731

3290

ADA108684

R and D CENTER

LABORATORY

TECHNICAL REPORT

NO. 12587

DESIGN SENSITIVITY ANALYSIS AND OPTIMIZATION OF CONSTRAINED DYNAMIC SYSTEMS

Interim Report

16 June 1981

Contract No. DAAK30-78-C-0096

N.C. Barman & E.J. Haug
College of Engineering
The University of Iowa
University of Iowa Rept. No. 56

Ronald R. Beck, Project Engineer, TACOM

20031211062

U.S. ARMY TANK-AUTOMOTIVE COMMAND
RESEARCH AND DEVELOPMENT CENTER
Warren, Michigan 48090

NOTICES

The findings in this report are not to be construed as an official Department of the Army position.

Mention of any trade names or manufacturers in this report shall not be construed as advertising nor as an official endorsement of approval of such products or companies by the U.S. Government.

Destroy this report when it is no longer needed. Do not return it to originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 12587	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Design Sensitivity Analysis and Optimization of Constrained Dynamic Systems		5. TYPE OF REPORT & PERIOD COVERED Interim to June 1979 6. PERFORMING ORG. REPORT NUMBER 56
7. AUTHOR(s) N.C. Barman & E.J. Haug University of Iowa Ronald R. Beck, TACOM		8. CONTRACT OR GRANT NUMBER(s) DAAK30-78-C-0096
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of Iowa College of Engineering Iowa City, IA 52242		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Tank-Automotive Command, R&D Center Tank-Automotive Concepts Lab, DRSTA-ZSA Warren, MI 48090		12. REPORT DATE 16 June 81 13. NUMBER OF PAGES 180
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Euler angles, Dynamics, Bodies, Response, Interaction, Stability, Lagrangian Functions, Optimization, Sensitivity

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

In this report, the technical objective is the derivation of a systematic and unified theory and organization of a corresponding general computer program for the design of constrained dynamic systems by judicious selection of the most suitable methods from the following branches of mathematics and mechanics: (a) Optimization Methods, (b) Rigid Body Mechanics, (c) Numerical Integration Methods, (d) Matrix Manipulation Methods. Accordingly, a method of formulating and automatically integrating the equations of motion and design sensitivity

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

adjoint equations for general constrained dynamic systems is presented. Design sensitivity analysis is carried out using a state-space method that has previously been used for design optimization of linear structural systems. Application of efficient sparse matrix computational methods is shown to be suitable for both dynamic and design sensitivity analyses and for interactive optimization. For dynamic analysis of planar systems each element of the constrained system is treated with three degrees of freedom. Algebraic equations prescribing constraints between various bodies are then written and a Lagrangian formulation is used to write the dynamical equations of motion for each body of the system. A stiff predictor-corrector numerical integration (GEAR) algorithm is used for numerical integration of mixed systems of nonlinear differential equations of motion and algebraic equations of constraint (together with spring-damper relations and other user-supplied equations). At each time step a corrector equation together with a very sparse Jacobian matrix is encountered and corrector convergence is obtained through Newton interaction and sparse matrix techniques. Results that are to be used in design sensitivity analysis are stored in a direct access disk.

A similar procedure is adopted for solution of the mixed system of linear differential and algebraic equations for adjoint variables. Since the time grid of the transient analysis need not coincide with that for adjoint analysis, interpolation of the solution variables is used to calculate the right-hand sides of the adjoint corrector equation. The Jacobian matrix for the adjoint corrector equation is the transpose of the Jacobian for transient analysis; so its elements are not recalculated but are read from the direct access disk. Solutions of the adjoint equations are then used to obtain design sensitivity coefficient matrices. It is noted that the extension of this technique to three-dimensional systems is straightforward, at least theoretically.

A computer code named DADS (Dynamic Analysis and Design System) that implements the method for planar systems is organized and described. Two numerical examples are treated with this program. The first example is a classified slider-crank mechanism and the second is a model of the 2500 semi-integral spring-reset trip-plow that is produced by John Deere. While the former undergoes continuous motion, the latter undergoes intermittent motion and is of a more complex nature.

ABSTRACT

In this report, the technical objective is the derivation of a systematic and unified theory and organization of a corresponding general computer program for the design of constrained dynamic systems by judicious selection of the most suitable methods from the following branches of mathematics and mechanics:

(a) Optimization Methods

(b) Rigid Body Mechanics

(c) Numerical Integration Methods

(d) Matrix Manipulation Methods.

Accordingly, a method of formulating and automatically integrating the equations of motion and design sensitivity adjoint equations for general constrained dynamic systems is presented. Design sensitivity analysis is carried out using a state-space method that has previously been used for design optimization of linear structural systems. Application of efficient sparse matrix computational methods is shown to be suitable for both dynamic and design sensitivity analyses and for iterative optimization. For dynamic analysis of planar systems each element of the constrained system is treated with three degrees of freedom. Algebraic equations prescribing constraints between various bodies are then written and a Lagrangian formulation is used to write the dynamical equations of motion for each body of the system. A stiff predictor-corrector numerical integration (GEAR) algorithm is used for numerical

integration of mixed systems of nonlinear differential equations of motion and algebraic equations of constraint (together with spring-damper relations and other user-supplied equations). At each time step a corrector equation together with a very sparse Jacobian matrix is encountered and corrector convergence is obtained through Newton iteration and sparse matrix techniques. Results that are to be used in design sensitivity analysis are stored in a direct access disk.

A similar procedure is adopted for solution of the mixed system of linear differential and algebraic equations for adjoint variables. Since the time grid of the transient analysis need not coincide with that for adjoint analysis, interpolation of the solution variables is used to calculate the right-hand sides of the adjoint corrector equation. The Jacobian matrix for the adjoint corrector equation is the transpose of the Jacobian for transient analysis; so its elements are not recalculated but are read from the direct access disk. Solutions of the adjoint equations are then used to obtain design sensitivity coefficient matrices. It is noted that the extension of this technique to three-dimensional systems is straightforward, at least theoretically.

A computer code named DADS (Dynamic Analysis and Design System) that implements the method for planar systems is organized and described. Two numerical examples are treated with this program. The first example is a classical slider-crank mechanism and the second is a model of the 2500 semi-integral spring-reset trip-plow that is produced by John Deere. While the former undergoes continuous motion, the latter undergoes intermittent motion and is of a more complex nature.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\underset{\sim}{A}$	general square matrix as in Eq. (2.54)
A	set of all constraint indices that are violated or ε-active
$A(I,J), a_{ij}$	elements of matrix $\underset{\sim}{A}$
$\underline{B}, \underline{y}$	vectors as in Eqs. (2.54) and (2.60)
$\underline{b}, b$	design parameter vector
b^I, b^L, b^U	initial estimate, lower and upper bounds for the design parameter vector
$\delta b^1, \delta b^2$	variation of b defined by Eqs. (4.33) and (4.34)
δb	small change of b defined by Eq. (4.35)
C^i_{jk}	damping constants for the i-th spring-damper pair connecting j-th and k-th bodies
C^i	damping constants with j,k suppressed
C_{ij}	damping constant for the spring-damper pair connecting i-th and jth bodies (the pair number is suppressed)
c_{zi}	coefficients of stiffly stable methods in canonical form
E	kinetic energy of a mechanical system
F	general vector function denoting a system of differential-algebraic equations as in Eq. (1.1)
F^i_0	extra constant force along the i-th spring-damper pair
$F', \Phi', \bar{\xi}'$	left-hand side functions in adjoint equations
f	vector of functions of several varialbes
G	vector of nonzero elements of the Jacobian matrix; the shear modulus
g_0, g_β	non-integral parts of the cost and constraint functionals

H	functions in design parameter constraints
$h, \hat{h}$	time-step for numerical integration
I	row index of a matrix
$\bar{I}$	a matrix as defined in Eq. (2.38)
$\underset{\sim}{I}$	identity matrix
i	suffix, prefix or superscript
J	column index of a matrix
$\bar{J}$	real cost functional
J_1	moment of inertia of the first body; Jacobian matrix of static analysis
J_i	moments of inertia
$\underset{\sim}{J}$	Jacobian matrix
j	suffix, prefix or superscript
K	step number in permutation and $\underset{\sim}{L}$ $\underset{\sim}{U}$ factorization process
K^i_{jk}	spring constants for the i-th spring-damper pair connecting j-th and k-th bodies
K^i	spring constants with j, k supressed
K_{jk}	spring constant for the spring-damper pair connecting j-th and k-th bodies (with the number of pair suppressed)
K^{mx}_{jk}	maximum value of K_{jk}
k	order of Gear algorithm; suffix, prefix or superscript
L	general symbol for a linear system
L_0, L_β	integrands of the integral parts of the cost and constraint functionals
$\underset{\sim}{L}$	lower triangular matrix
$\ell^i, v^i, F^i_X, F^i_Y$	spring-damper related variables (for i-th pair)

$\underset{\sim}{\mathcal{L}}^i$	spring-damper variable vector for the i-th pair
$\underset{\sim}{\mathcal{L}}^i_{jk}, \ell^i_{jk}, v^i_{jk}$ F^i_{Xjk}, F^i_{Yjk}	spring damper vectors and variables for the i-th pair connecting j-th and k-th bodies
$\underset{\sim}{\ell}^i$	vector of spring-damper related variables for the i-th body
ℓ^i_0	undeformed length of the i^{th} spring-damper pair
ℓ_{ij}	elements of lower triangular matrix; length of spring damper pair connecting i-th and j-th bodies
$\underset{\sim}{\ell}, \ell$	global spring-damper variable vector
$\ell^{\psi_0}, \ell^{\psi_\beta}$	sensitivity coefficient vectors
$\bar{\ell}$	truncated vector formed from $\underset{\sim}{\ell}$
M	the number of corrector iterations
$M_{\psi\psi}$	matrix defined by Eq. (4.13)
M_i	mass of the i-th body
m	number of joints
N	general symbol for a nonlinear system
$\bar{N}$	number of operations required in $\underset{\sim}{L}\ \underset{\sim}{U}$ factorization
n	dimension of a square matrix; number of bodies
$\underset{\sim}{P}(b)$	matrix function of design parameters as defined in Eq. (2.34)
$P_{ij}, P_{ij}(b)$	components of $\underset{\sim}{P}(b)$
P'_{ij}, P'_{ji}	points as defined in Fig. 2.4
$\underset{\sim}{Q}$	a matrix as defined in Eq. (2.63)
$Q_j, Q_{X_i}, Q_{Y_i}, Q_{\phi_i}$	generalized forces
$\underset{\sim}{q}^i$	vector of generalized coordinates for a rigid body (two-dimensional)
q_j	generalized coordinates

q_{ij}	elements of matrix $\underset{\sim}{Q}$
$\bar{R}_i, \bar{R}_{sij}$	position vectors
R^n	n-dimensional Euclidean space
r	suffix, prefix or superscript
$\bar{r}_p, \bar{r}_{ij}, \bar{r}_{sij}$	position vectors
S	suffix, prefix or superscript
s	number of spring-damper pairs
T	superscript to denote transpose of a matrix
$\underset{\sim}{T}$	transformation matrix
t	time parameter
t'	reverse time parameter for the adjoint case
U^i_j, V^i_j	coordinates of the point of attachment of the i--th spring-damper pair on j-th body (w.r.t. fixed reference frame)
$\underset{\sim}{U}$	upper triangular matrix
u_{ij}	elements of upper triangular matrix
$\underline{u}^i$	generalized velocity vector for a rigid body (two-dimensional)
u_j	first component of two-dimensional generalized velocity vector $\underline{u}^j$
v	velocity of spring-damper pair
W(I,J)	a weighting matrix used in optimal ordering of a matrix
W	weighting matrix appearing in Eq. (4.6)
X,Y,Z	coordinates of CM's of rigid bodies
X_i, Y_i, ϕ_i	generalized coordinates for a rigid body (two-dimensional)
x_i, y_i	coordinates of a point on a body w.r.t. body-fixed axes

