บรรณานุกรม

ภาษาไทย

ชาลี ตระการกูล. การออกแบบแม่พิมพ์ฉีด 1. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2536.

วิเชียร พาชยมัย. ระบบการจำแนกชนิดและการให้รหัสสำหรับชิ้นส่วนแม่พิมพ์ฉีดพลาสติก. วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2531.

ภาษาอังกฤษ

Austin, C. Moldflow design Principles. 1991.

Cracknell, P. S., and Dyson, R. W. Handbook of thermoplastics injection mold design. Great Britain : Blackie academic & professional, 1993.

Dassault Systemes, CATIA. Free form design API. 1996.

Dassault Systemes, Geometry interface quick reference. 1991.

DuBois, J. H., and Pribble, W. I. Plastics mold engineering handbook. 3rd ed. U.S.A. : Van Nostrad Reihold, 1978.

Ellis, J. W. Injection mold design. Petrleum and petrochemical college Chulalongkorn University, 1996.

Groover, M. P. Automation, production systems, and computer integrated manufacturing. U.S.A. : Prentice-Hall, 1987.

Sava, M., and Pusztai, J. Computer Numerical Control Programming. U.S.A. : Prentice-Hall, 1990.

Shina, S. G. Sucessful implementation of concurrent engineering products and processes. New York : Van Nostrad Reihold, 1994.

Zeid, I. CAD/CAM theory and practice. Singapore : McGraw-Hill, 1991.

ภาคผนวก ก.

ตารางแสดงรหัสเอ็มและจี

ตารางที่ ก.1 ตารางแสดงความหมายของรหัสเอ็ม

M-code	Mill/Turn
M00	Program Stop
M01	Optional Stop
M02	End of program and tape rewind
M03	Spindle start CW
M04	Spindle start CCW
M05	Spindle Stop
M06	Tool change
M08	Coolant ON
M09	Coolant OFF
M10	Polarity change(probing)
M11	Read X-measurement (probing)
M12	Read Y-measurement (probing)
M13	Read Z-measurement (probing)
M19	Spindle orient/stop
M20	-
M21	Mirrow image X
M22	Mirrow image Y
M23	Mirrow image OFF
M27	Reset register (probing)

ตารางที่ ก.1 ตารางแสดงความหมายของรหัสเอ็ม (ต่อ)

M30	End of program and memory rewind
M38	Measurement correction to CNC
M41	Low range(mill)
	Deviation measurement(probing)
M42	High range (mill)
	Comparative measurement (probing)
M44	Center location measurement(probing)
M45	Multisurface center locations(probing)
M46	Hole center(probing)
M47	Shaft center
M48	Override cancel OFF(mail)
M49	Override cancel ON(mail)
M60	-
M61	-
M62	-
M63	-
M64	-
M65	-
M66	-
M67	-
M68	-
M69	-
M70	-
M71	-
M74	-
M75	-

ตารางที่ ก.1 ตารางแสดงความหมายของรหัสเอ็ม (ต่อ)

M76	-
M77	-
M78	-
M80	-
M81	-
M82	-
M83	-
M84	-
M85	-
M88	-
M89	-
M90	-
M91	-
M93	-
M95	-
M96	-
M98	Go to subroutine
M99	Return from subroutine

ตารางที่ ก.2 ตารางแสดงความหมายของรหัสจี

G-code	Machining centers
G00	Positioning (rapid traverse)
G01	Linear interpolation(cutting feed)
G02	Circular/helical interpolation(clockwise)
G03	Circular/helical interpolation(counterclockwise)
G04	Dwell cycle
G09	Exact stop, deceleration
G10	Offset value setting by program
G14	-
G15	Polar coordinates command cancel
G16	Polar coordinates command
G17	X - Y plane selection
G18	Z - X plane selection
G19	Y - Z plane selection
G20	Inch data input (G70 on some systems)
G21	Metric data input (G71 on some systems)
G22	Safety Zone programming
G23	Programmed crossing though safety zone
G25	-
G27	Reference point return check
G28	Return to reference point
G29	Return from reference point
G30	Return to 2nd, 3rd, or 4th reference point
G31	Skip function
G32	-
G33	Thread cutting

