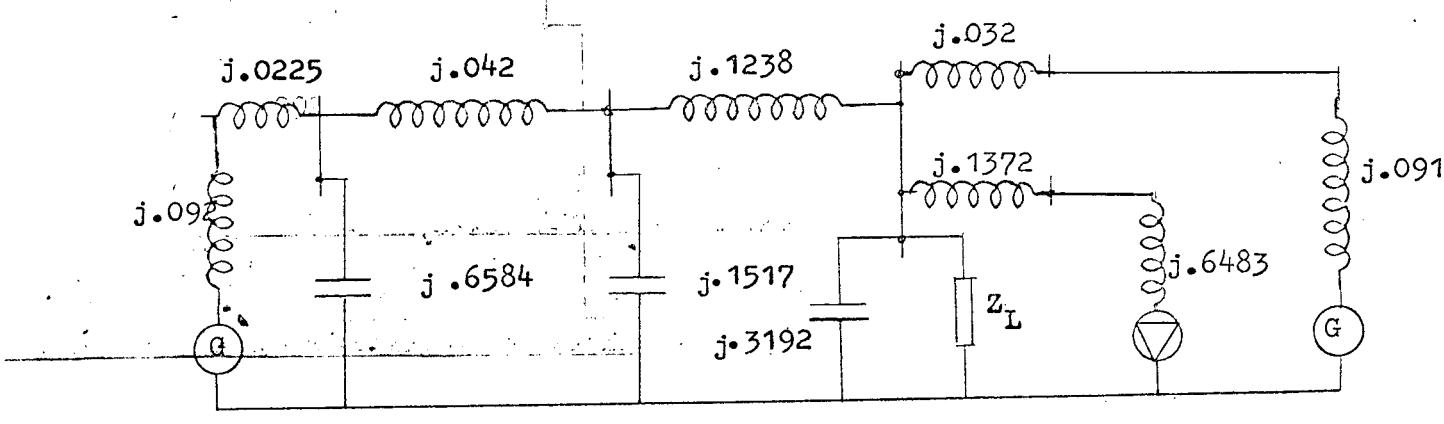
$$Z_c = \frac{Zab + Zbc + Zac}{ZacZbc}$$

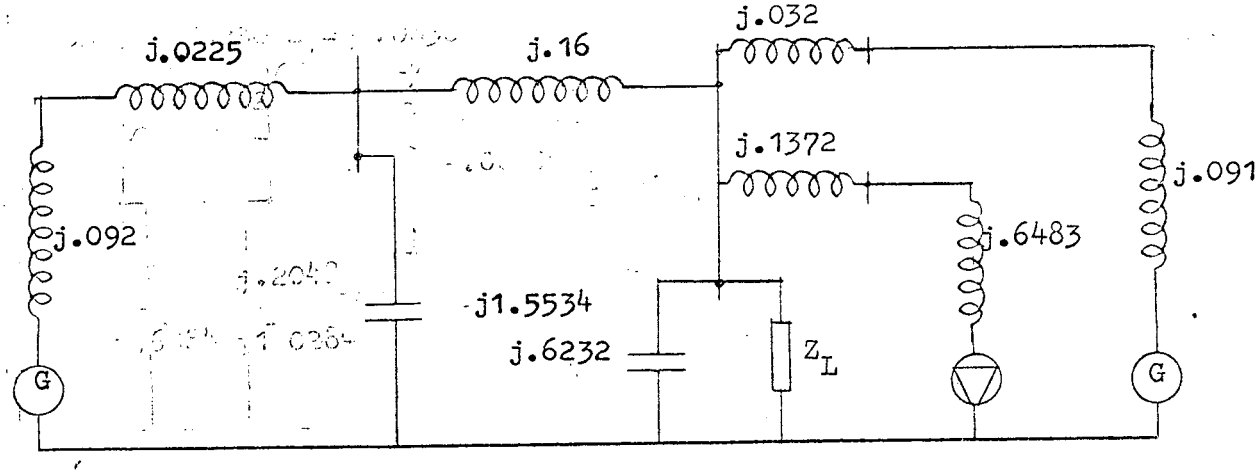
ดังนี้จะ Simplified diagram ได้ดังต่อไปนี้

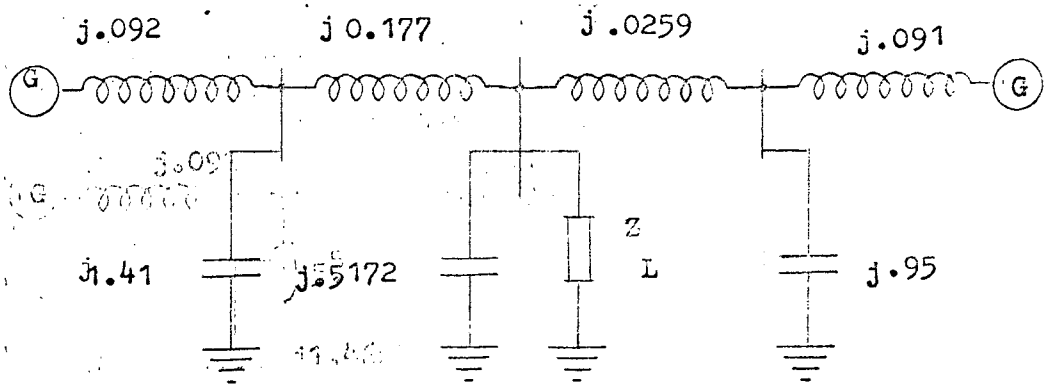



BB230 NS230 Z = j .0856, Z = j .00896
NB69
Z=j .042
Z=j .0532
j.032
Z = j.00907
j.2301
j.0225
j.1372
j.091
j.092
j.2048
j.0995
j.6483
j.6584
j1.0287
j.00176
$Z_L$
G


j.042
j.0856
j.0621
j.032
j.0225
j.3852
j.1083
j.1372
j.091
j.6584
j.2048
j.6483
j.092
j1.0287
j.0522
$Z_L$
G


j.032
j.0225
j.042
j.1238
j.091
j.1372
j.092
j.6584
j.1517
j.6483
j.3192
$Z_L$
G

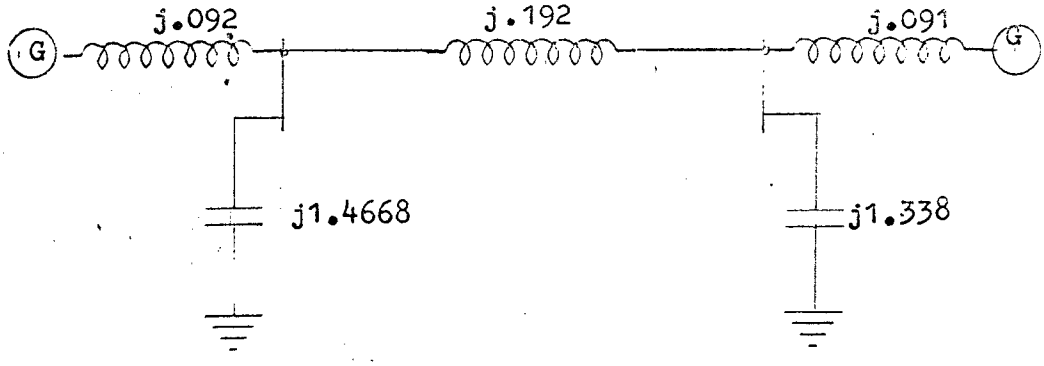

j.032
j.0225
j.16
j.1372
j.091
j.092
j.6483
j1.5534
j.6232
$Z_L$
G




$$Z_L = .352 + .244$$

$$= .428 \angle 34.8$$

$$V_L = 1.92 - j1.33$$



# Appendix D

## Result

// EXEC

YANKEE ELECTRICITY
SYSTEM PROTECTION AND CONM
SYSTEM PROTECTION

FAULT NEAR B.B. BUS CR

1. THREE PHASE FAULT AT BUS BB 230

FOR SUSTAINED FAULT

A. CONSIDER BUS BB 230 AS AN INFINITE BUS

INFINITE BUS VOLTAGE = 1.112 P.U.
INITIAL PHASE ANGLE = 0.339 RADIAN

B. CONSIDER BUS NB 230 AS AN INFINITE BUS

INFINITE BUS VOLTAGE = 1.108 P.U.
INITIAL PHASE ANGLE = 0.377 RADIAN

| T (SEC) | PAA (P.U.) | ANGA (RAD) | VELA (R/S) | PAB (P.U.) | A( ( |
|---|---|---|---|---|---|
| 0.0 | 1.50000000 | 0.33899999 | 0.0 | 2.87999916 | 0 |
| 0.01999991 | 1.50000000 | 0.77365905 | -0.02999999 | 2.87999916 | 0. |
| 0.03999980 | 1.50000000 | 2.07719026 | -0.05999997 | 2.87999916 | 2. |
| 0.05999969 | 1.50000000 | 4.25126171 | -0.08999908 | 2.87999916 | 5. |
| 0.07999843 | 1.50000000 | 7.29410267 | -0.11999810 | 2.87999916 | 8. |
| 0.09999698 | 1.50000000 | 11.20632172 | -0.14999712 | 2.87999916 | 13. |
| 0.11999553 | 1.50000000 | 15.98792076 | -0.17999613 | 2.87999916 | 19. |
| 0.13999408 | 1.50000000 | 21.63874817 | -0.20999515 | 2.87999916 | 26. |
| 0.15999264 | 1.50000000 | 28.15899081 | -0.23999417 | 2.87999916 | 34. |
| 0.17999119 | 1.50000000 | 35.54850769 | -0.26999319 | 2.87999916 | 42. |
| 0.19998974 | 1.50000000 | 43.80744934 | -0.29999220 | 2.87999916 | 52. |
| 0.21998829 | 1.50000000 | 52.93574524 | -0.32999122 | 2.87999916 | 64. |
| 0.23998684 | 1.50000000 | 62.93342590 | -0.35999024 | 2.87999916 | 76. |
| 0.25998539 | 1.50000000 | 73.80046082 | -0.38998926 | 2.87999916 | 89. |
| 0.27998394 | 1.50000000 | 85.53688049 | -0.41998827 | 2.87999916 | 103. |
| 0.29998249 | 1.50000000 | 98.14265442 | -0.44998729 | 2.87999916 | 118. |
| 0.31998104 | 1.50000000 | 111.61781311 | -0.47998631 | 2.87999916 | 135. |
| 0.33997959 | 1.50000000 | 125.96234131 | -0.50998533 | 2.87999916 | 152. |
| 0.35997814 | 1.50000000 | 141.17625427 | -0.53998435 | 2.87999916 | 170. |
| 0.37997669 | 1.50000000 | 157.25953674 | -0.56998336 | 2.87999916 | 190. |
| 0.39997524 | 1.50000000 | 174.21218872 | -0.59998238 | 2.87999916 | 210. |
| 0.41997379 | 1.50000000 | 192.03419495 | -0.62998140 | 2.87999916 | 232. |
| 0.43997234 | 1.50000000 | 210.72558594 | -0.65998042 | 2.87999916 | 254. |
| 0.45997089 | 1.50000000 | 230.28634644 | -0.68997943 | 2.87999916 | 278. |
| 0.47996944 | 1.50000000 | 250.71647644 | -0.71997845 | 2.87999916 | 303. |
| 0.49996799 | 1.50000000 | 272.01416016 | -0.74997747 | 2.87999916 | 329. |

THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.00100000
IT IS PRINTED EVERY 0.02000000
THE FINAL VALUE OF THIS INTEGRATION IS 0.50000000

AUTHORITY
NICATION DIVISION
SECTION

.O. - N.S. LINE

| GD (AU) | VELO (R/S) |
|---|---|
| 37699997 | 0.0 |
| 90288645 | -0.05759992 |
| 48033551 | -0.11519949 |
| 10995197 | -0.17279899 |
| 79113483 | -0.23039854 |
| 52408600 | -0.28799808 |
| 30873108 | -0.34559762 |
| 14906531 | -0.40319717 |
| 03318767 | -0.46079671 |
| 97305298 | -0.51839626 |
| 64469116 | -0.57599580 |
| 00811768 | -0.63359535 |
| 80330200 | -0.69119489 |
| 99022688 | -0.74879444 |
| 94894409 | -0.80639398 |
| 89944763 | -0.86399353 |
| 80167847 | -0.92159307 |
| 75569763 | -0.97919261 |
| 71339832 | -1.03678131 |
| 66859741 | -1.09436417 |
| 62724915 | -1.15194702 |
| 58738403 | -1.20952988 |
| 54895630 | -1.26711273 |
| 51181445 | -1.32469559 |
| 47221680 | -1.38227844 |
| 43588867 | -1.43986130 |

ENTER NEW SOURCE PROGRAM

FOR CLEARED FAULT

A. CONSIDER BUS 88 230 AS AN INFINITE BUS

INFINITE BUS VOLTAGE = 0.908 P.U.
INITIAL PHASE ANGLE = 0.399 RADIAN

B. CONSIDER BUS NO 230 AS AN INFINITE BUS

INFINITE BUS VOLTAGE = 0.84 P.U.
INITIAL PHASE ANGLE = 0.377 RADIAN

(1) FAULT CLEARED AT T=0.02 SEC

| T (SEC) | PAA (P.U.) | ANGA (RAD) | VELA (R/S) | PAB (P.U.) |
|---|---|---|---|---|
| 0.0 | 1.50000000 | 0.33899999 | 0.0 | 2.879999 |
| 0.00999996 | 1.50000000 | 0.44761457 | -0.01499999 | 2.879999 |
| 0.01999991 | 1.50000000 | 0.77369905 | -0.02999999 | 2.879999 |

THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.0
IT IS PRINTED EVERY 0.01000000
THE FINAL VALUE OF THIS INTEGRATION IS 0.02000000

| | | | | |
|---|---|---|---|---|
| 0.02999966 | -3.17401409 | 1.03300095 | -0.00307882 | -10.927013 |
| 0.03999980 | -1.38211632 | 0.87107056 | 0.02228726 | 2.732719 |
| 0.04999975 | 0.99109352 | 0.51918190 | 0.02270512 | 15.203214 |
| 0.05999969 | 1.38183594 | 0.27663517 | 0.00960964 | 3.298295 |
| 0.06999916 | 1.00718784 | 0.23033690 | -0.00256829 | -11.093060 |
| 0.07999843 | 0.49463093 | 0.32869685 | -0.01013022 | -9.613610 |
| 0.08999771 | -0.08464858 | 0.49786347 | -0.01223848 | 6.273674 |
| 0.09999698 | -0.69157473 | 0.65461653 | -0.00841976 | 12.693048 |
| 0.10999626 | -0.81881154 | 0.72165942 | -0.00053955 | -1.986827 |
| 0.11999553 | -0.38692975 | 0.67754203 | 0.00587808 | -10.401833 |
| 0.12999481 | 0.17531455 | 0.57877713 | 0.00678284 | -5.349198 |
| 0.13999408 | 0.42203695 | 0.50138223 | 0.00348123 | 10.097196 |
| 0.14999336 | 0.36348534 | 0.48154098 | -0.00063864 | 7.814046 |
| 0.15999264 | 0.15341169 | 0.51264703 | -0.00329425 | -6.944414 |
| 0.16999191 | -0.08234781 | 0.56568819 | -0.00362238 | -8.623013 |
| 0.17999119 | -0.23008722 | 0.60783374 | -0.00194069 | 2.352403 |
| 0.18999046 | -0.21496648 | 0.61860484 | 0.00042741 | 9.944218 |
| 0.19998974 | -0.07904493 | 0.59972948 | 0.00193601 | -0.050377 |
| 0.20998901 | 0.06890470 | 0.56999213 | 0.00192091 | -6.274482 |
| 0.21998829 | 0.13072312 | 0.54922819 | 0.00083964 | -3.625360 |
| 0.22998756 | 0.10446906 | 0.54635823 | -0.00039776 | 7.602812 |
| 0.23998684 | 0.03048889 | 0.55806953 | -0.00109146 | 5.114935 |
| 0.24998611 | -0.04118103 | 0.57423455 | -0.00101665 | -5.620388 |
| 0.25998539 | -0.07285684 | 0.58491510 | -0.00040333 | -6.097976 |
| 0.26998466 | -0.05571581 | 0.58562481 | 0.00027556 | 3.308256 |
| 0.27998394 | -0.01157369 | 0.57857990 | 0.00062068 | 6.915950 |
| 0.28998321 | 0.02697978 | 0.56978291 | 0.00052727 | -1.736020 |
| 0.29998249 | 0.03997233 | 0.56458515 | 0.00016799 | -6.311982 |
| 0.30998176 | 0.02763395 | 0.56487781 | -0.00018704 | -0.616602 |
| 0.31998104 | 0.00355201 | 0.56903762 | -0.00034573 | 6.248980 |
| 0.32998031 | -0.01621453 | 0.57376689 | -0.00027313 | 1.733626 |
| 0.33997959 | -0.02192578 | 0.57631636 | -0.00006907 | -5.118935 |
| 0.34997886 | -0.01397833 | 0.57585186 | 0.00011940 | -3.079178 |
| 0.35997814 | -0.00037446 | 0.57342780 | 0.00019174 | 4.209717 |
| 0.36997741 | 0.00978711 | 0.57090610 | 0.00013883 | 3.791413 |
| 0.37997669 | 0.01182963 | 0.56970495 | 0.00002348 | -3.084671 |
| 0.38997597 | 0.00683169 | 0.57013416 | -0.00007413 | -4.105326 |

| ANGU (RAD) | VELB (R/S) |
|---|---|
| 0.37699997 | 0.0 |
| 0.50847143 | -0.02879996 |
| 0.90288645 | -0.05759992 |

00000

| | |
|---|---|
| 0.92632282 | 0.05295182 |
| 0.03164023 | 0.11996537 |
| -0.63178462 | 0.00443021 |
| -0.06017405 | -0.10985017 |
| 0.77386260 | -0.04790503 |
| 0.70731519 | 0.06016219 |
| -0.11234113 | 0.09104776 |
| -0.46660715 | -0.02402835 |
| 0.18255061 | -0.09222031 |
| 0.72675270 | -0.01317866 |
| 0.39941669 | 0.07692904 |
| -0.29936016 | 0.04813249 |
| -0.23282677 | -0.09881833 |
| 0.43011504 | -0.05917960 |
| 0.58442271 | 0.02853694 |
| 0.04346800 | 0.07086003 |
| -0.31264037 | -0.00608450 |
| 0.10281140 | -0.06732351 |
| 0.53591669 | -0.01342584 |
| 0.30428916 | 0.05640848 |
| -0.19099784 | 0.03195326 |
| -0.10869449 | -0.04966271 |
| 0.36791056 | -0.03959212 |
| 0.42092115 | 0.02873862 |
| -0.00327938 | 0.04755713 |
| -0.17668402 | -0.01989147 |
| 0.18150008 | -0.04717825 |
| 0.42203856 | 0.00233784 |
| 0.16260934 | 0.04464141 |
| -0.13949198 | 0.00932939 |
| 0.03563614 | -0.03973453 |
| 0.35069340 | -0.01721438 |
| 0.26098593 | 0.03160540 |
| -0.04798249 | 0.02466121 |
| -0.04590308 | -0.02458154 |
| 0.24869585 | -0.02773615 |
| 0.31235033 | 0.01554471 |

| | | | | |
|---|---|---|---|---|
| 0.39997524 | -0.00059663 | 0.57152522 | -0.00010513 | 1.7663679 |
| 0.40997452 | -0.00574036 | 0.57285821 | -0.00007006 | 4.3500089 |
| 0.41997379 | -0.00636327 | 0.57341379 | -0.00000564 | -0.8419943 |
| 0.42997307 | -0.00330569 | 0.57309139 | 0.00004482 | -4.0508846 |
| 0.43997234 | 0.00076568 | 0.57230157 | 0.00005715 | -0.3394561 |
| 0.44997162 | 0.00333105 | 0.57160431 | 0.00003466 | 3.8877630 |
| 0.45997089 | 0.00337784 | 0.57135934 | -0.00000096 | 1.0211944 |
| 0.46997017 | 0.00155268 | 0.57158172 | -0.00002663 | -3.2742157 |
| 0.47996944 | -0.00064451 | 0.57202506 | -0.00003083 | -1.7972631 |
| 0.48996872 | -0.00190479 | 0.57238644 | -0.00001694 | 2.7325487 |
| 0.49996799 | -0.00178091 | 0.57248843 | 0.00000260 | 2.1882867 |

THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.00
IT IS PRINTED EVERY 0.01000000
THE FINAL VALUE OF THIS INTEGRATION IS 0.50000000

```
    0.05616825    0.03021785
   -0.06591159   -0.00812145
    0.14827507   -0.02951112
    0.30475098    0.00084964
    0.14497024    0.02764576
   -0.04052192    0.00547944
    0.07029748   -0.02467989
    0.26360259   -0.01009177
    0.20480263    0.02032579
    0.01062336    0.01424579
    0.02391524   -0.01619665
100000
```

ENTER NEW SOURCE PROGRAM
(1) FAULT CLEARED AT T=0.03 SEC

| T (SEC) | PAA (P.U.) | ANGA (RAD) | VELA (R/S) | PAB (P.U.) |
|---|---|---|---|---|
| 0.0 | 1.50000000 | 0.33899999 | 0.0 | 2.879999 |
| 0.00999996 | 1.50000000 | 0.44767457 | -0.01499999 | 2.879999 |
| 0.01999991 | 1.50000000 | 0.77369905 | -0.02999999 | 2.879999 |
| 0.02999986 | 1.50000000 | 1.31707001 | -0.04499998 | 2.879999 |

THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.0
IT IS PRINTED EVERY 0.01000000
THE FINAL VALUE OF THIS INTEGRATION IS 0.03000000

| | | | | |
|---|---|---|---|---|
| 0.03999980 | -6.90540981 | 1.49835972 | 0.02776280 | 4.5564699 |
| 0.04999975 | 0.85120678 | 0.70981896 | 0.00156084 | 0.7740281 |
| 0.05999969 | 3.01334095 | -0.01759684 | 0.03560393 | 5.7889841 |
| 0.06999916 | 2.43793011 | -0.32867143 | 0.00833418 | 1.8868401 |
| 0.07999843 | 1.55984688 | -0.29093432 | -0.01206703 | -2.4634361 |
| 0.08999771 | 0.59950924 | -0.02797592 | -0.02257834 | -0.7552489 |
| 0.09999698 | -0.07016063 | 0.32617605 | -0.02519074 | 13.7645387 |
| 0.10999626 | -1.02877522 | 0.66589695 | -0.02005709 | -12.0258426 |
| 0.11999553 | -1.81962013 | 0.85736895 | -0.00496717 | 2.3714466 |
| 0.12999481 | -1.09947205 | 0.80459869 | 0.01100314 | 14.7197704 |
| 0.13999408 | 0.29901516 | 0.60073769 | 0.01469427 | -8.1888761 |
| 0.14999336 | 0.86026669 | 0.42891961 | 0.00807836 | -2.1231355 |
| 0.15999264 | 0.74102253 | 0.37388158 | -0.00026634 | -1.7429327 |
| 0.16999191 | 0.38153321 | 0.42363358 | -0.00598578 | 3.5078906 |
| 0.17999119 | -0.04930216 | 0.52784938 | -0.00766291 | 13.6335866 |
| 0.18999046 | -0.41336554 | 0.62559897 | -0.00520247 | -9.1452846 |
| 0.19998974 | -0.49450976 | 0.66682905 | -0.00034724 | 1.9663744 |
| 0.20998901 | -0.24145830 | 0.64051116 | 0.00354097 | -13.3636927 |
| 0.21998829 | 0.09097332 | 0.58008718 | 0.00422566 | -2.2552306 |
| 0.22998756 | 0.25870508 | 0.53076202 | 0.00229654 | -9.0270919 |
| 0.23998684 | 0.23468469 | 0.51663965 | -0.00030411 | -0.9742641 |
| 0.24998611 | 0.10064381 | 0.53525132 | -0.00203431 | -0.9707155 |
| 0.25998539 | -0.05173522 | 0.56824166 | -0.00225762 | 1.1342554 |
| 0.26998466 | -0.14292300 | 0.59446782 | -0.00120568 | 4.6238880 |
| 0.27998394 | -0.13233685 | 0.60121012 | 0.00029635 | 2.0986566 |
| 0.28998321 | -0.04763740 | 0.58966291 | 0.00119116 | 5.9403343 |
| 0.29998249 | 0.04084202 | 0.57123929 | 0.00119959 | 1.8887453 |
| 0.30998176 | 0.08103216 | 0.55814165 | 0.00053958 | 4.0265669 |
| 0.31998104 | 0.06597793 | 0.55612653 | -0.00023519 | 0.9666652 |
| 0.32998031 | 0.01967907 | 0.56330609 | -0.00067609 | -0.1438236 |
| 0.33997959 | -0.02937061 | 0.57335794 | -0.00063398 | -0.5329809 |
| 0.34997886 | -0.04915223 | 0.58003938 | -0.00025441 | -6.1339529 |
| 0.35997814 | -0.03470713 | 0.58054060 | 0.00016703 | -2.2162752 |
| 0.36997741 | -0.00757414 | 0.57620955 | 0.00038397 | -12.0136871 |
| 0.37997669 | 0.01644269 | 0.57073891 | 0.00032983 | -2.3951988 |
| 0.38997597 | 0.02484585 | 0.56746143 | 0.00010818 | -13.8436317 |
| 0.39997524 | 0.01738039 | 0.56759501 | -0.00011372 | 1.0076284 |
| 0.40997452 | 0.00239614 | 0.57015908 | -0.00021444 | -9.7482233 |
| 0.41997379 | -0.00997687 | 0.57310367 | -0.00017078 | 7.7742853 |
| 0.42997307 | -0.01361850 | 0.57470822 | -0.00004451 | -2.3853635 |
| 0.43997234 | -0.00876491 | 0.57444072 | 0.00007299 | 13.6040725 |
| 0.44997162 | -0.00034391 | 0.57294476 | 0.00011097 | 2.7627582 |
| 0.45997089 | 0.00601904 | 0.57137305 | 0.00008702 | 10.8945035 |
| 0.46997017 | 0.00736420 | 0.57061285 | 0.00001599 | 1.6552734 |
| 0.47996944 | 0.00430090 | 0.57086736 | -0.00004547 | -2.3622751 |
| 0.48996872 | -0.00031914 | 0.57172740 | -0.00006532 | -2.3222198 |
| 0.49996799 | -0.00353977 | 0.57255894 | -0.00004355 | -13.3538885 |

THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.00
IT IS PRINTED EVERY 0.01000000
THE FINAL VALUE OF THIS INTEGRATION IS 0.50000000

| | ANGB (RAD) | VELB (R/S) |
|---|---|---|
| 6 | 0.37699997 | 0.0 |
| 6 | 0.50847143 | -0.02879996 |
| 6 | 0.90288645 | -0.05759992 |
| 6 | 1.56023979 | -0.08639967 |
| 100000 | | |
| 2 | 2.28064823 | -0.08766782 |
| 5 | 3.30705547 | -0.13015908 |
| 0 | 4.48468781 | -0.13639414 |
| 3 | 6.23191547 | -0.24604100 |
| 3 | 8.10699362 | -0.16267210 |
| 3 | 9.67853905 | -0.18292046 |
| 5 | 11.37090302 | -0.21630919 |
| 7 | 13.69479079 | -0.23054053 |
| 1 | 15.62693446 | -0.22241914 |
| 3 | 17.71307373 | -0.29999260 |
| 3 | 20.33877563 | -0.25434411 |
| 7 | 22.64086440 | -0.25461555 |
| 3 | 25.20758057 | -0.32399368 |
| 5 | 27.77523804 | -0.26574206 |
| 4 | 30.24357605 | -0.29240960 |
| 0 | 33.10380554 | -0.27888787 |
| 0 | 35.68316650 | -0.28404289 |
| 1 | 38.64385986 | -0.32501066 |
| 7 | 41.40971994 | -0.30242878 |
| 0 | 44.40055847 | -0.39623618 |
| 1 | 47.30902319 | -0.31381136 |
| 1 | 50.30747986 | -0.36853498 |
| 1 | 53.32357788 | -0.32115136 |
| 1 | 56.35787964 | -0.37437558 |
| 0 | 59.47789001 | -0.32924110 |
| 2 | 62.57885742 | -0.38074839 |
| 1 | 65.77294446 | -0.33656687 |
| 5 | 68.94186401 | -0.38702208 |
| 7 | 72.17082214 | -0.34184110 |
| 1 | 75.40139771 | -0.39179367 |
| 7 | 78.64593506 | -0.34619641 |
| 1 | 81.94615173 | -0.39324611 |
| 1 | 85.19859314 | -0.34926885 |
| 1 | 88.57275391 | -0.38773501 |
| 1 | 91.82173157 | -0.35002995 |
| 1 | 95.25660706 | -0.37486809 |
| 1 | 98.49719238 | -0.35135031 |
| 101 | 101.97257996 | -0.36133963 |
| 1 | 105.21920776 | -0.35949832 |
| 1 | 108.71672058 | -0.35510063 |
| 1 | 111.99900818 | -0.37825572 |
| 1 | 115.49752808 | -0.35833436 |
| 1 | 118.84762573 | -0.40129101 |
| 1 | 122.31906128 | -0.36460763 |
| 1 | 125.79514221 | -0.41074532 |
| 1 | 129.17263630 | -0.36519521 |
| 1 | 132.66475342 | -0.39632279 |
| 00000 | | |