$\underline{x}$	solution variable vector as in Eq. (2.54)
y	dependent variable vector
$\Delta y^{(m)}$	Newton difference of y at the m-th Newton iteration step at a certain time step
y_n	value of y at the n-th time step
$y_n^{\langle k \rangle}$	k-th derivative of y at the n-th time step
$\underline{z}, z$	global vector of generalized coordinates and velocities
$\underline{z}^i$	vector of two-dimensional generalized coordinates and velocities for the i-th body
$\underset{\sim}{z}_n$	Nordsieck vector (Eq. (2.46))
$\lambda(t)$	adjoint variable vector
$\bar{\lambda}, \bar{\bar{\lambda}}, \lambda'$	component vectors of $\lambda(t)$
Δ	symbol indicating Newton differences in integration algorithm and desired changes in optimization algorithm
δ	variation operator
δ_i	angle as indicated in Figure 2.4
γ	Lagrange multiplier vector in optimal design algorithm
γ^1, γ^2	parts of γ defined by Eqs. (4.15) and (4.16)
$\underline{\mu}, \mu$	Lagrand multiplier vector for dynamic analysis
α	a number less than unity (Section 2.2.1)
α_j, β_0	Gear coefficients
$\Phi_i, \Phi_{X_i}, \Phi_{Y_i}, \Phi_{\phi_i}, \Phi_n$	constraint functions in dynamic analysis
$\underline{\Phi}, \Phi$	vectors of constraint functions in dynamic analysis

ψ_0, ψ_β	cost and constraint functionals
Θ	functions in initial conditions
$\varepsilon_{\ell^i}, \varepsilon_{v^i}, \varepsilon_{F_X^i}, \varepsilon_{F_Y^i}$	functions defining spring-damper variables
ζ	six times the number of bodies
$\bar{\xi}, \xi$	functions as defined in Eqs. (2.36) and (2.37)
ρ	an integer
$\nu(b), \bar{\nu}(b)$	initial solution variables as functions of b
χ_γ	functions in design parameters constraints
η	a small number appearing in Eq. (4.6)
ε	small constant; belong to
ε_T	local truncation error
$\|\|\cdot\|\|$	L_2-Norm
$\|\cdot\|$	absolute value
$(\dot{\cdot})$	time derivative
$[\cdot,\cdot]$	closed interval
$<\cdot>, <<\cdot>>$	symbols for definition of special functions
$[\cdot]$	denotes reference numbers
*	notation for footnote; computer symbol for multiplication
DSIN,DCOS	sine and cosine functions with double precision
$\triangleq, \equiv$	definition and equivalence symbols respectively
$b^{(j)}, z^{(j)}, \mu^{(j)}, \ell^{(j)}$	values of the vectors b, z, μ, ℓ at the j-th iteration of optimization

CHAPTER I

INTRODUCTION

1.1 Motivation, Scope and Organization

1.1.1 Main Objective

The main objective of this research is to develop and demonstrate a systematic and unified theory and computational method for the design of large scale constrained dynamic mechanisms and machines. The key to meeting this objective lies in judicious selection of the most suitable methods from the following branches of mathematics and mechanics:

(a) Methods of Optimization

(b) Rigid Body Mechanics

(c) Numerical Integration Methods

(d) Matrix Manipulation Methods.

The objective can be illustrated by the following Venn diagram:

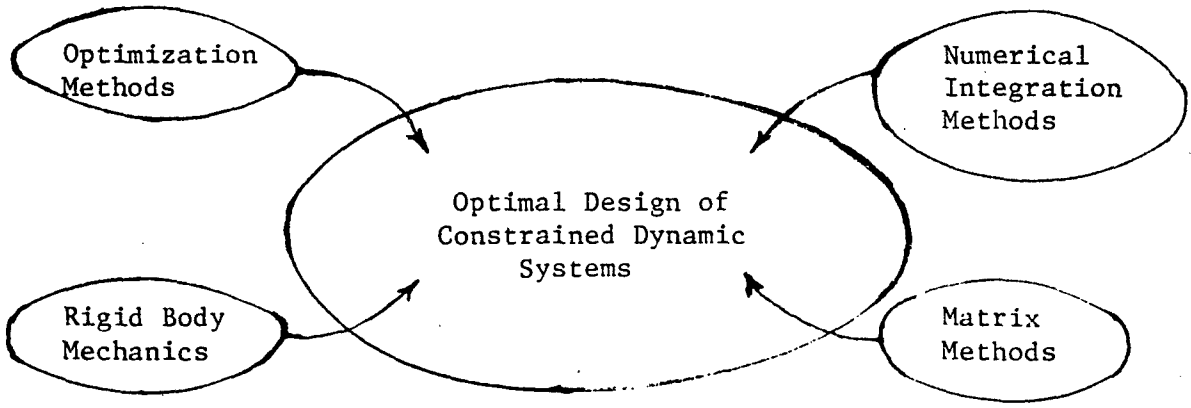


Figure 1.1 Venn Diagram for the Main Objective of the Research.

The arrows on the diagram may be viewed as the process of selecting from among a large number of alternatives in each area a method that is compatible with those of the other areas, together providing a qualitatively new design capability. Such a comprehensive treatment has never been attempted, so utmost possible care has been taken in the theoretical investigation to implement the most suitable computer algorithms associated with various branches of mechanics and mathematics noted above. Brief discussions of the above branches, together with the indication of the methods selected, are given below.

1.1.2 The Notion of Optimal Engineering Design

The job of "optimal engineering design" is to develop the best possible system for the given application, consistent with resources allocated to the development phase. Although the notion of optimization is inherent in the design process, optimization as a formalized approach to engineering design is a relatively new concept. It is in the judicious selection of a "quantitative measure of performance" of the system and in quantifying performance constraints that an optimal design is distinguished from a conventional design.

After quantification of the notions of a design process comes the role of selection of numerical methods for its solution, using the modern high-speed digital computer. Great strides have been made with digital computers in the past two decades, to allow for numerical analysis as a test of an idea or concept, rather than previous cut-and-try techniques. In this report, advantage is taken of advances in computer-aided design techniques [1].

The field of optimal design of constrained dynamic systems is of growing interest and importance. In the realm of dynamic mechanisms, both continuous (smooth) and intermittent (discontinuous) motion occurs. For many mechanisms, the logical sequence of events in intermittent motion is known in advance and the total period of motion under consideration can be divided into intervals of continuous motion. Also, some intermittent motion can be reduced to continuous motion by the introduction of artificial spring-damper systems (see Chapter VI). Thus, derivation of a unified technique for optimal design of dynamic systems with continuous motion is of major importance.

1.1.3 Methods of Optimization

The fundamental problem of infinite dimensional optimal design can be described by the problem of Bolza and its extended version that accounts for inequality functional constraints (see [1]). A corresponding problem for constrained dynamic systems with continuous motion is formulated in Chapter III. There are many indirect methods [1] based on a powerful Functional Analysis theorem of Liusternik and Sobolev [69] for the solution of such problems. But generally they pose serious computational hazards. In this dissertation a direct numerical method of solution is adopted.

The basic idea of the direct method of solving optimal design problems is to first construct an initial estimate of the solution and then to find small changes in the design parameters such that the modified design forms an improved estimate of the optimum, in some sense. Before design improvements can be determined, analysis of their effect

on the problem must be performed. This forms the sensitivity analysis part of the design process and is of extreme importance.

In the approach used here, design sensitivity analysis uses state-space methods (see [24]) in which the state variables and design parameters are first treated as independent variables and the elimination of the variations of the state variables is performed through use of adjoint differential and algebraic equations (see Chapter III and references [1,70]).

1.1.4 Rigid Body Mechanics

There are two general approaches to the subject of rigid body mechanics. They can be called the "Vectorial or Newtonian approach" and the "Analytical approach". Vectorial dynamics is based on a direct application of Newton's laws of motion and concentrates on the forces and motion associated with individual parts of the system, whereas analytical dynamics is concerned with the system as a whole and uses descriptive scalar functions such as kinetic and potential energies. The most direct analytical approach is the well-known Lagrangian formulation. For details, references [66,67,68] are recommended.

In most treatments of optimal design of dynamic systems in the literature, the number of generalized coordinates of a system is taken equal to the number of degrees of freedom of the whole system, so that with the application of Newton's laws, the number of first order ordinary differential equations for the state is equal to twice the number of physical degrees of freedom. In general, a dynamic mechanism may be very complicated and direct application of Newton's laws may result

in a highly nonlinear set of differential equations. Moreover, the eigenvalues for such a nonlinear system may vary over extreme ranges during the process of numerical integration. Therefore, there remains a danger of the numerical integration problem turning "stiff" [2,3,4, 5,6].

Orlandea [7] and Orlandea, Chace, and Calahan [8] have developed a node-analogous sparsity-oriented approach to the dynamic analysis of mechanical systems using the Lagrangian formulation of rigid body mechanics. By applying SPARSE MATRIX [9,64,65] and STIFF [10] integration algorithms, large sets of sparse linear equations can be efficiently solved with moderate expenditure of CPU time and the numerical instability [2,3,6,11] associated with widely split eigenvalues at any stage can be avoided. Both sparse matrix techniques and stiff integration algorithms are in the constant process of modification and refinement. These topics are treated briefly in Chapter II of this report (also see Sections 1.1.5 and 1.1.6).

These algorithms have been implemented by Orlandea [7] to generate a computer program "ADAMS" (Automatic Dynamic Analysis of Mechanical Systems). This program was developed for efficient simulation of dynamic mechanical systems (e.g., vehicles and machinery) using methods of numerical analysis developed for electrical circuits [10,12,13]. Orlandea's work has demonstrated that the sparse matrix formulation and numerical methods involving stiff numerical integration techniques can be effectively used for simulation of large three-dimensional mechanical systems. The advantages and disadvantages of these can be stated as follows:

Advantages:

1) No topological preprocessing is necessary to establish a set of independent variables; equations can be developed directly from the connection data, component by component.
2) High sparsity of the "Jacobian Matrix" is used in a stiff integration algorithm (see Chapter II).
3) All angular and displacement variables are retained as solution variables; none are eliminated in the interest of producing a reduced set of equations with fewer variables. In this way the total number of matrix operations and the number of operations for eliminating variables from nonlinear equations are kept at a minimum.
4) All joint reaction forces are determined directly in the solution and therefore the formulation is compatible with current methods of continuum mechanics for internal stress-analysis.
5) Frictional effects in joints are routinely handled.

Disadvantages:

1) Nodal formulation results in more equations than loop formulation.
2) Some time may be wasted in solving for variables of little interest to the designer.

ADAMS, IMP (Integrated Mechanism Program) [15,16], and DRAM (Dynamic Response of Articulated Machineries) [17,18] are the three main computer programs at present for dynamic simulation of mechanical systems. Both DRAM and IMP use relative coordinate systems for parts (or bodies or nodes) of a mechanism and consider independent loop

equations for the determination of constraint equations. These are complicated mathematical relations involving numerous matrix inversions. ADAMS, on the other hand, deals with a global inertial reference frame for all bodies in the system. It will be evident from the formulation and analytical procedure in Chapters II and III that the ADAMS program is extremely suitable for optimal design investigations. All the above mentioned programs, however, use Lagrange's equations of motion in the analysis.

There are other dynamic analysis programs, namely, IMP-UM (IMP at the University of Michigan), MEDUSA (Machine Dynamic Universal System Analyzer) [19], VECNET (Vector Network) [20], and DYMAC (Dynamics of Machinery) [21]. They deal with different methods of rigid body dynamics. A more detailed overview of all the programs can be obtained in reference [22].

It should be noted that, theoretically, any of the dynamic analysis programs discussed in the foregoing can be used to predict system dynamics. An apparent advantage inherent in all the programs except ADAMS lies in the fact that they deal with a minimum number of independent (generalized) coordinates and thus involve fewer equations than does the ADAMS method. The disadvantages of these methods are much more serious, however. The reduced set of equations with fewer variables is highly nonlinear. Consequently, the eigenvalues for such nonlinear systems may vary quite unpredictably during the integration process. Another major disadvantage lies in the fact that they do not evaluate the reaction forces simultaneously with the solution variables.

In a reasonable optimal design formulation, one generally places bounds on reaction forces at joints (see Chapters III and VI). All these problem areas are readily handled by the ADAMS modeling method, which makes the ADAMS method extremely suitable for optimal design investigations.

Wehage [14] has written a program for planar systems using constrained system formulation and sparse matrix methods. This program, after several modifications, has been extended and implemented as the analysis module of the computer program "DADS" (Dynamic Analysis and Design System) in this dissertation (see Chapters II, V).

