รายการอ้างอิง

1. G.M. Davies. Effect of Dimension and Shape Deviation of Steel Products on Automatic Processing Operation, Dimensional Control in Rolling Mills , pp 1-3. The Institutes of Metals in Co-operate with the Association of Iron and Steel Engineering : Bourne Press, 1990.

2. P.Habberley. Market Requirement of Metal Packaging, Dimensional Control in Rolling Mills , pp 8-12.The Institutes of Metals in Co-operate with the Association of Iron and Steel Engineering : Bourne Press, 1990

3. R.M. Guo. A Mathematical Model for Strip Profile Predictation Using Influence Coeffients and Boussinesq's Equation, ICEME-94 (ASME) Annual Conference , Chichago,1994.

4. V.B. Ginzburg. Definition of Geometrical Parameter, High-Quality Steel Rolling Theory and Pratice , pp 3-27. New York : Marcel Dekker, 1993.

5. R.M. Guo. Computer Model Simulation of Strip Crown and Shape Control , Journal of Iron and Steel Engineering , Vol 63.N0.11. 1988 : 35-42

6. V.B. Ginzburg. Effect of Strip Crown on Strip Flatness, , High-Quality Steel Rolling Theory and Pratice , pp 578-580. New York : Marcel Dekker, 1993.

7. M.D. Stone and R.Gray. Theory and Practice Aspects in Crown Control, Iron and Steel Engineers , Vol 42. 1965 : 657-674.

8. K.N. Shobet and N.A. Townsend. Roll Bending Method of Crown Control in Four-High Plate Mills , JISI , 1968 : 1088-1098.

9. P.Beer and E.Johnston. Vector Mechanics for Engineers .Second Edition. Singapore : McGraw-Hill , 1990.

10. P.Pinij. Differential Equation . Third Edition. Bangkok : Numagsorn , 1990.

11. V.B. Ginzburg . Effect of Roll Deformation on Strip Profile , High-Quality Steel Rolling Theory and Practice , pp 451-469. New York : Marcel Dekker, 1993.

12. P.D. Spooner and G.F. Bryant . Analysis of Shape and Discussion of Problem of Scheduling Setup and Shape Control , Proceeding 0f Metal Society Conference on Shape Control , Chester. UK.1976 : PP19-29

13. A. Foppl , Technische Mecanik , Forth Edition, Vol 5, (n.d.) : 350

14. R, Hill.,The Mathematical Theory of Plastic , Oxford at the Clarendon Press , 1950.

15. D.R. Bland and Ford, Proc.Inst.Mech.Engrs , pp 144-163 .London , 1948

16. F. Fujita, et al , Strip Rolling Theories and Practical Applications , Japan : The Iron and Steel

Institute of Japan, 1998.

17. W.L. Roberts, Flat Processing of Steel , New York : Marcel Dekker, 1988.

18. W.L. Roberts, Cold Rolling of Steel , New York : Marcel Dekker, 1988.

ภาคผนวก

ภาคผนวก ก

Source code ของแบบจำลองการหา Strip profile และ shape control สำหรับแท่นรีดชนิด 6 ลูกรีด

โปรแกรม Data in

```
------------------------------------------------------------------------------------------------------------------------
------
  Sub datain(i)
'C
'C       DB:BACK-UP ROLL DIAMETER (MM)
'C       DW:WORK ROLL DIAMETER    (MM)
'C       L :ROLL BARREL LENGTH    (MM)
'C       LJ:WORK ROLL BENDER ARM LENGTH (MM)
'C       AMMU:UPPER SIDE MILL MODULUS WITHOUT ROLL DEFLECTION (TON/MM)
'C       AMML:LOWER SIDE MILL MODULUS WITHOUT ROLL DEFLECTION (TON/MM)
'C       CRBC:BACK-UP ROLL CROWN;DIAMETER (MM)
'C       CRWC:WORK ROLL CROWN   ;DIAMETER (MM)
'C       TP  :WORK ROLL TAPER   ;RADIUS   (MM)
'C       SWD: STRIP Width(MM)
'C       H0  :THICKNESS OF HOT BAND  (MM)
'C       HIM :THICKNESS OF INLET STRIP (MM)
'C       HOM :THICKNESS OF OUTLET THICKNESS (MM)
'C       PT  :ROLLING LOAD (TON)
'C       ASK,BSK,CSK :DEFORMATION REGISTANCE ASK*(BSK+EPS)**CSK (KG/MM**2)
'C       JW  :WORK ROLL BENDER FORCE (TON)
'C       LSFT:WORK ROLL SIFT LENGTH  (MM)
'C       DXX :WIDTH OF DEVIDED PARTS ;NORMAL SECTION (MM)
'C       DXE :                  ;FIRST PART FROM STRIP EDGE (MM)
'C       DXE2:                  ;SECOND
'C       ALPH1:SHAPE CHANGE RATIO (SHAPE/(PROFILE CHANGE))
'C       LG  :LENGTH OF THE RANGE OF PLANE STRESS CONDITION (MM)
'C       TB  :BACKWARD TENSION (KG/MM**2)
'C       TF  :FORWARD TENSION  (KG/MM**2)
'C       GAP :ROLL GAP FOR INITIAL CONDITION (MM)
'C       EPSP:LIMIT RATIO OF ROLLING LOAD FOR CONVERGENT
'C************************************************************************

  PAI = Application.Pi()
  DB = Cells(16, 13).Value
  DBN = Cells(16, 12).Value
  DW = Cells(28, 12).Value
  LW = Cells(40, 7).Value
```

```
    LB = Cells(10, 7).Value
    LTB = Cells(9, 7).Value
    LNB = (LTB - LB) / 2
    LJ = Cells(42, 6).Value
    TJWL = Cells(45, 5).Value
    TJWR = Cells(45, 10).Value
    GAPR = Cells(46, 10).Value
    GAPL = Cells(46, 5).Value
    AMM = Cells(6, 5).Value
'
    DI = Cells(23, 12).Value
    LI = Cells(33, 8).Value
    LJI = Cells(34, 7).Value
    TJIL = Cells(37, 6).Value
    TJIR = Cells(37, 11).Value
    GAPIL = Cells(7, 5).Value
    GAPIR = Cells(7, 10).Value
'
    EB = Cells(3, 3).Value
    CRBC = Cells(4, 3).Value
'
    EI = Cells(7, 3).Value
    CRIC = Cells(8, 3).Value
    TPIL = Cells(9, 3).Value
    TPIR = Cells(10, 3).Value
    LTIL1 = Cells(11, 3).Value
    LTIR1 = Cells(12, 3).Value
    LTIL2 = Cells(13, 3).Value
    LTIR2 = Cells(14, 3).Value
'
    EW = Cells(17, 3).Value
    CRWC = Cells(18, 3).Value
    TPWL = Cells(19, 3).Value
    TPWR = Cells(20, 3).Value
    LTWL1 = Cells(21, 3).Value
    LTWR1 = Cells(22, 3).Value
    LTWL2 = Cells(23, 3).Value
    LTWR2 = Cells(24, 3).Value
'
    ES = Cells(27, 3).Value
    swd = Cells(28, 3).Value
    HIM0 = Cells(29, 3).Value
    HIM = Cells(30, 3).Value
    HOM = Cells(31, 3).Value
    scr = Cells(32, 3).Value
    DHE = Cells(33, 3).Value
    LEDRP = Cells(34, 3).Value
    ALPH1 = Cells(35, 3).Value
```

```
'
  AMU = Cells(38, 3).Value
  PT = Cells(39, 3).Value
'
  LSI = Cells(42, 3).Value
  LSFT = Cells(43, 3).Value
  LSFTS = Cells(44, 3).Value

    LG = 0
'
  GB = EB / (2 * (1 + ANU))
  GI = EI / (2 * (1 + ANU))
  GW = EW / (2 * (1 + ANU))
'
    ASK = 70
    BSK = 0.001
    CSK = 0.21
'
    CCSB = 1
    CCSI = 1
    CCSIS = 1
    CCSW = 1
'
    m = 1
  For i = 1 To 25
    DX(i) = Cells(3 + i, 15).Value
      ISWWS(i) = 0
    If DX(i) <> 0 Then
      m = i
    End If
  Next i
'
  If (AMM = 0) Then AMM = 1E+20
  AMM = AMM / 2 * 1000
  If (GAMP >= HOM * 0.6) Then GAMP = HOM * 0.6
'
    JWL = TJWL * 1000
    JWR = TJWR * 1000
    JIL = TJIL * 1000
    JIR = TJIR * 1000
'    LSR = LB / 2 - LW / 2 + LSFT
    LSS = LW / 2 - swd / 2 + LSFTS
'
      LSBB = LI / 2 - LB / 2
      LSRR = LI / 2 - LW / 2
      LEBB = LSBB + LB
      LERR = LSRR + LW
'
```

```
      LES = LSS + swd
      LER = LSR + LW
      LEI = LSI + LI
      LTW = LW + (2 * LJ)
      LTI = LI + (2 * LJI)
'
      ISR = 1
      IER = m
      ISS = 1
      IES = m
'
      ILSS = Int(LSS * 100 + 0.01)
      ILES = Int(LES * 100 - 0.01)
'
'************* Make  Division of Contact of BUR-INR (dz) and INR-WR  (du)  **************
'
    Call CLEAR1(DXXI, 25)
    Call CLEAR1(DUW, 25)
    Call CLEAR1(DUB, 25)
    Call CLEAR1(DZ, 25)
    Call CLEAR1(ISWBI, 25)
    Call CLEAR1(ISWIB, 25)
    Call CLEAR1(ISWIW, 25)
    Call CLEAR1(ISWWI, 25)
    Call CLEAR1(ISWWS, 25)
    Call Division(DXXI, xxi, LW, LI, 0, LSI, nxxi, ISWWI, nxxic, 25)
    Call Division(DUW, uw, LI, LW, LSI, 0, nuw, ISWIW, nuwc, 26)
    Call Division(DUB, ub, LI, LB, LSI, 0, nub, ISWIB, nubc, 7)
    Call Division(DZ, z, LB, LI, 0, LSI, nz, ISWBI, nzc, 28)
    For i = 1 To nxxi
        Cells(3 + i, 17) = DXXI(i)
        Cells(3 + i, 18) = xxi(i)
    Next i
    For i = 1 To nuw
        Cells(33 + i, 15) = DUW(i)
        Cells(33 + i, 16) = uw(i)
    Next i
    For i = 1 To nub
        Cells(33 + i, 19) = DUB(i)
        Cells(33 + i, 20) = ub(i)
    Next i
    For i = 1 To nz
        Cells(63 + i, 15) = DZ(i)
        Cells(63 + i, 16) = z(i)
    Next i

                xx = 0
    For i = 1 To m
```

```
        XM = xx
        IXM = Int(xx * 100)
        xx = xx + DX(i) / 2
        XP = xx + DX(i) / 2
        IXP = Int(XP * 100)
        x(i) = xx
        xx = XP
    '   GAP(i) = GAPL + (GAPR - GAPL) * (LNB + x(i)) / LTB
          Cells(150 + i, 16).Value = LNB
          Cells(150 + i, 17).Value = LTB
          Cells(150 + i, 18).Value = x(i)
    '     Cells(70 + i, 19).Value = GAP(i)
    '
    '   If (XM <= LSR And XP > LSR) Then ISR = i
        If (IXM <= ILSS And IXP > ILSS) Then ISS = i
    '   If (XM < LSR + LW And XP >= LSR + LW) Then IER = i
        If (IXM < ILES And IXP >= ILES) Then IES = i
          Cells(150 + i, 2).Value = DX(i)
          Cells(150 + i, 3).Value = x(i)
          Cells(3 + i, 16).Value = x(i)
          Cells(175 + i, 2).Value = XM
          Cells(175 + i, 3).Value = XP
     Next i
    nctotal = nxxic + nubc + (IES - ISS + 1)
        For i = 1 To m
          ISWWS(i) = 0
          If (i >= ISS And i <= IES) Then ISWWS(i) = 1
        Next i
For i = 1 To m
  Cells(100 + i, 5) = ISWBI(i)
  Cells(100 + i, 6) = ISWIB(i)
  Cells(100 + i, 7) = ISWIW(i)
  Cells(100 + i, 8) = ISWWI(i)
  Cells(100 + i, 9) = ISWWS(i)
Next i

    If (swd <> 0) Then
      PM = PT / swd * 1000
  Call ROLFOC(P, DW, HIM, HOM, TB, TF, ISWCH)
    SKM = SK
    CT = 1
      If (PM <> 0) Then CT = PM / P
    End If
'
    IAB = PAI / 64 * DB ^ 4
    IABN = PAI / 64 * DBN ^ 4
    IAI = PAI / 64 * DI ^ 4
    IAW = PAI / 64 * DW ^ 4
```

```
'
  Call CLEAR1(DEPSI, 25)
'
  Call scrown(SC, scr, LW, swd, DHE, LEDRP, x, m)
'
    For i = ISS To IES
      xx = (x(i) - LW / 2#)
      HINP(i) = -SC(i)
      DEPSI(i) = 0#
    Next i
      HIMD = 0#
    For i = ISS To IES
      HIMD = HIMD + HINP(i) * DX(i)
    Next i
      HIMD = HIMD / swd
      DHIM = HIMD - HIM
    For i = ISS To IES
      HINP(i) = HINP(i) - DHIM
    Next i
'C
      HIM = 0#
    For i = ISS To IES
      HIM = HIM + HINP(i) * DX(i)
    Next i
      HIM = HIM / swd
'C
      Call CLEAR1(ALPH, 25)
      Call CLEAR1(G, 25)
      LGR = swd - LG
    For i = ISS To IES
      xx = x(i) - LSS
      ALPH(i) = ALPH1
      If (LG <> 0#) Then
       If (xx >= LG And xx <= LGR) Then G(i) = 0#
       If (xx < LG) Then G(i) = ((LG - xx) / LG) ^ 2
       If (xx > LGR) Then G(i) = ((xx - LGR) / LG) ^ 2
      End If
    Next i
'C
'C      INITIAL SX0(I) (STRESS DISTRIBUTION TO ROLLING DIRECTION)
'C           WIDTH STRESS       (SZ0(I))
'C
  AK = SK / 2# * CT
  SXE = -Sqr(3#) * AK * ALD / (HOM * 8#)
  SXE = SXE * 2# * AMU
  SXE = SXE + (2# * TB + TF) / 3#
  SZE = 0#
  If (ALD <> 0#) Then SYC = -PM / ALD
```

```
   SXC = SYC + 2# * AK
   SZC = (SXC + SYC) / 2#
'C
For i = ISS To IES
   SX0(i) = G(i) * SXE + (1# - G(i)) * SXC
   SZ0(i) = G(i) * SZE + (1# - G(i)) * SZC
   SY0(i) = (SX0(i) + SZ0(i)) / 2# - Sqr(3# * ((2# * AK) ^ 2 - (SX0(i) - SZ0(i)) ^ 2)) / 2#
'C
'    Cells(150 + i, 7).Value = HINP(i)
'    Cells(150 + i, 8).Value = HOM
'C
   EPSY0(i) = -Application.LN(HOM / HIM) * (2# * SY0(i) - SZ0(i) - SX0(i)) / (2# * SX0(i) - SY0(i) - SZ0
(i))
   HOF0(i) = HINP(i) * Exp(EPSY0(i))
   PS(i) = -SY0(i) * ALD
Next i
'C
   Call CLEAR1(CRB, nz)
   Call CLEAR1(CRIB, nub)
   Call CLEAR1(CRIW, nuw)
   Call CLEAR1(CRWI, nxxi)
   Call CLEAR1(CRWS, m)
   Call CLEAR1(CRPX, m)
   Call CLEAR1(YVC, m)
   Call CROWN(CRB, LB, CRBC, 0, TPBL, LTBL1, LTBL2, TPBR, LTBR1, LTBR2, z, nz, DZ)
   Call CROWN(CRIB, LI, CRIC, 0, TPIL, LTIL1, LTIL2, TPIR, LTIR1, LTIR2, ub, nub, DUB)
   Call CROWN(CRIW, LI, CRIC, 0, TPIL, LTIL1, LTIL2, TPIR, LTIR1, LTIR2, uw, nuw, DUW)
   Call CROWN(CRWI, LW, CRWC, 0, TPWL, LTWL1, LTWL2, TPWR, LTWR1, LTWR2, xxi, nxxi, DXXI)
   Call CROWN(CRWS, LW, CRWC, 0, TPWL, LTWL1, LTWL2, TPWR, LTWR1, LTWR2, x, m, DX)

       Cells(63, 17).Value = "CRB (i) "
    For i = 1 To nz
       Cells(63 + i, 17).Value = CRB(i)
    Next i
    For i = 1 To nub
       Cells(33 + i, 21).Value = CRIB(i)
    Next i
    For i = 1 To nuw
       Cells(33 + i, 17).Value = CRIW(i)
    Next i
    For i = 1 To nxxi
       Cells(3 + i, 19).Value = CRWI(i)
    Next i
    For i = 1 To m
       Cells(3 + i, 20).Value = CRWS(i)
    Next i

       Cells(3, 21).Value = "HINP (i)"
```

```
        Cells(3, 22).Value = "HOM"
     For i = ISS To IES
        Cells(3 + i, 21).Value = HINP(i)
        Cells(3 + i, 22).Value = HOM
     Next i
'
     If ((LI / 2) - LSI >= LB / 2) Then
       LCIB = LB
     Else
       LCIB = (LB / 2) + (LI / 2) - LSI
     End If
       PMB = PT / LCIB * 1000
'
     If ((LI / 2) - LSI >= LW / 2) Then
       LCIW = LW
     Else
       LCIW = (LW / 2) + (LI / 2) - LSI
     End If
       PMR = PT / LCIW * 1000
       For i = 1 To nz
          PBZ(i) = PMB
       Next i
       For i = 1 To nub
          PBU(i) = PMB
       Next i
       For i = 1 To nuw
          PRU(i) = PMR
       Next i
       For i = 1 To nxxi
          PRX(i) = PMR
       Next I

End Sub
```

