THE APPLICATION OF MATHEMATICAL PROGRAMMING FOR THE DESIGN OF AXIALLY LOADED MEMBERS (II)

BY

DR. SOAD MOHAMED SERAG

ABSTRACT:

In a previous paper we outlined the application of geometric programming in the design of axially loaded members. In this paper we discuss the solution technique and extend the formulation to the general design problem based or mainly kind of material, type of machinability and (design cost), the performance (displacement).

I- INTRODUCTION:

In our previous paper we concluded that the optimal design problem can be cited as follows.

Max. P.E

$$\text{displacent} = II/2 \left(\frac{Sc^2}{(1+p)^2 E}\right) \frac{dr^4}{N2\ Kr^2} \quad \frac{L1}{d1^2} + \frac{L2}{d2^2}$$

Subject to :-

$$\left.\begin{array}{l} d1 \leq d1max \\ d2 \leq d2max \\ L1+L2 \leq Ltmax \\ L1+L2 \geq Ltmin \end{array}\right\} \text{....... design}$$

$$\sum_{1} Coi\ d1^{\alpha 1i} . d2^{\alpha 2i} . L1^{B1i} . L2^{B2i} \leq C \quad \Big\} \text{Cost}$$

II. SOLUTION TECHNIQUE:

The following section restate the G.P. again and our case is a signomial optimization problem. Fig. (1) outline the flow chart of the solution technique followed by sub-program of solution for the linearisd coefficient system.

* Dr. Soad Mohamed Serag, B.Sc., Ph.D., Lecturer, Production Engineering & Machine Design Department, Faculty of Engineering and Technology, Menoufia University.

GEOMETRIC PROGRAMMING COMPUTER PROGRAM:

As the number of variables and terms increase either in the objective equations or constraints, it becomes impractical to solve by the regular method. A computer program was developed. The logic diagram of the computer program is indicated in Figure 1. For the purpose of completion, the mathematical model of geometric programming will be written again.

Geometric programming finds the minimun of a multivariable, nonlinear function of geometric form:

$$\text{Minimize} \quad Y_O(\underline{x}) = \sum_{t=1}^{T_O} \sigma_{Ot} C_{Ot} \prod_{n=1}^{N} \left(x_n \right)^{a_{Otn}}$$

subject to constraints of geometric form

$$\sum_{t=1}^{T_m} \sigma_{mt} \, C_{mt} \prod_{n=1}^{N} \left(x_n \right)^{a_{mtn}} \leq \quad m$$

σ_{Ot} and $\sigma_{mt} = \pm 1$ (the sign of each term in the objective function and m^{th} constraint, respectively)

C_{Ot} and $C_{mt} > 0$ (the coefficients of each term in the objective function and m^{th} constraint, respectively)

$X_n > 0$ (the independent variables)

$\sigma_m = \pm 1$ (the constant bound of the m^{th} constraint

a_{Otn} and a_{mtn} are the exponents of the n^{th} independent variable of the t^{th} term of the objective function and m^{th} constraint, respectively

M is the number of constraints

T_O is the number of terms in the objective function

$T_1, T_2, \ldots, T_m$ are the number of terms in each constraint, 1 to M, respectively.

$\sigma = \pm 1$ assumed sign of the bojective function

As the equation developed mostly confirm with the geometric programming model, the geometric programming technique will be used for the optimization design system of axially loaded members the logic diagram of the computer program that will be used is indicated in figure 1.