1.1.5 Numerical Integration Methods

The solution of linear dynamic state equations can always be expressed in analytic forms. But when the equations are nonlinear, the analytic forms of solutions seldom exist and one is compelled to resort to graphical [42] or numerical methods. The most serious shortcoming of graphical methods lies in their inapplicability for higher order, nonlinear mechanical systems. On the other hand, numerical methods are valid and appropriate for nonlinear systems of any order. In view of their greater generality and ease of implementation on a digital computer, numerical methods are widely applied. In the following, most discussions will be confined to nonlinear systems. For detailed analytic theory of numerical methods, references [2,3,4,6,11, 43,44,45] are recommended.

In the Lagrangian formulation of the equations of motion of a constrained mechanical system, when all the generalized coordinates

and Lagrange multipliers are taken to be independent [66,67], nonlinear algebraic equations enter into the system of equations (see Section 2.1). As put forward by Gear [46], such a system can be written as:

$$\underline{F}(\dot{\underline{y}},\underline{y},t) = \underline{0} \tag{1.1}$$

where $\underline{y}$ is the solution variable vector, t is the time parameter, and $\dot{\underline{y}}$ is the vector of time derivatives. A member of Eq. (1.1) may be either a differential or an algebraic equation, according a $\partial\underline{F}/\partial\dot{\underline{y}}$ is nonzero or zero. Such a system of equations is called a simultaneous system of Differential and Algebraic Equations (DAE's). To the system of equations (1.1) one must add the appropriate initial conditions of the form

$$\bar{\underline{y}}(0) = \bar{\underline{y}}_o \tag{1.2}$$

where $\bar{\underline{y}}$ is a vector of the subset of solution variables having time derivatives in the system of Eq. (1.1).

Although there exist many algorithms for the solution of initial value problems, most of them are based on two basic approaches: (1) the Taylor series expansion approach and (2) polynomial approximation approach. Algorithms based on the first approach are generally called Runge-Kutta algorithms and those based on the second one are usually called numerical integration algorithms (see references [2,6]). The former are single-step algorithms, whereas the latter are multistep algorithms that use information from previous time steps. Adams-Bashforth and Adams-Moulton algorithms are examples of the second category [2,6].

In most of the treatments found in the literature, these algorithms deal with the system of equations of the form

$$\dot{\underline{y}} = \underline{f}(\underline{y},t)$$
$$\underline{y}(0) = \underline{y}_0 \qquad (1.3)$$

which do not include algebraic equations. Gear [2,46,48] appears to be the first person to present a multistep algorithm that deals with a simultaneous system of differential and algebraic equations, together with the difficult concepts of stability, convergence and automatic change of order and step-size (see references [2,6]). Hachtel, et al. [49] have used this algorithm for electrical network analysis and design. Calahan and Orlandea [7,8,9] have modified the Gear algorithm and used it in the ADAMS computer program for the solution of the dynamic system equations. The Gear algorithm has been used in this dissertation for the solution of linear and/or nonlinear sets of differential and algebraic equations (see Chapter II, III and V).

1.1.6 Matrix Methods

In the process of numerical integration, solution of simultaneous linear equations is inevitable at every time step. Theoretically, any method of solution by the $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization procedure (see [6,11,44,55]) can be adopted.

By the sparse matrix formulation of the dynamical equations, difficulties with nonlinearities in the differential equations can be avoided and so by the application of Gear algorithms and sparse matrix

techniques, large dynamic systems can be simulated very effectively. Thus sparse matrix approaches have been adopted in this dissertation.

Section 2.3 of this dissertation deals briefly with sparse matrix techniques. Some modifications necessary for the adjoint analysis (see Chapter III) are noted in that section.

1.1.7 Thesis Organization

In the remaining part of this chapter (i.e., in Section 1.2), a brief literature survey on optimization of dynamic systems is made.

In Chapter II a sparse matrix formulation of the equations of dynamical systems and the topics of stiff integration and sparse matrix techniques are discussed. The formulation of the general optimal design problem and the corresponding design sensitivity analysis and optimization algorithms are described in Chapters III and IV.

In the sensitivity analysis, two types of state variables occur. In mechanical problems, the first type corresponds to variables like displacements and velocities and are called "Primary" state variables. The second type corresponds to Lagrange multipliers of the constrained motion and the spring-damper associated variables. These are termed "Secondary" state variables (see Chapter II). Chapter V deals with the organiztion and description of the computer program DADS. Numerical results of application of this program to slider-crank mechanism and spring reset plow share mechanism are presented in Chapter VI. Finally, some discussion, conclusions and recommendations are given in Chapter VII.

1.2 Literature Survey

A general survey of mechanical design optimization is presented by Seireg [23]. Haug and Arora [24] have given a description of state space techniques for solving optimal mechanical design problems. Also a fundamental treatment of the problem of optimal design of constrained dynamic systems can be found in AMCP 706-192 [1]. Sevin and Pilkey [25] have used the penalty function technique [26] to obtain min-max response of dynamic systems with incompletely prescribed input functions. Sevin, Pilkey, and Kalinowski [27] have formulated problems of optimal design of mechanical systems subjected to dynamic loads in mathematical programming terms. These can be found in reference [28]. They have studied three types of variational problems: (1) Extreme disturbance analysis with bounds on performance index for a given system when the inputs are described as a class of unspecified waveforms; (2) optimum system performance dealing with bounding a performance index for a class of inputs when certain system elements are unspecified and constraints are imposed on the system response; and (3) optimum system design concerning identification of parameters that uniquely specify the system, so that a performance index is minimized for the worst disturbance among the class of admissible inputs. For these problems, solution techniques are based on linear, nonlinear, and dynamic programming [29].

Brock [30], Den Hartog [31], Hamad [32], Arora, Rim, and Kwak [33], Afimiwala and Mayne [34], Willmart and Fox [35], McMunn [36], and Kwak [37] have considered various aspects of optimization in vibration

absorbers and vehicular models. Hsiao [38] has considered similar problems with "P-Norm approximation" and/or "Equivalent Functional Treatment" of cost functional and performance constraints. Haug and Arora [39] have made further modifications of Hsiao's treatments. However, all these problems involve either a small number of degrees of freedom or they are relatively easy to formulate by Newton's laws (owing to restricted dynamic motions).

In the field of intermittent motion of dynamical systems, very little work has been done so far. The general problem of optimization of mechanical systems with this type of motion has been treated in reference 1. Huang [40] and Huang, Haug, and Andrews (41] have developed a state-space method of optimal design of mechanical systems with intermittent motion, which has been applied to a cammed, three mass system. None of these methods has gone into the consideration of the problem of instabilities associated with a highly nonlinear set of differential equations and with widely split eigenvalues. Thus none has taken advantage of the modern methods of stiff integration (Gear) and sparse matrix algorithms.

CHAPTER II

SPARSE MATRIX FORMULATION OF EQUATIONS OF MOTION, THE STIFF INTEGRATION (GEAR) ALGORITHM, AND SPARSE MATRIX TECHNIQUES

2.1 Sparse Matrix Formulation of the Mechanical Systems Equations*

2.1.1 Introduction

The principal objective of this section is to present a sparsity-oriented formulation of dynamical equations using the idea of writing the equations of motion for each element (or component) of a mechanism (machine) separately. This idea originated in reference [7] and is discussed further in reference [8]. In this approach, the Lagrangian formulation of equations of motion is adopted and constraint equations for various joints and spring-damper relations are written separately. No attempt is made to reduce the number of solution variables through the process of elimination. Thereby, the order of nonlinearity in the equations is kept at a minimum. The equations are then solved numerically using a stiff integration algorithm [2,6] and sparse matrix [9,64,65] techniques. These methods are discussed briefly in Sections 2.2 and 2.3.

The results presented here are confined to two-dimensional mechanical systems. Thus, the sparse matrix formulation of dynamical

---

*Companion reading of references [7,14,63] is suggested for the reading of this section.

equations for such systems is presented here. A similar formulation for three dimensional systems is available in reference [7].

2.1.2 Two Dimensional Systems

In the Lagrangian formulation [66,67,68] the general form of the equations of motion can be written (for constrained or unconstrained systems) as:

$$\frac{d}{dt}\left(\frac{\partial E}{\partial \dot{q}_j}\right) - \frac{\partial E}{\partial q_j} - Q_j = 0 \qquad (2.1)$$

where

E = Kinetic energy of the system,

q_j = generalized coordinates,

$\dot{q}_j$ = generalized velocities,

Q_j = generalized forces (conservative or non-conservative including reactions and spring-damper forces),

$j = 1,2,\ldots,k$, (where k represents the number of degrees of freedom),

t = time.

The system of Eq. (2.1) can be equivalently written as

$$\frac{d}{dt}\left(\frac{\partial E}{\partial u_j}\right) - \frac{\partial E}{\partial q_j} - Q_j = 0 \quad , \qquad j = 1,\ldots,k \qquad (2.2)$$

$$u_j - \dot{q}_j = 0 \quad , \qquad j = 1,\ldots,k \qquad (2.3)$$

For constrained systems, the Lagrange multiplier method [66,67] is adopted and Eqs. (2.2) and (2.3) can be replaced by

$$F_{1j} \equiv \frac{d}{dt}\left(\frac{\partial E}{\partial u_j}\right) - \frac{\partial E}{\partial q_j} - Q_j + \sum_{i=1}^{p} \frac{\partial \Phi_i}{\partial q_j} \mu_i = 0 \qquad (2.4a)$$

$$F_{2j} \equiv u_j - \dot{q}_j = 0 \quad , \qquad j = 1,2,\ldots,k \qquad (2.4b)$$

and

$$\Phi_i = 0 \quad , \qquad i = 1,2,\ldots,p \qquad (2.4c)$$

where Φ_i are geometrical constraint functions, Q_j are generalized forces (excluding constraint reactions), and μ_i are Lagrange multipliers.

2.1.2.1 Choice of Coordinates

In two space dimensions, let OX,OY represent a set of coordinates fixed in an inertial reference frame and $O_i x_i$, $O_i y_i$ represent a set of body-fixed rectangular coordinate axes fixed at O_i in the i-th body of the system. Let X_i, Y_i, ϕ_i be the translational and angular generalized coordinates and $u_i \equiv \dot{X}_i$, $v_i \equiv \dot{Y}_i$, and $w_i \equiv \dot{\phi}_i$ the corresponding generalized velocities for the i-th body. These coordinate systems are illustrated in Figure 2.1. Let

$$\left.\begin{aligned} \underline{q}^i &\equiv [X_i, Y_i, \phi_i]^T \\ \text{and} \\ \underline{u}^i &\equiv [u_i, v_i, w_i]^T \end{aligned}\right\} \qquad (2.5)$$

Then, one can write the vector $\underline{Z}^i(t)$ of generalized coordinates and velocities as

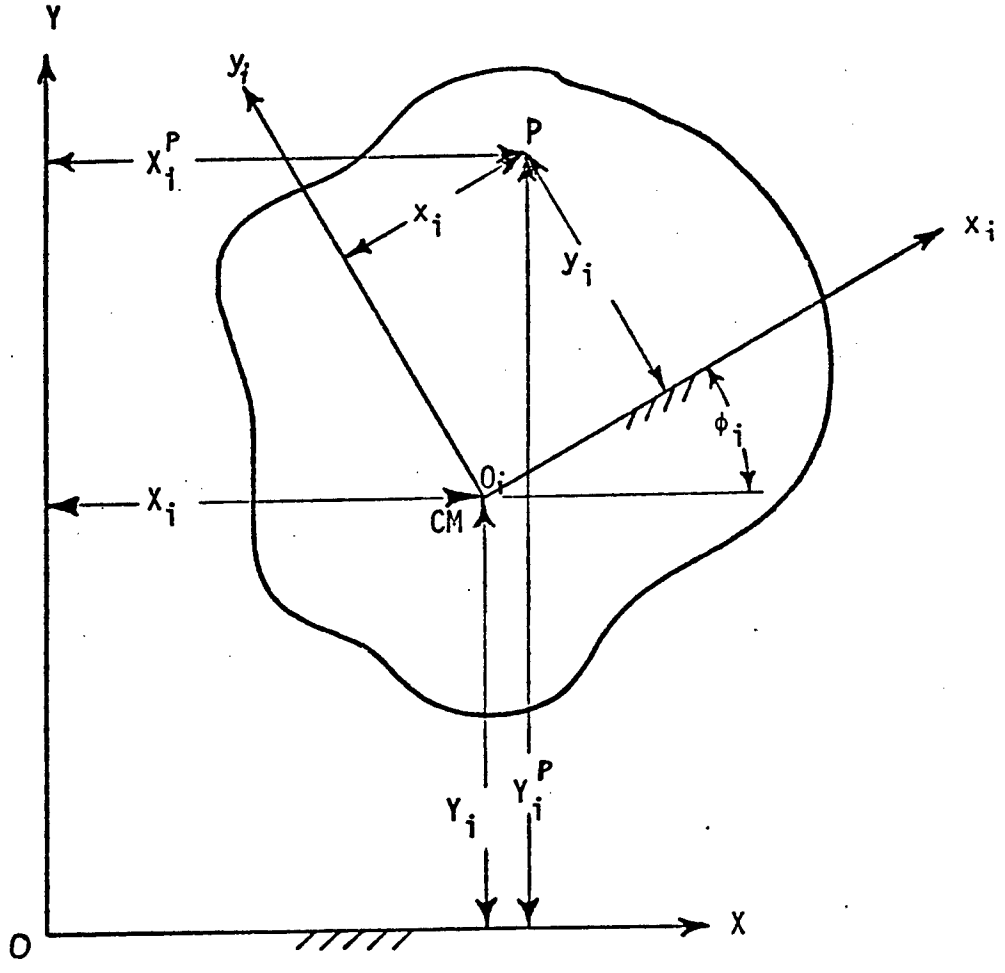


Figure 2.1 Definition of the Generalized Coordinates for the i-th Body.