ENTER NEW SOURCE PROGRAM
(1) FAULT CLEARED AT T=0.04 SEC

| T (SEC) | PAA (P.U.) | ANGA (RAD) | VELA (R/S) | PAB (P.U.) |
|---|---|---|---|---|
| 0.0 | 1.50000000 | 0.33899999 | 0.0 | 2.879999 |
| 0.01999991 | 1.50000000 | 0.77369905 | -0.02999999 | 2.879999 |
| 0.03999980 | 1.50000000 | 2.07779026 | -0.05999997 | 2.879999 |
| THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.0 | | | | |
| IT IS PRINTED EVERY 0.02000000 | | | | |
| THE FINAL VALUE OF THIS INTEGRATION IS 0.04000000 | | | | |
| 0.05999969 | -1.53997421 | 1.03302002 | 0.11421967 | 15.169130 |
| 0.07999843 | 6.91378880 | -1.09903431 | 0.01689184 | 4.743885 |
| 0.09999698 | -0.65616035 | -0.08562106 | -0.05510866 | -13.616096 |
| 0.11999553 | -4.04332447 | 1.11569971 | -0.01582459 | 15.727578 |
| 0.13999408 | 0.90084678 | 0.56933181 | 0.03263399 | 1.972086 |
| 0.15999264 | 1.37350655 | 0.09956497 | 0.00064099 | -2.745163 |
| 0.17999119 | 0.20233816 | 0.36857390 | -0.01529900 | 2.150868 |
| 0.19998974 | -1.04850156 | 0.73812699 | -0.00575122 | 2.910463 |
| 0.21998829 | -0.06417090 | 0.63692755 | 0.00909352 | -1.116702 |
| 0.23998684 | 0.52525955 | 0.49696066 | 0.00140453 | -3.535355 |
| 0.25998539 | 0.03155583 | 0.53074110 | -0.00470178 | -4.708312 |
| 0.27998394 | -0.31121933 | 0.63000512 | -0.00086983 | -4.517643 |
| 0.29998249 | 0.01145011 | 0.58682412 | 0.00269054 | -2.436700 |
| 0.31998104 | 0.15936765 | 0.53803152 | 0.00018210 | 0.603745 |
| 0.33997959 | -0.01323801 | 0.56524390 | -0.00145343 | 2.869631 |
| 0.35997814 | -0.08817798 | 0.59044760 | -0.00002953 | 2.866620 |
| 0.37997669 | 0.01446917 | 0.57416642 | 0.00078528 | 0.007301 |
| 0.39997524 | 0.04566379 | 0.58189632 | -0.00004624 | -2.684679 |
| 0.41997379 | -0.01051500 | 0.57155800 | -0.00042293 | -0.523097 |
| 0.43997234 | -0.02425173 | 0.57759386 | 0.00005197 | 1.096537 |
| 0.45997089 | 0.00748769 | 0.57196474 | 0.00022471 | 13.155217 |
| 0.47996944 | 0.01248636 | 0.56911540 | -0.00004408 | 8.971336 |
| 0.49996799 | -0.00486000 | 0.57235187 | -0.00011844 | -6.385744 |
| THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.0 | | | | |
| IT IS PRINTED EVERY 0.02000000 | | | | |
| THE FINAL VALUE OF THIS INTEGRATION IS 0.50000000 | | | | |

```
     ANGB            VELB
     (RAD)           (R/S)
6    0.37699997      0.0
6    0.90288645     -0.05759992
6    2.48093551     -0.11919945
100000


3    5.11267757     -0.19765967
9    9.04493182     -0.19882780
0   13.41135693     -0.26924696
6   17.79516602     -0.27972335
1   23.12956238     -0.28350866
3   28.73323740     -0.28963894
2   34.31025696     -0.29086512
3   39.98361206     -0.29754585
8   45.94033813     -0.31491077
8   52.09463501     -0.32131046
3   58.29716492     -0.32510114
3   64.58050537     -0.33277357
2   71.00424194     -0.33881685
6   77.52339172     -0.34160751
7   84.08952332     -0.34597969
6   90.71578979     -0.35328446
3   97.41479492     -0.35871977
3  104.16241455     -0.35834646
4  110.93699646     -0.35669041
9  117.74481201     -0.36342442
5  124.60501099     -0.36384819
6  131.52871704     -0.40759856
9  138.49928284     -0.41129935
100000
```

```
        ENTER NEW SOURCE PROGRAM
 (1) FAULT CLEARED AT T=0.09 SEC

   T            PAA          ANGA          VELA          PAB
 (SEC)        (P.U.)        (RAD)         (R/S)        (P.U.)
0.0          1.50000000   0.33899999    0.0           2.879999
0.01999991   1.50000000   0.77369905   -0.02999999    2.879999
0.03999980   1.50000000   2.07779026   -0.05999997    2.879999
       THE ABOVE INTEGRATION IS BASED ON INCREMENT OF     0.0
       IT IS PRINTED EVERY     0.02000000
       THE FINAL VALUE OF THIS INTEGRATION IS     0.05000000
0.05999969  -1.53997421   1.03302002    0.11421967   15.169130
0.07999843   6.91378880  -1.09903431    0.01689184    4.743885
0.09999098  -0.65616035  -0.08962106   -0.05510866  -13.616696
0.11999553  -4.04332447   1.11565971   -0.01582459   15.727978
0.13999408   0.90084678   0.58933181    0.03263399    1.972086
0.15999264   1.37350655   0.09956497    0.00064095   -2.749163
0.17999119   0.20233816   0.36857390   -0.01529900    2.150868
0.19998974  -1.04890156   0.73812699   -0.00575122    2.910463
0.21998829  -0.06417090   0.63692755    0.00909359   -3.116702
0.23998684   0.52529955   0.45696086    0.00140453   -3.539355
0.25998539   0.03155583   0.53074110   -0.00478178   -4.708312
0.27998394  -0.31121933   0.63000512   -0.00086983   -4.517643
0.29998249   0.01145011   0.58682412    0.00269054   -2.436700
0.31998104   0.15938765   0.53803152    0.00016210    0.603745
0.33997959  -0.01323801   0.56524390   -0.00145343    2.869631
0.35997814  -0.08817798   0.59044760   -0.00002953    2.866620
0.37997669   0.01446917   0.57416642    0.00078928    0.007301
0.39997524   0.04566375   0.56189632   -0.00004624   -2.684679
0.41997379  -0.01051500   0.57155800   -0.00042293   -0.523097
0.43997234  -0.02425173   0.57759386    0.00005197    7.096531
0.45997089   0.00748769   0.57196474    0.00022471   13.755217
0.47996944   0.01248636   0.56911540   -0.00004408    8.971336
       THE ABOVE INTEGRATION IS BASED ON INCREMENT OF     0.0
       IT IS PRINTED EVERY     0.02000000
       THE FINAL VALUE OF THIS INTEGRATION IS     0.50000000
```

| ANGO (RAD) | VEL (R/S) |
|---|---|
| 0.37699997 | 0.0 |
| 0.90288645 | -0.05759992 |
| 2.46093551 | -0.11515945 |

00000

| | |
|---|---|
| 9.11267757 | -0.19765967 |
| 9.04453182 | -0.19862780 |
| 13.41135693 | -0.26924896 |
| 17.79516602 | -0.27972335 |
| 23.12956238 | -0.28350866 |
| 28.73333740 | -0.28963894 |
| 34.31029696 | -0.29086912 |
| 39.98361206 | -0.29754585 |
| 45.94033813 | -0.31491077 |
| 52.09463501 | -0.32131046 |
| 58.29716492 | -0.32510114 |
| 64.58050537 | -0.33277357 |
| 71.00424194 | -0.33881885 |
| 77.52339172 | -0.34160751 |
| 84.08952332 | -0.34597969 |
| 90.71578979 | -0.35328466 |
| 97.41479492 | -0.35871977 |
| 104.16241455 | -0.35834646 |
| 110.93699646 | -0.35669041 |
| 117.74481201 | -0.36342442 |
| 124.60501099 | -0.38384819 |
| 131.52871704 | -0.40755856 |

100000

ENTER NEW SOURCE PROGRAM
(1) FAULT CLEARED AT T=0.06 SEC

| T (SEC) | PAA (P.U.) | ANGA (RAD) | VELA (R/S) | PAB (P.U.) |
|---|---|---|---|---|
| 0.0 | 1.50000000 | 0.33899999 | 0.0 | 2.8799991 |
| 0.00999996 | 1.50000000 | 0.44767457 | -0.01499999 | 2.8799991 |
| 0.01999991 | 1.50000000 | 0.77369905 | -0.02999999 | 2.8799991 |
| 0.02999986 | 1.50000000 | 1.31707001 | -0.04499998 | 2.8799991 |
| 0.03999980 | 1.50000000 | 2.07779026 | -0.05999997 | 2.8799991 |
| 0.04999975 | 1.50000000 | 3.05585480 | -0.07499957 | 2.8799991 |
| 0.05999969 | 1.50000000 | 4.25126171 | -0.08999908 | 2.8799991 |
| THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.00 | | | | |
| IT IS PRINTED EVERY 0.01000000 | | | | |
| THE FINAL VALUE OF THIS INTEGRATION IS 0.06000000 | | | | |
| 0.07999843 | -8.68287563 | 7.89680576 | -0.09036231 | 5.4869604 |
| 0.09999698 | -5.26446342 | 7.61542416 | 0.10019392 | -5.3460769 |
| 0.11999553 | 5.76689243 | 5.35867405 | 0.03021341 | 12.9000253 |
| 0.13999408 | 0.08568478 | 5.98808098 | -0.05056941 | -1.4993734 |
| 0.15999264 | -2.89760780 | 7.23474216 | -0.02684822 | 0.6453514 |
| 0.17999119 | -0.39520615 | 7.05205154 | 0.03012141 | 2.5637989 |
| 0.19998974 | 1.45898288 | 6.49538998 | 0.00579941 | 1.972459. |
| 0.21998829 | 0.38388294 | 6.61253452 | -0.01301368 | 2.5458249 |
| 0.23998684 | -0.85728526 | 6.97868633 | -0.00793368 | 3.6990799 |
| 0.25998539 | -0.27950418 | 6.95097637 | 0.00172461 | 3.4737262 |
| 0.27998394 | 0.49672282 | 6.76031208 | 0.00280472 | 2.1114301 |
| 0.29998249 | 0.11124319 | 6.79940605 | -0.00414652 | 0.5829429 |
| 0.31998104 | -0.28030646 | 6.90287971 | -0.00162997 | -1.165678. |
| 0.33997959 | -0.04359389 | 6.87936401 | 0.00239674 | -2.6297471 |
| 0.35997814 | 0.15203500 | 6.82603455 | 0.00061574 | -0.438892. |
| 0.37997669 | 0.01405193 | 6.84331894 | -0.00132097 | 6.0812494 |
| 0.39997524 | -0.08447975 | 6.87134171 | -0.00027386 | 12.8910417 |
| 0.41997379 | -0.00130135 | 6.86034107 | 0.00073578 | 12.2219309 |
| 0.43997234 | 0.04516974 | 6.84610081 | 0.00008809 | -0.3217654 |
| 0.45997089 | -0.00217151 | 6.85307884 | -0.00040289 | -12.708784. |
| 0.47996944 | -0.02445672 | 6.66033916 | -0.00002276 | -11.646322. |
| 0.49996799 | 0.00295103 | 6.85608101 | 0.00021889 | -1.9924707 |
| THE ABOVE INTEGRATION IS BASED ON INCREMENT OF 0.0 | | | | |
| IT IS PRINTED EVERY 0.02000000 | | | | |
| THE FINAL VALUE OF THIS INTEGRATION IS 0.50000000 | | | | |

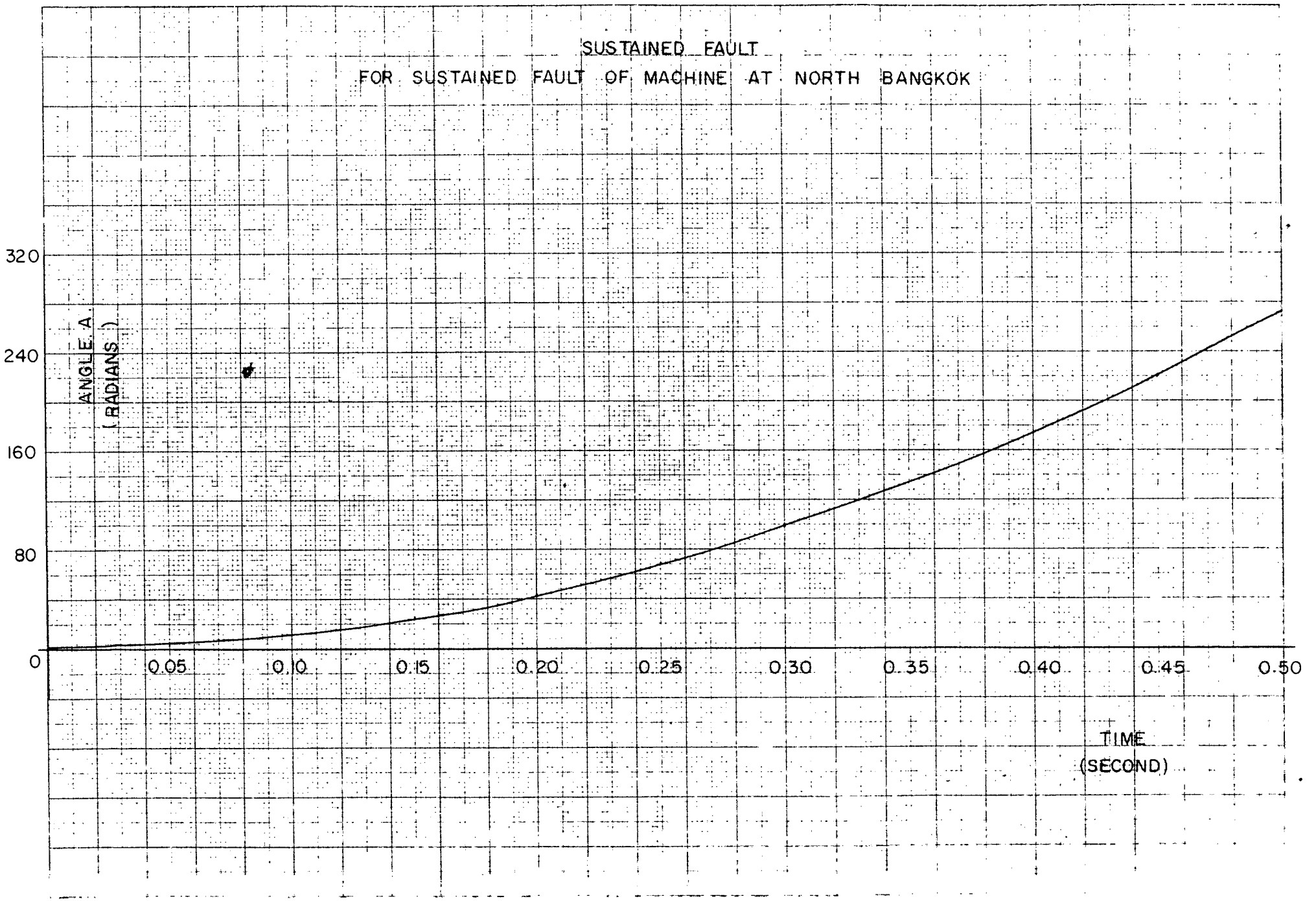
| | ANGU (RAD) | VELB (R/S) |
|---|---|---|
| 6 | 0.37699997 | 0.0 |
| 6 | 0.50847143 | -0.02879996 |
| 6 | 0.90288645 | -0.05759992 |
| 6 | 1.96023979 | -0.08639967 |
| 6 | 2.48093591 | -0.11519945 |
| 6 | 3.66377258 | -0.14399922 |
| 6 | 5.10995197 | -0.17279899 |