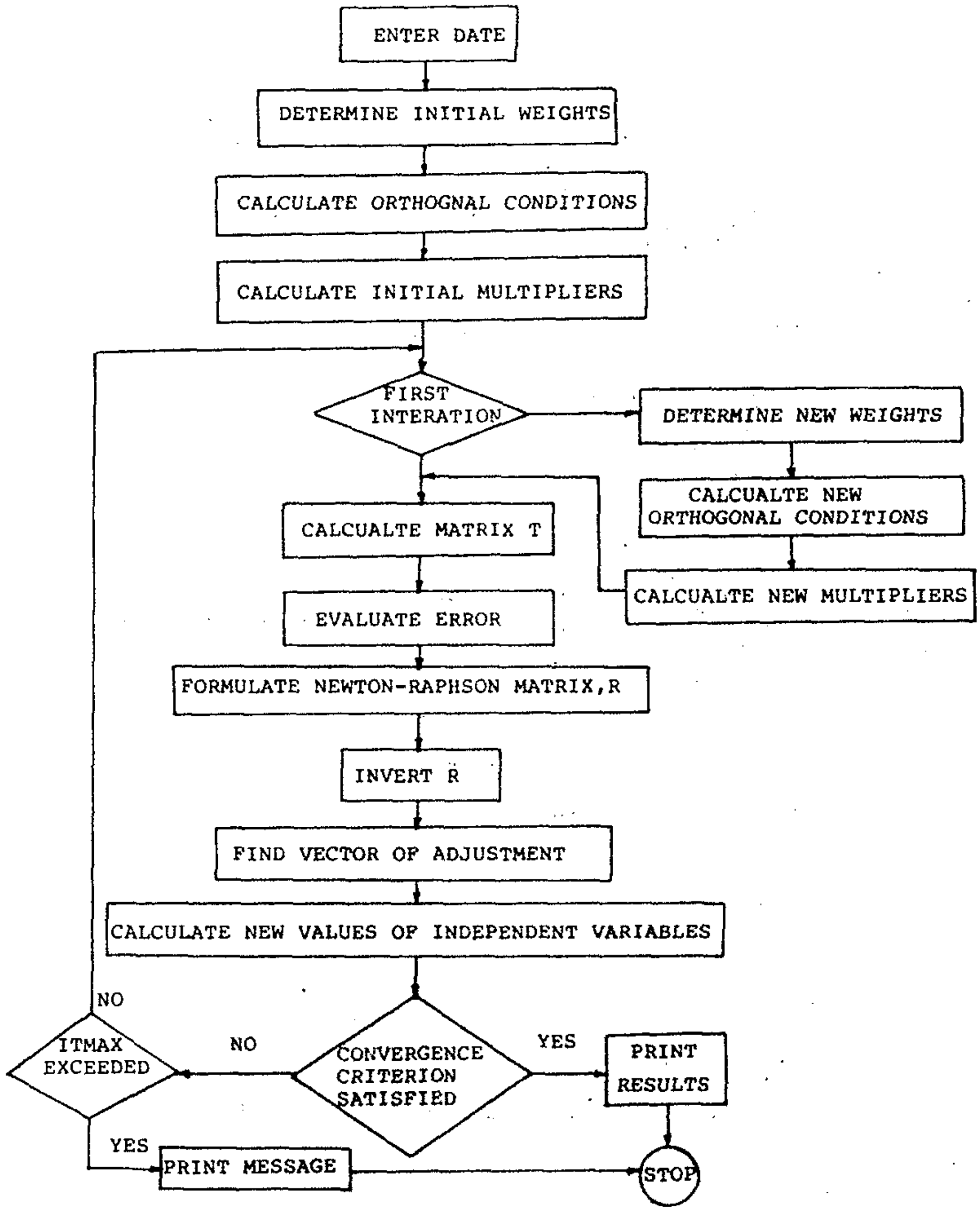

FIGURE 1. GEOMETRIC PROGRAMMING (GEOMTRY ALGORITHM) LOGIC DIAGRAM

```
      SUBROUTINE IMPRUV (NN, A, UL, B, X, DIGITS)
      DIMENSION A(30,30), UL(30,30), B(30), X(30), R(30), DX(30)
C     USES ABS(), AMAX1(), ALOG10()
      DOUBLE PRECISION SUM
      N = NN
C
      EPS = 1.0E-8
      ITMAX = 16
C     ***  EPS AND ITMAX ARE MACHINE DEPENDENT. ***
C
      XNORM = 0.0
      DO 1 I = 1,N
    1    XNORM = AMAX1(XNORM,ABS(X(I)))
      IF (XNORM) 3,2,3
    2    DIGITS = -ALOG10(EPS)
         GO TO 10
C
    3 DO 9 ITER = 1,ITMAX
         DO 5 I = 1,N
            SUM = 0.0
            DO 4 J = 1,N
    4          SUM = SUM + A(I,J)*X(J)
            SUM = B(I) - SUM
    5    R(I) = SUM
C        *** IT IS ESSENTIAL THAT A(I,J)*X(J) YIELD A DOUBLE PRECISION
C             RESULT AND THAT THE ABOVE + AND - BE DOUBLE PRECISION.  ***
         CALL SOLVE (N,UL,R,DX)
         DXNORM = 0.0
         DO 6 I = 1,N
            T = X(I)
            X(I) = X(I) + DX(I)
            DXNORM = AMAX1(DXNORM,ABS(X(I)-T))
    6    CONTINUE
         IF (ITER-1) 8,7,8
    7       DIGITS = -ALOG10(AMAX1(DXNORM/XNORM,EPS))
    8    IF (DXNORM-EPS*XNORM) 10,10,9
    9 CONTINUE
C     ITERATION DID NOT CONVERGE
      CALL SING(3)
   10 RETURN
      END
```

```
      SUBROUTINE SING (IWHY)
   11 FORMAT(54H0MATRIX WITH ZERO ROW IN DECOMPOSE.                    )
   12 FORMAT(54H0SINGULAR MATRIX IN DECOMPOSE.  ZERO DIVIDE IN SOLVE.  )
   13 FORMAT(54H0NO CONVERGENCE IN IMPRUV. MATRIX IS NEARLY SINGULAR.  )
      NOUT = 3
C        NOUT = STANDARD OUTPUT UNIT
      GO TO (1,2,3),IWHY
    1 WRITE (NOUT,11)
      GO TO 10
    2 WRITE (NOUT,12)
      GO TO 10
    3 WRITE (NOUT,13)
   10 RETURN
      END
```

```
      SUBROUTINE DECOMP (NN, A, UL)
      DIMENSION A(30,30), UL(30,30), SCALES(30), IPS(30)
      COMMON IPS
      N = NN
C
C     INITIALIZE IPS, UL AND SCALES
      DO 5 I = 1,N
         IPS(I) = I