$$\underline{z}^i \equiv [\underline{u}^{iT}, \underline{q}^{iT}]^T \equiv [u_i, v_i, w_i, X_i, Y_i, \phi_i]^T \tag{2.6}$$

2.1.2.2 Spring-damper Parameters and Variables and Related Equations

To the system of equations (2.4), one must add spring-damper equations. For the k-th spring-damper pair in two-dimensional systems, four variables are defined; ℓ^k is the spring-damper length, v^k is the velocity associated with damping and F_X^k and F_Y^k are the spring-damper force components. One can define a vector $\underline{\mathcal{L}}^k(t)$ with these variables as its components, i.e.

$$\underline{\mathcal{L}}^k(t) \equiv [\ell^k, v^k, F_X^k, F_Y^k]^T \tag{2.7}$$

A companion vector $\underline{\mathcal{L}}_{ij}^k(t)$ for the k-th spring-damper pair connecting i-th and j-th bodies can be defined in the following way:

$$\underline{\mathcal{L}}_{ij}^k(t) \equiv [\ell_{ij}^k, v_{ij}^k, F_{Xij}^k, F_{Yij}^k]^T \tag{2.8}$$

Figure 2.2 shows the spring-damper variables and parameters for a spring-damper pair. Superscripts identifying the number of spring-damper pair have been deleted for the sake of simplicity. The vectors $\bar{R}_i, \bar{R}_j, \bar{r}_{sij}, \bar{r}_{sji}$, and $\bar{R}_{sij}$ are position vectors and s_{ij} and s_{ji} are points of attachment on the i-th and j-th bodies, respectively. The angle α is measured between $\bar{R}_{sij}$ and the positive X-axis. The constants K_{ij}, C_{ij} are spring and damping coefficients, respectively. Although $K_{ij}^k, C_{ij}^k, \ell_{ij}^k, v_{ij}^k, F_{Xij}^k$, and F_{Yij}^k are complete notations for the spring-damper parameters and variables for the k-th pair

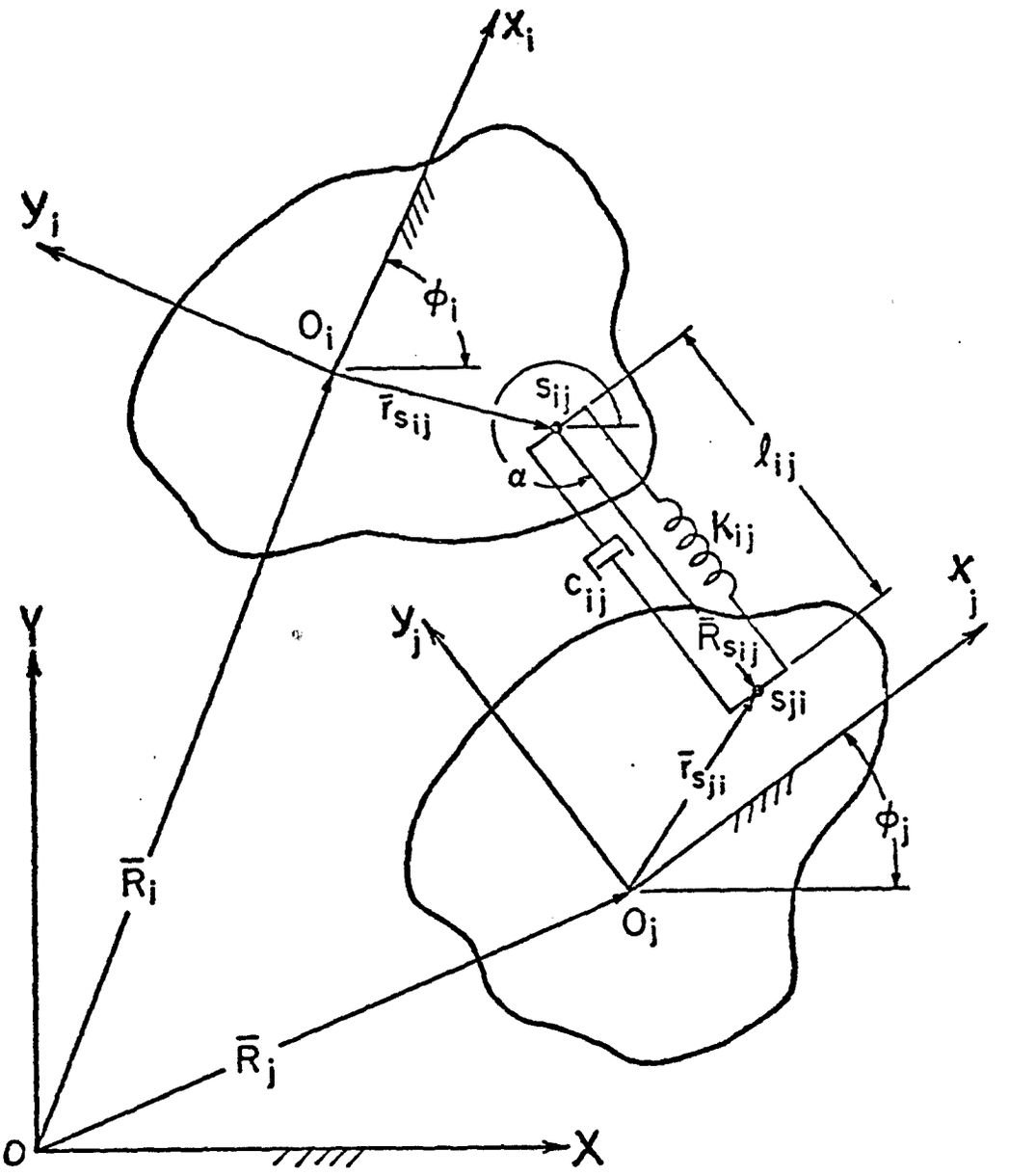


Figure 2.2 Variables and Parameters for a Spring-Damper Combination.

connecting the i-th and j-th bodies, superscripts, subscripts, or both may sometimes be suppressed in the ensuing discussions, for the sake of notational simplicity.

Explicit defining expressions for the spring-damper variables are given by (see [14])

$$\varepsilon_{\ell^k} \equiv \sqrt{(U_j^k - U_i^k)^2 + (V_j^k - V_i^k)^2} - \ell^k = 0 \tag{2.9}$$

$$\varepsilon_{v^k} \equiv \dot{\ell}^k - v^k = 0 \tag{2.10}$$

$$\varepsilon_{F_X^k} \equiv [K^k(\ell^k - \ell_0^k) + C^k v^k + F_0^k]\ [U_j^k - U_i^k]/\ell^k - F_X^k = 0 \tag{2.11}$$

$$\varepsilon_{F_Y^k} \equiv [K^k(\ell^k - \ell_0^k) + C^k v^k + F_0^k]\ [V_j^k - V_i^k]/\ell^k - F_Y^k = 0 \tag{2.12}$$

where (U_i^k, V_i^k) and (U_j^k, V_j^k) are the global coordinates of the points of attachment of the k-th spring-damper pair on bodies i and j, respectively; K^k and C^k are the spring-damper constants; and ℓ_0^k and F_0^k are the initial length and constant force along the spring-damper. The explicit expressions for U_i^k, V_i^k, U_j^k, and V_j^k are given by

$$U_i^k = X_i + x_{sij} \cos \phi_i - y_{sij} \sin \phi_i \tag{2.13}$$

$$V_i^k = Y_i + x_{sij} \sin \phi_i + y_{sij} \cos \phi_i \tag{2.14}$$

$$U_j^k = X_j + x_{sji} \cos \phi_j - y_{sji} \sin \phi_j \tag{2.15}$$

$$V_j^k = Y_j + x_{sji} \sin \phi_j + y_{sji} \cos \phi_j \qquad (2.16)$$

where (x_{sij}, y_{sij}) and (x_{sji}, y_{sji}) are the coordinates of the points s_{ij} and s_{ji}, with respect to their respective body-fixed coordinate axes.

The functional symbols $\varepsilon_{\ell^k}, \varepsilon_{V^k}, \varepsilon_{F_X^k}, \varepsilon_{F_Y^k}$ in Eqs. (2.9)-(2.12) are written to indicate the small values the expressions will take during Newton iterations in the dynamic analysis (see Section 2.2 and reference [6]).

2.1.2.3 Constraint Equations

In two dimensions there are two principal types of joints; revolute and translational. Figure 2.3 shows parameters and coordinates associated with a revolute joint. In the figure, $\bar{r}_p$ is the position vector of a point P_{ji} on body j with respect to a point P_{ij} on body i. When $\bar{r}_p = \bar{0}$, P_{ij} and P_{ji} coincide and they define a revolute joint. The loop closure relation of the position vectors gives.

$$\bar{R}_i + \bar{r}_{ij} + \bar{r}_p - \bar{r}_{ji} - \bar{R}_j = \bar{0} \qquad (2.17)$$

Let (x_i, y_i) and (x_j, y_j) be the coordinates of P_{ij} and P_{ji} respectively with respect to body-fixed axes. With $\bar{r}_p = \bar{0}$, the constraint equations for a revolute joint between bodies i and j are obtained from Eq. (2.17) as,