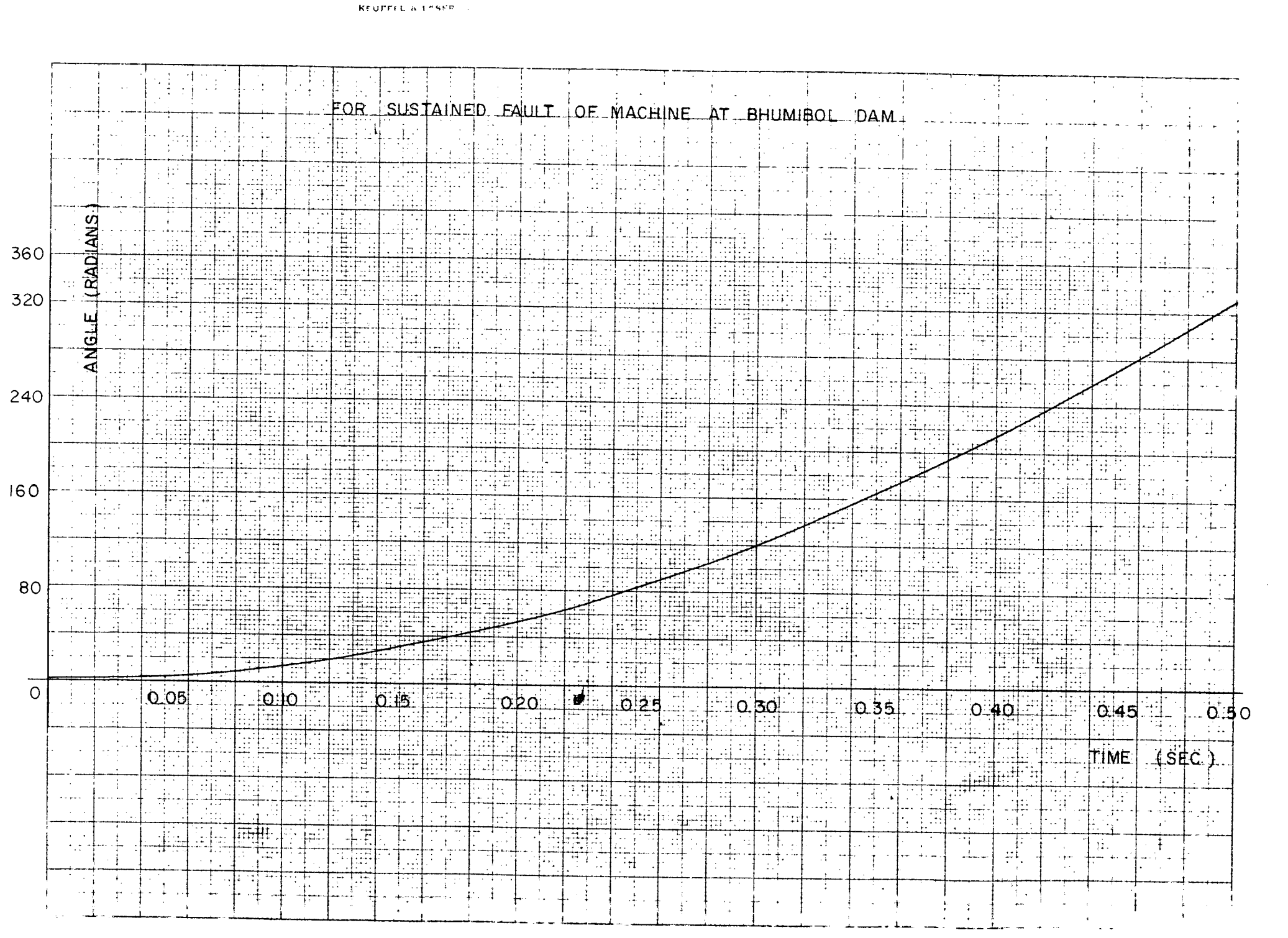
100000

| | | |
|---|---|---|
| 1 | 8.81253052 | -0.18140566 |
| 7 | 12.90320396 | -0.29579580 |
| 7 | 17.50711060 | -0.26796758 |
| 4 | 22.85470581 | -0.28003794 |
| 1 | 28.39230347 | -0.29591244 |
| 0 | 34.34214783 | -0.32854557 |
| 2 | 40.59054565 | -0.32800949 |
| 5 | 46.73957825 | -0.32021981 |
| 1 | 52.86616516 | -0.32462847 |
| 7 | 59.19319153 | -0.33978558 |
| 2 | 65.04113159 | -0.34555459 |
| 1 | 72.70297241 | -0.34618104 |
| 0 | 78.74600220 | -0.35014099 |
| 9 | 85.38836670 | -0.35344005 |
| 6 | 92.09971619 | -0.35293442 |
| 2 | 98.83341980 | -0.35749525 |
| 6 | 105.61001587 | -0.37439537 |
| 1 | 112.46432495 | -0.39886275 |
| 6 | 119.38508606 | -0.41162425 |
| 0 | 126.32489014 | -0.39929998 |
| 5 | 133.25022888 | -0.37991965 |
| 4 | 140.16274207 | -0.36399567 |

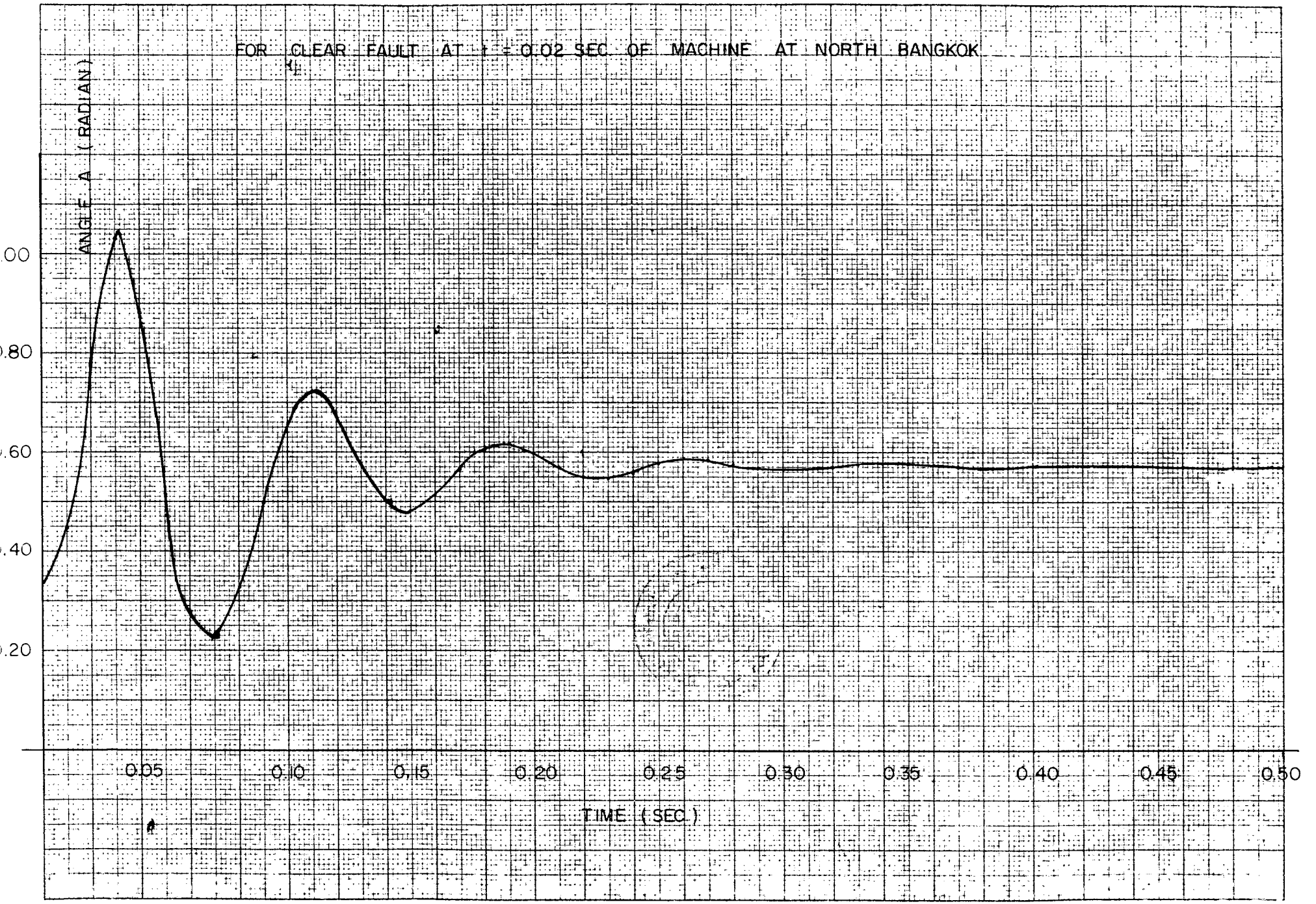
100000

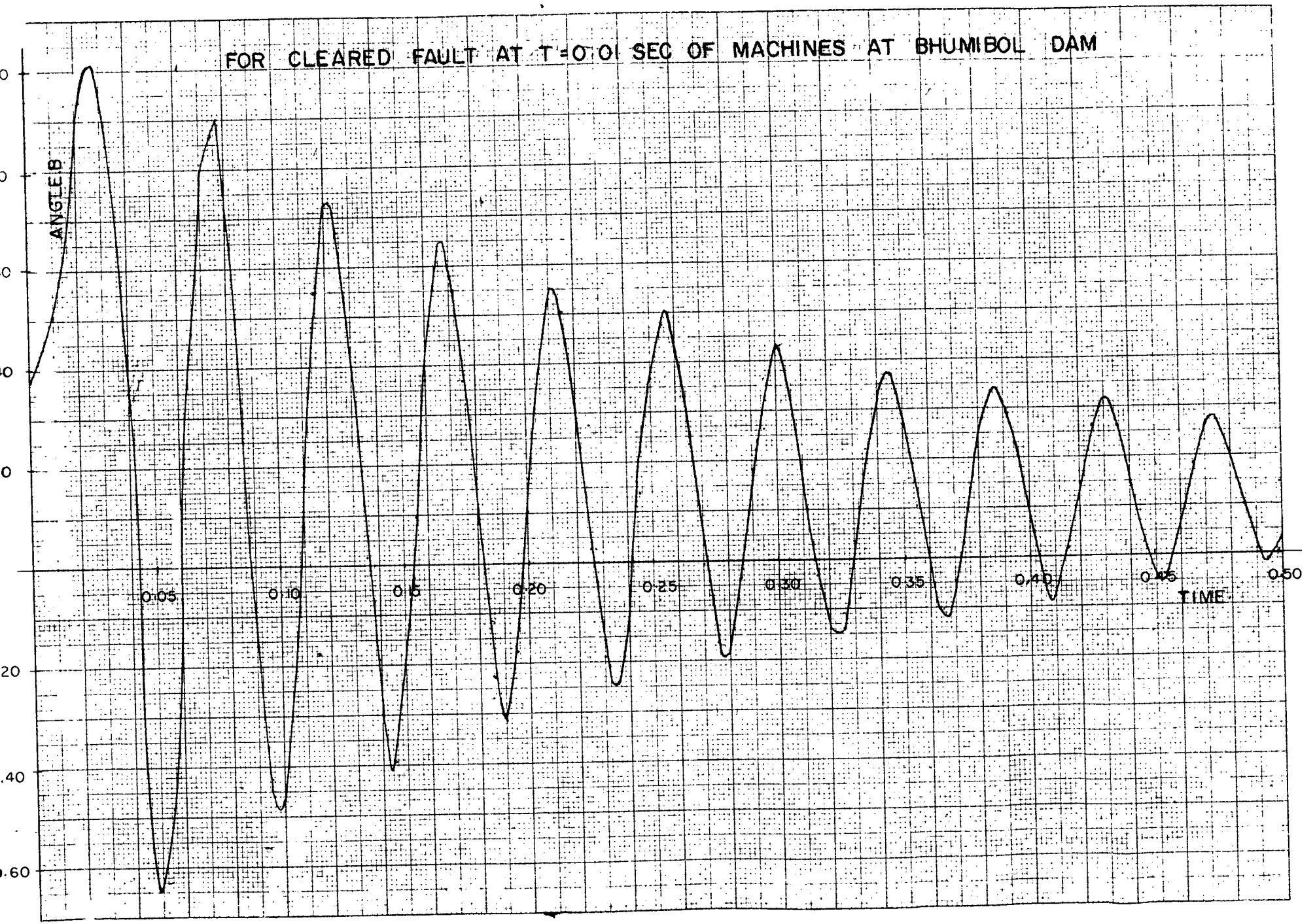
KRUFFEL & ESSER

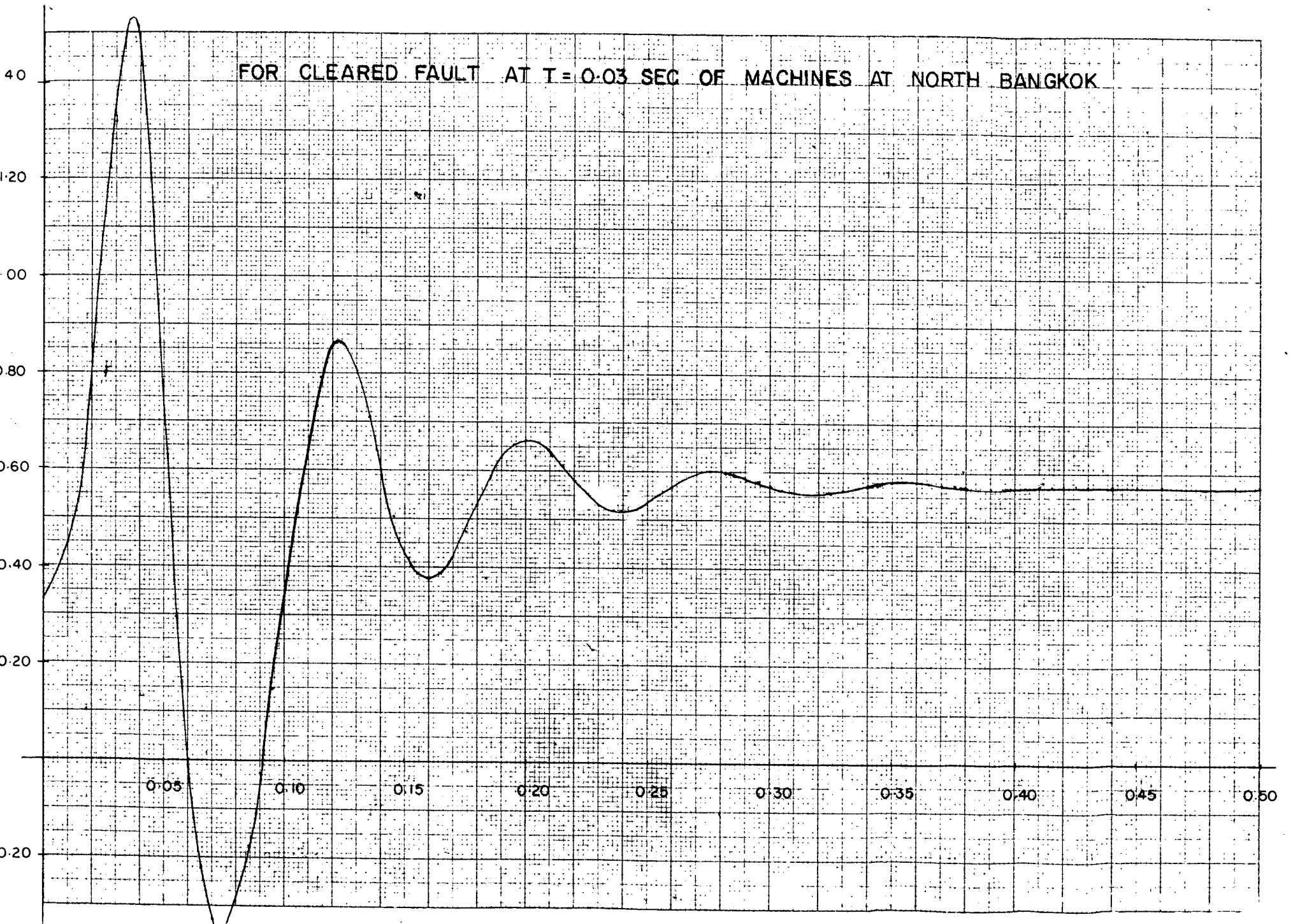



FOR SUSTAINED FAULT OF MACHINE AT BHUMIBOL DAM
ANGLE (RADIANS)
360
320
240
160
80
0
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
TIME (SEC.)