         ROWNRM = 0.0
         DO 2 J = 1,N
            UL(I,J) = A(I,J)
            IF(ROWNRM-ABS(UL(I,J))) 1,2,2
    1       ROWNRM = ABS(UL(I,J))
    2    CONTINUE
         IF (ROWNRM) 3,4,3
    3    SCALES(I) = 1.0/ROWNRM
         GO TO 5
    4    CALL SING(1)
         SCALES(I) = 0.
    5 CONTINUE
```

```
C
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
      NM1 = N-1
      DO 17 K = 1,NM1
         BIG = 0.0
         DO 11 I = K,N
            IP = IPS(I)
            SIZE = ABS(UL(IP,K))*SCALES(IP)
            IF (SIZE-BIG) 11,11,10
   10          BIG = SIZE
               IDXPIV = I
   11    CONTINUE
         IF (BIG) 13,12,13
   12       CALL SING(2)
            GO TO 17
   13    IF (IDXPIV-K) 14,15,14
   14       J = IPS(K)
            IPS(K) = IPS(IDXPIV)
            IPS(IDXPIV) = J
   15    KP = IPS(K)
         PIVOT = UL(KP,K)
         KP1 = K+1
         DO 16 I = KP1,N
            IP = IPS(I)
            EM = -UL(IP,K)/PIVOT
            UL(IP,K) = -EM
            DO 16 J = KP1,N
               UL(IP,J) = UL(IP,J) + EM*UL(KP,J)
C           INNER LOOP.  USE MACHINE LANGUAGE CODING IF COMPILER
C           DOES NOT PRODUCE EFFICIENT CODE.
   16    CONTINUE
   17 CONTINUE
      KP = IPS(N)
      IF (UL(KP,N)) 19,18,19
   18 CALL SING(2)
   19 RETURN
      END
```

```
      SUBROUTINE SOLVE (NN, UL, B, X)
      DIMENSION UL(30,30), B(30), X(30), IPS(30)
      COMMON IPS
      N = NN
      NP1 = N+1
C
      IP = IPS(1)
      X(1) = B(IP)
      DO 2 I = 2,N
         IP = IPS(I)
         IM1 = I-1
         SUM = 0.0
         DO 1 J = 1,IM1
    1       SUM = SUM + UL(IP,J)*X(J)
    2 X(I) = B(IP) - SUM
C
      IP = IPS(N)
      X(N) = X(N)/UL(IP,N)
      DO 4 IBACK = 2,N
      I = NP1-IBACK
C        I GOES (N-1),....,1
         IP = IPS(I)
         IP1 = I+1
         SUM = 0.0
         DO 3 J = IP1,N
    3       SUM = SUM + UL(IP,J)*X(J)
    4 X(I) = (X(I)-SUM)/UL(IP,I)
      RETURN
      END
```