โปรแกรม main เป็นโปรแกรมหลักซึ่งเรียกใช้โปรแกรมย่อยต่างๆ

```
Dim ROLF, DBT, DBNT, DWT, LBT, LWT, LNBT, JWL, JWR, TJWL, TJWR
Dim DFNC(104, 104) As Double, FNC(104) As Double, answer(104) As Double
Dim DFNCD(104, 104) As Double, FNCD(104) As Double
Dim DPDHH(25), SHAPE(25), PMM(25), DSHAPE(25), DPDT(25), AKK(25)
Dim DPDHE(25), DTDHE(25), DPDHC(25), DPDTE(25), ALDD(25), SX(25), SY(25), SZ(25), DT(25),
PE0(25), PC0(25)
Dim LAMDQ, LAMDE, LAMDQL, LAMDQR, LAMDEL, LAMDER
Dim YBA(25), YWA(25), YBAI(25), YWAL(25)
Dim YBS(25), YWSR(25), YWSS(25), YWSSS(25)
Dim YBSR(25), YWSRR(25), DYBW(25), DYWW(25), YBG(25), REDUC(25)
Dim ISWBI(25) As Integer, ISWIB(25) As Integer, ISWIW(25) As Integer
Dim ISWWI(25) As Integer, ISWWS(25) As Integer, ICASE As Integer
'c
Dim PAI, ANU, EW, EB, GB, GW, ES
Dim DX(25), x(25), DXS(25), HINP(25), HOF(25), HOF0(25)
Dim PR(25), PS(25), SX0(25), SY0(25), SZ0(25), EPSY0(25), QPP(25)
Dim CAG(25, 25), CCSB, CCSW, CVC(25, 25)
Dim DHDP(25), HEQ(25), DEPSI(25), CRPX(25), YVC(25), SC(25)
Dim DB, DBN, DW, JW, LSFT, LSFTS, LW, LB, LJ, LNB, AMM, LTB, LTW
Dim swd, HIM0, HIM, HOM, PT, PM, PMR, DH, RD, ALD, QP, AMU, SK, CT, AK, IAB, IAW, IABN,
ASK, BSK, CSK, TB, TF, CEQ, V, TMP, SKM
Dim ALPH(25), G(25), LS, LSS, EPSP, RATIO, DHE, scr, GAPM, GAP(25), GAPL, GAPR
Dim NS, ISS, IES, m, ISR, IER, NM, NLIM, ICTR, IQ1, IQ2
Dim CRBC, CRWC, TPBL, LTBL1, LTBL2, TPBR, LTBR1, LTBR2, TPWL, LTWL1, LTWL2, TPWR,
LTWR1, LTWR2
Dim LVC, HVC, PVC
Dim DI, LI, IAI, RI, LSFTI, LSI, LTI
Dim TJIL, TJIR, LJI
Dim EI, CRIC, TPIL, TPIR, LTIL1, LTIR1, LTIL2, LTIR2
Dim GI, CCSI, CCSIS
Dim PB(25), CRB(25), CRIB(25), CRIW(25), CRWI(25), CRWS(25)
'Dim ISIW, IEIW, ISIB, IEIB, ISR2, IER2
Dim DZ(25), DUW(25), DUB(25), DXXI(25), zux(25), DZUX(25)
Dim z(25), ub(25), uw(25), xxi(25)
Dim nz As Integer, nub As Integer, nuw As Integer, nxxi As Integer
Dim nzc As Integer, nubc As Integer, nuwc As Integer, nxxic As Integer, nctotal As Integer
Dim CSB(25, 25), CSIB(25, 25), CSIW(25, 25), CSWI(25, 25), CSWS(25, 25)
Dim CAB(25, 25), CAW(25, 25)
Dim CAIBB(25, 25), CAIBW(25, 25), CAIWB(25, 25), CAIWW(25, 25)
Dim CAWII(25, 25), CAWIS(25, 25), CAWSI(25, 25), CAWSS(25, 25), PMB, CRBZ(25), GAPI(25)
Dim JIL, JIR
Dim LSBB, LSRR, LEBB, LERR
Dim PBZ(25), PBU(25), PRU(25), PRX(25)
Dim PBB(25), PBI(25), PRI(25), PRW(25)
Dim YABB(25), YAIB(25), YAIW(25), YGI(25), YG(25), YAWI(25), YAWS(25)
Dim DEL(25), h(25), EL(25), ALPH1, lunit(25)
```

```
Sub main()

     Range("Q4:y32").Value = " "
    ' Range("b71:y100") = " "
     Range("q34:y100") = " "
     Range("B151:y198").Value = " "
     Range("d200:y250").Value = " "
     Range("o34:y59").Value = " "
     Range("o64:y90").Value = " "
  NLIM = 10
  EPSP = 0.01
  ANU = 0.3       ' Poison ratio
  TB = 20
  TF = 5
'
  Call datain(i)
'
  RB = DB / 2
  RI = DI / 2
  RW = DW / 2
  RBN = DBN / 2


  Range("a801,y900").Value = " "

  Call CLEAR(CAB, 25, 25)
  Call CLEAR(CAG, 25, 25)
  Call CLEAR(CSB, 25, 25)
  Call CLEAR(CAIBB, 25, 25)
  Call CLEAR(CAIBW, 25, 25)
  Call CLEAR(CAIWB, 25, 25)
  Call CLEAR(CAIWW, 25, 25)
  Call CLEAR(CSIB, 25, 25)
  Call CLEAR(CSIW, 25, 25)
  Call CLEAR(CAWII, 25, 25)
  Call CLEAR(CAWIS, 25, 25)
  Call CLEAR(CAWSI, 25, 25)
  Call CLEAR(CAWSS, 25, 25)
  Call CLEAR(CSWI, 25, 25)
  Call CLEAR(CSWS, 25, 25)
  Call CABSUB(CAB, IAB, IABN, RB, RBN, LB, LNB, EB, GB, z, nz, DZ, z, nz, DZ)
  Call CAGSUB(CAG, LB, LNB, AMM, z, nz, DZ)
```

```
    Call CASUBD(CAIBB, IAI, RI, LI, LJI, 0, EI, GI, ub, nub, DUB, ub, nub, DUB)
    Call CASUBD(CAIBW, IAI, RI, LI, LJI, 0, EI, GI, ub, nub, DUB, uw, nuw, DUW)
    Call CASUBD(CAIWB, IAI, RI, LI, LJI, 0, EI, GI, uw, nuw, DUW, ub, nub, DUB)
    Call CASUBD(CAIWW, IAI, RI, LI, LJI, 0, EI, GI, uw, nuw, DUW, uw, nuw, DUW)
    Call CASUBD(CAWII, IAW, RW, LW, LJ, 0, EW, GW, xxi, nxxi, DXXI, xxi, nxxi, DXXI)
    Call CASUBD(CAWIS, IAW, RW, LW, LJ, 0, EW, GW, xxi, nxxi, DXXI, x, m, DX)
    Call CASUBD(CAWSI, IAW, RW, LW, LJ, 0, EW, GW, x, m, DX, xxi, nxxi, DXXI)
    Call CASUBD(CAWSS, IAW, RW, LW, LJ, 0, EW, GW, x, m, DX, x, m, DX)
    Call CSRSUB(CSB, CVC, DB, DI, EB, EI, PBZ, 0, PMB, z, nz, DZ)
    Call CSRSUB(CSIB, CVC, DI, DB, EI, EB, PBU, 0, PMB, ub, nub, DUB)
    Call CSRSUB(CSIW, CVC, DI, DW, EI, EW, PRU, 0, PMR, uw, nuw, DUW)
    Call CSRSUB(CSWI, CVC, DW, DI, EW, EI, PRX, 0, PMR, xxi, nxxi, DXXI)
    Call CSSSUB(CSWS, DW, EW, PS)

 '  Cells(1, 36) = "CAB"
 '  Call output2(CAB, nz, nz, 2, 36)
 '  Cells(19, 36) = "CAIBB"
 '  Call output2(CAIBB, nub, nub, 20, 36)
 '  Cells(39, 36) = "CAIBW"
 '  Call output2(CAIBW, nub, nuw, 40, 36)
 '  Cells(59, 36) = "CAIWB"
 '  Call output2(CAIWB, nuw, nub, 60, 36)
 '  Cells(79, 36) = "CAIWW"
 '  Call output2(CAIWW, nuw, nuw, 80, 36)
 '  Cells(99, 36) = "CAWII"
 '  Call output2(CAWII, nxxi, nxxi, 100, 36)
 '  Cells(119, 36) = "CAWIS"
 '  Call output2(CAWIS, nxxi, m, 120, 36)
 '  Cells(139, 36) = "CAWSS"
 '  Call output2(CAWSS, m, m, 140, 36)
 '  Cells(1, 66) = "CSB"
 '  Call output2(CSB, nz, nz, 2, 66)
 '  Cells(19, 66) = "CSIB"
 '  Call output2(CSIB, nub, nub, 20, 66)
 '  Cells(39, 66) = "CSIW"
 '  Call output2(CSIW, nuw, nuw, 40, 66)
 '  Cells(59, 66) = "CSWI"
 '  Call output2(CSWI, nxxi, nxxi, 60, 66)
 '  Cells(79, 66) = "CSWS"
 '  Call output2(CSWS, m, m, 80, 66)
'C
'C HERZT'S DEFORMATION
'C
'

   CBI = 8# / PAI * (1# - ANU ^ 2) * (1# / EB + 1# / EI)
   BBI = Sqr(CBI * PMB * DB * DI / (DB + DI))
   AABI = CBI / 8# * PMB * (2# / 3# + Application.LN(2# * DB / BBI) + Application.LN(2# * DI / BBI))
'
```

```
   CIW = 8 / PAI * (1 - ANU ^ 2) * (1 / EI + 1 / EW)
   BIW = Sqr(CIW * PMR * DI * DW / (DI + DW))
   AAIW = CIW / 8 * PMR * (2 / 3 + Application.LN(2 * DI / BIW) + Application.LN(2 * DW / BIW))
 '
   ADB = 0
   ADBI = 0
   ADIW = 0
   ADW = 0
   MD2 = m / 2
   nzD2 = nz / 2
   nubD2 = nub / 2
   nuwD2 = nuw / 2
   nxxiD2 = nxxi / 2
 '
   For j = 1 To nz
     ADB = ADB + CSB(nzD2, j)
   Next j
   For j = 1 To nub
     ADBI = ADBI + CSIB(nubD2, j)
   Next j
   For j = 1 To nuwD2
     ADIW = ADIW + CSIW(nuwD2, j)
   Next j
   For j = 1 To nxxiD2
     ADW = ADW + CSWI(nxxiD2, j)
   Next j
 '
   ADB = ADB * PMB
   ADBI = ADBI * PMB
   ADIW = ADIW * PMR
   ADW = ADW * PMR
 '
   CBH = AABI / ADB * (1 / 3 + Application.LN(2 * DB / BBI)) / (2 / 3 + Application.LN(2 * DB / BBI) +
Application.LN(2 * DI / BBI))
   CIH = AABI / ADBI * (1 / 3 + Application.LN(2 * DI / BBI)) / (2 / 3 + Application.LN(2 * DB / BBI) +
Application.LN(2 * DI / BBI))
   CIWH = AAIW / ADIW * (1 / 3 + Application.LN(2 * DI / BIW)) / (2 / 3 + Application.LN(2 * DI / BIW)
+ Application.LN(2 * DW / BIW))
   CWH = AAIW / ADW * (1 / 3 + Application.LN(2 * DW / BIW)) / (2 / 3 + Application.LN(2 * DI / BIW)
+ Application.LN(2 * DW / BIW))
 '
   Call CLEAR1(DHDP, m)
   Call CLEAR1(DPDHH, m)
   Call CLEAR1(QPP, m)
   Call CLEAR1(HEQ, m)
   Call CLEAR1(DT, m)
 '
     DT0 = (2# * TB + TF) / 3#
   For i = ISS To IES
      DPDHH(i) = DPDH(HINP(i), HOF(i), HOF0(i), ALPH(i), G(i), SX0(i), SY0(i), SX0(i), DT0, EPSYO(i),
DW, PE0(i), PC0(i), DPDHE(i), DTDHE(i), DPDHC(i), DPDT(i), DPDTE(i), ISWCH, 0)
      DHDP(i) = 1# / DPDHH(i)
      PMM(i) = PS(i) - DPDT(i) * ES * DEPSI(i) / 2#
      QPP(i) = QP
      AKK(i) = AK
      ALDD(i) = ALD
   Next i
      DPDHM = 0#
      TPMM = 0#
   For i = ISS To IES
      DPDHM = DPDHM + DPDHH(i) * DX(i)
      TPMM = TPMM + PMM(i) * DX(i)
   Next i
      DTPM = (PT * 1000# - TPMM) / swd
         IYS = 150
   For i = ISS To IES
      PMM(i) = PMM(i) + DTPM
      HEQ(i) = HINP(i) * Exp(EPSY0(i)) - PMM(i) * DHDP(i)
       Cells(IYS + i, 10).Value = EPSY0(i)
       Cells(IYS + i, 11).Value = PMM(i)
       Cells(IYS + i, 12).Value = HEQ(i)
   Next i

      icn = 0
3000:
'      NMB = 0
'      NMW = 0
'      NMS = 0

'   For i = 1 To m
'      Cells(100 + i, 5).Value = ISWBI(i)
'      Cells(100 + i, 6).Value = ISWIB(i)
'      Cells(100 + i, 7).Value = ISWIW(i)
'      Cells(100 + i, 8).Value = ISWWI(i)
'      Cells(100 + i, 9).Value = ISWWS(i)
'   Next i
'      M3 = m * 3
'      NM = NMB + NMW + NMS
'
'      Cells(96, 4).Value = M3
'      Cells(96, 5).Value = NMS
'      Cells(96, 6).Value = NM
'      Cells(96, 11).Value = ISS
'      Cells(96, 12).Value = IES
'
1000:
    icn = icn + 1
    icnp = 0
 1111:
    icnp = icnp + 1
'
  Call CLEAR(DFNC, 104, 104)
  Call DFUNC(DFNC)
  Call FUNC(FNC)

    Cells(200, 3) = nctotal
  For i = 1 To nctotal + 5
    Cells(200 + i, 3) = i
      For j = 1 To nctotal + 5
        Cells(200 + i, 5 + j) = DFNC(i, j)
      Next j
    Cells(200 + i, 5 + nctotal + 7) = FNC(i)
  Next i

'**************************************************
  Call gauss(DFNC, FNC, answer, nctotal + 5)

  For j = 1 To nctotal + 5
    Cells(255, 5 + j) = answer(j)
  Next j
'**************************************************
  Call CLEAR1(PBB, 25)
  Call CLEAR1(PBI, 25)
  Call CLEAR1(PRI, 25)
  Call CLEAR1(PRW, 25)
  Call CLEAR1(PS, 25)
  Call CLEAR1(HOF, 25)
'
    j = 0
  For i = 1 To nz
    If (ISWBI(i) <> 0) Then
      j = j + 1
      PBB(i) = answer(j)
    End If
  Next i
   cc = 0
  For i = 1 To nub
    If (ISWIB(i) <> 0) Then
      cc = cc + 1
      PBI(i) = answer(cc)
    End If
  Next i
'
   cc = j
   For i = 1 To nuw
      If (ISWIW(i) <> 0) Then
         cc = cc + 1
         PRI(i) = answer(cc)
      End If
   Next i
   For i = 1 To nxxi
      If (ISWWI(i) <> 0) Then
         j = j + 1
         PRW(i) = answer(j)
      End If
   Next i
'
   For i = 1 To m
      If (ISWWS(i) <> 0) Then
         j = j + 1
         PS(i) = answer(j)
      End If
   Next i
         GIL = answer(j + 1)
         GIR = answer(j + 2)
         GL = answer(j + 3)
         GR = answer(j + 4)
         gapp = answer(j + 5) / 2
'
   For i = 1 To 25
      Cells(100 + i, 11) = PBB(i)
      Cells(100 + i, 12) = PBI(i)
      Cells(100 + i, 13) = PRI(i)
      Cells(100 + i, 14) = PRW(i)
      Cells(100 + i, 15) = PS(i)
   Next i
      Cells(126, 11) = gapp
      Cells(126, 12) = GIL
      Cells(127, 12) = GIR
      Cells(126, 14) = GL
      Cells(127, 14) = GR
'
'**************************************************
    Call CLEAR1(YABB, 25)
    Call CLEAR1(YAIB, 25)
    Call CLEAR1(YAIW, 25)
    Call CLEAR1(YGI, 25)
    Call CLEAR1(YG, 25)
    Call CLEAR1(YAWI, 25)
    Call CLEAR1(YAWS, 25)
    Call CLEAR1(h, 25)
'
```

```
'*******  Total deflection of Buck-up roll   **************
'
    For i = 1 To nz
        YABB(i) = 0
                For j = 1 To nz
                        YABB(i) = YABB(i) + CAB(i, j) * PBB(j) * DZ(j) + CSB(i, j) * PBB(j)
                 Next j
                        YABB(i) = YABB(i) + CRB(i) + gapp
     Next i
'
'******  Total deflection of Intermediate roll (upper surface) *******
'
    For i = 1 To nub
        YAIB(i) = 0
        YGI(i) = GIL * (1 - (ub(i) + LJI) / LTI) + GIR * ((ub(i) + LJI) / LTI) + gapp
            For j = 1 To nub
                    YAIB(i) = YAIB(i) - (CAIBB(i, j)) * PBI(j) * DUB(j) - CSIB(i, j) * PBI(j)
            Next j
            For j = 1 To nuw
                    YAIB(i) = YAIB(i) + ((CAIBW(i, j)) * PRI(j) * DUW(j))
            Next j
            YAIB(i) = YAIB(i) + YGI(i) + CRIB(i)
    Next i
'
'******  Total deflection of Intermediate roll (under surface) **********
'
    For i = 1 To nuw
        YAIW(i) = 0
        YG(i) = GIL * (1 - (uw(i) + LJI) / LTI) + GIR * ((uw(i) + LJI) / LTI) + gapp
          For j = 1 To nub
              YAIW(i) = YAIW(i) - (CAIWB(i, j) * PBI(j)) * DUB(j)
          Next j
          For j = 1 To nuw
              YAIW(i) = YAIW(i) + (CAIWW(i, j) * DUW(j) + CSIW(i, j)) * PRI(j)
          Next j
          YAIW(i) = YAIW(i) + YG(i) + CRIW(i)
     Next i
'
'******* Total deflection of Work roll ( upper surface) ******************
'
   For i = 1 To nxxi
     YAWI(i) = 0
     YG(i) = GL * (1 - (xxi(i) + LJ) / LTW) + GR * ((xxi(i) + LJ) / LTW) + gapp
       For j = 1 To nxxi
         YAWI(i) = YAWI(i) - (CAWII(i, j) * DXXI(j) + CSWI(i, j)) * PRW(j)
       Next j
       For j = 1 To m
         YAWI(i) = YAWI(i) + CAWIS(i, j) * DX(j) * PS(j)
```

```
          Next j
          YAWI(i) = YAWI(i) + YG(i) + CRWI(i)
    Next i
'
'*******  Total deflection of work roll (under surface)  *****************
'
    For i = 1 To m
          YAWS(i) = 0
          YG(i) = gapp + GL * (1 - (x(i) + LJ) / LTW) + GR * ((x(i) + LJ) / LTW)
              For j = 1 To nxxi
                    YAWS(i) = YAWS(i) - (CAWSI(i, j)) * PRW(j) * DXXI(j)
              Next j
              For j = 1 To m
                    YAWS(i) = YAWS(i) + (CAWSS(i, j)) * PS(j) * DX(j) + CSWS(i, j) * PS(j)
              Next j
              YAWS(i) = YAWS(i) + YG(i) + CRWS(i)
    Next i
'
'******** Thickness ******************************
'
    For i = ISS To IES
          h(i) = YAWS(i) + YAWS(m + 1 - i)
    Next i

          Cells(100, 17).Value = "YAIB (i) "
          Cells(100, 18).Value = "YAIW (i)"
          Cells(100, 19).Value = "YAWI (i)"
          Cells(100, 20).Value = "YAWS (i)"
          Cells(100, 21).Value = "h (i)"
      For i = 1 To 25
          Cells(100 + i, 16).Value = YABB(i)
          Cells(100 + i, 17).Value = YAIB(i)
          Cells(100 + i, 18).Value = YAIW(i)
          Cells(100 + i, 19).Value = YAWI(i)
          Cells(100 + i, 20).Value = YAWS(i)
          Cells(100 + i, 21).Value = h(i)
      Next i
'
' *************************************************************************
'
   Call CLEAR1(EL, 25)
   Call CLEAR1(DEL, 25)
'
          rr = 0
    For i = ISS To IES
        EL(i) = Log(HINP(i) / h(i))
        rr = rr + (EL(i) * DX(i))
    Next i
```

```
'
    EMEA = rr / swd
    DELMIN = 0
  For i = ISS To IES
    DEL(i) = (EL(i) - EMEA) * ALPH1
    If DEL(i) < DELMIN Then DELMIN = DEL(i)
  Next i
  For i = ISS To IES
    Iunit(i) = (DEL(i) - DELMIN) * 100000
  Next i
'
    Cells(100, 22).Value = "EL"
    Cells(100, 23).Value = "DEL"
    Cells(100, 24).Value = "I-unit"
    Cells(100, 25).Value = "x(i)"
  For i = ISS To IES
    Cells(100 + i, 22).Value = EL(i)
    Cells(100 + i, 23).Value = DEL(i)
    Cells(100 + i, 24).Value = Iunit(i)
    Cells(100 + i, 25).Value = x(i)
  Next I

End Sub
```