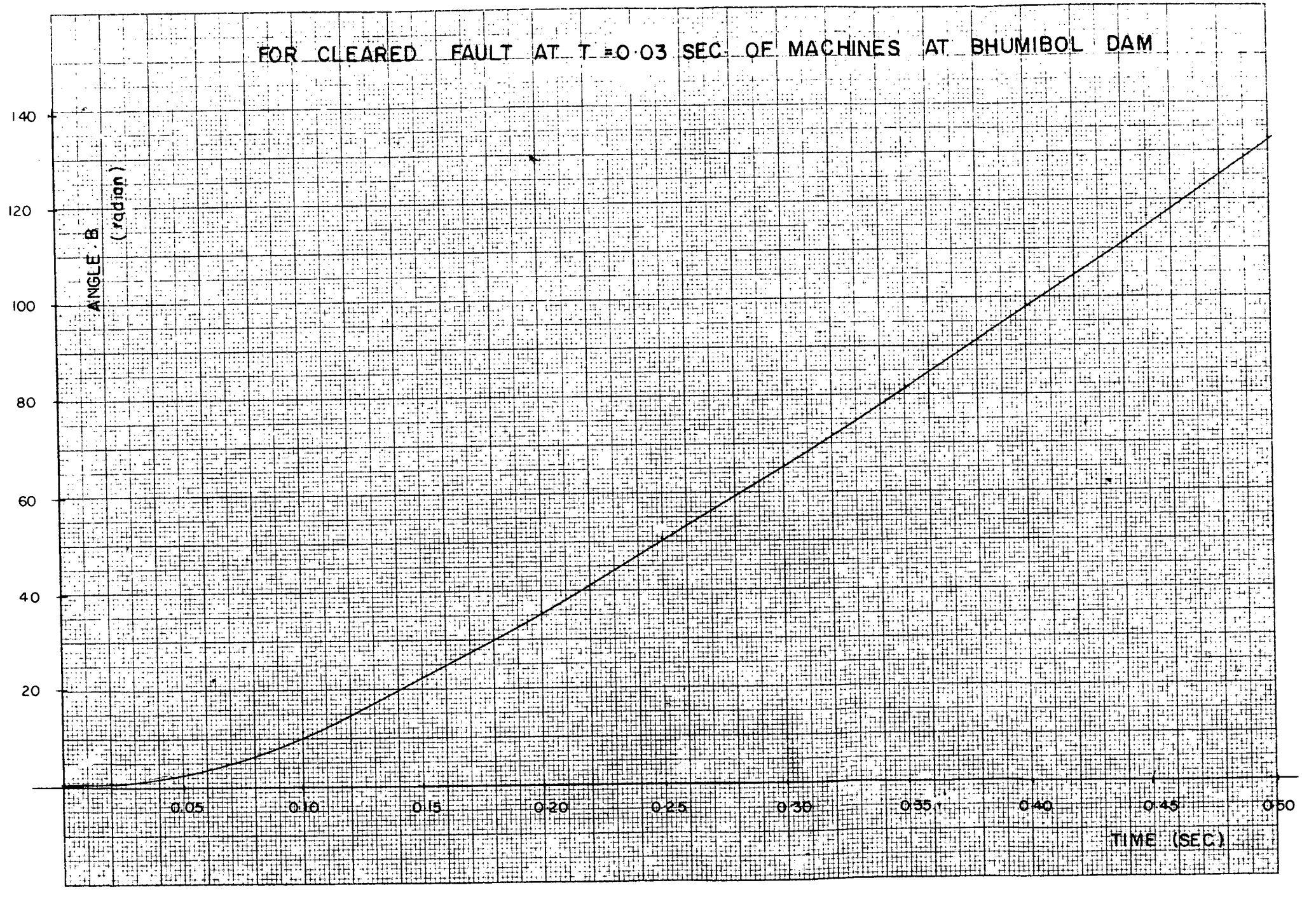
KEUFFEL & ESSER CO.

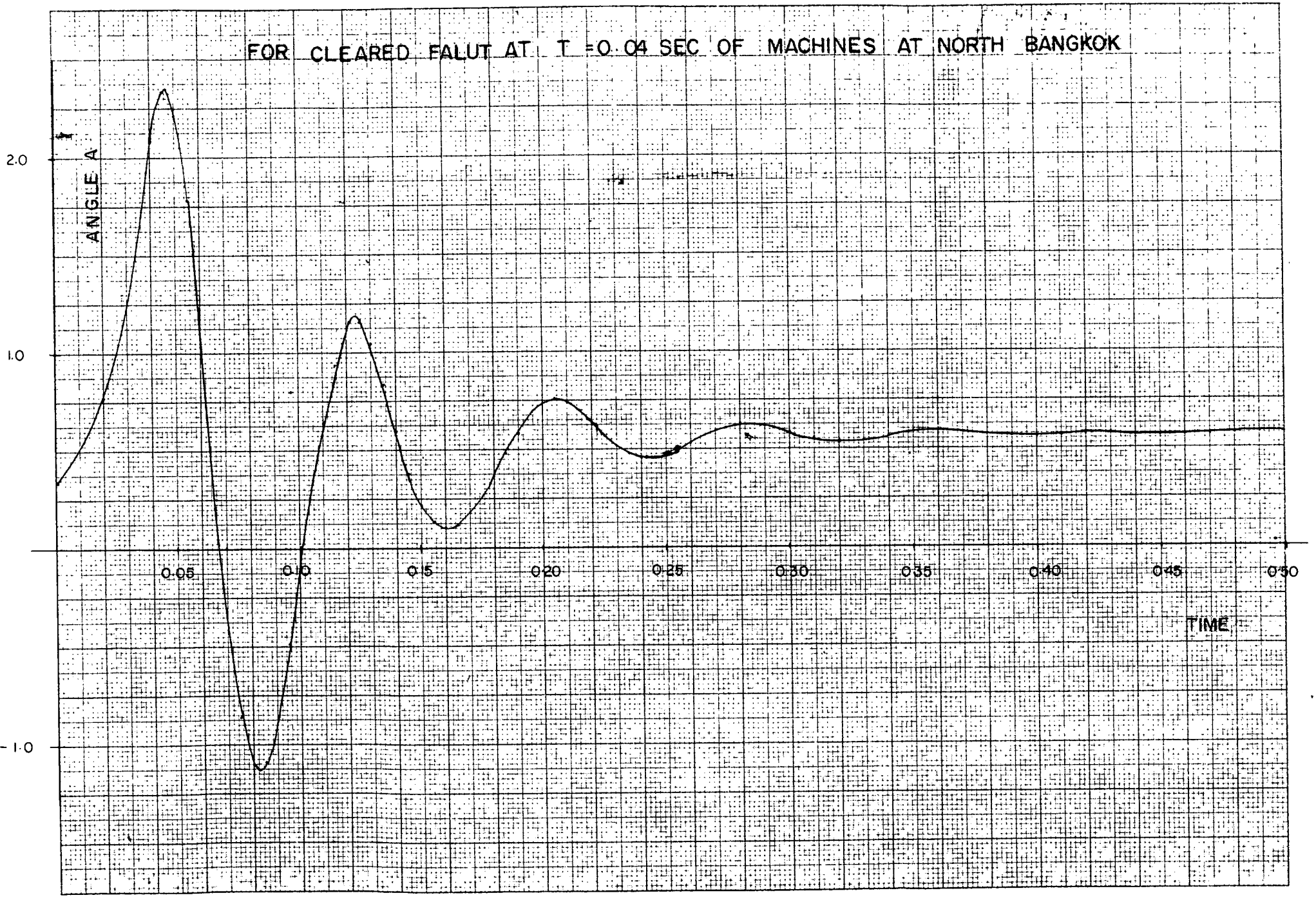




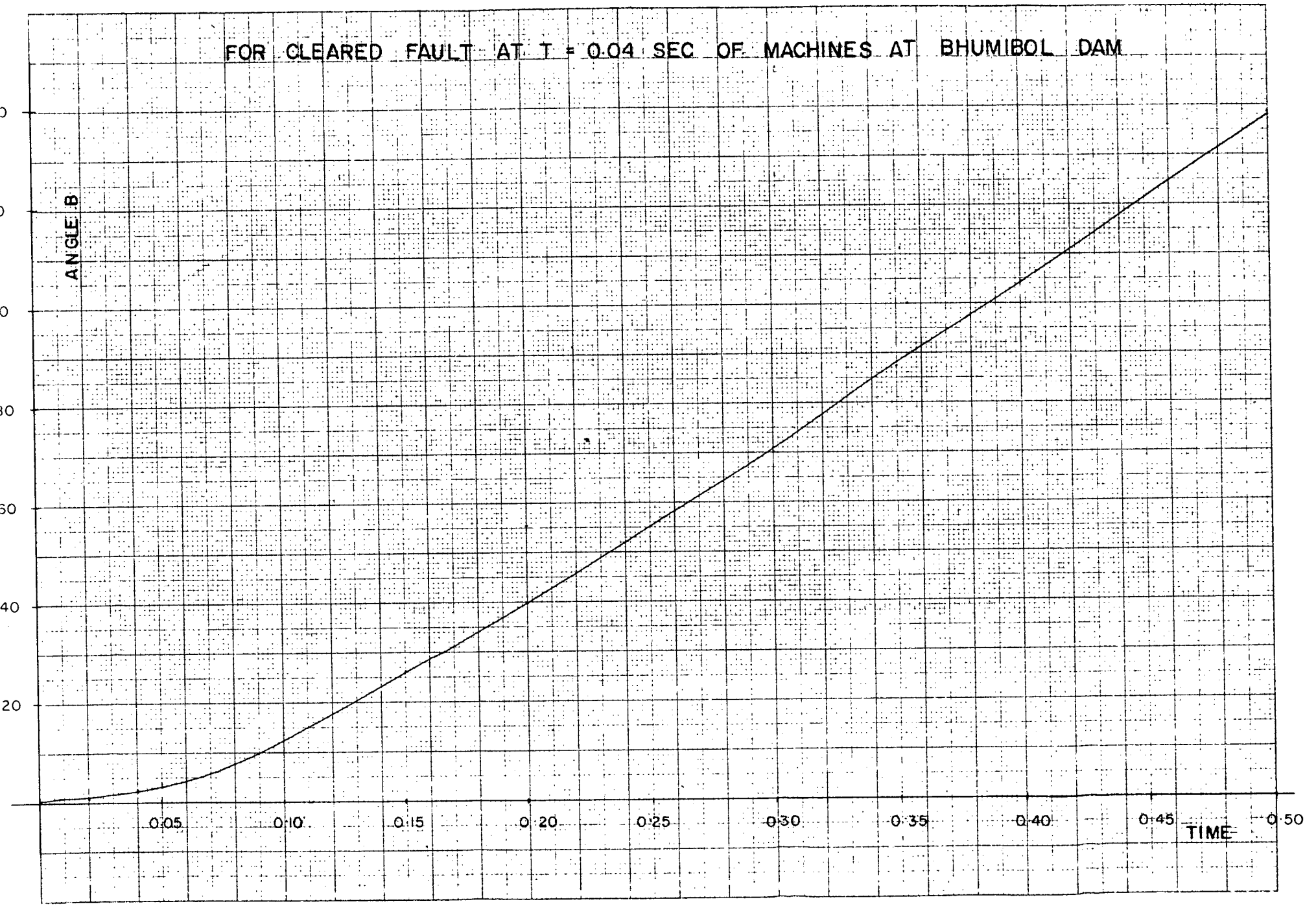


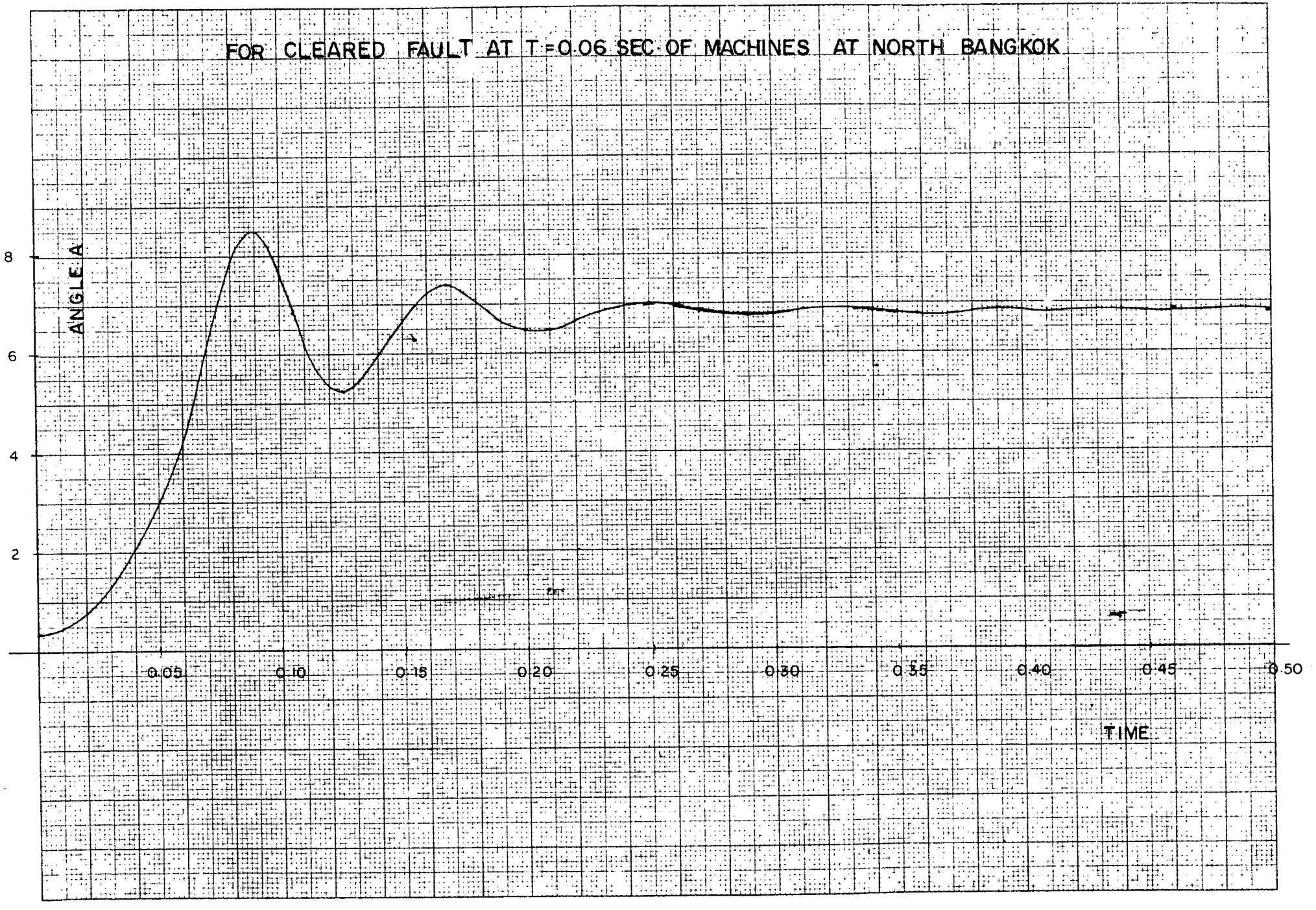
KEUFFEL & ESSER

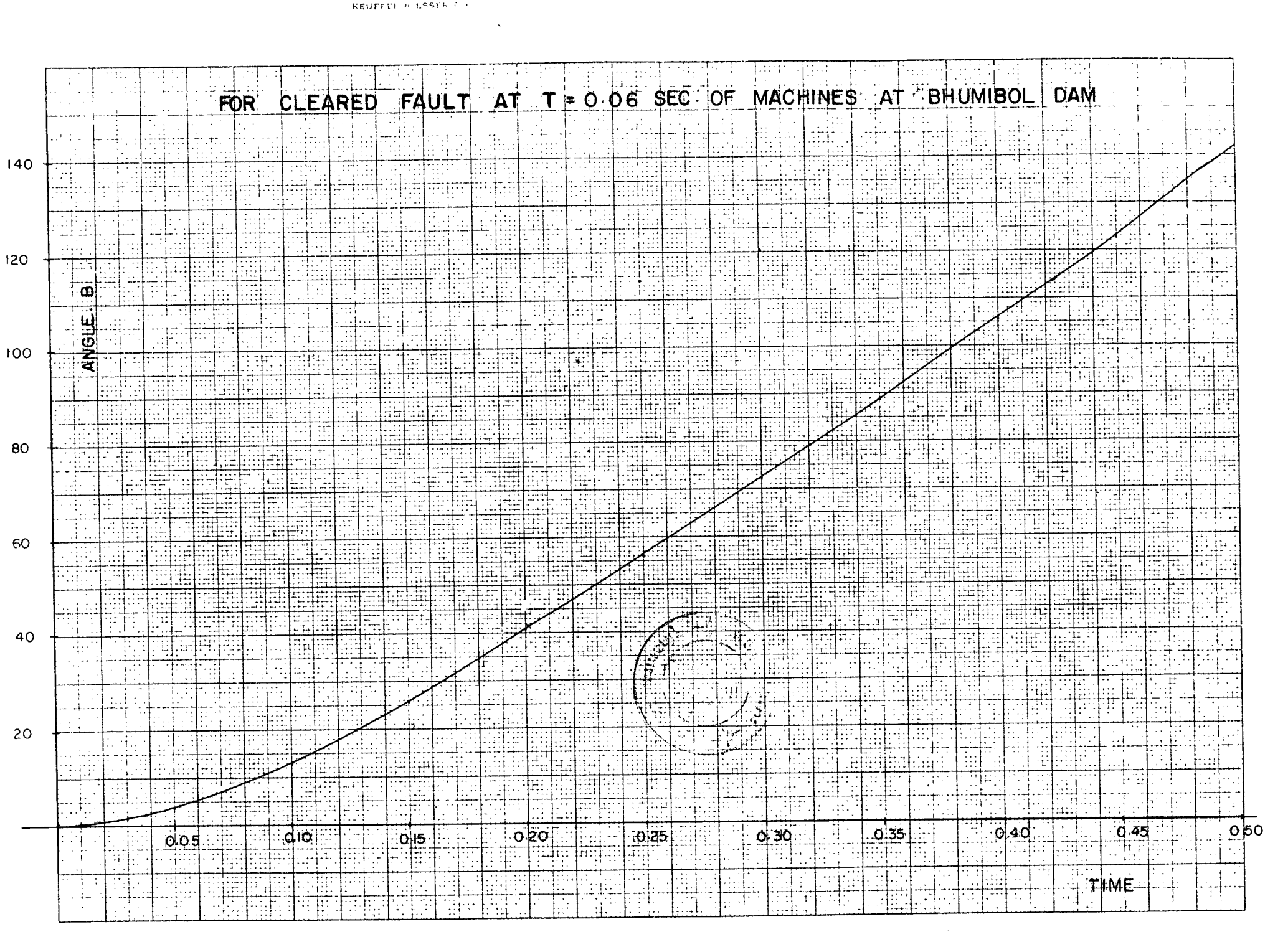




KRUFFEL & ESSER CO.