THE GENERAL DESIGN PROBLEM

If we consider a mechanism composed of N members (or links, although each link might be divided into several members), the design problem is to find the cross-sectional sizes of the members, characterized by the variables y_i for $i = 1,2, \ldots, N$ such that the total volume:

$$V = \sum_{i=1}^{N} A_i L_i \qquad \ldots\ldots\ldots\ldots\ldots(1)$$

is minimized , while the stress in the links due to inertia effects (or external loading) is limited by:

$$|\sigma_i| \leq |\bar{\sigma}_i| \qquad i=1,2,\ldots,N \qquad \ldots\ldots\ldots\ldots\ldots(2)$$

and the displacements at the joints (or anywhere along the links) are limited by:

$$u_j \leq \bar{u}_j \qquad j = 1,2,\ldots,J \qquad \ldots\ldots\ldots\ldots\ldots(3)$$

where A_i and L_i are the area and length of the ith member, σ_i is the maximum stress in the ith member during the mechanism'sentire motion, $\bar{\sigma}_i$ is the allowable stress, u_j is the maximum displacement for some point j on the mechanism during its rotation, and $\bar{u}_j$ is the allowable displacement at this point. It is assumed that the cross-sectional size of each member is completely specified by the single variable y_i. This variable could be area A_i , the diameter for circular members, or a similar quantity. It is further assumed that once this variable is known, then all other cross-sectional properties such as area, moment of inertia, etc., can be obtained from it As a result, the area A_i in equation (1) can be written as a function of the variables y_i to the bth power:

$$A_i = C y_i^{b} \qquad \ldots\ldots\ldots\ldots\ldots(4)$$

where C is some known constant.

where C is some known constant.

If the displacements and stresses in the mechanism are periodic functions of time (in the steady state) then constraints (2) and (3) might be decretized for K positions of the mechanism are periodic function of time (in the steady state) then constraints (2) and (3) might be decretized for K positions of mechanism:

$$|\sigma_{ik}| \leq \bar{\sigma} \qquad i=1,2,\dots,N \quad k=1,2,\dots,k \qquad \dots\dots(5)$$

$$u_{jk} \leq \bar{u}_j \qquad j=1,2,\dots,J \quad k=1,2,\dots,K \qquad \dots\dots(6)$$

Stress constraints. If only stress constraints exist on the problem then using the Kuhn-Tucker conditions, the following stresss ratio formula can be obtained for redesign of the cross-sectional areas:

$$(A_i)_{\nu+1} = \left[\frac{\max_k |\sigma_{ik}|}{\bar{\sigma}_i}\right]^{\eta}_{\nu} (A_i)_{\nu} \qquad \dots\dots(7)$$

where ν is the iteration counter η is a relaxation parameter. The parameter η controls the stability of the method and speed of convergence. For all mechanism design problems considered $\eta = 1$ gave the best results. Of this relation as well as its to mechanism design.

Displacement constraints. If only displacement constraints exist on the problem, then we can define the functional:

$$\phi = V + \sum_{j=1}^{j} \sum_{k=1}^{k} \lambda_{ik} (u_{jk} - \bar{u}_j)$$

from which the Kuhn-Tucker necessary conditions for an optimal design are:

$$\frac{\partial V}{\partial y_i} + \sum_{j=1}^{j}\sum_{k=1}^{K} \frac{\partial u_{jk}}{\partial y_i}\lambda_{jk} = 0 \;, \; i = 1,2,\dots,N \qquad \dots\dots(8)$$

$$u_{jk} - \bar{u}_j \leq 0, \quad \lambda_{jk} \geq 0 \qquad j = 1,2,\dots,J \quad K = 1,2,\dots,K \qquad \dots\dots(9)$$

where λ_{jk} are lagranage multipliers.
If we assume that the pth displacement constraint at the qth sicrete position of the mechanism is active, and all other constraints are not active, then equations (8) and (9) become:

$$\frac{\partial v}{\partial yi} + \lambda_{pq} \frac{\partial upq}{\partial yi} = 0 \qquad i=1,2,\ldots..N \qquad \ldots\ldots\ldots\ldots\ldots(10)$$

$$upq - \bar{u}p = 0 \quad \lambda_{pq} > 0 \qquad \ldots\ldots\ldots\ldots\ldots(11)$$