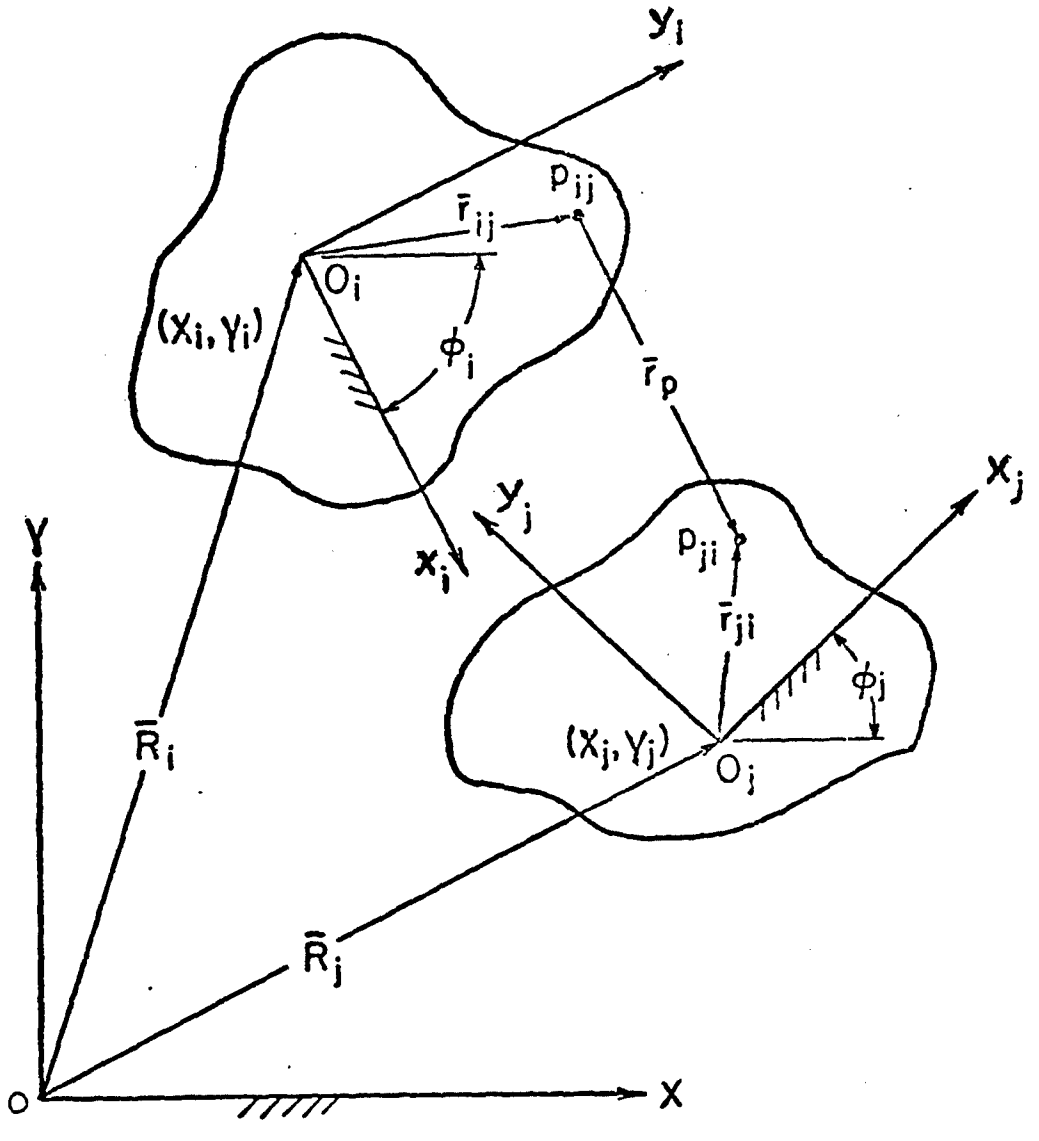


Figure 2.3 Joint Coordinates ($r_p = 0$ for revolute joint).

$$\Phi_X \equiv X_i + x_i \cos\phi_i - y_i \sin\phi_i - X_j - x_j \cos\phi_j + y_j \sin\phi_j = 0 \tag{2.18}$$

and

$$\Phi_Y \equiv Y_i + x_i \sin\phi_i + y_i \cos\phi_i - Y_j - x_j \sin\phi_j - y_j \cos\phi_j = 0 \tag{2.19}$$

Figure 2.4 shows parameters and coordinates associated with a translational joint. Here P'_{ij} and P'_{ji} are the points of intersection of perpendiculars drawn from the origins (centers of mass) 0_i and 0_j onto a straight line that is parallel to the line of relative motion between the bodies. The vectors $\bar{r}_{ij}, \bar{r}_{ji}, \bar{r}_p$ have the same meaning as before and δ_i and δ_j are angles between the vectors $\bar{r}_{ij}$ and $\bar{r}_{ji}$ with the body-fixed axes $0_i x_i$ and $0_j x_j$, respectively. The loop closure condition of the position vectors again gives, after elimination of $\bar{r}_p$, the following constraint equations for a translational joint between bodies i and j (see [14,63])

$$\Phi_n \equiv X_i \cos(\phi_i + \delta_i) + Y_i \sin(\phi_i + \delta_i) + \sqrt{x_i^2 + y_i^2}$$
$$- X_j \cos(\phi_j + \delta_j) - Y_j \sin(\phi_j + \delta_j) - \sqrt{x_j^2 + y_j^2} = 0 \tag{2.20}$$

$$\Phi_\phi \equiv \phi_i + \delta_i - \phi_j - \delta_j = 0 \tag{2.21}$$

In Eqs. (2.18) to (2.21), the coordinates (x_i, y_i) on body i and (x_j, y_j) on body j (hence the derived parameters δ_i and δ_j) depend on design parameters that define the geometry of the bodies.

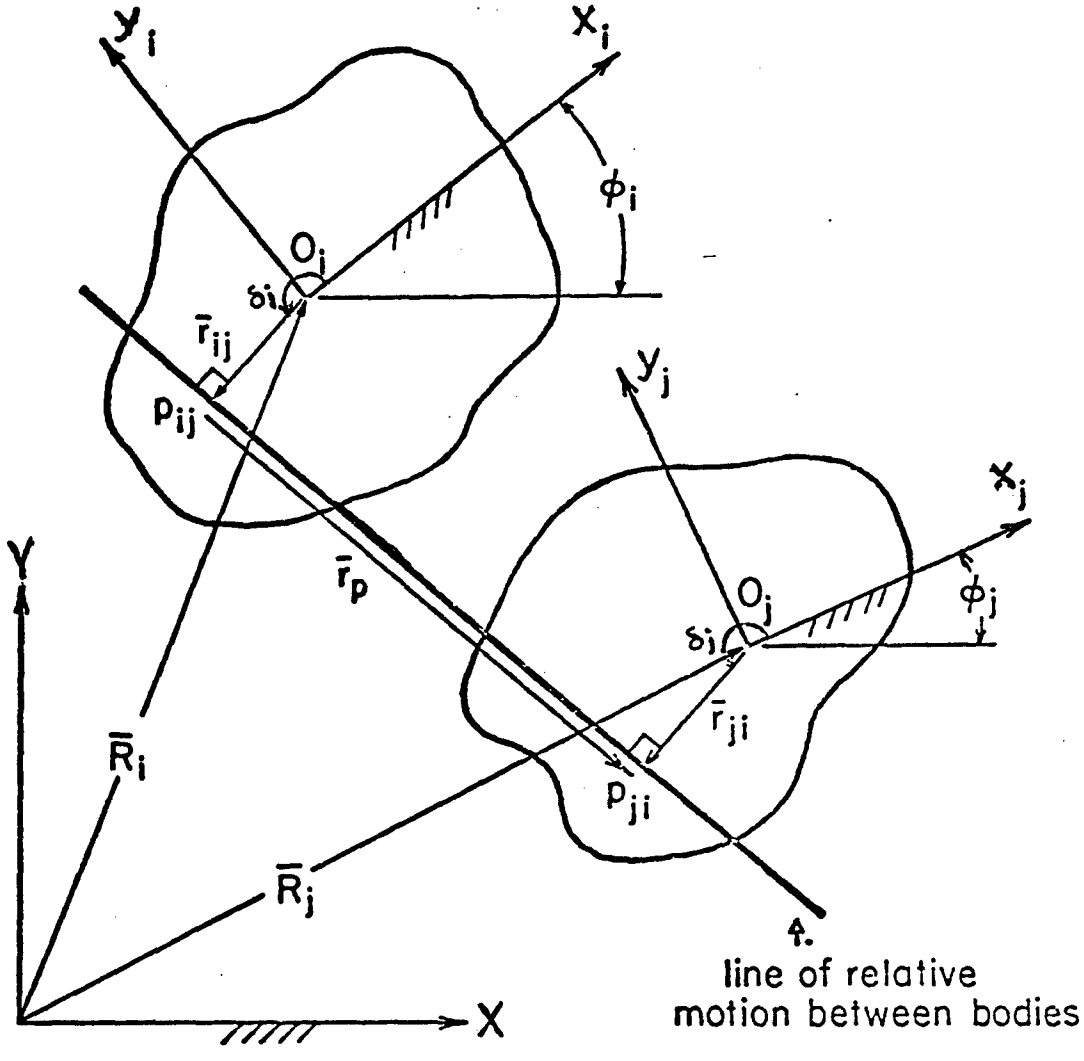


Figure 2.4 Translational Joint.

2.1.2.4 Primary and Secondary State Variables, Design Parameters

In the problem of optimal design of constrained dynamic systems, state variables and design parameters are encountered [1]. In the present formulation, two kinds of state variables are defined:

Primary State Variables: Variables whose time derivatives appear in the eqution of motion or in related equations are called "Primary State Variables".

For the planar systems treated here, $\underline{u}^{i^T}$, $\underline{q}^{i^T}$, and $\ell^{j_1}, \ell^{j_2}, \ldots, \ell^{j_k}$ (the last k terms being the lengths of the k spring-damper pairs connecting i-th body) are Primary State Variables related to the i-th body.*

Secondary State Variables: Variables that appear without their time derivatives (i.e., appear algebraically only) in the equations of motion and related equations are called "Secondary State Variables".

Let $\underline{\mu}^i \equiv \underline{\mu}^i(t)$ denote the Lagrange multiplier vector corresponding to the constraints on the i-th body. Then the variables $\underline{\mu}^{i^T}, v^{j_1}, v^{j_2}, \ldots, v^{j_k}, F_X^{j_1}, F_X^{j_2}, \ldots, F_X^{j_k}, F_Y^{j_1}, F_Y^{j_2}, \ldots F_Y^{j_k}$ are secondary state variables related to the i-th body.

The vector $\underline{b}$ denotes the design parameter vector for the entire system.

2.1.2.5 Element State Equations of Motion

In the Lagrangian formulation [66,67,68] of the equations of motion of a two dimensional constrained mechanical system, when all

*ℓ^k is kept with other spring-damper variables in $\underline{\chi}^k(t)$ for advantages in computer programming.

the generalized coordinates and Lagrange multipliers are taken to be independent, the state equations of motion for the i-th body of the system can be written as (cf. Eqs. (2.1) and (2.4)),

$$\underset{\sim}{P}^{i}(\underline{b})\dot{\underline{z}}^{i} + \underline{f}_{i} = 0 \tag{2.22}$$

where

$$\underset{\sim}{P}^{i}(\underline{b}) \equiv \begin{bmatrix} M_i & & & & \underset{\sim}{0} & \\ & M_i & & & & \\ & & J_i & & & \\ & & & -1 & & \\ & & & & -1 & \\ & \underset{\sim}{0} & & & & -1 \end{bmatrix} \tag{2.23}$$

M_i and J_i are the mass and moment of inertia of the i-th body,

$$\underline{f}_{i} \equiv -\left[\left(Q_{X_i} - \sum_{k=1}^{p_i} \frac{\partial \Phi_k}{\partial X_i}\mu_k\right), \left(Q_{Y_i} - \sum_{k=1}^{p_i} \frac{\partial \Phi_k}{\partial Y_i}\mu_k\right),\right.$$

$$\left.\left(Q_{\phi_i} - \sum_{k=1}^{p_i} \frac{\partial \Phi_k}{\partial \phi_i}\mu_k\right), -u_i, -v_i, -w_i\right]^{T} \tag{2.24}$$

μ_k, $k = 1,2,\ldots,p_i$, are the components of the Lagrange multiplier vector $\underline{\mu}^{i}(t)$, and Q_{X_i}, Q_{Y_i}, Q_{ϕ_i} are generalized forces (including the contributions from spring-dampers). Equation (2.23) indicates that M_i and J_i may be taken as design parameters. Note that the last three equations of Eq. (2.22) contain $\dot{\underline{q}}^{i}$ explicitly. This structure is introduced, intentionally, in order to increase sparsity (see Section 2.3).

2.1.2.6 Element State Equations for a Slider Crank Mechanism

To illustrate the procedure for writing the system equations of motion, a slider crank mechanism is considered in this subsection. The radial slider-crank mechanism is a complex of rigid bodies that move in a plane. Figure 2.5 shows the approximate initial position of such a mechanism. Link 1 is ground, link 2 is the crank shaft, link 3 is the connecting rod or coupler, and link 4 is the piston (or slider). A spring damper pair is attached between link 4 and ground (Figure 2.5). There is a translational joint (type 2) between bodies 4 and 1 and a revolute joint (type 1) between each of the following pairs of bodies: 1 and 2, 2 and 3, and 3 and 4. The figure also indicates the following design parameters:

b_1 = The spring constant $K^1 \equiv K_{41}$ of the spring

b_2 = Height of the points of attachment of the spring

b_3 = Half of the length of the uniform coupler.

Gravitational forces are excluded from the present simulation of the mechanism. To illustrate the use of the present technique, only the equations of motion and constraint involving body 4 are written.