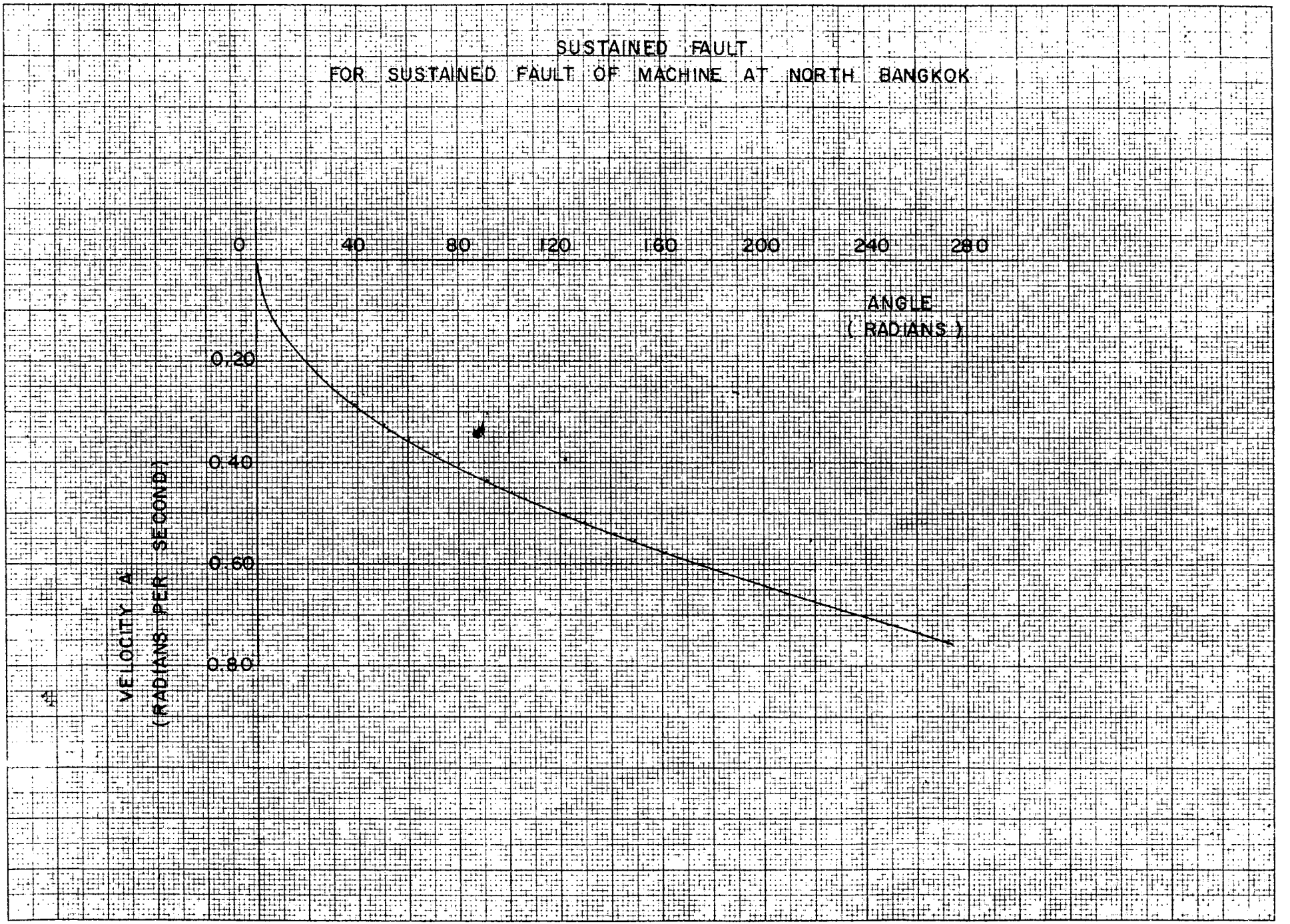


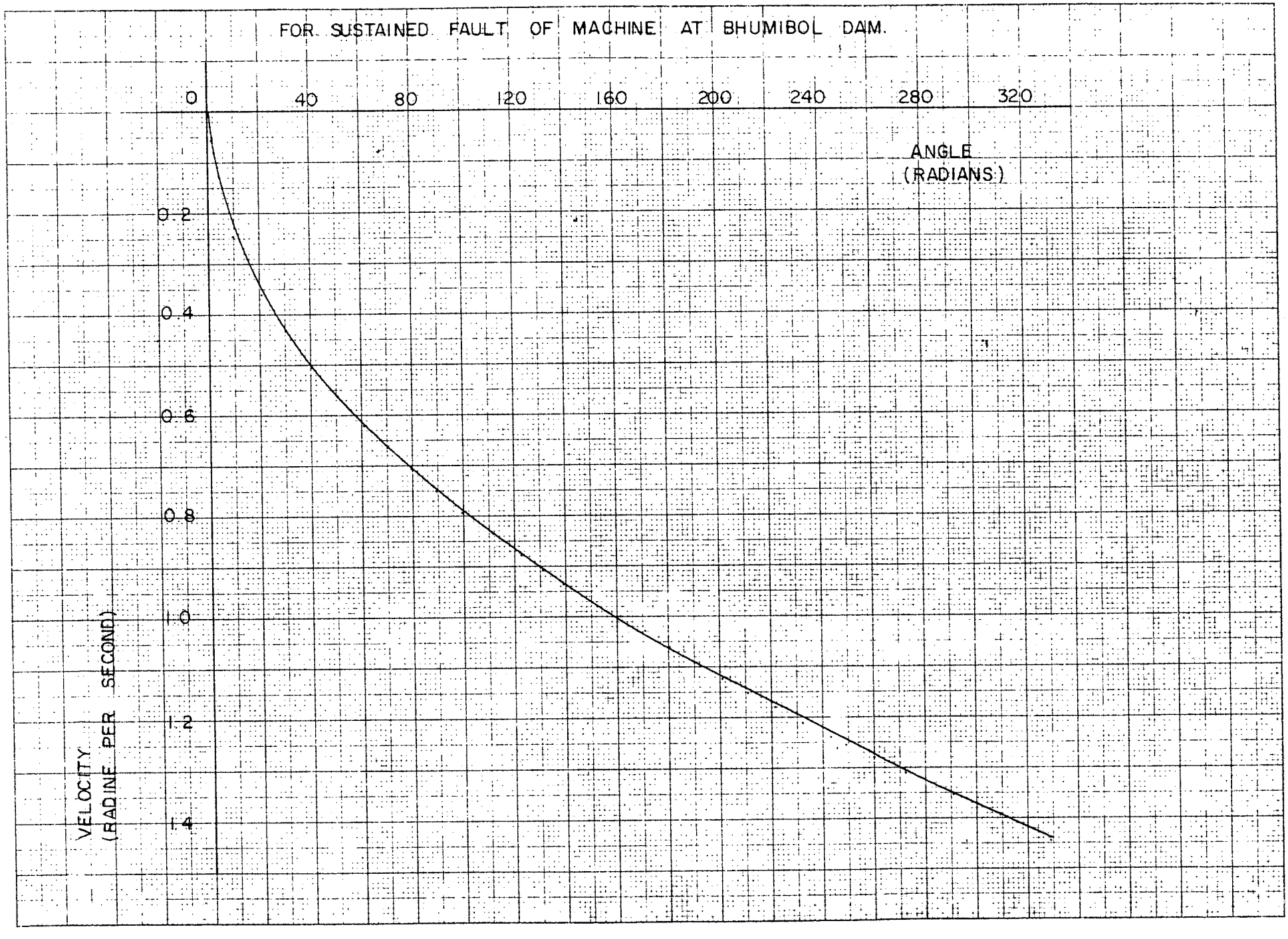

FOR CLEARED FAULT AT T = 0.06 SEC OF MACHINES AT BHUMIBOL DAM
ANGLE, B
140
120
100
80
60
40
20
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
TIME