The derivative of the displacement upq with respect to the design variable yi can obtained from the vibrational equations describing the motion. For example, using the equations

$$M\ddot{\vec{X}} + K\vec{X} = \vec{F} \qquad \ldots\ldots\ldots\ldots\ldots(12)$$

the derivative with respect to the design variable yi is:

$$\frac{\partial M}{\partial yi} \ddot{\vec{X}} + M \frac{\ddot{\vec{X}}}{\partial yi} + \frac{\partial K}{\partial yi} \vec{X} + K \frac{\partial X}{\partial yi} = \frac{\partial \vec{F}}{\partial yi}$$

Rearranging gives:

$$M \frac{\partial \ddot{\vec{X}}}{\partial yi} + K \frac{\partial \vec{X}}{\partial yi} = \frac{\partial \vec{F}}{yi} - \frac{\partial M}{\partial yi} \ddot{\vec{X}} - \frac{\partial K}{\partial yi} \vec{X} \qquad \ldots\ldots\ldots\ldots\ldots(13)$$

which is a differential equation that can be solved for $\partial\vec{X}/\partial yi$
One of the components of this vector will be the required $upq/\partial yi$

Needed in equation (9). Experience has shown, however, that the terms is equation (13) involving mass are small compared with the remaining terms. Thus equation (13) could be.
written as:

$$\frac{\partial \vec{X}}{\partial yi} = K^{-1} \left[\frac{\partial \vec{F}}{\partial yi} - \frac{\partial K}{\partial yi} \vec{X} \right] \qquad \ldots\ldots\ldots\ldots\ldots(14)$$

Knowing the value of $\partial upq/\partial yi$, substituting the expression for the volume into equation (10), and summing over all the members gives:

$$\lambda_{pq} = -\frac{Cb \sum_{j=1}^{N} Lj yj^{b-1}}{\sum_{i=1}^{N} \frac{\partial upq}{\partial yi}} \qquad \ldots\ldots\ldots\ldots\ldots (15)$$

Substituting this back into equation (10) produces:

$$1 = \left(\frac{\sum_{i=1}^{N} Lj yj^{b-1}}{Lj yi^{\ b-1}} \right) \left(\frac{\frac{\partial upq}{\partial yi}}{\sum_{i=1}^{N} \frac{\partial upq}{\partial yi}} \right) \quad i = 1,2,\ldots,N \qquad \ldots\ldots (16)$$

Also, equation (11) can be written as:

$$1 = \frac{upa}{\bar{u}p} \qquad \ldots\ldots\ldots\ldots\ldots (17)$$

Equations (16) and (17) are expressions which must be satisfied at the optimal design. If a particular design is not optimal then the right-hand sides of these equations will not equal one. Thus we might form a recursion relation based on the right-hand sides of these equations which will change the design from one iteration to the next. It is observed that the elements Kij of the stiffness matrix K in the vibrational equations (12) are approximately linear functions of the moment of inertia of the cross section for each member i.e.,

$$Kij \cong \alpha ij1I1 + \alpha ij2I2 + \ldots + \alpha ijNIN$$

Where $\alpha ij1, \ldots, \alpha ijN$ are constants. Also, the elements Fj of the forcing function F in equation (12) are approximately linear functions of the cross-sectional areas of the members, i.e.,

$$Fj \cong Bj1A1 + Bj2A2 + \ldots + BjNAN$$

where Bj1,....,Bjn are constants. Thus for common crossectional shapes (take a circular shape for example where I=A2/4) the defiection upq is inversely related to the areas of the members, since:

$$\vec{X} \cong K^{-1} \quad F$$

ingnoring the mass and acceleration terms in equation (12). More specifically, it is observed that upq is approximately linearly related to 1/Ai for circular shapes. Thus from equation (17), if upq/$\bar{u}$p is greater than one for a particular design, then the areas of the members should be increased. As a result, an iterative equation migh be formed based on this ratio, i.e.,

$$(Ai)_{p+1} = \frac{Upa}{\bar{U}p} (Ai)_{\nu} \quad \text{..............(18)}$$

where we have assumed a linear relationship between upq/ūp and 1/Ai However this iterative formula does not take into account the other optimality equations (16), which are also related to the areas. Since the displacement upq is proportional to 1/A, then the derivative ∂upq/∂yi will be approximately proportional to 1/Ai2 thus the second term in equation (16), i.e.,