โปรแกรม Division สำหรับทำการแบ่งช่วงบนลูกรีด

```
    Sub Division(DV, V, L1, L2, shift1, shift2, n, ISW, nc, j)

            LRRR = L1 / 2 - L2 / 2
            LX1 = (L1 / 2) - shift1
            LX2 = (L2 / 2) - shift2
            LY1 = (L1 / 2) + shift1
            LY2 = (L2 / 2) + shift2
'
            LL = min(LX1, LX2) + min(LY1, LY2)
            n2 = Int(LL / 200)
            DL = (LL - n2 * 200) / 2
'           Cells(1, j) = LL
'           Cells(2, j) = n2
'           Cells(3, j) = LX1
'           Cells(4, j) = LX2
'           Cells(5, j) = LY1
'           Cells(6, j) = LY2
        If LX2 >= LX1 Then
                   nc = 0
                For i = 1 To n2 * 2
                    DV(i) = 100
                    ISW(i) = 1
                    nc = nc + 1
                Next i
                    DV(n2) = 100 + DL
                    DV(n2 + 1) = 100 + DL
            If LY1 > LY2 Then
                DV(n2 * 2 + 1) = LY1 - LY2
                   ISW(n2 * 2 + 1) = 0
                n = n2 * 2 + 1
            Else
                n = n2 * 2
            End If
        Else
            DV(1) = LX1 - LX2
                   ISW(1) = 0
                   nc = 0
                For i = 1 To n2 * 2
                DV(i + 1) = 100
                   ISW(i + 1) = 1
                   nc = nc + 1
            Next i
                DV(n2 + 1) = 100 + DL
                DV(n2 + 2) = 100 + DL
            If LY1 > LY2 Then
```

```
                DV(n2 * 2 + 2) = LY1 - LY2
                   ISW(n2 * 2 + 2) = 0
                n = n2 * 2 + 2
            Else
                n = n2 * 2 + 1
            End If
        End If
            vv = 0
        For i = 1 To n
            vv = vv + DV(i) / 2
            V(i) = vv
            vv = vv + DV(i) / 2
        Next i
End Sub
```