For body 4, there will be 4 constraint equations corresponding to (revolute) joint 3 and (translational) joint 4. They can be written as (using notations of previous subsections),

$$\bar{\Phi}_1 \equiv \Phi_{X_4} \equiv X_3 + x_3 \cos \phi_3 - y_3 \sin \phi_3 - X_4 - x_4 \cos \phi_4 + y_4 \sin \phi_4 = 0$$

(2.25 cont.)

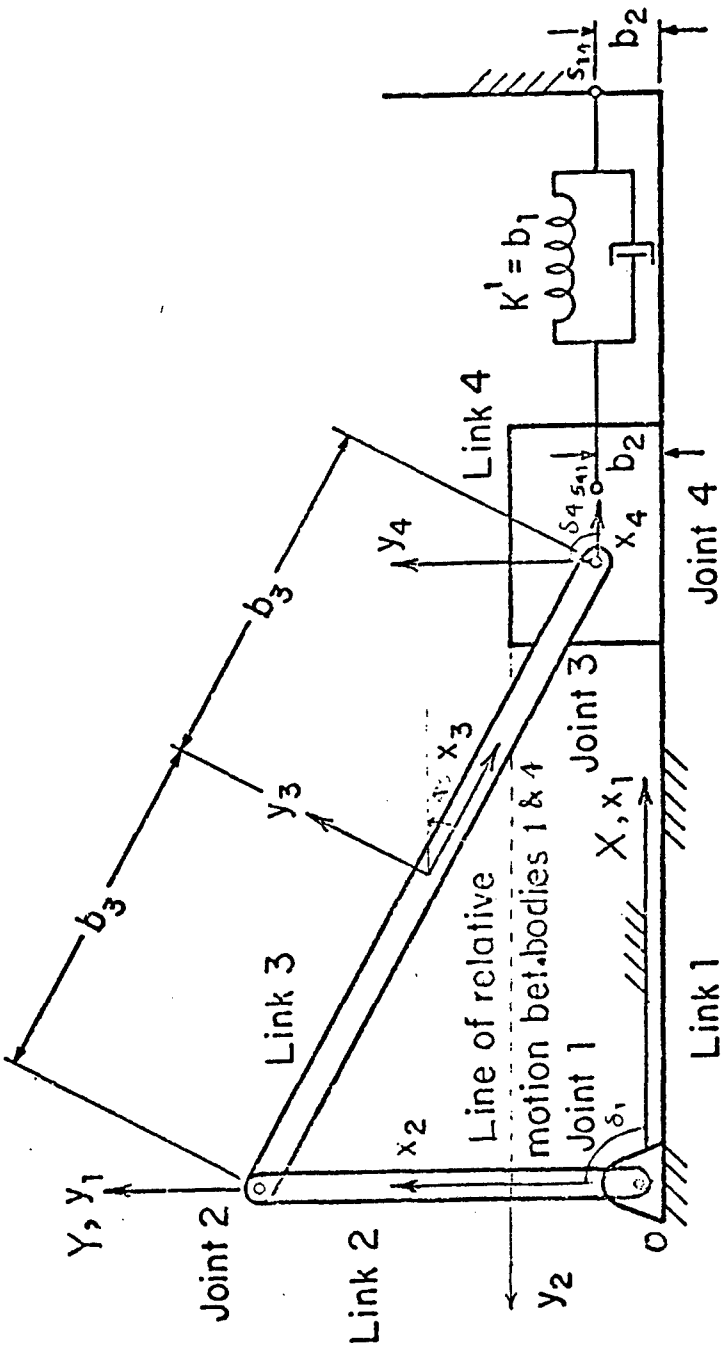


Figure 2.5 Approximate Initial Configuration of the Slider-Crank Mechanism.

$$\bar{\Phi}_2 \equiv \Phi_{Y_4} \equiv Y_3 + x_3 \sin\phi_3 + y_3 \cos\phi_3 - Y_4 - x_4 \sin\phi_4 - y_4 \cos\phi_4 = 0$$

$$\bar{\Phi}_3 \equiv \Phi_{n_4} \equiv X_4 \cos(\phi_4 + \delta_4) + Y_4 \sin(\phi_4 + \delta_4) + \sqrt{x_4^2 + y_4^2}$$

$$- X_1 \cos(\phi_1 + \delta_1) - Y_1 \sin(\phi_1 + \delta_1) - \sqrt{x_1^2 + y_1^2} = 0$$

$$\bar{\Phi}_4 \equiv \Phi_{\phi_4} \equiv \phi_4 + \delta_4 - \phi_1 - \delta_1 = 0 \qquad (2.25)$$

The spring-damper relations for the pair connecting bodies 1 and 4 can be written as

$$\varepsilon_{\ell^1} \equiv \sqrt{(U_1^1 - U_4^1)^2 + (V_1^1 - V_4^1)^2} - \ell^1 = 0$$

$$\varepsilon_{V^1} \equiv \dot{\ell}^1 - v^1 = 0 \qquad (2.26)$$

$$\varepsilon_{F_X^1} \equiv [K^1(\ell^1 - \ell_0^1) + C^1 v^1 + F_0^1][U_1^1 - U_4^1]/\ell^1 - F_X^1 = 0$$

$$\varepsilon_{F_Y^1} \equiv [K^1(\ell^1 - \ell_0^1) + C^1 v^1 + F_0^1][V_1^1 - V_4^1]/\ell^1 - F_Y^1 = 0$$

where

$$U_1^1 = X_1 + x_{s14} \cos\phi_1 - y_{s14} \sin\phi_1$$

$$V_1^1 = Y_1 + x_{s14} \sin\phi_1 + y_{s14} \cos\phi_1 \qquad (2.27)$$

$$U_4^1 = X_4 + x_{s41} \sin\phi_4 - y_{s41} \cos\phi_4$$

$$V_4^1 = Y_4 + x_{s41} \sin\phi_4 + y_{s41} \cos\phi_4$$

(x_{s14}, y_{s14}), (x_{s41}, y_{s41}) are respectively the coordinates of the points s_{14} and s_{41} of attachment of the spring-damper pair.

In this problem, contributions to the generalized forces Q_{X_4}, Q_{Y_4}, and Q_{ϕ_4} come only from the spring-damper forces. Thus,

$$
\begin{aligned}
Q_{X_4} &= F_X^1 (= -Q_{X_1}) \\
Q_{Y_4} &= F_Y^1 (= -Q_{Y_1}) \\
Q_{\phi_4} &= -F_X^1 (x_{s41} \sin\phi_4 + y_{s41} \cos\phi_4) + F_Y^1 (x_{s41} \cos\phi_4 - y_{s41} \sin\phi_4)
\end{aligned}
\tag{2.28}
$$

From Eqs. (2.22) to (2.24), the state equations of motion for the 4-th body can be written as,

$$
\begin{bmatrix}
M_4 & & & & & \\
 & M_4 & \underset{\sim}{0} & & & \\
 & & J_4 & & & \\
 & & & -1 & & \\
 & \underset{\sim}{0} & & & -1 & \\
 & & & & & -1
\end{bmatrix}
\begin{bmatrix} \dot{u}_4 \\ \dot{v}_4 \\ \dot{w}_4 \\ \dot{X}_4 \\ \dot{Y}_4 \\ \dot{\phi}_4 \end{bmatrix}
-
\begin{bmatrix}
Q_{X_4} - \sum_{k=1}^{4} \frac{\partial \bar{\Phi}_k}{\partial X_4} \mu_k \\
Q_{Y_4} - \sum_{k=1}^{4} \frac{\partial \bar{\Phi}_k}{\partial Y_4} \mu_k \\
Q_{\phi_4} - \sum_{k=1}^{4} \frac{\partial \bar{\Phi}_k}{\partial \phi_4} \mu_k \\
-u_4 \\
-v_4 \\
-w_4
\end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\tag{2.29}
$$

The equations of motion and constraint relations for other bodies of the mechanism can be written in a similar manner. However, manual formulation of the equations is unnecessary, since the computer program ADAMS-2D [14] automatically generates the necessary code for all the expressions and equations. Only data defining parameters such as mass, moments of inertia, locations of centers of mass, location of joints and points of attachment of spring-damper pairs, and spring-damper constants, (see [14,63]) are to be provided by the user.

2.1.2.7 Goblal System Equations

For a mechanical system of n bodies, m joints, and s spring-dampers, the global vector of generalized coordinates and velocities may be denoted by z, having components $z_1, z_2, \ldots, z_\zeta$, where $\zeta = 6n$.

$$\underline{z} \equiv \left[\underline{z}^{1^T}, \underline{z}^{2^T}, \ldots, \underline{z}^{n^T}\right]^T$$

$$\equiv [z_1, z_2, \ldots, z_\zeta]^T \tag{2.30}$$

The global vector of Lagrange multipliers may be denoted by $\underline{\mu}$ which, with its components $\mu_1, \mu_2, \ldots$, is

$$\underline{\mu} \equiv [\mu_1, \mu_2, \ldots, \mu_{2m-1}, \mu_{2m}]^T \tag{2.31}$$

The global vector of spring-damper variables may be denoted by $\underline{\ell}$, which with its components $\ell_1, \ell_2, \ldots$, is given by

$$\underline{\ell} \equiv \left[\underline{\ell}^{1^T}, \underline{\ell}^{2^T}, \ldots, \underline{\ell}^{s^T}\right]^T \equiv \left[\ell_1, \ell_2, \ldots, \ell_{4s-1}, \ell_{4s}\right]^T \tag{2.32}$$

Now the state equations of motion for the entire system can be written in the form (deleting for simplicity the underlines of the vector variables)

$$F(t,\dot{z},z,\ell,\mu,b) \equiv P(b)\dot{z} + f(t,z,\ell,\mu, b) = 0 \tag{2.33}$$

where

$$P(b) \equiv \begin{bmatrix} P^1(b) & & 0 \\ & P^2(b) & \\ 0 & & \ddots \; P^n(n) \end{bmatrix}_{(6n \times 6n)} \tag{2.34}$$

$$f \equiv \left[f_1^T, f_2^T, \ldots, f_n^T\right]^T$$

From equations (2.18) to (2.21) one may summarize the constraint equations in the form

$$\Phi(z,b) = 0 \tag{2.35}$$

From equations (2.9) to (2.12), one may write the equations related to spring-damper pairs as

$$\bar{\xi} \equiv \bar{I}\dot{\ell} + \xi(z,\ell,b) = 0 \tag{2.36}$$

where

$$\bar{\xi} \equiv \left[\varepsilon_{\ell^1}, \varepsilon_{v^1}, \varepsilon_{F_X^1}, \varepsilon_{F_Y^1}, \varepsilon_{\ell^2}, \ldots, \varepsilon_{F_Y^s}\right]^T \tag{2.37}$$

ξ is the algebraic part of $\bar{\xi}$, and $\bar{I}$ is defined as

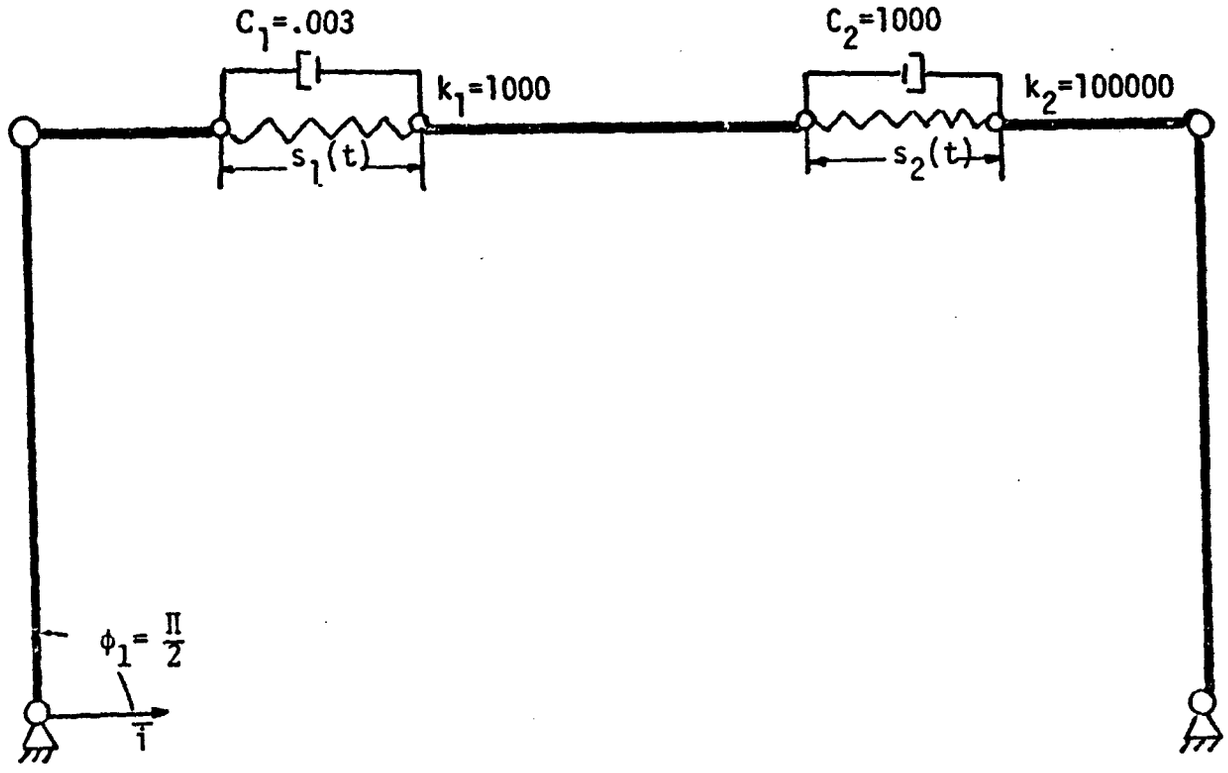
$$\bar{I} \equiv \begin{bmatrix} 0 & 0 & & & & & & & 0 \\ 1 & 0 & & & & & & & \\ & & 0 & & & & & & \\ & & & 0 & & & & & \\ & & & & 0 & 0 & & & \\ & & & & 1 & 0 & & & \\ & & & & & & 0 & & \\ 0 & & & & & & & 0 & \\ & & & & & & & & \ddots \end{bmatrix}_{(4s \times 4s)} \tag{2.38}$$