$$\frac{\frac{\partial upq}{\partial yi}}{\sum_{j=1}^{N} \frac{\partial upq}{\partial yi}}$$

will generally be reduced if the area Ai is increased. Of course this is a nonlinear relationship. Similarly the first term in equation (16) i.e.,

$$\frac{\sum_{j=1}^{N} Ljyj^{b-1}}{Liyi^{b-1}}$$

will decrease with increasing Ai, since Ai = Cyi^{b}. Thus an iterative equation might be formed from equation (16) similar to that of equation (18) except it would be nonlinear. However these equation might be combined to form:

$$(A_i)_{u+1} = \left\{ \left| \frac{u_{pq}}{u_p} \right| \left(\left[\frac{\sum_{i=1}^{N} L_j y_j^{b-1}}{L_i y_i^{b-1}} \right] \left[\frac{\frac{\partial u_{pq}}{\partial y_i}}{\sum_{j=1}^{N} \frac{U_{PQ}}{Y_I}} \right] \right)^{\eta} A_i \right\} \quad \text{...........(19)}$$

which is the primary recursion relation for redesign of mechanisms involving only displacement constraints.

The exponent η, called the relaxation parameter, takes into account the nonlinear relationship between the area Ai and the right-hand side of equation (16). Experience has shown that for mechanism design problems values between 0.001 and 0.2 gave excellent results. For small values of, the technique converges very slowly but for larger values of n stability problems sometimes occur, where the areas oscillated from one iteration to the next. It is observed that the iterative equation (19) takes into account both optimality conditions (16) and (17). The design variables (through Ai) will continue to change as long as either condition is not satisfied.

In the derivation of the interative equation (19), it was assumed that only one displacement constraint was most active (or most violated). At the optimal design, it is possible, even likely, that more than one constraint will be active, however this point, for real design problems, is almost never reached by currenly available nonlinear optimization methods. There will be only one most critical or violated constraint. The iterative formula (19) derived here simply takes advantage of this characteristic. It is assumed that only one displacement constraint at some input crank angle position is most active, all other constraints are considered inactive. In the special case where two or more constraints are exactly equal because or symmetry or other limitations, these exactly equal displactly equal displacements are treated as one active constraint.

Based on the iterative equation (19) a design algorithm for displacement constrained mechanism design problems can be stated:

1. Choose a design yi for i = 1,2,....,N (and calculate areas Ai). Choose a value of the relaxation parameter η (between 0.05 and 0.15 is suggested).

2. Evaluate the constrained displacements at K discrete positions of the mechanism during its motion.

3. Record the positions of the mechanism where each constrained displacement reaches its maximum.

4. Find the most critical displacement upq, i.e., the one for which upq/up is maximum.

5. Compute the derivatives ∂upq/∂yi.

6. Use equation (19) to resize the elements.

7. If the volumes of two successive iterations are very close (saybetween 0.001 percent and 0.01 percent difference) and the displacement constraints are satisfied to within a given tolerance (say 0.005 percent) then stop; otherwise go to step 8.

8. Determine the displacements for the new design only at the recorded positions from step 3 and continue from step 4.

This procedure has been applied to a number of examples and has been found to be effective on displacement constrained mechanism design problems. Step 8 in the process was used to save computational time. It was found that in almost all cases, no matter what the starting design, the positions of the mechanism at which the maximum displacements occurred did not change throughout the optimization. Thus, after the first iteration the displacements need only be calculated at the recorded maximum positions. A final complete analysis for the optimal design could be used to verify these maximums.

Stress and Displacement Constraints. More practical problems in mechanism design occur when both stress and displacement constraints are included. The stress recursion formula of equation (7) can be combined with the displacement recursion formula of equation (19) to produce the following design procedure:

1. Choose a design yi for i=1,2,....,N (and calculate areas Ai). Choose a value of the relaxation parameter η (between 0.005 and 0.15).

2. Calculate the stesses in the links, and the displacements at those locations on the links which are constrained, at the K discrete positions for the mechanism during its motion.

3. Record the positions of the mechanism where each displacement and stress is maximum.

4. Find the most critical displacement upq, i.e., the one for which/Upq/Uq/is maximum.

5. Compute the constraint derivatives $\partial upq/\partial yi$.

6. Group the members as follows:

a. if $\max_k /\sigma ik/ > \bar{\sigma} ij$ then member i belongs to group G1.

b. Otherwise member i belongs to group G2. Note that either group could be empty.