ตารางที่ ก.2 ตารางแสดงความหมายของรหัสจี (ต่อ)

G37	Tool length automatic measurement
G38	Cutter diameter compensation vector change
G39	Cutter diameter compensation corner rounding
G40	Cutter diameter compensation cancel
G41	Cutter diameter compensation left
G42	Cutter diameter compensation right
G43	Tool length compensation + direction
G44	Tool length compensation - direction
G45	Tool offset increase
G46	Tool offset decrease
G47	Tool offset double increase
G48	Tool offset double decrease
G49	Tool length compensation cancel
G50	Scaling cancel
G51	Scaling
G51.1	Programmable mirror image cancel
G51.1	Programmable mirror image
G52	Local coordinate system setting
G53	Machining coordinate system selection
G54	Work coordinate system 1 selection
G55	Work coordinate system 2 selection
G56	Work coordinate system 3 selection
G57	Work coordinate system 4 selection
G58	Work coordinate system 5 selection
G59	Work coordinate system 6 selection
G60	Single direction positioning

ตารางที่ ก.2 ตารางแสดงความหมายของรหัสจี (ต่อ)

G61	Exact stop mode
G62	Automatic corner override mode
G63	Tapping mode
G64	Cutting mode
G65	Transfer to subroutine (macro call)
G66	-
G68	coordinate system rotation
G69	coordinate system rotation cancel
G70	-
G71	-
G73	Peak-drilling canned cycle
G74	Counter tapping canned cycle
G76	Fine boring canned cycle
G80	Canned cycle cancel
G81	Drilling/spot drilling canned cycle
G82	Drilling/counter boring canned cycle
G83	Peak drilling canned cycle
G84	Tapping canned cycle
G85	Boring canned cycle
G86	Boring canned cycle
G87	Back boring canned cycle
G88	Boring canned cycle
G89	Boring canned cycle
G90	Absolute data input
G91	Incremental data input
G92	Work coordinate change-absolute zero point

ตารางที่ ก.2 ตารางแสดงความหมายของรหัสจี (ต่อ)

G93	-
G94	Feed per minute
G95	Feed per revolution
G96	Constant surface speed control
G97	Constant surface speed control cancel
G98	Canned cycle -return to initial level
G99	Canned cycle -return to R- level

ภาคผนวก ข

โปรแกรมที่เขียนขึ้นเพื่อใช้ในวิทยานิพนธ์

โปรแกรมพรีโปรเซสเซอร์ MKPT

```
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      CHARACTER*25 PROJ
      CHARACTER*8  GROUP,USER,ACC,PASSW,FILE
      CHARACTER*80 TIT
      REAL CO_X,CO_Y,CO_Z
      INTEGER TEP
      DIMENSION  PT0(3),V1(3),V2(3)
      DIMENSION  PT1(3),PT2(3)
C
C---- FILL IN PROJECT, GROUP, USER, PASSWORD AND FILE DDNAME.
C---- THE FILE DDNAME MUST BE REFERENCED IN THE MEMBER 'MODEL'
C---- OF THE SYSDYN FILE.
C
      DATA PROJ /'        '/
      DATA GROUP/'catia   '/
      DATA USER /'catadm  '/
      DATA ACC  /'        '/
      DATA PASSW/'catadm  '/
      DATA FILE /'CUUA    '/
```

```
C
   10 FORMAT(I1,T3,F5.2,T9,F5.2,T14,F6.2)
   20 FORMAT(I2,T4,F5.2,T10,F5.2,T15,F6.2)
   30 FORMAT(I3,T5,F5.2,T11,F5.2,T16,F6.2)
C
      OPEN(3,FILE='SCANNER',STATUS='OLD')
C
      CALL CATGEO
      CALL GLOGON (PROJ,GROUP,USER,ACC,PASSW,IER,*100)
      CALL GIFALL ('MODEL   ',FILE,ISHR,ICATI,IER,*100)
C
C--------------------------------------------------------------------
C     NUMERICAL VALUES
C--------------------------------------------------------------------
C
      SIZE  = 0.D0
      UNIT  = 1.D0
      ITRU  = 1
      LELEM = 0
      LDESC = 0
      MNUM  = 1
      PI    = DACOS (-1.D0)
C
C--------------------------------------------------------------------
C     CREATION OF MODEL
C--------------------------------------------------------------------
      CALL GIMCRW (MNUM,SIZE,UNIT,LELEM,LDESC,IER,*100)
      IMOD = 1
      CALL GICAXM (IMOD,IER,*100)
C
C-------------------------------------------------------------------
C     BODY
C-------------------------------------------------------------------
C
      TEP = 1
   50 IF(TEP.LT.9) THEN
        READ(3,10,END=60) TEP,CO_X,CO_Y,CO_Z
      ELSE
         IF(TEP.LT.99) THEN
          READ(3,20,END=60) TEP,CO_X,CO_Y,CO_Z
         ELSE
          READ(3,30,END=60) TEP,CO_X,CO_Y,CO_Z
         END IF
      END IF

      IF(CO_Z.LT.0) THEN
       PT1(1) = CO_X
       PT1(2) = CO_Y
       PT1(3) = CO_Z
       CALL GIWPT (MNUM,PT1,JPT1,IER,*100)
      END IF
      GOTO 50
C-------------------------------------------------------------------
C     FILE THE MODEL
C-------------------------------------------------------------------
   60 TIT = 'POP_EYE'
```

```
      CALL GIMWRI (MNUM,FILE,TIT,ITRU,ITRU,IER,*100)
C
C-----------------------------------------------------------------------
C      END : ERROR MESSAGE (OR OK)
C-----------------------------------------------------------------------
 100  CALL GILERR (IER)
      STOP
      END
```