The system of equations (2.33), (2.35), and (2.36) is solved by Gear's predictor-corrector algorithm and sparse matrix techniques discussed briefly in Sections 2.2 and 2.3.

2.2 Stiff Integration (Gear) Algorithm

2.2.1 Introduction

Any system of differential equations that has widely split eigenvalues, at least locally, is called a stiff system. If a nonlinear system N is approximated by a linear system L around some point t, the eigenvalues of L are the local eigenvalues of N at t with respect to L. The mechanical system of Fig. 2.6 represents a stiff mechanical system. As indicated in Chapter I, mechanical systems like the ones discussed in Chapter VI may not be stiff initially, but they may unpredictably become stiff. It has been shown in references [2,6] that neither Runge-Kutta nor Adams-Bashforth nor Adams-Moulton algorithms are suitable for the solutions of such systems, for stability reasons [2,6].



$\dot{\phi}(0)$=1000 rad/sec

$\dot{s}_1(0)=0$

$\dot{s}_2(0)=0$

$s_1(0)$=1"

$s_2(0)$=1"

Figure 2.6 Stiff Mechanical System.

In 1969, Gear developed stiffly stable [2,6,47,48] multistep algorithms, which are well suited for the solution of stiff systems. Originally he considered systems of the form of Eq. (1.3) and employed his criteria of stiff stability to derive the algorithm. Later he showed [46] that the same algorithm can be used for mixed systems of differential and algebraic equations. One must understand the technical details of the subject of multistep numerical predictor-corrector algorithms and the concepts of stability and convergence and automatic change of order and step-size in order to have proper command of the Gear algorithms and their implementation in computer program DIFSUB (see [7]). These are, however, availabe in standard references [2,6]. and will not be treated here. The basic idea of automatic control of order and step size can be stated briefly as follows:

Let $t = t_n$ represent the current time instant and h and k be the current step size and order of the numerical integration. Let ε_T be the local truncation error [6] and ε_{max} the maximum allowable ε_T.

The basic step control algorithm is then to execute one step and test whether the relation

$$\varepsilon_T \leq \varepsilon_{max} \tag{2.39}$$

is satisfied. If it is, the step is accepted. Otherwise, the step is rejected and a smaller step size $\hat{h} = \alpha h (\alpha < 1)$ is used. The exact step size to use for executing the next step, or for repeating the rejected step, is given by the choice of α as the maximum value computed from three expressions [6] for local truncation errors

corresponding to the orders k, k - 1, and k + 1. The maximum of the three α's gives the maximum allowable step size and the corresponding order is the optimum order to compute values at $t = t_{n+1}$.

2.2.2 The Gear Algorithm and the Mixed System of Differential and Algebraic Equations

The k-th order Gear algorithm for the solution of the mixed system of differential and algebraic equations (1.1) and (1.2) is an implicit algorithm of the form

$$\underline{y}_{n+1} = -\beta_0 h\, \dot{\underline{y}}_{n+1} - (\alpha_0 \underline{y}_n + \alpha_1 \underline{y}_{n-1} + \ldots + \alpha_{k-1} \underline{y}_{n-k+1})$$

$$= -\beta_0 h\, \dot{\underline{y}}_{n+1} - \sum_{j=0}^{k-1} \alpha_j\, \underline{y}_{n-j} \qquad (2.40)$$

where h is the time step, $\underline{y}_{n+1}, \underline{y}_n, \ldots$ are the values of $\underline{y}$ at time instants $t_{n+1}, t_n, \ldots$, and $\alpha_0, \alpha_1, \ldots, \alpha_{k-1}, \beta_0$ are the (k + 1) coefficients known as Gear coefficients for this multistep algorithm [2,6].

One proceeds from the n^{th} to the $(n + 1)^{st}$ time step by solving Eq. (2.40), together with Eq. (1.1) at $t = t_{n+1}$, i.e.

$$\underline{F}(\dot{\underline{y}}_{n+1}, \underline{y}_{n+1}, t_{n+1}) = 0 \qquad (2.41)$$

Linearization of Eq. (2.41) gives the Newton formula [6] at the $(n + 1)^{st}$ time step:

$$\frac{\partial \underline{F}^{(m)}}{\partial \underline{y}} \Delta \underline{y}^{(m)} + \frac{\partial \underline{F}^{(m)}}{\partial \dot{\underline{y}}} \Delta \dot{\underline{y}}^{(m)} = -\underline{F}^{(m)} \qquad (2.42)$$

where

$$\Delta \underline{y}^{(m)} = \underline{y}^{(m+1)} - \underline{y}^{(m)}$$
$$\Delta \dot{\underline{y}}^{(m)} = \dot{\underline{y}}^{(m+1)} - \dot{\underline{y}}^{(m)} \tag{2.43}$$

m being the iteration number in the Newton method of solving the algebraic equations. The time step counter n has been dropped, for simplicity of notation.

Since $\sum_{j=0}^{k-1} \alpha_j \underline{y}_{n-j}$ remains invariant for Newton's iteration at the $(n + 1)^{st}$ time step, one obtains from Eq. (2.40)

$$\Delta \underline{y}^{(m)} = -\beta_0 h \, \Delta \dot{\underline{y}}^{(m)}$$
or
$$\Delta \dot{\underline{y}}^{(m)} = -\frac{1}{\beta_0 h} \Delta \underline{y}^{(m)} \tag{2.44}$$

Hence, Eq. (2.42) becomes (after the substitution of the second of Eqs. (2.44))

$$\left[\frac{\partial \underline{F}^{(m)}}{\partial \underline{y}} - \frac{1}{\beta_0 h} \frac{\partial \underline{F}^{(m)}}{\partial \dot{\underline{y}}} \right] \Delta \underline{y}^{(m)} = -\underline{F}^{(m)} \tag{2.45}$$

This is called the "corrector formula" for $\underline{y}$ at the $(n + 1)^{st}$ time step.

Equation (2.45) together with the second equation of Eq. (2.44), updates both $\underline{y}$ and $\dot{\underline{y}}$, which are required in $\underline{F}^{(m)}$. The iteration is continued until the right-hand side of Eq. (2.45) is less than a pre-assigned small quantity. Since the Jacobian matrix on the left-hand

side of Eq. (2.45) is of the same structure for each iteration, this procedure matches ideally the requirements for code generation for sparse matrix algorithms discussed in Section 2.3.

It should be noted that in this procedure even a linear differential equation will generally require more than one Newton iteration for corrector convergence. Moreover, this procedure may be adopted for any implicit algorithm.

The above procedure for the evaluation of $\underline{y}_{n+1}$ and $\dot{\underline{y}}_{n+1}$ is due to Calahan and Orlandea [7,8,9] and has been used in the ADAMS program [7].

For various advantages in computer programming, implicit multistep predictor-corrector algorithms are recast into canonical matrix representations (see references [2,6,7]). For that purpose, the Gear algorithm is recast into such a representation with the help of Nordsieck vector $\underline{z}_n$ [2,6,7] that is defined as

$$\underline{z}_n \equiv [y_n, hy_n', h^2y_n''/2!, \ldots, h^k y_n^{\langle k \rangle}/k!]^T \qquad (2.46)$$

where

$$y_n', y_n'', \ldots, y_n^{\langle k \rangle}$$

are the 1st, 2nd, ..., k^{th} derivative respectively of a single component y at $t = t_n$. The Nordsieck array $\underline{z}_n$ is defined by

$$\underline{z}_n \equiv [\underline{y}_n, h\underline{y}_n', h^2\underline{y}_n''/2!, \ldots, h^k \underline{y}_n^{\langle k \rangle}/k!] \qquad (2.47)$$

All the updated values of the Nordsieck vectors are required for prediction of the solution variables that are used as initial estimates in the corrector iterations. It should be noted here that Eqs. (2.44) and (2.45) give only the first two components of the Nordsieck vectors.

Orlandea [7] has shown that all the components of the Nordsieck vectors can be obtained from the corrector iteration formulas of the form:

$$\left[\frac{\partial \underline{F}^{(m)}}{\partial \underline{y}} - \frac{1}{\beta_0 h}\,\frac{\partial \underline{F}^{(m)}}{\partial \dot{\underline{y}}}\right] \Delta \underline{z}^{(m)(i)} = -\frac{c_{zi}}{c_{z1}}\,\underline{F}^{(m)} \qquad (2.48)$$

where the time step counter n has again been suppressed for simplicity, the vector $\underline{z}^{(m)(i)}$ represents the vector of the i-th components of $\underline{z}_n$ at the m-th Newton iteration, and c_{zi}, i = 1,2,...,k+1, are the coefficients of the transformed Gear algorithm. Their values are given in Table 2.1 (see reference [6]).

The present form of DIFSUB, however, does not iterate for the values of the Nordsieck vector components other than the first two. They are evaluated from Eq. (2.48) after corrector convergence for the first two components.

2.2.3 Starting of Multistep Algorithms

Multistep numerical algorithms are not self starting. Generally a single step algorithm is used at least k times before a multistep algorithm can be initiated.

In the ADAMS program, The Gear algorithm is implemented through the subroutine DIFSUB and initially the order is taken to be 1. The

Table 2.1

Coefficients of Stiffly Stable Methods
in Canonical Form

k	2	3	4	5	6
c_{z1}	$\frac{2}{3}$	$\frac{6}{11}$	$\frac{25}{50}$	$\frac{120}{274}$	$\frac{720}{1764}$
c_{z2}	$\frac{3}{3}$	$\frac{11}{11}$	$\frac{50}{50}$	$\frac{274}{274}$	$\frac{1764}{1764}$
c_{z3}	$\frac{1}{3}$	$\frac{6}{11}$	$\frac{35}{50}$	$\frac{225}{274}$	$\frac{1624}{1764}$
c_{z4}		$\frac{1}{11}$	$\frac{10}{50}$	$\frac{85}{274}$	$\frac{735}{1764}$
c_{z5}			$\frac{1}{50}$	$\frac{15}{274}$	$\frac{175}{1764}$
c_{z6}				$\frac{1}{274}$	$\frac{21}{1764}$
c_{z7}					$\frac{1}{1764}$

first order Gear algorithm, being exactly the backward Euler algorithm, is itself a single step algorithm that serves the purpose of initialization.