โปรแกรม scrown สำหรับคำนวณคราวน์ของแผ่นโลหะขาเข้า

```
    Sub scrown(SC, scr, LW, swd, deh, ldeh, x, m)


    x1 = swd / 2# - ldeh
    x2 = swd / 2#
    a1 = scr / x1 ^ 2
    dydx = 2 * a1 * x1
      Cells(50, 4).Value = x1
      Cells(50, 5).Value = x2
      Cells(50, 6).Value = 11
      Cells(50, 7).Value = dydx
    a2 = 0#
    a3 = 0#
  If (deh = 0#) Then GoTo 100
    a4 = (dydx * (x2 ^ 2 - x1 ^ 2) - 2 * x1 * deh) / (2 * (dydx * (x2 - x1) - deh))
  If (x1 - a4 = 0) Then
    a3 = deh / ldeh ^ 2
  Else
    a3 = dydx / (2 * (x1 - a4))
  End If
    a2 = scr - a3 * (x1 - a4) ^ 2
100:
  For i = 1 To m
    xx = x(i) - LW / 2
    SC(i) = a1 * xx ^ 2
  If (Abs(xx) > x1) Then SC(i) = a2 + a3 * (Abs(xx) - a4) ^ 2
  Next i
End Sub
```

โปรแกรมหา Roll force

```
   Sub ROLFOC(P, DW, hi, HO, TBB, TFF, ISWCH)

   CNST = 16# * (1# - ANU ^ 2) / (Application.Pi() * EW)
   DH = hi - HO
   RED = DH / hi
   EPS = Application.LN(hi / HO)
   EPS0 = Application.LN(HIM0 / HO)
   SK = 1.15 * ASK * (EPS0 + BSK) ^ CSK
   PD = 0#
   IC = 0
1000:
   IC = IC + 1
 If (IC > 20) Then GoTo 300
   RD = DW / 2# * (1# + PD * CNST / DH)
'C    WRITE(6,*) RD,PD"
   ALD = Sqr(DH * RD)
'C
'C      HOT STRIP MILL
'C   EPSD = EPS / ALD * (V * 1000# / 60#)
'C   AKD = 0.126 - 1.75 * CEQ + 0.594 * CEQ ^ 2
'C   AK = 2851# + 2968# * CEQ - 1120# * CEQ ^ 2
'C   SK = 1.15 * Exp(AKD + AK / (TMP + 273#)) * EPS ^ 0.21 * EPSD ^ 0.13
'C   FR = 0.45 * RED + 0.04
'C   QP = 0.8 + FR * (sqr(RD / HIM) - 0.5)
'C   QPA = QP - (TBB * 2# + TFF) / (3# * SK)
'C
'C      COLD STRIP MILL

   PHI1 = Application.Acos(1# - DH / 2# / RD)
   QP = 1.08 + 1.79 * RED * AMU / PHI1 * Sqr(RED) - 1.02 * RED
'C
   cc = 1# - (TBB * 2# + TFF) / (3# * SK)
   QPA = QP * cc
   P = SK * ALD * QPA
If (Abs(P - PD) > 0.01) Then
   PD = P
 GoTo 1000
End If
300:
End Sub
```

โปรแกรมหา Roll force

```
   Sub ROLFOE(PE, SXM, hi, HO, T, AMU, ALD, AK, QPE, ISWCH)

   HM = (hi + HO * 2#) / 3#
   PHH = Sqr(3#) * AK
'   If (ISWCH = "COLD") Then GoTo 100
'   SXM = T - PHH * ALD / (8# * HM)
'   If (PHH * ALD / (4# * HM) < T) Then GoTo 10
'   PE = PHH * ALD - 2# * HM * T ^ 2 / PH
'GoTo 200
'10:
'   PE = PHH * ALD * (1# + ALD / (8# * HM)) - T * ALD
'GoTo 200
100:
    PC = PHH - T
   If (PC * ALD * AMU / (2# * HM) >= T) Then
    PE = PHH * ALD - HM * T ^ 2 / (PC * AMU)
   Else
      PE = PC * ALD * (1# + AMU * ALD / (4# * HM))
   End If
    SXM = T - PE * AMU / (4# * HM)
200:
   If (T < 0#) Then PE = PHH * ALD
   QPE = PE / (PHH - T)
End Sub
```

โปรแกรม CROWN สำหรับหาคราวน์

```
  Sub CROWN(CR, L, CRC, LS, TPL, LTPL1, LTPL2, TPR, LTPR1, LTPR2, zux, nzux, DZUX)

  AC = 2# * CRC / L ^ 2
  CRNTL = AC * (LTPL2 - L / 2#) ^ 2
  CRNTR = AC * (LTPR2 - L / 2#) ^ 2
For i = 1 To nzux
  xx = zux(i) - LS
 If (xx >= 0#) Then
  If (xx <= L) Then
   If (xx >= LTPL2 And (L - xx) >= LTPR2) Then GoTo 200
   If (LTPL2 <> LTPL1) Then
    If (xx <= LTPL1) Then CR(i) = CRNTL + TPL
    If (xx > LTPL1 And xx < LTPL2) Then CR(i) = CRNTL - TPL * (xx - LTPL2) / (LTPL2 - LTPL1)
   End If
 If (LTPR2 = LTPR1) Then GoTo 100
 If ((L - xx) <= LTPR1) Then CR(i) = CRNTL + TPL
 If ((L - xx) > LTPR1 And (L - xx) < LTPR2) Then CR(i) = CRNTR - TPR * (L - xx - LTPR2) / (LTPR2 -
LTPR1)
 GoTo 100
200:
   CR(i) = AC * (xx - L / 2#) ^ 2
  End If
 End If
100:
Next i
End Sub
```