โปรแกรมพรีโปรเซสเซอร์ MKSUR

```
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      CHARACTER*17 PROJ
      CHARACTER*8  GROUP,USER,ACC,PASSW,FILE
      CHARACTER*80 TIT,INFILE
      CHARACTER*1  INDEX
      CHARACTER*33 CARD
      REAL*8       PT(4,13,13)           ! USED BY GCWBSF
      REAL*8       PTS(4,13)
      REAL*8       PT1(3),PT2(3),PT3(3),DEVMAX
      REAL*8       SMAX,SMIN,STEP
      INTEGER*2    IBLBS(4,10)           !B-SPLINE SURFACE BLOCK.
      REAL*8       BLBS(10)              !B-SPLINE SURFACE BLOCK.
      EQUIVALENCE  (BLBS,IBLBS)          !B-SPLINE SURFACE BLOCK.
      INTEGER*2    IBLBSS(4,5)           !B-SPLINE SURFACE BLOCK.
      REAL*8       BLBSS(5)              !B-SPLINE SURFACE BLOCK.
      EQUIVALENCE  (BLBSS,IBLBSS)        !B-SPLINE SURFACE BLOCK.
      INTEGER*4    NPU(200)              !B-SPLINE SURFACE BLOCK.
      INTEGER*2    LEN                   !B-SPLINE SURFACE BLOCK.
      INTEGER*4    NP,NPUU,NPVV
      INTEGER*2    NPT,NPTU,NPTV
      INTEGER*2    LENS
      INTEGER*2    NDEGU,IRATU,ITYPNU          !B-SPLINE SURFACE BLOCK.
      INTEGER*2    NDEGV,IRATV,ITYPNV          !B-SPLINE SURFACE BLOCK.
      REAL*8       VDEBU,VFINU,VDEBV,VFINV !B-SPLINE SURFACE BLOCK.
      INTEGER*4    JCST,IER
      INTEGER*4    ISAVPT(1000),JXPT
      CHARACTER*1  C1,C2,C3,C4,C5,C6          !FOR READ Y/N.
      CHARACTER*3  STR                        !FOR OLD,NEW
      DATA STR /'OLD'/
      DATA PROJ /'        '/
      DATA GROUP/'catia   '/
      DATA USER /'catadm  '/
      DATA ACC  /'        '/
      DATA PASSW/'catadm  '/
      DATA FILE /'CUUA    '/
C
C +-------------------- DEFINE VARIABLES ----------------------+
C
C     READ (03,'(7X,BN,I8)')LELEM
C     READ (03,'(7X,BN,I8)')LDESC
C
C +-- LELEM read from MKEBSPL ALLOCALE (MAX 280000)------------+
C +-- LDESC read from MKEBSPL ALLOCATE ( about LELEM * 4 )-----+
C
      PI    = DACOS (-1.D0)
      SIZE  = 0.D0
      UNIT  = 1.D0
      ITRU  = 1
      LELEM = 0
      LDESC = 0
      MNUM  = 1
      IMOD  = 1
      LENS  = 5
      LEN    = 10      ! 10 lines of Bspline block.
      NDEGU  = 3       ! 1 - 15 CAN USED FOR THIS VARIABLE.
      IRATU  = 3       ! 3 = polynomial form 4 = rational form
      ITYPNU = 1       ! 0 = non-uniform 1 = uniform.
      VDEBU  = -1000.D0
      VFINU  = 1000.D0
C
      NDEGV  = 3     ! 1 - 15 CAN USED FOR THIS VARIABLE.
      IRATV  = 3    ! 3 = polynomial form 4 = rational form
      ITYPNV = 1    ! 0 = non-uniform 1 = uniform.
      VDEBV  = -1000.D0
      VFINV  = 1000.D0
C
      IBLBS(1,1) = LEN
      IBLBS(3,1) = NDEGU
      IBLBS(2,2) = IRATU
      IBLBS(1,3) = ITYPNU
      BLBS(4)    = VDEBU
      BLBS(5)    = VFINU
      IBLBS(3,6) = NDEGV
      IBLBS(2,7) = IRATV
      IBLBS(1,8) = ITYPNV
      BLBS(9)    = VDEBV
      BLBS(10)   = VFINV
C
      IBLBSS(1,1) = LENS
      IBLBSS(3,1) = NDEGU
      IBLBSS(2.2) = IRATU
      IBLBSS(1,3) = ITYPNU
      BLBSS(4)    = VDEBU
      BLBSS(5)    = VFINU
C
C +------------ INPUT MODEL NAME AND FILE NAME ----------------+
C
      WRITE (*,'(" CREATE THE NEW MODEL (Y/N) ?")')