7. Use the recursion formula:

$$(Ai)\ (\nu)_{i+1} = \left[\left(\frac{\max_k /\sigma(ik)/}{\bar{\sigma}_{(i)}}\right) A(i)\right]_\nu$$

to resize the members of G1. Use the formula:

$$(Ai)^\nu i+1 = \left\{\left|\frac{upq}{up}\right| \left[\left(\frac{\sum_{j=1}^{N} Ljyj^{b-1}}{Liyi^{b-1}}\right) \frac{\frac{\partial upq}{\partial yi}}{\sum_{i=1}^{N} {}^{u}pq/yi}\right]^\eta A_i\right\}_\nu$$

for those members of G2.

8. If the volumes of two successive iterations are approximately the same (say between 0.001 percent and 0.01 percent) and all constraints (both stress and displacement) are satisfied to within a prescribed tolerance (between 0.001 percent and 0.005 percent) then stop; otherwise go to step 9.

9. If the change in volume in this iteration is of different sign from that of the previous iteration, i.e.,

$$(vi+1 - Vi)\ (Vi-Vi-1) < 0 \qquad \text{..............(20)}$$

and n has not previously been changed, then reduce n by one half.

10. Determine the stresses and displacements for the new design only at the recorded postions from step 3 and continue from step 4.

As mentioned previously the value of n controls the speed of convergence and stability of the method. Step 9 allows the uses to start with a larger value of n, so that the method will converge rapidly towards the optimal design. However with a large n the method may overshoot the optimum resulting in oscillation, which sometimes can become unstable. As soon as oscillations are first detected using equation (20), the value of n is reduced by one half its initial value and the procedure continued. Experience has shown that this was sufficient to stabilize the procedure (assuming n is started between 0.05 and 0.15 as suggested) and thus no further reduction in n was necessary. An alternate approach, if additional stability problems are encountered, would be to continue to reduce n as long as oscillations are occuring.

REFERENCES

1. DUFFIN, R.Y, G.L PETERSON AND C.Zener "GEOMETRIC PROGRAMMING" N.Y 1967

2. WILLARD, Y ZANGWILL "NON-LINEAR PROGRAMMING" PRENICE HALL INC. 1967.

3. U. PASSY "MODULAR. DESIGN-AN APPLICATION OF STRUCTURE GEOMETRIC PROGRAMMING" Jr. ORSA VOL. 18 No.3 PP 471-480 1970.

4. BEN ISRAEL, LUIS PASCAL "VECTOR VALUED CRITERION IN GEOMETRIC PROGRAMMING" Jr. ORSA VOL. 19 No. 1 PP 98-104.

5. NASSER, SEFIN "OPTIMUM DESIGN OF THE GEAR-BOX FOR MACHINE TOOL BY GEOMETRIC PROGRAMMING" ENG. RES. BULLETIN, MENOUFIA UNIVERSITY. VOL II., PART I 1979.

6. THORNTON, WILLIMERT, KHAN "MECHANISM OPTIMIZATION VIA OPTIMALITY CRITERION TECHNIQUES". HSAME, VOL 101 JULY 1979.

7. SOAD SERAG "THE APPLICATION OF MATHEMATICAL PROGRAMMING IN THE DESIGN OF AXIALLY LOADED MEMBERS (I).

تطبيقات في استخدام البرامج الرياضية

في تصميم الأطراف محورية الحمل (٢)

الدكتورة / سعاد سراج

في هذا البحث تم استخدام البرنامج المناسب لحل مسألة البرمجة الهندسية التي تمت صياغتها في البحث السابق (استخدام البرامج الرياضية في تصميم الأطراف محورية الحمل ٣ ") كما تم تعميم الدراسة لتشمل مسألة التصميم في صورتها العامة آخذين في الاعتبار الاجهادات الحملية والازاحة ونوع المعدن وتكاليف المنشأ .

وقد تبين أن اضافة قيود التكاليف كدالة في المادة وطريقة تشغيلها تغير في هيكل الحل عن الطرق التقليدية - كما أنه في حالة تعدد الاجزاء فان الطرق التقليدية لا يمكن أن تؤدي الى تصميمات مثلى .