โปรแกรม CABSUB สำหรับการหา Influence coefficient

```
'C*******************************************************************************************
'C                                                                              *
   Sub CABSUB(CA, IA, IAN, R, RN, LR, LN, E, G, zux, nzux, DZUX, zux2, nzux2, DZUX2)
'C   ROLL AXIAL DEFLECTION BY DISTRIBUTED LOAD (WITH NECK)
'C                                                                              *
'C*******************************************************************************************
'C
   a = R ^ 2 * PAI
   AN = RN ^ 2 * PAI
   L = LR + 2# * LN
For i = 1 To nzux
   XI = zux(i) + LN
For j = 1 To nzux2
   XJ = zux2(j) + LN
   L1 = XJ - DZUX2(j) / 2#
   L2 = L - XJ - DZUX2(j) / 2#
   L3 = DZUX2(j)
   E1 = E
   E3 = E
   E2 = E
'C
   C2 = (1# - IA / IAN) * L3 / L * (L2 + 0.5 * L3) * LN ^ 3 / 3#
   C3 = C2
   C4 = (1# - IA / IAN) * L3 / L * (L1 + 0.5 * L3) * LN ^ 3 / 3#
'C
   B3 = -1# / 6# * (L + L2 + 0.5 * L3) * (L1 + 0.5 * L3) + 1# / 24# * L3 ^ 2 + (C4 - C3) / L3 / (L2 + 0.5 * L3)
   B2 = L3 / L * (L2 + 0.5 * L3) * B3
   B4 = L3 / L * (L2 + 0.5 * L3) * (B3 + 0.5 * L * (L1 + 0.5 * L3)) - 1# / 24# * L3 ^ 3
   B5 = IAN / IA * B4 + (1# - IAN / IA) * (L3 / L * (L1 + 0.5 * L3) * LN ^ 2 / 2#)
   B1 = IAN / IA * B2 - (1# - IAN / IA) * (L3 / L * (L2 + 0.5 * L3) * LN ^ 2 / 2#)
'C
 If (XI >= 0# And XI < LN) Then
   CA(i, j) = -1# / E1 / IAN * (L3 / L * (L2 + 0.5 * L3) * XI ^ 3 / 6# + B1 * XI)
   AM = (1# - XJ / L) * XI * DZUX2(j)
 End If
'C
 If (XI >= LN And XI < L1) Then
   CA(i, j) = -1# / E3 / IA * (L3 / L * (L2 + 0.5 * L3) * XI ^ 3 / 6# + B2 * XI + C2)
   AM = (1# - XJ / L) * XI * DZUX2(j)
 End If
'C
 If (XI >= L1 And XI < L1 + L3) Then
   CA(i, j) = -1# / E3 / IA * (L3 / L * (L2 + 0.5 * L3) * XI * (XI ^ 2 / 6# + B3) - 1# / 24# * (XI - L1) ^ 4 + C3)
```

```
    AM = (1# - XJ / L) * XI * DZUX2(j) - (XI - L1) ^ 2 / 2#
  End If
'C
  If (XI >= L1 + L3 And XI < L - LN) Then
    CA(i, j) = -1# / E3 / IA * (-1# * L3 / L * (L1 + 0.5 * L3) * (XI - L) ^ 3 / 6# + B4 * (XI - L) + C4)
    AM = (1# - XI / L) * XJ * DZUX2(j)
  End If
'C
  If (XI >= L - LN And XI <= L) Then
    CA(i, j) = -1# / E2 / IAN * (-1# * L3 / L * (L1 + 0.5 * L3) * (XI - L) ^ 3 / 6# + B5 * (XI - L))
    AM = (1# - XI / L) * XJ * DZUX2(j)
  End If
'C
    CA(i, j) = CA(i, j) + AM / (a * G) - R * R * ANU * AM / (2# * E * IA)
    CA(i, j) = CA(i, j) / DZUX2(j)
Next j
Next i
'C
End Sub
'C
'C************************************************************
    Sub CAGSUB(CA, LR, LN, AMM, zux, nzux, DZUX)
*
'C************************************************************
'C
'C
    L = LR + 2# * LN
For i = 1 To nzux
    xx = zux(i) + LN
For j = 1 To nzux
    zbb = zux(j) + LN
    CA(i, j) = (L - zbb) * (L - xx) / (L ^ 2 * AMM * 2) + zbb * xx / (L ^ 2 * AMM * 2)
Next j
Next i
End Sub
```

โปรแกรม CSRSUB สำหรับคำนวณหา Influence coefficient ที่ผิวสัมผัสระหว่างลูกรีด

```
'
   Sub CSRSUB(CSR, CSV, D1, D2, E1, E2, P, ISW, PMH, zux, nzux, DZUX)
'
'        Distribution function of surface

   C = 8 / PAI * (1# - ANU ^ 2) * (1# / E1 + 1# / E2)
For i = 1 To nzux
For j = 1 To nzux
   pp = P(j)
   If (pp <= 0#) Then pp = PMH
   b = Sqr(C * pp * D1 * D2 / (D1 + D2))
   x1 = Abs(zux(i) - zux(j))
   CSR(i, j) = (1# - ANU ^ 2) / (PAI * E1) * FFD(x1, b / 2#, DZUX(j)) - (1# + ANU) / (2# * PAI * E1) *
FFD1(x1, D1, DZUX(j), ANU)
   If (ISW = 1) Then CSR(i, j) = CSR(i, j) + CSV(i, j)
Next j
Next i
End Sub
```

โปรแกรม CSSSUB

```
   Sub CSSSUB(CSS, D, E, P)
'
   C = 8 / PAI * (1# - ANU ^ 2) * 2# / E
   b = ALD
For i = 1 To m
For j = 1 To m
   pp = P(j)
   If (pp <= 0#) Then pp = PMR
   BR = Sqr(C * pp * D / 2)
   x1 = Abs(x(i) - x(j))
   If (j < ISS Or j > IES Or swd = 0#) Then
      CSS(i, j) = (1# - ANU ^ 2) / (PAI * E) * FFD(x1, BR / 2#, DX(j)) - (1# + ANU) / (2# * PAI * E) *
FFD1(x1, D, DX(j), ANU)
   Else
      CSS(i, j) = (1# - ANU ^ 2) / (PAI * E) * FFD(x1, b, DX(j)) - (1# + ANU) / (2# * PAI * E) * FFD1(x1,
D, DX(j), ANU)
   End If
Next j
Next i
End Sub
```

Function FDD เป็นฟังชั่นใช้คำนวณในการหา influence coefficient

---

```
'************************************************************
   Function FFD(x, b, DZ)
'************************************************************
'
   bb = b
   Y1 = x + DZ / 2#
   Y2 = x - DZ / 2#
   FF = Application.LN((Sqr(bb ^ 2 + Y1 ^ 2) + Y1) / (Sqr(bb ^ 2 + Y2 ^ 2) + Y2)) + Y1 *
Application.LN((Sqr(Y1 ^ 2 + bb ^ 2) + bb) / Abs(Y1)) / bb - Y2 * Application.LN((Sqr(Y2 ^ 2 + bb ^
2) + bb) / Abs(Y2)) / bb
   FFD = FF
End Function
```

Function FDD1 เป็นฟังชั่นใช้คำนวณในการหา influence coefficient

---

```
'
'************************************************************
   Function FFD1(x, D, DZ, ANU)
'************************************************************
'
   YY = D * 0.75
   x1 = x - DZ / 2#
   x2 = x + DZ / 2#
   FF = 2# * (1# - ANU) * Application.LN((x2 + Sqr(x2 ^ 2 + YY ^ 2)) / (x1 + Sqr(x1 ^ 2 + YY ^ 2))) +
x2 / Sqr(x2 ^ 2 + YY ^ 2) - x1 / Sqr(x1 ^ 2 + YY ^ 2)
   FFD1 = FF
End Function
```

โปรแกรม DPDH สำหรับคำนวณหาความสัมพันธ์ระหว่างแรงรีดและความหนา

```
   Function DPDH(hi, HO, HO0, ALPH, G, SX, SY, SX0, DT, EPSY, DW, PE0, PC0, DPDHE, DTDHE,
DPDHC, DPDTC, DPDTE, ISWCH, ICAL)

If (ICAL <> 1) Then
   DDH = -HO0 * 0.02
   HOD = HO0 + DDH
   DT0 = (TB * 2# + TF) / 3#
   DTT = 1#
   TBD = TB + DTT
   TFD = TF + DTT
   Call ROLFOC(PDDH, DW, hi, HOD, TB, TF, ISWCH)
   PDDH = PDDH * CT
   Call ROLFOC(PDDT, DW, hi, HO0, TBD, TFD, ISWCH)
   PDDT = PDDT * CT
   Call ROLFOC(PC0, DW, hi, HO0, TB, TF, ISWCH)
   PC0 = PC0 * CT
   Call ROLFOE(PE, SXM, hi, HO0, DT0 + DTT, AMU, ALD, AK, QPE, ISWCH)
   Call ROLFOE(PE0, SXM, hi, HO0, DT0, AMU, ALD, AK, QPE, ISWCH)
   DPDHHH = (PDDH - PC0) / DDH
   DPDTC = (PDDT - PC0) / DTT
   DPDTE = (PE - PE0) / DTT
   DPDTT = 1# / ((1# - G) / DPDTC + G / DPDTE)
   DTDHC = ALPH * ES / hi
   DEY = Application.LN(-DDH / hi)
   EY = Application.LN(HO0 / hi)
   EX = EY * (2# * SX - SY) / (2# * SY - SX)
   a = (2# * SY - SX) / ES + (EY + DEY) - EX
   b = DEY * (2# * SX - SY) / ES
   DEX = (a + Sqr(a ^ 2 - 4# * b)) / 2#
   ALPHE = -DEX / DEY
   DTDHE = ALPHE * ES / hi
   DPDHC = DPDHHH + DPDTC * DTDHC
   DPDHE = DPDHHH + DPDTE * DTDHE
   DPDH = 1# / ((1# - G) / DPDHC + G / DPDHE)
Else
   TBD = DT - DT0 + TB
   TFD = DT - DT0 + TF
   If (SY >= 0#) Then
   EPSX0 = -Application.LN(HOM / HIM)
   EPSY = Application.LN(HO / hi)
   DEX = (SX - SX0) / ES
   SY = (2# * EPSY + (EPSX0 + DEX)) / (2# * (EPSX0 + DEX) + EPSY) * SX / 2#
   DPDHE = (-SY * ALD - PE0) / (HO - HO0)
'C      WRITE(6,*) SY,DPDHE"
Else
```

```
End If
   DPDH = 1# / ((1# - G) / DPDHC + G / DPDHE)
End If
'C
End Function
```

โปรแกรม CLEAR 1 สำหรับล้างข้อมูลใน Array 1 มิติ

```
'************************************************************
   Sub CLEAR1(F, m)
'************************************************************
'
 For i = 1 To m
   F(i) = 0#
 Next i
End Sub
```

โปรแกรม CLEAR สำหรับล้างข้อมูลใน Array แบบ 2 มิติ

```
'************************************************************
   Sub CLEAR(F, m, n)
'************************************************************
 For i = 1 To m
  For j = 1 To n
   F(i, j) = 0#
  Next j
 Next i
End Sub
```

โปรแกรม FUNC

```
Sub FUNC(FNC)

  ii = 0
    For i = 1 To nz
      If (ISWBI(i) = 1) Then
        ii = ii + 1
        FNC(ii) = -CRB(i)
      End If
    Next i
  ii = 0
    For i = 1 To nub
      If (ISWIB(i) = 1) Then
        ii = ii + 1
        FNC(ii) = FNC(ii) - CRIB(i)
      End If
    Next i
  ii = nubc
    For i = 1 To nuw
      If (ISWIW(i) = 1) Then
        ii = ii + 1
        FNC(ii) = -CRIW(i)
      End If
    Next i
  ii = nubc
    For i = 1 To nxxi
      If (ISWWI(i) = 1) Then
        ii = ii + 1
        FNC(ii) = FNC(ii) - CRWI(i)
      End If
    Next i
  ii = nubc + nuwc
    For i = 1 To m
      If (ISWWS(i) = 1) Then
        ii = ii + 1
        FNC(ii) = HEQ(i) - CRWS(i) - CRWS(m + 1 - i)
      End If
    Next i
        FNC(ii + 1) = -JIL - JIR
        FNC(ii + 2) = -LTI * JIR
        FNC(ii + 3) = -JWL - JWR
        FNC(ii + 4) = -LTW * JWR
        FNC(ii + 5) = PT * 1000

  End Sub
```