      WRITE (*,'(" Y if create new model.")')
      WRITE (*,'(" N if added  old model.")')
      CALL YESNO(C1)
      IF(C1 .EQ.'Y') THEN
        STR = 'NEW'
      ENDIF
      WRITE (*,'(" INPUT DATA FILE NAME ?")')
      READ  (*,'(A80)') INFILE
      WRITE (*,'(" INPUT ",A3," MODEL NAME ?")')STR
      READ  (*,'(A80)') TIT
      WRITE (*,'(" DO YOU WANT TO CREATE B-SPLINE TOO (Y/N) ?")')
      CALL YESNO(C2)
      WRITE (*,'(" CALCULATING... PLEASE WAIT.")')
C
      CALL CATGEO
      CALL GLOGON (PROJ,GROUP,USER,ACC,PASSW,IER,*1000)
      CALL GIFALL ('MODEL   ',FILE,ISHR,ICATI,IER,*1000)
C
      IF(C1 .EQ.'Y') THEN     !yes, create new model.
        CALL GIMCRW (MNUM,SIZE,UNIT,LELEM,LDESC,IER,*1000)
        CALL GICAXM (IMOD,IER,*1000)
      ELSE                    !no, added old model.
        CALL GIMSIZ (FILE,TIT,LELEM1,LDESC1,IER,*1000)
        CALL GIMREA (MNUM,FILE,TIT,LELEM,LDESC,IER,*1000)
      ENDIF
C
C +------------------------- BODY -----------------------------+
C
      OPEN(1,FILE=INFILE,STATUS='OLD')
      I = 1
      J = 1
      NPVV = 0
      NPU(I) = 0
 100  READ(1,'(A1,A33)',END=200) INDEX,CARD       ! READ SCANNED POINT
C
      IF (INDEX.EQ.'A') THEN
        READ(CARD,'(F10.4,1X,F10.4,1X,F10.4)')(PT(K,J,I),K = 1,3)
        PT(4,J,I) = 1.D0
        WRITE(*,'(F10.4,1X,F10.4,1X,F10.4)')(PT(K,J,I),K = 1,3)
        J = J+1
        NPU(I) = NPU(I)+1
      ENDIF
      IF (INDEX.EQ.'i') THEN
        IF (C2 .EQ. 'Y') THEN
          NP = NPU(I)
          NPT = NP
          DO 40, L = 1,NP
            PTS(1,L) = PT(1,L,I)
            PTS(2,L) = PT(2,L,I)
```

```
             PTS(3,L) = PT(3,L,I)
             PTS(4,L) = 1.D0
   40       CONTINUE
           IBLBSS(2,1) = NPT
           CALL GCWBCV (MNUM,NP,PTS,BLBSS,IBCV,IER,*1000)
         ENDIF
         I = I+1
         J = 1
         NPVV = NPVV+1
         NPU(I) = 0
      ENDIF
      IF ((INDEX.EQ.'d').OR.(INDEX.EQ.'.')) THEN
         GOTO 200
      ENDIF
      GOTO 100
C
 200  NPUU = NPU(1)
      NPVV = NPVV+1
      NPTU = NPUU
      NPTV = NPVV
      IBLBS(2,1) = NPTU
      IBLBS(2,6) = NPTV
      IF((NPTU.EQ.0).OR.(NPTV.EQ.0)) THEN
        GOTO 300
      ELSE
        CALL GCWBSF (MNUM,NPUU,NPVV,PT,BLBS,IBCV,IER,*1000) !MAKE B-SUR.
      ENDIF
C
```

```
C +------------------- FILE OF THE MODEL ----------------------+
C
 300  CONTINUE
C
      IF( C1 .NE.'Y' ) THEN
        WRITE(*,'(" CHANGE THE MODEL NAME (Y/N) ?")')
        CALL YESNO(C4)
        IF( C4 .EQ.'Y' ) THEN
          WRITE(*,'("INPUT NEW NAME ?")')
          READ(*,'(A80)')TIT
        ENDIF
      ENDIF
      WRITE(*,'(" WRITING...")')
      CALL GIMWRI (MNUM,FILE,TIT,ITRU,ITRU,IER,*1000)
C +---------------- END : ERROR MESSAGE (OR OK) ---------------+
1000  CALL GILERR (IER)
      STOP
      END
C +---------------------- SUBROUTINE -------------------------+
      SUBROUTINE YESNO(CH)
      CHARACTER*1 CH
      READ (*,'(A1)')CH
      DO WHILE (CH .NE.'Y'.AND. CH .NE.'N')
        WRITE(*,'(" Y OR N ONLY! TRY AGAIN.")')
        READ (*,'(A1)')CH
      END DO
      RETURN
      END
```