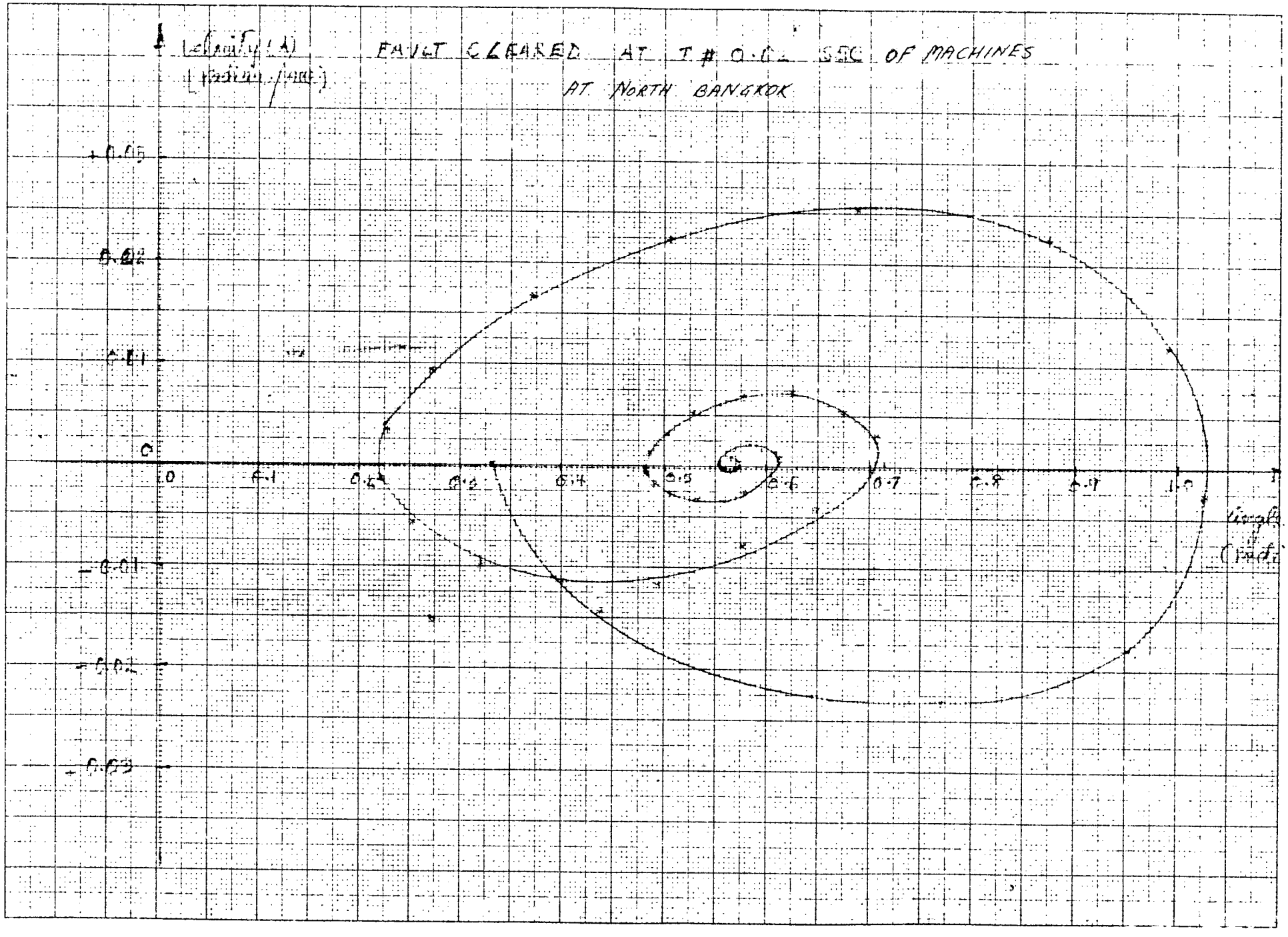
KEUFFEL & ESSER CO

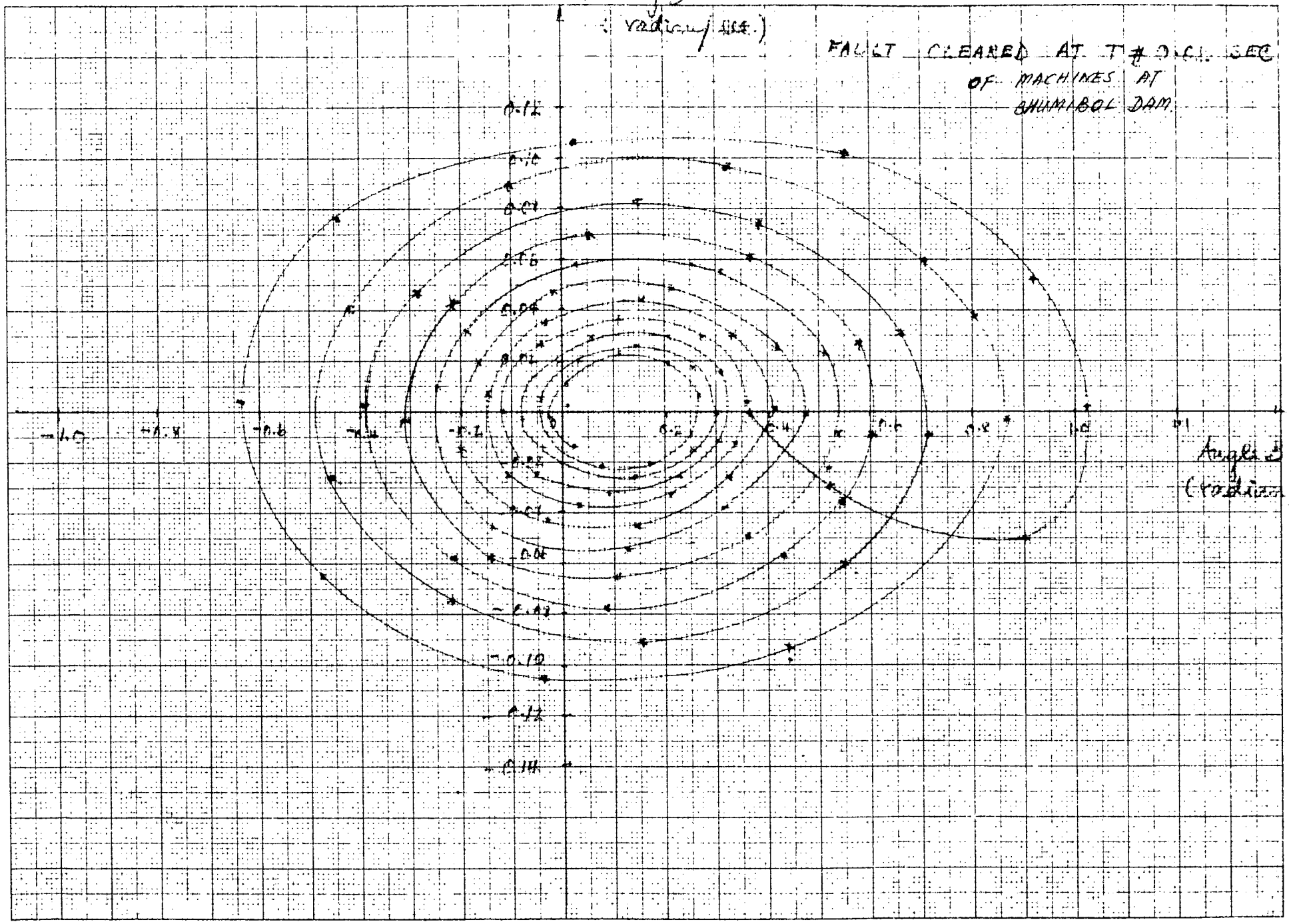
SUSTAINED FAULT

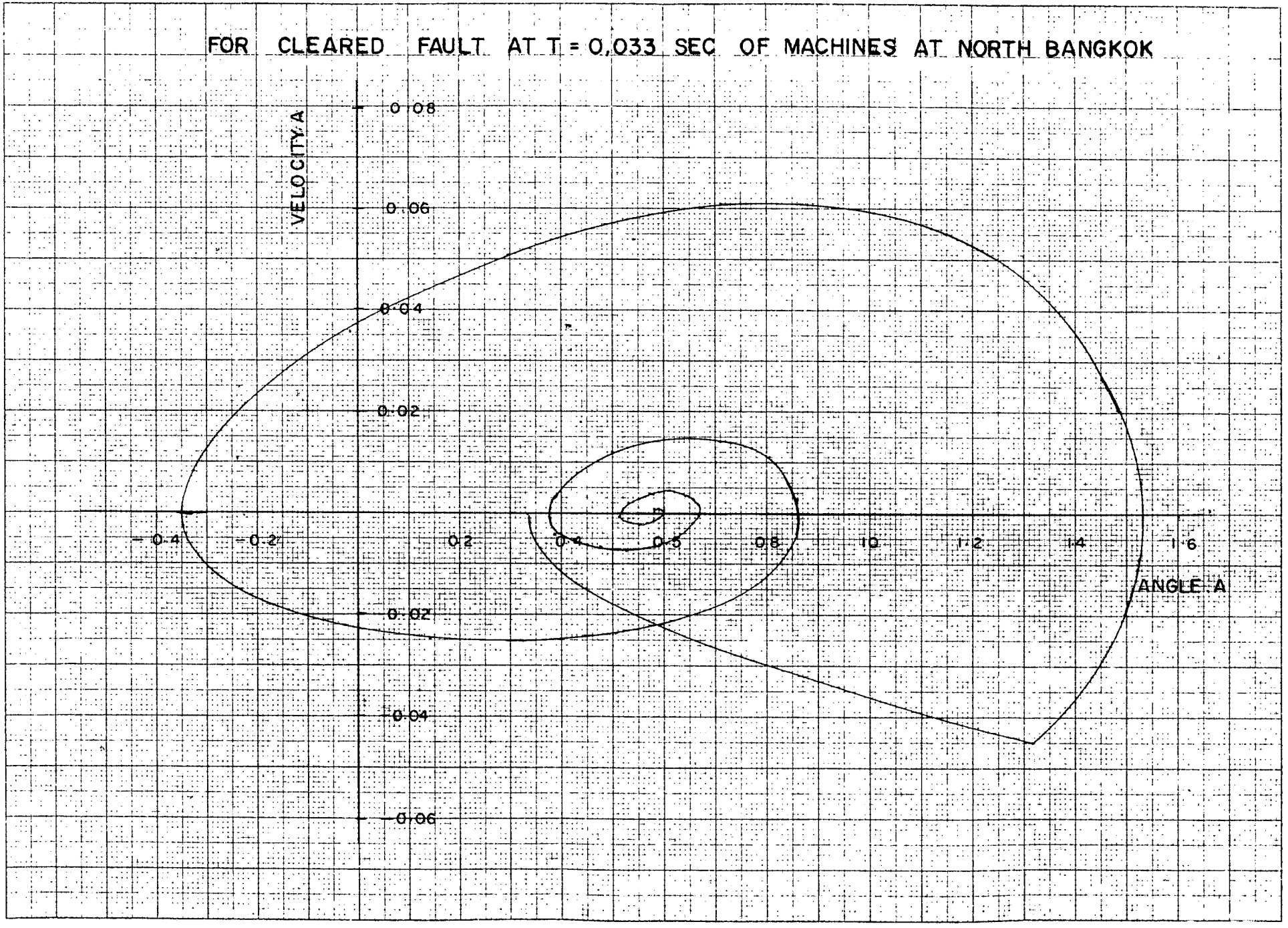
FOR SUSTAINED FAULT OF MACHINE AT NORTH BANGKOK



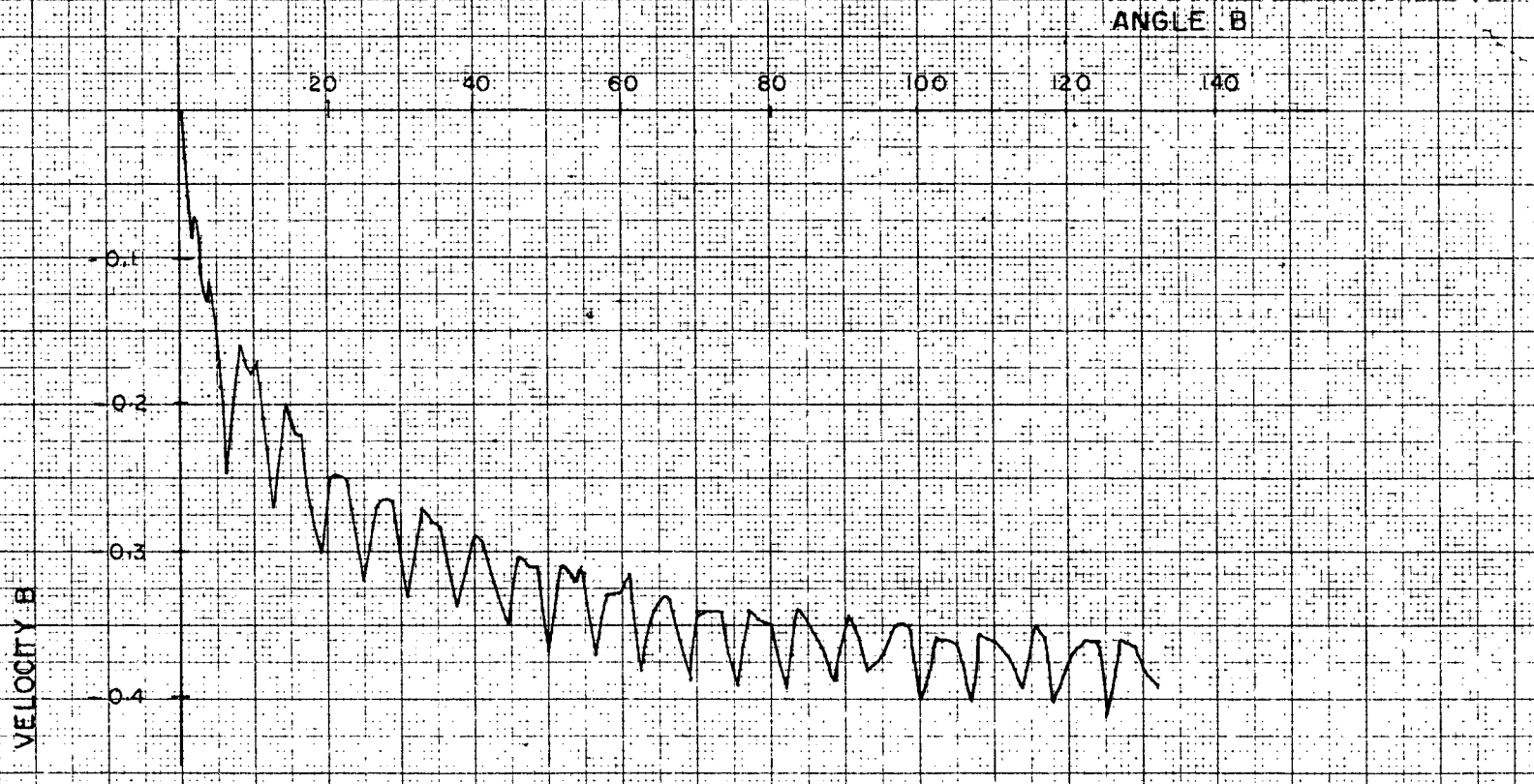

FOR SUSTAINED FAULT OF MACHINE AT BHUMIBOL DAM.
0
40
80
120
160
200
240
280
320
ANGLE
(RADIANS)
0.2
0.4
0.6
0.8
1.0
1.2
1.4
VELOCITY
(RADINE PER SECOND)


FAULT CLEARED AT T # 0.0 SEC OF MACHINES
AT NORTH BANGKOK
+0.05
0.02
0.01
0
-0.01
-0.02
-0.03
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

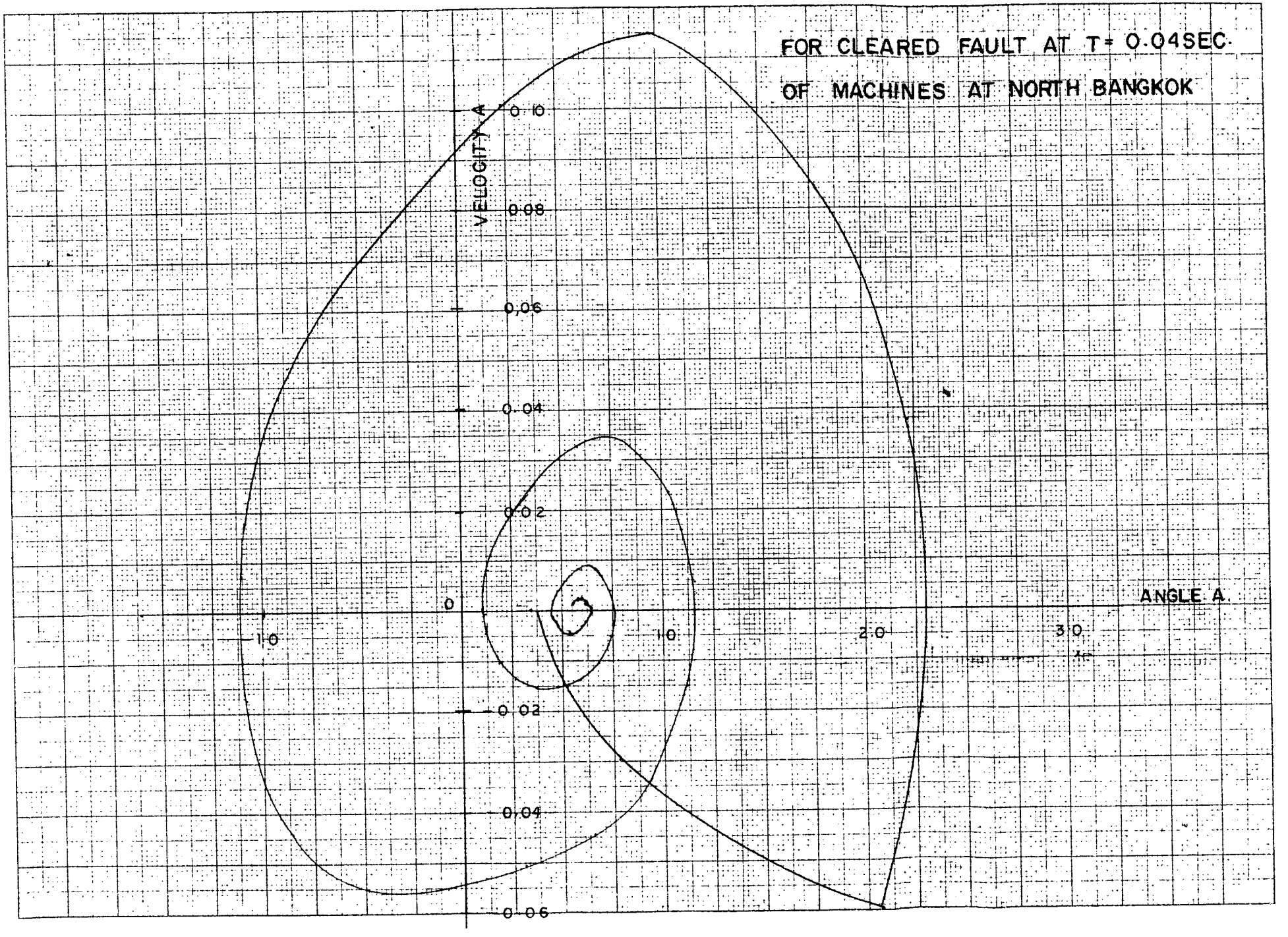

FAULT CLEARED AT T # 0.1 SEC
OF MACHINES AT
BHUMIBOL DAM
Angle δ
(radians)
0.12
0.10
0.08
0.06
0.04
0.02
-0.02
-0.04
-0.06
-0.08
-0.10
-0.12
-0.14
-1.0
-0.8
-0.6
-0.4
-0.2
0.2
0.4
0.6
0.8
1.0

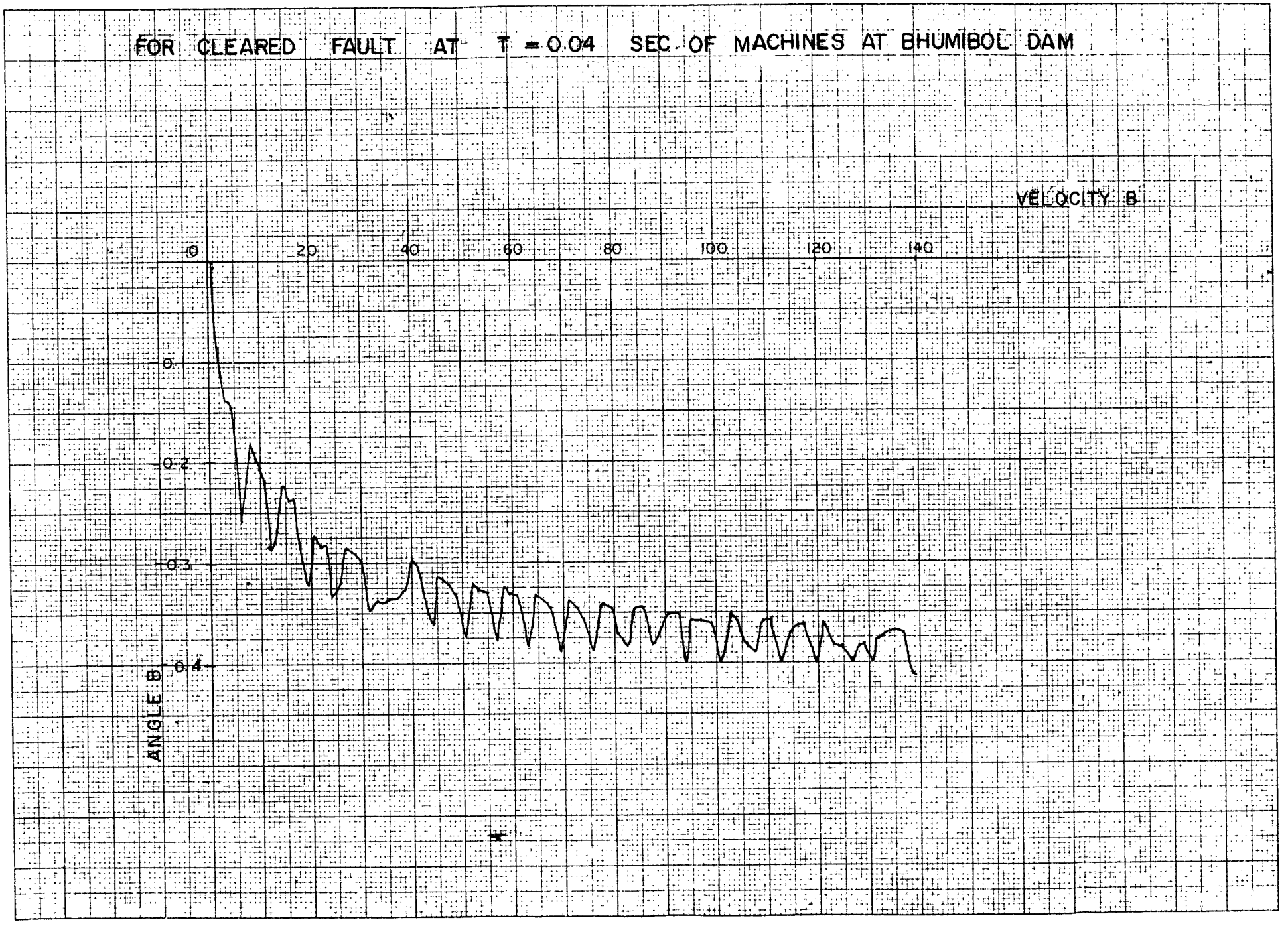

FOR CLEARED FAULT AT T = 0.033 SEC OF MACHINES AT NORTH BANGKOK
VELOCITY, A
0.08
0.06
0.04
0.02
-0.02
-0.04
-0.06
-0.4
-0.2
0.2
0.4
0.5
0.8
1.0
1.2
1.4
1.6
ANGLE, A