โปรแกรม DFUNC สำหรับนำสมการหลักมาสร้างเมตริกซ์

```
'**************************************************
   Sub DFUNC(DFNC)
'**************************************************
'*** Contact bewteen Back-up roll and Intermediate roll ******
'

      ii = 0
   For i = 1 To nz
      If (ISWBI(i) = 1) Then
         ii = ii + 1
         jj = 0
            For j = 1 To nz
               If (ISWBI(j) = 1) Then
                  jj = jj + 1
                  DFNC(ii, jj) = CAB(i, j) * DZ(j) + CSB(i, j)
               End If
            Next j
      End If
   Next i
      ii = 0
   For i = 1 To nub
      If (ISWIB(i) = 1) Then
         ii = ii + 1
         jj = 0
            For j = 1 To nub
               If (ISWIB(j) = 1) Then
                  jj = jj + 1
                  DFNC(ii, jj) = DFNC(ii, jj) + CAIBB(i, j) * DUB(j) + CSIB(i, j)
               End If
            Next j
            For j = 1 To nuw
               If (ISWIW(j) = 1) Then
                  jj = jj + 1
                  DFNC(ii, jj) = -CAIBW(i, j) * DUW(j)
               End If
            Next j
                  DFNC(ii, nctotal + 1) = -(1 - (ub(i) + LJI) / LTI)
                  DFNC(ii, nctotal + 2) = -(ub(i) + LJI) / LTI
      End If
   Next i
'
'*** Contact bewteen Intermediate roll and Work roll ******
'

      ii = nubc
   For i = 1 To nuw
      If (ISWIW(i) = 1) Then
```

```
            ii = ii + 1
            jj = 0
                For j = 1 To nub
                    If (ISWIB(j) = 1) Then
                        jj = jj + 1
                        DFNC(ii, jj) = -CAIWB(i, j) * DUB(j)
                    End If
                Next j
                For j = 1 To nuw
                    If (ISWIW(j) = 1) Then
                        jj = jj + 1
                        DFNC(ii, jj) = CAIWW(i, j) * DUW(j) + CSIW(i, j)
                    End If
                Next j
                        DFNC(ii, nctotal + 1) = (1 - (uw(i) + LJI) / LTI)
                        DFNC(ii, nctotal + 2) = (uw(i) + LJI) / LTI
        End If
    Next i
        ii = nubc
    For i = 1 To nxxi
        If (ISWWI(i) = 1) Then
            ii = ii + 1
            jj = nubc
                For j = 1 To nxxi
                    If (ISWWI(j) = 1) Then
                        jj = jj + 1
                        DFNC(ii, jj) = DFNC(ii, jj) + CAWII(i, j) * DXXI(j) + CSWI(i, j)
                    End If
                Next j
                For j = 1 To m
                    If (ISWWS(j) = 1) Then
                        jj = jj + 1
                        DFNC(ii, jj) = -CAWIS(i, j) * DX(j)
                    End If
                Next j
                        DFNC(ii, nctotal + 3) = -(1 - (xxi(i) + LJ) / LTW)
                        DFNC(ii, nctotal + 4) = -(xxi(i) + LJ) / LTW
        End If
    Next i
'
'****** Thickness ***
'

    For i = 1 To m
        If (ISWWS(i) = 1) Then
            ii = ii + 1
            jj = nubc
                For j = 1 To nxxi
                    If (ISWWI(j) = 1) Then
```

```
                  jj = jj + 1
                  DFNC(ii, jj) = -CAWSI(i, j) * DXXI(j) - CAWSI(m + 1 - i, j) * DXXI(j)
               End If
            Next j
            For j = 1 To m
               If (ISWWS(j) = 1) Then
                  jj = jj + 1
                  DFNC(ii, jj) = CAWSS(i, j) * DX(j) + CAWSS(m + 1 - i, j) * DX(j) + CSWS(i, j) + CSWS(m
+ 1 - i, j)

               End If
            Next j
                  DFNC(ii, nctotal + 3) = (1 - (x(i) + LJ) / LTW) + (1 - (x(m + 1 - i) + LJ) / LTW)
                  DFNC(ii, nctotal + 4) = (x(i) + LJ) / LTW + (x(m + 1 - i) + LJ) / LTW
                  DFNC(ii, nctotal + 5) = 1
                  DFNC(ii, ii) = DFNC(ii, ii) - DHDP(i)
      End If
   Next i
'
'*** Force balance for I-R
'
   ii = ii + 1
   jj = 0
      For j = 1 To nub
         If (ISWIB(j) = 1) Then
            jj = jj + 1
            DFNC(ii, jj) = -DUB(j)
         End If
      Next j
      For j = 1 To nuw
         If (ISWIW(j) = 1) Then
            jj = jj + 1
            DFNC(ii, jj) = DUW(j)
         End If
      Next j
'
'*** Moment balance for I-R
'
   ii = ii + 1
   jj = 0
      For j = 1 To nub
         If (ISWIB(j) = 1) Then
            jj = jj + 1
            DFNC(ii, jj) = -DUB(j) * (ub(j) + LJI)
         End If
      Next j
      For j = 1 To nuw
         If (ISWIW(j) = 1) Then
```

```
        jj = jj + 1
        DFNC(ii, jj) = DUW(j) * (uw(j) + LJI)
      End If
    Next j
'
'*** Force balance for w-R
'
  ii = ii + 1
  jj = nubc
    For j = 1 To nxxi
      If (ISWWI(j) = 1) Then
        jj = jj + 1
        DFNC(ii, jj) = -DXXI(j)
      End If
    Next j
    For j = 1 To m
      If (ISWWS(j) = 1) Then
        jj = jj + 1
        DFNC(ii, jj) = DX(j)
      End If
    Next j
'
'*** Moment balance for W-R
'
  ii = ii + 1
  jj = nubc
    For j = 1 To nxxi
      If (ISWWI(j) = 1) Then
        jj = jj + 1
        DFNC(ii, jj) = -DXXI(j) * (xxi(j) + LJ)
      End If
    Next j
    For j = 1 To m
      If (ISWWS(j) = 1) Then
        jj = jj + 1
        DFNC(ii, jj) = DX(j) * (x(j) + LJ)
      End If
    Next j
'*** Total force
  ii = ii + 1
  jj = nubc + nuwc
    For j = 1 To m
      If (ISWWS(j) = 1) Then
        jj = jj + 1
        DFNC(ii, jj) = DX(j)
      End If
    Next j
  End Sub
```

โปรแกรม guass สำหรับคำนวณหาคำตอบของระบบเมตริกซ์ด้วยระเบียบวิธีการกำจัดแบบเกาส์

```
'**********************************************************************
   Sub gauss(a, b, xxx, nnn)
'**********************************************************************
Static m, i, k, n
Static C As Double, amax As Double, wb As Double, w As Double
n = nnn
For i = 1 To n
 amax = Abs(a(i, i))
 m = i
 k = i + 1
 If k > n Then GoTo calc
strt:
 If amax >= Abs(a(k, i)) Then GoTo nxt
  amax = Abs(a(k, i))
  m = k
nxt:
 If k >= n Then GoTo chng
 k = k + 1
 GoTo strt
chng:
 If m = i Then GoTo calc
 k = i
 wb = b(k)
 b(k) = b(m)
 b(m) = wb
nxtk:
 w = a(i, k)
 a(i, k) = a(m, k)
 a(m, k) = w
 If k = n Then GoTo calc
 k = k + 1
 GoTo nxtk
```

```
calc:
  C = a(i, i)
 For j = i To n
  a(i, j) = a(i, j) / C
 Next j
  b(i) = b(i) / C
 For k = 1 To n
 If k = i Then GoTo endk
   C = a(k, i)
 For j = i To n
  a(k, j) = a(k, j) - C * a(i, j)
 Next j
  b(k) = b(k) - C * b(i)
endk:
 Next k
 Next i
 For i = 1 To n
  xxx(i) = b(i)
 Next i
End Sub
'
```

โปรแกรม output2 สำหรับแสดงผลของการคำนวณ

```
' ****************************************
  Sub output2(a, m, n, nl, nc)
' ****************************************
    For i = 1 To m
      For j = 1 To n
        Cells(nl + i - 1, nc + j - 1) = a(i, j)
      Next j
    Next i
  End Sub
```

Function min

```
'****************************************
  Function min(x, y)
'****************************************
    If x < y Then
      min = x
    Else
      min = y
    End If
  End Function
```