โปรแกรมโพสโปรเซสเซอร์ APT.for

```
C*****THIS PROGRAM CONVERTS CATIA APT OUTPUT TO G-CODE INPUT *****C
C*****Note 1)Unit is in MM.                  *****C
C*****      2)Input file must end with END  *****C
      INTEGER LINENO,FRMFLG,BLOCK,SPIN,ARCNO,START,POINT
      REAL ZMAX,ZCURR
      CHARACTER*70 LINDAT
      CHARACTER*6 APTWRD
      CHARACTER*10 FILENAME
      COMMON/INTS/LINENO,FRMFLG,BLOCK,SPIN,ARCNO
    1 FORMAT(A6,A70)
    2 FORMAT('N',I5,' SKIP TO 30000')
    3 FORMAT('N',I5,' ;BLOCK ',I3)
    4 FORMAT('N',I5)
    5 FORMAT('N',I5,' GO Z',F8.4)
    6 FORMAT('N',I5,'M05')
    7 FORMAT('N',I5,'M03')
    8 FORMAT(A10)
      WRITE(*,*) 'Please type APT_filename :'
      READ(*,8) FILENAME
      OPEN(3,FILE=FILENAME,STATUS='OLD')
      OPEN(4,FILE='DYNFIL',STATUS='NEW')
      POINT=0
      START=0
      LINENO=0
      ARCNO=0
      FRMFLG=0
      BLOCK=2
```

```
READ(3,1) APTWRD,LINDAT
DO WHILE(APTWRD.NE.'FINI')
 IF(LINENO.LT.99989) THEN
  CALL PROCESS(APTWRD,LINDAT,ZMAX,ZCURR)
 ELSE
  WRITE(4,5) LINENO,ZMAX
  LINENO=LINENO+1
  IF(SPIN.EQ.1) THEN
   WRITE(4,6) LINENO
   LINENO=LINENO+1
  ENDIF
  WRITE(4,2) LINENO
  LINENO=100000
  WRITE(4,4) LINENO
  LINENO=0
  WRITE(4,3) LINENO,BLOCK
  BLOCK=BLOCK+1
  LINENO=LINENO+1
  IF(SPIN.EQ.1) THEN
   WRITE(4,7) LINENO
   LINENO=LINENO+1
  ENDIF
  WRITE(4,5) LINENO,ZCURR
  LINENO=LINENO+1
 ENDIF
END DO
CLOSE(3)
CLOSE(4)
```

```
      END
C*****END MAIN PROGRAM*****C

      SUBROUTINE PROCESS(APWRD,LIDAT,ZMX,ZCUR)

      INTEGER LINNO,FROMFL,BLK,SPI,TOOLNO,ARNO,RPM,STARTT,POINTT
      CHARACTER*70 LIDAT
      CHARACTER*6 APWRD
      REAL X,Y,Z,FEDRAT,ZMX,ZCUR
      REAL R,XCHECK,YCHECK,X_CEN,Y_CEN,PH_V,PH_R
      REAL DIRVX,DIRVY,X2,Y2,X3,Y3
      REAL X0,Y0,Z0,F0
      CHARACTER*2 STATUS
      CHARACTER*2 COOL
      CHARACTER*3 CODE
      COMMON/INTS/LINNO,FROMFL,BLK,SPI,ARNO

    1 FORMAT(1X,A2,1X,I3)
    2 FORMAT(2X,F9.4)
    3 FORMAT(1X,F11.5,1X,F11.5,1X,F11.5)
    4 FORMAT(A6,A70)
    5 FORMAT('N',I5,'F',F9.4)
    6 FORMAT('N',I5,'G01X',F9.4,'Y',F9.4,'Z',F9.4)