FOR CLEARED FAULT AT T=0·03 SEC. OF MACHINES AT BHUMIBOL DAM



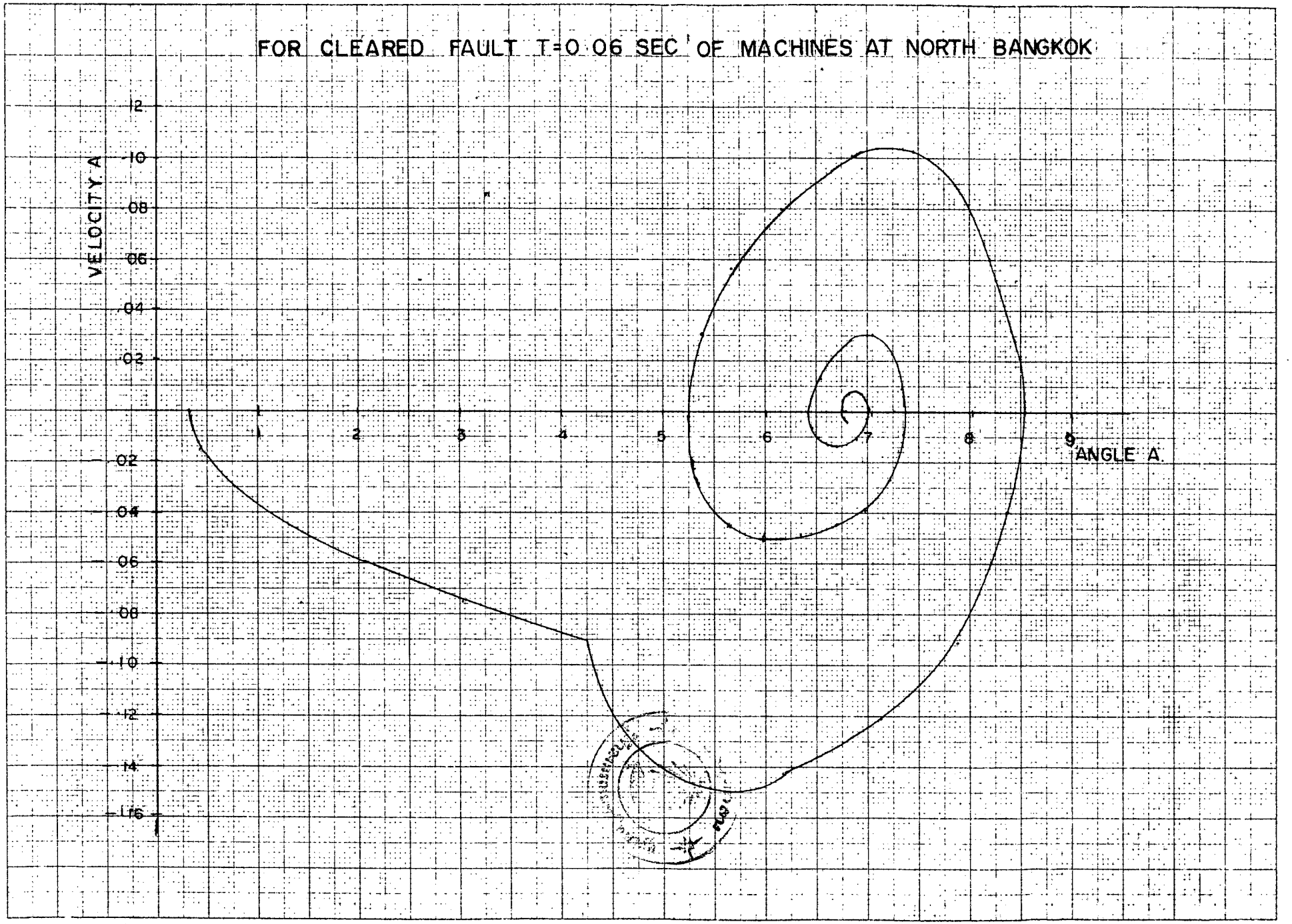
KEUFFEL & ESSER CO.

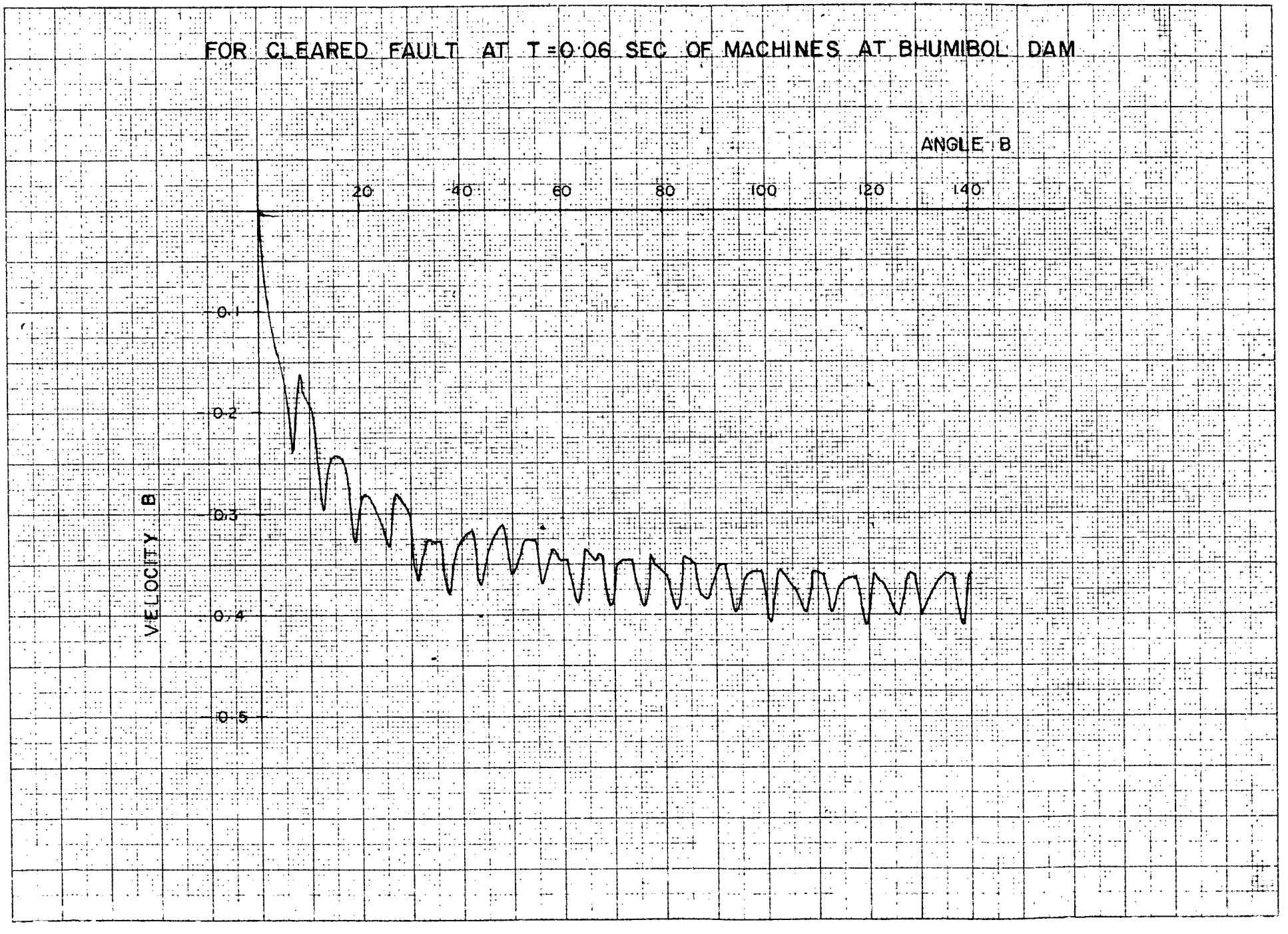




KEUFFEL & ESSER CO.

FOR CLEARED FAULT T=0.06 SEC OF MACHINES AT NORTH BANGKOK

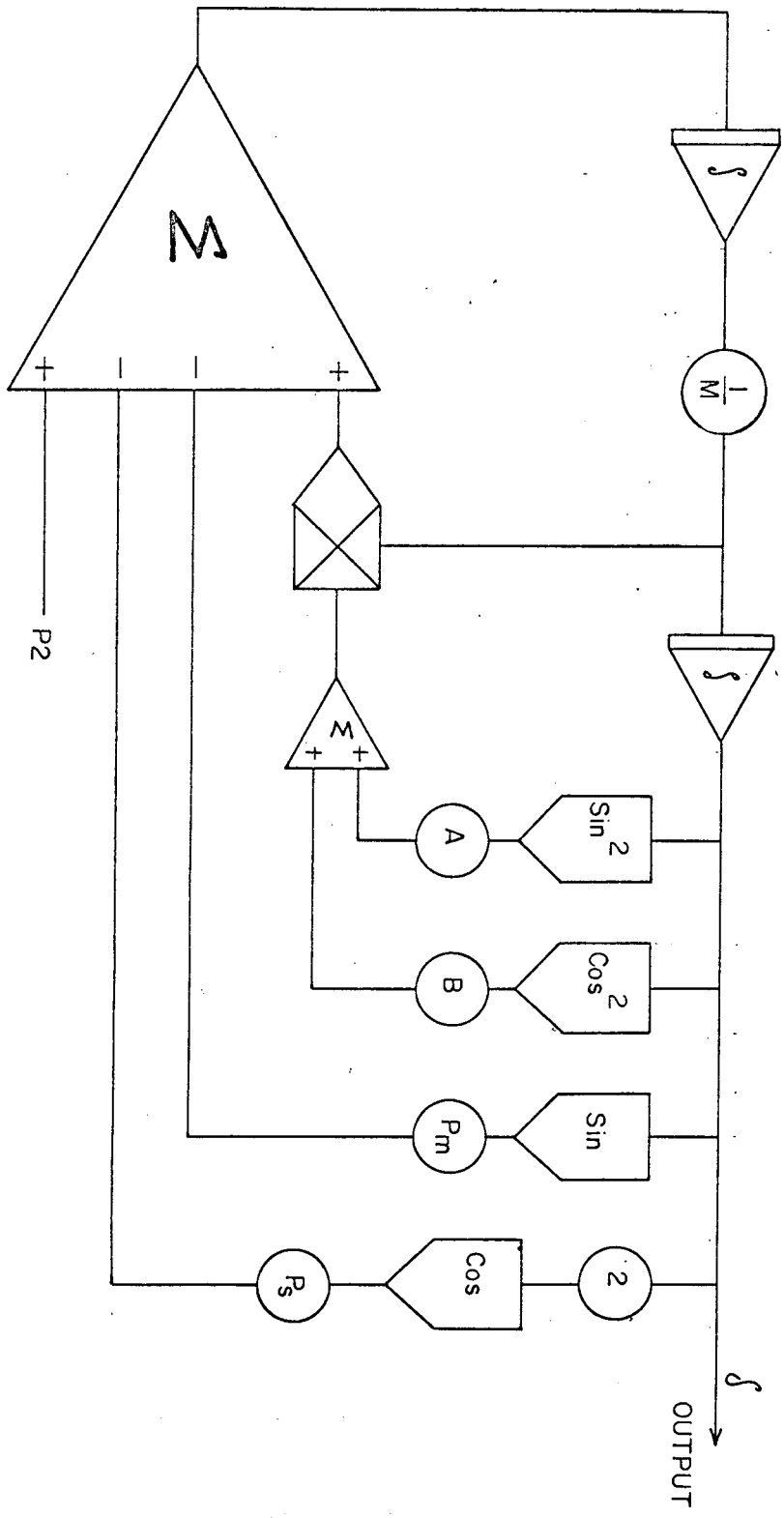




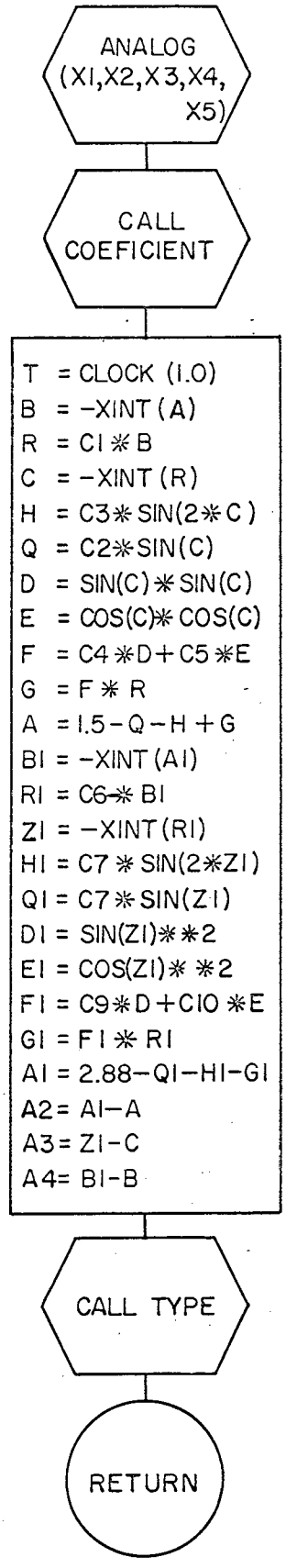
# Appendix E

## Power System Flow Chart Diagram

# ANALOG FLOW CHART



# ANALOG SURROUTINE IN POWER SYSTEM



```
      SUBROUTINE ARALEG(A,A1,Y2,Y3,Y4)
      DOUBLE PRECISION DBLES(200)
      REAL REALS(400)
      INTEGER INTS(200)
      COMMON DBLES,REALS,INTS
      CALL COEFF(1.,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10)
      T = CLOCK(1.0)
      B=-XINT(A)
      R=C1*B
      C=-XINT(R)
      F=C3*SIN(2*C)
      Q=C2*SIN(C)
      D=SIN(C)*SIN(C)
      E=COS(C)*COS(C)
      F=C4*D+C5*E
      G=F*A
      A=1.5C-Q-F+G
      B1=-XINT(A1)
      R1=C6*B1
      Z1=-XINT(R1)
      F1=C7*SIN(2*Z1)
      Q1=C7*SIN(Z1)
      D1=SIN(Z1)**2
      E1=COS(Z1)**2
      F1=C9*D+C10*E
      G1=F1*A1
      A1=2.5C-Q1-F1+G1
      A2=A1-A
      A3=Z1-C
      A4=B1-B
      CALL TYPE(7,T,A,C,G,A1,Z1,G1,A3,AF,A4)
      RETURN
      END
```