ภาคผนวก ข

ข้อมูลการรีดจริงจากโรงงานเหล็กแผ่นรีดเย็นไทย

ชุดที่ 1

```
 268:MATERIAL COIL WIDTH           (width   ) [1237]
 269:MATERIAL COIL LENGTH           (length  ) [741]
 270:MATERIAL COIL WEIGHT           (weight  ) [2016]
 271:MATERIAL COIL OUTSIDE DIAMETER  (odia    ) [1795]
 283:HSM C %                    (crate   ) [40]
 285:HSM Si %                   (sirate  ) [10]
 287:HSM Mn %                    (mnrate  ) [260]
 289:HSM Cr %                   (crrate  ) [50]
 291:HSM Mo %                    (morate  ) [0]
 293:HSM P %                    (prate   ) [150]
 295:HSM MANUFACTURE TEMPERATURE     (mtemp   ) [560]
332:PRODUCT COIL THICKNESS (TARGET) (pthick  ) [821]
 333:PRODUCT COIL WIDTH (TARGET)     (pwidth  ) [1225]
 335:PRODUCT COIL LENGTH (TARGET)    (plen    ) [2541]
 347:MAX DIAMETER FOR ROLL RESULT    (mxodia  ) [2101]
 348:MIN DIAMETER FOR ROLL RESULT    (mnodia  ) [770]
 349:MAX WEIGHT FOR ROLL RESULT     (mxwei   ) [3053]
 350:MIN WEIGHT FOR ROLL RESULT     (mnwei   ) [166]
 373:THICKNESS TOLERANCE (MAX)      (thtomx  ) [853]
 374:THICKNESS TOLERANCE (MIN)      (thtomn  ) [797]
 375:WIDTH TOLERANCE (MAX)         (wdtomx  ) [1227]
 376:WIDTH TOLERANCE (MIN)         (wdtomn  ) [1225]
 380:THICKNESS TOLERANCE (MIN) LOG   (thmnlog ) [765]
 381:THICKNESS TOLERANCE (MAX) LOG   (thmxlog ) [835]
 382:WIDTH TOLERANCE (MIN)    LOG   (wdmnlog ) [1219]
 383:WIDTH TOLERANCE (MAX)    LOG   (wdmxlog ) [1226]
**********************************
* MLDTC.PCOIL.COMM              *
**********************************
  51:SHAPE TOP AVERAGE 1 (WS)       (shpt1   ) [570]
  52:SHAPE TOP AVERAGE 2 (CENTER)    (shpt2   ) [680]
  53:SHAPE TOP AVERAGE 3 (DS)       (shpt3   ) [790]
  54:SHAPE MID AVERAGE 1 (WS)       (shpm1   ) [500]
  55:SHAPE MID AVERAGE 2 (CENTER)    (shpm2   ) [660]
  56:SHAPE MID AVERAGE 3 (DS)       (shpm3   ) [860]
**********************************
* MLDT.TCMPR.PCS                *
**********************************
```

```
2313:WIDTH                    (b      ) [1225.000000]
2315:THICKNESS       ENT.  - #1STD  (h      ) [2.744000]
2317:             #1STD - #2STD  (       ) [2.103754]
2319:             #2STD - #3STD  (       ) [1.533351]
2321:             #3STD - #4STD  (       ) [1.132120]
2323:             #4STD - #5STD  (       ) [0.859686]
2325:             #5STD - DEL.   (       ) [0.821000]
2341:REDUCTION              #1STD  (r      ) [0.233326]
2343:                   #2STD  (       ) [0.271136]
2345:                   #3STD  (       ) [0.261669]
2347:                   #4STD  (       ) [0.240641]
2349:                   #5STD  (       ) [0.045000]
2351:UNIT TENSION     ENT.  - #1STD  (utb     ) [3.734795]
2353:             #1STD - #2STD  (       ) [16.000000]
2355:             #2STD - #3STD  (       ) [15.500000]
2357:             #3STD - #4STD  (       ) [14.500000]
2359:             #4STD - #5STD  (       ) [15.500000]
2361:             #5STD - DEL.   (       ) [4.000000]
2385:DYNAMIC DEFORM.RESIST.   #1STD  (kt      ) [6.141077]
2387:                   #2STD  (       ) [2.239280]
2389:                   #3STD  (       ) [1.002473]
2391:                   #4STD  (       ) [1.152276]
2393:                   #5STD  (       ) [1.264676]
2395:TOTAL TENSION     ENT.  - #1STD  (tb      ) [11764.899414]
2397:             #1STD - #2STD  (       ) [41233.574219]
2399:             #2STD - #3STD  (       ) [29114.501953]
2401:             #3STD - #4STD  (       ) [20109.277344]
2403:             #4STD - #5STD  (       ) [16323.285156]
2405:             #5STD - DEL.   (       ) [4022.899658]
2417:STATIC  DEFORM.RESIST.   #1STD  (ks      ) [41.371029]
2419:                   #2STD  (       ) [54.562637]
2421:                   #3STD  (       ) [61.341343]
2423:                   #4STD  (       ) [65.795235]
2425:                   #5STD  (       ) [67.864632]
2427:FORWARD SLIP            #1STD  (f      ) [0.053707]
2429:                   #2STD  (       ) [0.011289]
2431:                   #3STD  (       ) [0.003074]
2433:                   #4STD  (       ) [0.002150]
2435:                   #5STD  (       ) [0.006222]
2447:WORK ROLL RADIUS         #1STD  (R      ) [273.609985]
2449:                   #2STD  (       ) [274.075012]
2451:                   #3STD  (       ) [270.607513]
2453:                   #4STD  (       ) [273.442505]
2455:                   #5STD  (       ) [202.167496]
```

```
2457:AVE. DEFORM. RESIST.    #1STD  (kp     ) [54.862255]
2459:                   #2STD  (       ) [65.589203]
2461:                   #3STD  (       ) [71.988441]
2463:                   #4STD  (       ) [77.304329]
2465:                   #5STD  (       ) [79.823647]
2467:MATERIAL SPEED   ENT.  - #1STD  (vo     ) [240.510223]
2469:             #1STD - #2STD  (       ) [313.705933]
2471:             #2STD - #3STD  (       ) [430.403778]
2473:             #3STD - #4STD  (       ) [582.941895]
2475:             #4STD - #5STD  (       ) [767.675842]
2477:             #5STD - DEL.   (       ) [803.849060]
2489:ROLL FORCE            #1STD  (P      ) [903120.375000]
2491:                   #2STD  (       ) [967130.187500]
2493:                   #3STD  (       ) [929034.750000]
2495:                   #4STD  (       ) [895110.937500]
2497:                   #5STD  (       ) [734776.125000]
2499:DEFORMED WR RADIUS      #1STD  (Rd     ) [330.246063]
2501:                   #2STD  (       ) [342.441284]
2503:                   #3STD  (       ) [361.639374]
2505:                   #4STD  (       ) [401.690643]
2507:                   #5STD  (       ) [598.226135]
2509:SCREW POSITION         #1STD  (S      ) [-1.652963]
2511:                   #2STD  (       ) [-1.549813]
2513:                   #3STD  (       ) [-1.128394]
2515:                   #4STD  (       ) [-0.975709]
2517:                   #5STD  (       ) [-1.386101]
2519:ROLL GAP (DELTA)       #1STD  (dS     ) [-0.020078]
2521:                   #2STD  (       ) [-0.052407]
2523:                   #3STD  (       ) [-0.087939]
2525:                   #4STD  (       ) [-0.059917]
2527:                   #5STD  (       ) [-0.027662]
2529:ROLLING SPEED          #1STD  (v      ) [297.716431]
2531:                   #2STD  (       ) [425.599304]
2533:                   #3STD  (       ) [581.155151]
2535:                   #4STD  (       ) [766.028503]
2537:                   #5STD  (       ) [798.878418]
2539:MILL MODULUS           #1STD  (K      ) [528986.312500]
2541:                   #2STD  (       ) [512987.125000]
2543:                   #3STD  (       ) [518986.812500]
2545:                   #4STD  (       ) [536985.875000]
2547:                   #5STD  (       ) [405946.656250]
2589:FRICTION COEFFICIENT    #1STD  (myu    ) [0.068972]
2591:                   #2STD  (       ) [0.045471]
2593:                   #3STD  (       ) [0.028666]
```

```
2595:                  #4STD  (       ) [0.020000]
2597:                  #5STD  (       ) [0.130000]
2701:ROLLING TORQUE          #1STD  (gr      ) [2903522.000000]
2703:                  #2STD  (       ) [11770645.000000]
2705:                  #3STD  (       ) [9398454.000000]
2707:                  #4STD  (       ) [6818226.000000]
2709:                  #5STD  (       ) [2944913.500000]
2753:LOW ROLL FORCE CORRECT.  #1STD  (Sa      ) [0.656000]
2755:                  #2STD  (       ) [0.893000]
2757:                  #3STD  (       ) [0.762000]
2759:                  #4STD  (       ) [0.688000]
2761:                  #5STD  (       ) [0.800000]
 2813:CONTACT LENGTH OF IMR    SP     (X       ) [1750.000000]
2815:                  SP     (       ) [1750.000000]
2817:                  SP     (       ) [1750.000000]
2819:                  SP     (       ) [1750.000000]
2821:                  #5STD  (       ) [1385.000000]
2833:DEFORM.RESIST.PARA L           (kpl     ) [65.478226]
2835:DEFORM.RESIST.PARA M           (kpm     ) [0.002803]
2837:DEFORM.RESIST.PARA N           (kpn     ) [0.209465]
2839:BOLZMAN CONSTANT               (Bz      ) [0.000086]
2841:INITIAL UNIT TENSION FOR TR    (iut     ) [1.300000]
2843:INITIAL TOTAL TENSION FOR TR   (it      ) [5229.769531]
 2857:AVE. BR DIAMETER        #1STD  (DB      ) [1362.324951]
2859:                  #2STD  (       ) [1361.145020]
2861:                  #3STD  (       ) [1368.239990]
2863:                  #4STD  (       ) [1366.849976]
2865:                  #5STD  (       ) [1366.050049]
2867:AVE. BR DIAMETER AT TEST #1STD  (DB0     ) [1420.000000]
2869:                  #2STD  (       ) [1420.000000]
2871:                  #3STD  (       ) [1420.000000]
2873:                  #4STD  (       ) [1420.000000]
2875:                  #5STD  (       ) [1420.000000]
2877:WR BENDER            #1STD  (wrb     ) [36020.710938]
2879:                  #2STD  (       ) [37298.796875]
2881:                  #3STD  (       ) [38577.296875]
2883:                  #4STD  (       ) [41598.847656]
2885:                  #5STD  (       ) [22976.998047]
2887:IMR BENDER           #5STD  (imrb    ) [18043.271484]
2889:IMR SHIFT          #5STD  (imrsh   ) [182.500000]
2903:TAPER END DIAMETER            (tpedia  ) [1679.016602]
2905:TAPER END WEIGHT COEFFICIENT    (ctp     ) [0.950000]
2912:ROLL ROUGHNESS          #1STD  (ra      ) [75]
2913:                  #2STD  (       ) [50]
```

```
2914:                 #3STD (      ) [42]
2915:                 #4STD (      ) [37]
2916:                 #5STD (      ) [425]
2917:WR BENDER CHANGE FORCE   #1STD  (wrbcf   ) [-291.205444]
2919:                 #2STD (      ) [-7.593815]
2921:                 #3STD (      ) [633.853210]
2923:                 #4STD (      ) [449.636078]
2925:                 #5STD (      ) [38.552563]
2927:IMR BENDER CHANGE FORCE  #5STD  (imrbcf  ) [47.502815]
2969:ROLL FORCE & WR BENDER   #1STD  (pwrb    ) [917528.687500]
2977:                 #5STD (      ) [743966.937500]
3065:UNIT TENSION     ENT.  - #1STD  (utb_tr  ) [0.000000]
3117:ROLL FORCE - BACK TENS.  #1STD  (ptb     ) [-12.202811]
3119:                 #2STD (      ) [-12.146875]
3121:                 #3STD (      ) [-10.464516]
3123:                 #4STD (      ) [-9.568966]
3125:                 #5STD (      ) [-10.095967]
3127:ROLL FORCE - FRICTION    #1STD  (pmyu    ) [3013.805420]
3129:                 #2STD (      ) [4468.373535]
3131:                 #3STD (      ) [5628.267090]
3133:                 #4STD (      ) [7098.437500]
3135:                 #5STD (      ) [3804.807617]
3137:ROLL FORCE FUNC. PARA    1    (A       ) [1.104400]
3139:                 2   (      ) [0.123460]
3141:                 3   (      ) [0.459410]
3143:                 4   (      ) [-0.135390]
3149:YOUNG'S MODULUS              (EE      ) [21700.000000]
3151:POISSON'S RATIO              (vv      ) [0.300000]
3153:HITCHCOOK'S COEFFICIENT      (CH      ) [0.000214]
3155:UNIT MILL MODULUS            (KW      ) [5000000.000000]
3273:BARREL LENGTH            #1STD  (BL      ) [1750.000000]
3275:                 #2STD (      ) [1750.000000]
3277:                 #3STD (      ) [1750.000000]
3279:                 #4STD (      ) [1750.000000]
3281:                 #5STD (      ) [1750.000000]
3283:ROLL GAP CALC        PARA  1 (S_FC    ) [81500.000000]
3285:                 PARA  2 (      ) [117000.000000]
3287:                 PARA  3 (      ) [112750.000000]
3289:                 PARA  4 (      ) [110625.000000]
3291:                 PARA  5 (      ) [89625.000000]]
3319:IMR SHIFT DELTA              (delta   ) [30.000000]
3333:                 #2STD (      ) [340.000000]
3335:                 #3STD (      ) [557.000000]
3337:                 #4STD (      ) [804.000000]
```

```
3339:                 #5STD  (      ) [1179.000000]
3431:DEFORMATION RESIST FOR LMN CALC.(k_km    ) [50.851181]
3463:WRB (PRES)           #1STD  (wrb_p   ) [76.084030]
3465:                 #2STD  (      ) [79.327057]
3467:                 #3STD  (      ) [82.571167]
3469:                 #4STD  (      ) [90.238144]
3471:                 #5STD  (      ) [77.129097]
3473:WRBCF (PRES)          #1STD  (wrbcf_p ) [-0.738913]
3475:                 #2STD  (      ) [-0.019269]
3477:                 #3STD  (      ) [1.608356]
3479:                 #4STD  (      ) [1.140919]
3481:                 #5STD  (      ) [0.154581]
3483:IMRB (PRES)                 (imrb_p  ) [72.129097]
3485:IMRBCF (PRES)               (imrbcf_p) [0.154581]
3487:WRB (OPERATOR)        #1STD  (wr_op   ) [0.000000]
3489:                 #2STD  (      ) [0.000000]
3491:                 #3STD  (      ) [0.000000]
3493:                 #4STD  (      ) [0.000000]
3495:                 #5STD  (      ) [0.000000]
3497:IMRB (OPERATOR)              (imr_op  ) [0.000000]
3499:UNIT TENSION (NORMAL)  ENT - #1 (utb0    ) [3.500000]
3501:                #1  - #2 (      ) [16.000000]
3503:                #2  - #3 (      ) [15.500000]
3505:                #3  - #4 (      ) [14.500000]
3507:                #4  - #5 (      ) [15.500000]
3509:                #5  - DEL(      ) [4.000000]
3511:IMR BL LENGTH                (bl_imr  ) [1650.000000]
```