    7 FORMAT('N',I5,'     Y',F9.4)
    8 FORMAT('N',I5)
    9 FORMAT('N',I5,'     Z',F9.4)
   10 FORMAT('N',I5,'S',I3,'M03')
   11 FORMAT('N',I5,'M05')
```

```
12 FORMAT('N',I5,'M30')

13 FORMAT('~~~~~~~~~~~~~~~~~~~~%')

14 FORMAT('O0001')

15 FORMAT(1X,I2)

16 FORMAT('N',I5,'T',I2)

17 FORMAT('THE FEEDRATE IS OUT OF RANGE AT LINE: ',I3)

18 FORMAT('N',I5,' GR X',F9.4)

19 FORMAT('N',I5,' ;BLOCK ',I3)

20 FORMAT('N',I5,' SKIP TO 900')

21 FORMAT('N',I5,' GO Z',F9.4)

22 FORMAT(1X,A9)

23 FORMAT(1X,F11.5,1X,F11.5)

24 FORMAT(A6,19X,F13.5,1X,F13.5)

25 FORMAT(1X,F13.5)

26 FORMAT(1X,2A)

27 FORMAT('N',I5,'M08')

28 FORMAT('N',I5,'M09')

29 FORMAT('N',I1,'G90 G00 G40 G80 G17 G49')

30 FORMAT('N',I1,'G54')

31 FORMAT('N',I5,'F1500')

32 FORMAT(25X,F13.5,1X,F13.5)

33 FORMAT('N',I5,A3,'X',F8.3,'Y',F8.3,'R',F9.3)

34 FORMAT('?N',I5,'G91 G30 Z0 M19')

35 FORMAT('?N',I5,'G91 G30 X0')

36 FORMAT('?N',I5,'M06')

37 FORMAT('?N',I5,'G91 G28 X0')

38 FORMAT('?N',I5,'G90 G54')

39 FORMAT('N',I5,'G00X',F9.4,'Y',F9.4,'S',I3,'M03')
```

```
 40 FORMAT('N',I5,'X',F9.4,'Y',F9.4)
 41 FORMAT('N',I5,'Z',F9.4)
 42 FORMAT('?N',I5,'G91 G28 Z0')
 43 FORMAT('?N',I5,'G91 G28 X0 Y0')
 44 FORMAT('?N',I5,'G80 G00')

C*****CHECK FOR GOTO AND GODLTA*****C

      IF((APWRD.EQ.'GOTO').OR.(APWRD.EQ.'GODLTA'))THEN
       READ(LIDAT,3) X,Y,Z
       IF(LINNO.LE.99986) THEN
        IF(APWRD.EQ.'GOTO') THEN
         IF(POINTT.EQ.0) THEN
           WRITE(4,40) LINNO,X,Y
           X0=X
           Y0=Y
           Z0=Z
           POINTT=1
         ELSE
           WRITE(4,6) LINNO,X,Y,Z
         ENDIF
        ELSE
         X=X+XX
         Y=Y+YY
         Z=Z+ZZ
         WRITE(4,6) LINNO,X,Y,Z
        ENDIF
        LINNO=LINNO+1
 IF(APWRD.EQ.'GOTO') THEN
  ZCUR=Z
 ELSE
  ZCUR=Z+ZCUR
 ENDIF
ELSE
 WRITE(4,21) LINNO,ZMX
 LINNO=LINNO+1
 IF(SPI.EQ.1) THEN
  WRITE(4,11) LINNO
  LINNO=LINNO+1
 ENDIF
 WRITE(4,20) LINNO
 LINNO=100000
 WRITE(4,8) LINNO
 LINNO=0
 WRITE(4,19) LINNO,BLK
 BLK=BLK+1
 LINNO=LINNO+1
 IF(SPI.EQ.1) THEN
  WRITE(4,10) LINNO
  LINNO=LINNO+1
 ENDIF
 WRITE(4,21) LINNO,ZCUR
 LINNO=LINNO+1
 IF(APWRD.EQ.'GOTO') THEN
  WRITE(4,6) LINNO,X
 ELSE
```

```
      WRITE(4,18) LINNO,X
     ENDIF
     LINNO=LINNO+1
     WRITE(4,7) LINNO,Y
     LINNO=LINNO+1
     WRITE(4,9) LINNO,Z
     LINNO=LINNO+1
     IF(APWRD.EQ.'GOTO') THEN
      ZCUR=Z
     ELSE
      ZCUR=Z+ZCUR
     ENDIF
    ENDIF
    XX=X
    YY=Y
    ZZ=Z
    GOTO 100
   ENDIF

C*****CHECK FOR ARC*****C

   IF(APWRD.EQ.'INDIRV') THEN
     READ(LIDAT,23) DIRVX,DIRVY
     READ(3,24) APWRD,X_CEN,Y_CEN
     READ(3,25) R
     READ(3,32) X2,Y2
     XCHECK=X2-X
     YCHECK=Y2-Y
```

```
      PH_V=ATAN2(DIRVY,DIRVX)
      PH_R=ATAN2(YCHECK,XCHECK)
       PH_R=PH_R-PH_V
       IF(PH_R.LT.-3.1416) THEN
        PH_R=PH_R+6.2832
       ENDIF
       IF(PH_R.GT.3.1416) THEN
        PH_R=PH_R-6.2832
       ENDIF
      IF(PH_R.LT.0) THEN
        CODE='G02'
      ELSE
        CODE='G03'
      ENDIF
C******** Case of the arc more than half revolute **C
       IF((X2.EQ.X_CEN).AND.(Y2.EQ.Y_CEN)) THEN
         READ(3,32) X3,Y3
C*********** Case of the arc is one revolute **C
         IF((X3.EQ.X).AND.(Y3.EQ.Y)) THEN
           X2=X_CEN+X_CEN-X3
           Y2=Y_CEN+Y_CEN-Y3
           WRITE(4,33) LINNO,CODE,X2,Y2,R
           LINNO=LINNO+1
         ENDIF
           X2=X3
           Y2=Y3
       ENDIF
```

```
        WRITE(4,33) LINNO,CODE,X2,Y2,R
        LINNO=LINNO+1
        X=X2
        Y=Y2
      ENDIF

C*****CHECK FOR FEDRAT*****C

      IF(APWRD.EQ.'FEDRAT') THEN
        READ(LIDAT,2) FEDRAT
       IF((FEDRAT.LT.0.1).OR.(FEDRAT.GT.15000.0)) THEN
        WRITE(4,17) LINNO
        APWRD='FINI'
        GOTO 101
       ENDIF
       IF(FROMFL.NE.0) THEN
        IF(POINTT.EQ.0) THEN
          F0=FEDRAT
        ELSE
          FEDRAT=FEDRAT
          WRITE(4,5) LINNO,FEDRAT
          LINNO=LINNO+1
        ENDIF
       ENDIF
       GOTO 100
      ENDIF
```

```
C*****CHECK FOR RAPID*****C

      IF(APWRD.EQ.'RAPID ') THEN
        WRITE(4,31) LINNO
        LINNO=LINNO+1
      ENDIF