ชุดที่ 2

```
268:MATERIAL COIL WIDTH           (width   ) [1527]
269:MATERIAL COIL LENGTH           (length  ) [673]
270:MATERIAL COIL WEIGHT           (weight  ) [2261]
271:MATERIAL COIL OUTSIDE DIAMETER  (odia    ) [1726]
283:HSM C %                    (crate   ) [40]
285:HSM Si %                   (sirate  ) [10]
287:HSM Mn %                    (mnrate  ) [240]
289:HSM Cr %                   (crrate  ) [40]
291:HSM Mo %                    (morate  ) [0]
293:HSM P %                    (prate   ) [110]
295:HSM MANUFACTURE TEMPERATURE    (mtemp   ) [560]
```

```
 332:PRODUCT COIL THICKNESS (TARGET) (pthick  ) [1027]
 333:PRODUCT COIL WIDTH (TARGET)     (pwidth  ) [1515]
  335:PRODUCT COIL LENGTH (TARGET)    (plen    ) [1845]
 373:THICKNESS TOLERANCE (MAX)       (thtomx  ) [1068]
 374:THICKNESS TOLERANCE (MIN)       (thtomn  ) [997]
 375:WIDTH TOLERANCE (MAX)           (wdtomx  ) [1517]
 376:WIDTH TOLERANCE (MIN)           (wdtomn  ) [1515]
  51:SHAPE TOP AVERAGE 1 (WS)        (shpt1   ) [410]
  52:SHAPE TOP AVERAGE 2 (CENTER)    (shpt2   ) [1230]
  53:SHAPE TOP AVERAGE 3 (DS)        (shpt3   ) [760]
  54:SHAPE MID AVERAGE 1 (WS)        (shpm1   ) [380]
  55:SHAPE MID AVERAGE 2 (CENTER)    (shpm2   ) [1140]
  56:SHAPE MID AVERAGE 3 (DS)        (shpm3   ) [750]
2313:WIDTH                       (b      ) [1515.000000]
2315:THICKNESS       ENT. - #1STD  (h      ) [2.744000]
2317:            #1STD - #2STD  (       ) [2.167760]
2319:            #2STD - #3STD  (       ) [1.707044]
2321:            #3STD - #4STD  (       ) [1.349256]
2323:            #4STD - #5STD  (       ) [1.081053]
2325:            #5STD - DEL.   (       ) [1.027000]
2341:REDUCTION           #1STD  (r      ) [0.210000]
2343:                    #2STD  (       ) [0.212531]
2345:                    #3STD  (       ) [0.209595]
2347:                    #4STD  (       ) [0.198779]
2349:                    #5STD  (       ) [0.050000]
2351:UNIT TENSION    ENT. - #1STD  (utb    ) [3.722301]
2353:            #1STD - #2STD  (       ) [16.000000]
2355:            #2STD - #3STD  (       ) [15.500000]
2357:            #3STD - #4STD  (       ) [14.500000]
2359:            #4STD - #5STD  (       ) [15.500000]
2361:            #5STD - DEL.   (       ) [4.000000]
2385:DYNAMIC DEFORM.RESIST.  #1STD  (kt     ) [6.211871]
2387:                    #2STD  (       ) [2.273945]
2389:                    #3STD  (       ) [0.982258]
2391:                    #4STD  (       ) [1.094926]
2393:                    #5STD  (       ) [1.170279]
2395:TOTAL TENSION   ENT. - #1STD  (tb     ) [14550.059570]
2397:            #1STD - #2STD  (       ) [52546.500000]
2399:            #2STD - #3STD  (       ) [40085.664062]
2401:            #3STD - #4STD  (       ) [29639.777344]
2403:            #4STD - #5STD  (       ) [25385.816406]
2405:            #5STD - DEL.   (       ) [6223.619629]
2417:STATIC  DEFORM.RESIST.  #1STD  (ks     ) [39.920021]
2419:                    #2STD  (       ) [52.032059]
```

```
2421:                  #3STD  (        ) [58.015789]
2423:                  #4STD  (        ) [62.152069]
2425:                  #5STD  (        ) [64.218895]
2427:FORWARD SLIP          #1STD  (f       ) [0.055040]
2429:                  #2STD  (        ) [0.018295]
2431:                  #3STD  (        ) [0.009070]
2433:                  #4STD  (        ) [0.005130]
2435:                  #5STD  (        ) [0.006239]
2447:WORK ROLL RADIUS        #1STD  (R       ) [273.609985]
2449:                  #2STD  (        ) [274.075012]
2451:                  #3STD  (        ) [272.885010]
2453:                  #4STD  (        ) [267.295013]
2455:                  #5STD  (        ) [200.812500]
2457:AVE. DEFORM. RESIST.    #1STD  (kp      ) [53.268520]
2459:                  #2STD  (        ) [62.707172]
2461:                  #3STD  (        ) [68.125076]
2463:                  #4STD  (        ) [73.031342]
2465:                  #5STD  (        ) [75.504913]
2489:ROLL FORCE            #1STD  (P       ) [1103408.375000]
2491:                  #2STD  (        ) [1160331.000000]
2493:                  #3STD  (        ) [1140574.375000]
2495:                  #4STD  (        ) [1131524.250000]
2497:                  #5STD  (        ) [844135.437500]
2499:DEFORMED WR RADIUS      #1STD  (Rd      ) [335.205139]
2501:                  #2STD  (        ) [354.134247]
2503:                  #3STD  (        ) [372.764618]
2505:                  #4STD  (        ) [394.902466]
2507:                  #5STD  (        ) [473.341309]
2509:SCREW POSITION        #1STD  (S       ) [-1.037833]
2511:                  #2STD  (        ) [-1.014397]
2513:                  #3STD  (        ) [-0.779581]
2515:                  #4STD  (        ) [-0.014182]
2517:                  #5STD  (        ) [-0.524868]
2519:ROLL GAP (DELTA)      #1STD  (dS      ) [-0.009527]
2521:                  #2STD  (        ) [-0.163719]
2523:                  #3STD  (        ) [-0.027459]
2525:                  #4STD  (        ) [-0.151531]
2527:                  #5STD  (        ) [-0.094622]
2539:MILL MODULUS          #1STD  (K       ) [528995.062500]
2541:                  #2STD  (        ) [512995.343750]
2543:                  #3STD  (        ) [518995.218750]
2545:                  #4STD  (        ) [536994.875000]
2547:                  #5STD  (        ) [428879.093750]
2567:                  #5STD  (        ) [0.100000]
```

```
2589:FRICTION COEFFICIENT     #1STD  (myu     ) [0.070000]
2591:                    #2STD  (        ) [0.060388]
2593:                    #3STD  (        ) [0.037302]
2595:                    #4STD  (        ) [0.023514]
2597:                    #5STD  (        ) [0.130000]
2701:ROLLING TORQUE          #1STD  (gr      ) [1454686.250000]
2703:                    #2STD  (        ) [11501342.000000]
2705:                    #3STD  (        ) [10422274.000000]
2707:                    #4STD  (        ) [8127517.500000]
2709:                    #5STD  (        ) [4222197.000000]
2731:TENSION RATIO    ENT.  - #1STD  (tbrate  ) [1.007702]
2733:               #1STD - #2STD  (        ) [0.863744]
2735:               #2STD - #3STD  (        ) [0.836752]
2737:               #3STD - #4STD  (        ) [0.782768]
2739:               #4STD - #5STD  (        ) [0.787532]
2741:               #5STD - DEL.   (        ) [0.531535]
2753:LOW ROLL FORCE CORRECT.  #1STD  (Sa      ) [0.656000]
2755:                    #2STD  (        ) [0.893000]
2757:                    #3STD  (        ) [0.762000]
2759:                    #4STD  (        ) [0.688000]
2761:                    #5STD  (        ) [0.800000]
2783:ELONGATION          .  #1STD  (el      ) [0.000000]
2785:                    #2STD  (        ) [0.238931]
2787:                    #3STD  (        ) [0.474141]
2789:                    #4STD  (        ) [0.695759]
2791:                    #5STD  (        ) [0.000000]
2813:CONTACT LENGTH OF IMR    SP     (X       ) [1750.000000]
2815:                    SP     (        ) [1750.000000]
2817:                    SP     (        ) [1750.000000]
2819:                    SP     (        ) [1750.000000]
2821:                    #5STD  (        ) [1675.000000]
2833:DEFORM.RESIST.PARA L            (kpl     ) [64.769684]
2835:DEFORM.RESIST.PARA M            (kpm     ) [0.003164]
2837:DEFORM.RESIST.PARA N            (kpn     ) [0.210289]
2839:BOLZMAN CONSTANT                (Bz      ) [0.000086]
2841:INITIAL UNIT TENSION FOR TR     (iut     ) [1.300000]
2843:INITIAL TOTAL TENSION FOR TR    (it      ) [8090.705078]
2857:AVE. BR DIAMETER        #1STD  (DB      ) [1362.324951]
2859:                    #2STD  (        ) [1361.145020]
2861:                    #3STD  (        ) [1368.239990]
2863:                    #4STD  (        ) [1366.849976]
2865:                    #5STD  (        ) [1366.050049]
2867:AVE. BR DIAMETER AT TEST #1STD  (DB0     ) [1420.000000]
2869:                    #2STD  (        ) [1420.000000]
```

```
2871:                 #3STD  (       ) [1420.000000]
2873:                 #4STD  (       ) [1420.000000]
2875:                 #5STD  (       ) [1420.000000]
2877:WR BENDER              #1STD  (wrb     ) [37564.000000]
2879:                 #2STD  (       ) [37564.000000]
2881:                 #3STD  (       ) [37564.000000]
2883:                 #4STD  (       ) [37564.000000]
2885:                 #5STD  (       ) [23693.000000]
2887:IMR BENDER             #5STD  (imrb    ) [20462.000000]
2889:IMR SHIFT            #5STD  (imrsh   ) [37.500000]
2891:MATERIAL TEMPERATURE     #1STD  (xt      ) [325.267792]
2893:                 #2STD  (       ) [401.848389]
2895:                 #3STD  (       ) [468.857697]
2897:                 #4STD  (       ) [468.857697]
2899:                 #5STD  (       ) [468.857697]
2903:TAPER END DIAMETER            (tpedia  ) [1610.754272]
2905:TAPER END WEIGHT COEFFICIENT    (ctp     ) [0.950000]
2912:ROLL ROUGHNESS          #1STD  (ra      ) [75]
2913:                 #2STD  (       ) [50]
2914:                 #3STD  (       ) [42]
2915:                 #4STD  (       ) [40]
2916:                 #5STD  (       ) [426]
2917:WR BENDER CHANGE FORCE   #1STD  (wrbcf   ) [-245.775543]
2919:                 #2STD  (       ) [3813.099854]
2921:                 #3STD  (       ) [2944.833496]
2923:                 #4STD  (       ) [5329.907227]
2925:                 #5STD  (       ) [1014.799255]
2927:IMR BENDER CHANGE FORCE  #5STD  (imrbcf  ) [1250.392090]
2969:ROLL FORCE & WR BENDER   #1STD  (pwrb    ) [1118335.625000]
2971:                 #2STD  (       ) [1176881.875000]
2973:                 #3STD  (       ) [1156777.875000]
2975:                 #4STD  (       ) [1148681.875000]
2977:                 #5STD  (       ) [854018.562500]
3107:FORWARD SLIP - FRICTION  #1STD  (fmyu    ) [-0.109103]
3109:                 #2STD  (       ) [0.369444]
3111:                 #3STD  (       ) [0.614647]
3113:                 #4STD  (       ) [0.845958]
3115:                 #5STD  (       ) [0.037811]
3117:ROLL FORCE - BACK TENS.  #1STD  (ptb     ) [-15.531390]
3119:                 #2STD  (       ) [-15.710938]
3121:                 #3STD  (       ) [-13.862904]
3123:                 #4STD  (       ) [-12.925000]
3125:                 #5STD  (       ) [-10.764516]
3127:ROLL FORCE - FRICTION    #1STD  (pmyu    ) [3345.000000]
```

```
3129:                 #2STD  (       ) [4197.430664]
3131:                 #3STD  (       ) [4800.005859]
3133:                 #4STD  (       ) [6302.674316]
3135:                 #5STD  (       ) [2932.259521]
3149:YOUNG'S MODULUS               (EE     ) [21700.000000]
3151:POISSON'S RATIO             (vv     ) [0.300000]
3153:HITCHCOOK'S COEFFICIENT        (CH     ) [0.000214]
3155:UNIT MILL MODULUS             (KW     ) [5000000.000000]
3273:BARREL LENGTH           #1STD  (BL     ) [1750.000000]
3275:                 #2STD  (       ) [1750.000000]
3277:                 #3STD  (       ) [1750.000000]
3279:                 #4STD  (       ) [1750.000000]
3281:                 #5STD  (       ) [1750.000000]
3319:IMR SHIFT DELTA             (delta  ) [30.000000]
3431:DEFORMATION RESIST FOR LMN CALC.(k_km    ) [50.285431]
3463:WRB (PRES)            #1STD  (wrb_p   ) [80.000000]
3465:                 #2STD  (       ) [80.000000]
3467:                 #3STD  (       ) [80.000000]
3469:                 #4STD  (       ) [80.000000]
3471:                 #5STD  (       ) [80.000000]
3473:WRBCF (PRES)          #1STD  (wrbcf_p ) [-0.623637]
3475:                 #2STD  (       ) [9.675463]
3477:                 #3STD  (       ) [7.472300]
3479:                 #4STD  (       ) [13.524250]
3481:                 #5STD  (       ) [4.068963]
3483:IMRB (PRES)               (imrb_p  ) [80.000000]
3485:IMRBCF (PRES)              (imrbcf_p) [4.068963]
3487:WRB (OPERATOR)         #1STD  (wr_op   ) [0.000000]
3489:                 #2STD  (       ) [0.000000]
3491:                 #3STD  (       ) [0.000000]
3493:                 #4STD  (       ) [0.000000]
3495:                 #5STD  (       ) [0.000000]
3497:IMRB (OPERATOR)             (imr_op  ) [0.000000]
3499:UNIT TENSION (NORMAL)  ENT - #1 (utb0    ) [3.500000]
3501:                #1  - #2 (       ) [16.000000]
3503:                #2  - #3 (       ) [15.500000]
3505:                #3  - #4 (       ) [14.500000]
3507:                #4  - #5 (       ) [15.500000]
3509:                #5  - DEL(       ) [4.000000]
3511:IMR BL LENGTH               (bl_imr  ) [1650.000000]
3513:MATERIAL COIL NUMBER          (mno     ) [............]
3519:TRACKING INDEX              (ti      ) [0]
3520:SPARE                   (sp5     ) [2 bytes]
```

ประวัติผู้เขียนวิทยานิพนธ์

นายเนติวุฒิ ม้ารุ่งอรุณ เกิดเมื่อวันที่ 25 มีนาคม พ.ศ.2513 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาชั้นมัธยมศึกษาปีที่ 6 จากโรงเรียนสวนกุหลาบวิทยาลัย กรุงเทพมหานคร สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล จากมหาวิทยาลัยเกษตรศาสตร์เมื่อปี พ.ศ.2542 และเข้าศึกษาต่อ หลักสูตรปริญญาวิศวกรรมศาตรมหาบัณฑิต สาขาวิศวกรรมโลหการ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ.2544