C*****CHECK FOR SPINDLE*****C

      IF(APWRD.EQ.'SPINDL') THEN
       READ(LIDAT,1) STATUS,RPM
       IF(STATUS.EQ.'ON') THEN
        SPI=1
        WRITE(4,39) LINNO,X0,Y0,RPM
        LINNO=LINNO+1
        WRITE(4,41) LINNO,Z0
        LINNO=LINNO+1
        WRITE(4,5) LINNO,F0
       ELSE
        SPI=0
        WRITE(4,11) LINNO
       ENDIF
       LINNO=LINNO+1
       GOTO 100
      ENDIF

C*****CHECK FOR LOADTL*****C
```

```
IF(APWRD.EQ.'LOADTL') THEN
 READ(LIDAT,15) TOOLNO
 WRITE(4,28) LINNO
 LINNO=LINNO+1
 WRITE(4,11) LINNO
 LINNO=LINNO+1
 WRITE(4,16) LINNO,TOOLNO
 LINNO=LINNO+1
 WRITE(4,34) LINNO
 LINNO=LINNO+1
 WRITE(4,35) LINNO
 LINNO=LINNO+1
 WRITE(4,36) LINNO
 LINNO=LINNO+1
 WRITE(4,37) LINNO
 LINNO=LINNO+1
 WRITE(4,38) LINNO
 LINNO=LINNO+1
 WRITE(4,39) LINNO,X0,Y0,RPM
 LINNO=LINNO+1
 WRITE(4,41) LINNO,Z0
 LINNO=LINNO+1
 WRITE(4,27) LINNO
 LINNO=LINNO+1
 GOTO 100
ENDIF
```

```
C*****CHECK FOR COOLANT*****C

      IF(APWRD.EQ.'COOLNT') THEN
       READ(LIDAT,26) COOL
       IF(COOL.EQ.'ON') THEN
        WRITE(4,27) LINNO
       ELSE
        WRITE(4,28) LINNO
       ENDIF
       LINNO=LINNO+1
       GOTO 100
      ENDIF

C*****CHECK FOR END*****C

      IF(APWRD.EQ.'END') THEN
       WRITE(4,42) LINNO
       LINNO=LINNO+1
       WRITE(4,43) LINNO
       LINNO=LINNO+1
       WRITE(4,44) LINNO
       LINNO=LINNO+1
       WRITE(4,12) LINNO
       GOTO 100
      ENDIF

C*****CHECK FOR FROM*****C
```

```
     IF(STARTT.EQ.0) THEN
      FROMFL=1
      WRITE(4,13)
      WRITE(4,14)
      LINNO=LINNO+1
      WRITE(4,29) LINNO
      LINNO=LINNO+1
      WRITE(4,30) LINNO
      LINNO=LINNO+1
      ZMX=Z
      ZCUR=Z
      STARTT=1
     ENDIF

C*****READ IN NEW LINE FROM INPUT*****C

  100 READ(3,4)APWRD,LIDAT
  101 RETURN
     END
```

ประวัติผู้เขียน

นายสมบูรณ์ อนันตธนะสาร เกิดเมื่อวันที่ 3 สิงหาคม พ.ศ. 2514 ที่อำเภอพระนคร จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมเครื่องกล จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535 และได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย ในปีพ.ศ. 2536