2.2.4 Corrector Formulas for the Dynamical System Equations

For the state equations of motion given by Eqs. (2.33), together with the constraint equations (2.35) and the spring-damper relations

(2.36), the corrector formula (2.45) can be written as (deleting underlines and superscripts),

$$\begin{bmatrix} -\frac{1}{\beta_0 h} P(b) + \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \mu} & \frac{\partial f}{\partial \ell} \\ \frac{\partial \Phi}{\partial z} & 0 & 0 \\ \frac{\partial \xi}{\partial z} & 0 & -\frac{1}{\beta_0 h} \bar{I} + \frac{\partial \xi}{\partial \ell} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \mu \\ \Delta \ell \end{bmatrix} = - \begin{bmatrix} F \\ \Phi \\ \bar{\xi} \end{bmatrix} \tag{2.49}$$

where the following relations have been used:

$$\frac{\partial F}{\partial z} = \frac{\partial f}{\partial z} , \qquad \frac{\partial \bar{\xi}}{\partial z} = \frac{\partial \xi}{\partial z}$$

$$\frac{\partial F}{\partial \mu} = \frac{\partial f}{\partial \mu} , \qquad \frac{\partial \bar{\xi}}{\partial \ell} = \frac{\partial \xi}{\partial \ell} \tag{2.50}$$

$$\frac{\partial F}{\partial \ell} = \frac{\partial f}{\partial \ell}$$

Moreover, when $z^i \equiv [X_i, Y_i, \phi_i, u_i, v_i, w_i]^T$ instead of $[u_i, v_i, w_i, X_i, Y_i, \phi_i]^T$, one obtains (see Eq. (2.24)),

$$\frac{\partial f}{\partial \mu} = \left(\frac{\partial \phi}{\partial z} \right)^T \tag{2.51}$$

In that special case, Eq. (2.49) can be written as

$$\begin{bmatrix} -\frac{1}{\beta_0 h} P(b) + \frac{\partial f}{\partial z} & \left(\frac{\partial \Phi}{\partial z}\right)^T & \frac{\partial f}{\partial \ell} \\ \frac{\partial \Phi}{\partial z} & 0 & 0 \\ \frac{\partial \xi}{\partial z} & 0 & -\frac{1}{\beta_0 h} \bar{I} + \frac{\partial \xi}{\partial \ell} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \mu \\ \Delta \ell \end{bmatrix} = - \begin{bmatrix} F \\ \Phi \\ \bar{\xi} \end{bmatrix}$$

(2.52)

A descriptive tableau form for the corrector formula of Eq. (2.49) can be given as

Body 1	$\underset{\sim}{0}$	constraint related sparse submatrices with elements of $\left(\frac{\partial \Phi}{\partial \underline{q}}\right)^T$	spring-damper related submatrices with elements of $\frac{\partial \underline{F}}{\partial \underline{\ell}}$
$\underset{\sim}{0}$	Body n		
constraint related sparse submatrices with elements of $\frac{\partial \Phi}{\partial \underline{q}}$		$\underset{\sim}{0}$	
spring-damper related submatrices of the form: $\frac{\partial \bar{\xi}}{\partial \underline{u}}$, $\frac{\partial \bar{\xi}}{\partial \underline{q}}$, $\frac{\partial \bar{\xi}}{\partial \underline{\ell}}$			

$$\begin{bmatrix} \begin{pmatrix} \Delta \underline{u}^1 \\ \Delta \underline{q}^1 \end{pmatrix} \\ \begin{pmatrix} \Delta \underline{u}^2 \\ \Delta \underline{q}^2 \end{pmatrix} \\ \Delta \underline{\mu} \\ \Delta \underline{\ell} \end{bmatrix} = \begin{bmatrix} -\underline{F} \\ -\underline{\Phi} \\ -\bar{\xi} \end{bmatrix}$$

(2.53)

The matrix in the left-hand side of this corrector formula is known as the corrector Jacobian matrix. A detailed description of the nonzero positions in the Jacobian matrix of such a tableau for the four-bar slider-crank mechanism of Fig. 2.5 is given by Fig. 2.7.

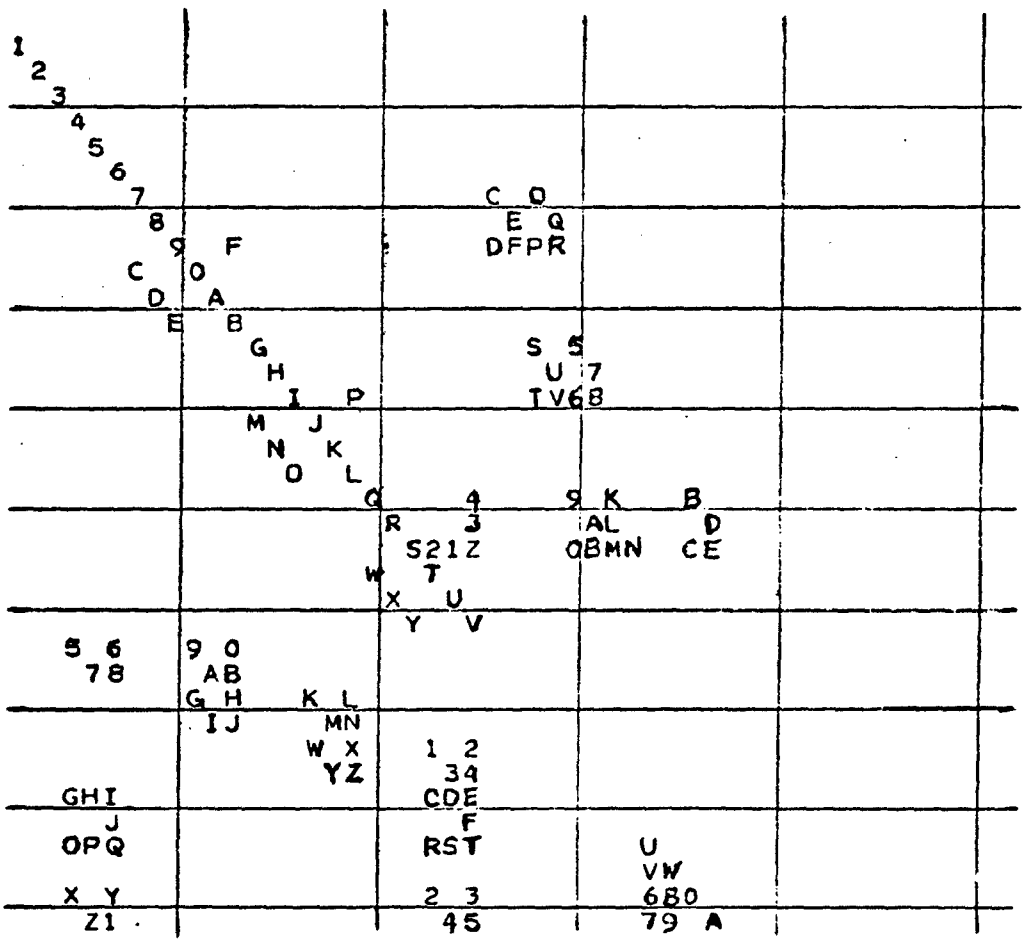


Figure 2.7 Symbolic Listing of the Nonzero Entries in the Jacobian Matrix for the Example Slider-Crank Mechanism.

2.3 Sparse Matrix Techniques

2.3.1 Introduction

In order to solve the system of simultaneous linear equations of Eq. (2.52), some matrix method must be employed. Sparse matrix techniques enhance speed of computation in such systems. These techniques are described briefly in the following. If less than 30% of the entries in a square matrix $\underset{\sim}{A}_{nxn}$ are nonzero, the matrix $\underset{\sim}{A}$ is considered to be sparse and storing the matrix as a two-dimensional array becomes inefficient. Consideration of matrix sparsity is extremely important for speed of computation in problems of mechanical system analysis (particularly dynamic systems considered later in this dissertation). This consideration outweighs the difficulties encountered in solving a large set of simultaneous linear equations to which many physical systems can be ultimately reduced [50,51].

Sparse systems of simultaneous linear equations generally result from the solution of the following classes of equations:

(1) Ordinary differential equations and/or algebraic equations, where after time discretization and/or linearization an irregularly structured matrix is encountered (cf. Fig. 2.6);

(2) Partial differential equations, where after discretization of the spatial variables by finite difference or finite element techniques [52,53,54] a regularly structured matrix is handled.

This research is concerned only with problems of the first class (as is evident from the last section and the developments in the subsequent chapters).

In sparse matrix methods, operations involving zeros are avoided by using structural information (the positions of nonzero entries) that is stored in a compacted form. One way of doing this is to store the row and column indices of each nonzero element in two vectors I and J and the value of the elements in a third vector G. This method is called "i-j" ordering. According to Calahan [10] this is the most convenient method and can be easily converted to other methods of compacting the data, such as the threaded list method and the bit map method, which are discussed in detail in [10].

2.3.2 Solution of Simultaneous Linear Algebraic Equations

There are two general methods for solving a set of simultaneous linear algebraic equations

$$\underset{\sim}{A}\underline{x} = \underline{B} \tag{2.54}$$

These are (1) the Gaussian elimination method and (2) $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization method [6,11,44]. Although theoretically they are somewhat interconnected, the solution techniques involve two different algorithms. The $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization method is preferable for sparse matrix techniques. Since the inverse of a sparse matrix may be full, whereas $\underset{\sim}{L}\,\underset{\sim}{U}$ factors may retain sparsity [10], the number of operations in the first method is much larger than that in the second [6]. The method of $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization is now briefly described.

Let $\underset{\sim}{L}$ and $\underset{\sim}{U}$ be of the form:

$$\underset{\sim}{L} = \begin{bmatrix} \ell_{11} & & \underset{\sim}{0} \\ \vdots & \ddots & \\ \ell_{i1} & \cdots \ell_{ii} & \\ \vdots & & \ddots \\ \ell_{n1} & \cdots \ell_{ni} & \cdots \ell_{nn} \end{bmatrix}, \quad \underset{\sim}{U} = \begin{bmatrix} 1 & \cdots u_{1j} & \cdots u_{1n} \\ & \ddots & \vdots \\ & & u_{in} \\ & & \vdots \\ \underset{\sim}{0} & & 1 \end{bmatrix} \tag{2.55}$$

Then to get

$$\underset{\sim}{A} = \underset{\sim}{L}\ \underset{\sim}{U} \tag{2.56}$$

one has with a_{ij} as the elements of $\underset{\sim}{A}$,

$$\ell_{ij} = a_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\, u_{kj} \quad , \qquad i \geq j \tag{2.57}$$

and

$$u_{ij} = \left[a_{ij} - \sum_{k-1}^{i-1} \ell_{ik}\, u_{kj} \right] \Big/ \ell_{ii} \quad , \qquad i < j \tag{2.58}$$

which determine the matrices $\underset{\sim}{L}$ and $\underset{\sim}{U}$ for the matrix $\underset{\sim}{A}_{(n \times n)}$. This method is employed in the Crout algorithm [55] for $\underset{\sim}{L}$, $\underset{\sim}{U}$ factorization.

After $\underset{\sim}{L}\ \underset{\sim}{U}$ factorization of $\underset{\sim}{A}$, the solution of Eq. (2.54) is obtained as follows:

$$\underset{\sim}{L}\ \underset{\sim}{U}\ \underline{x} = \underline{B} \tag{2.59}$$

Denoting

$$\underset{\sim}{U}\,\underline{x} = \underline{y} \tag{2.60}$$

one has

$$\underset{\sim}{L}\,\underline{y} = \underline{B} \tag{2.61}$$

Now, $\underline{y}$ can be determined from Eq. (2.61) by forward substitution and $\underline{x}$ can be obtained from Eq. (2.60) by backward substitution. The number $\bar{N}$ of operations required by the $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization method is given by [6].

$$\bar{N} = \frac{n^3}{3} + n^2 - \frac{n}{3} \tag{2.62}$$

It has been shown in [54] that if only $\underline{B}$ changes, then only n^2 operations must be performed to get a new solution.

2.3.3 Sparsity and Optimal Ordering

Let an auxiliary matrix $\underset{\sim}{Q}$ be defined as

$$\underset{\sim}{Q} \equiv L + (\underset{\sim}{U} - \underset{\sim}{I}) \tag{2.63}$$

where $\underset{\sim}{A} = \underset{\sim}{L}\,\underset{\sim}{U}$ and $\underset{\sim}{I}$ is the identity matrix. Let q_{ij} and a_{ij} denote elements of $\underset{\sim}{Q}$ and $\underset{\sim}{A}$, respectively. Then if $q_{ij} \neq 0$ whenever $a_{ij} = 0$, the element q_{ij} is said to be a "fill" that is generated in $\underset{\sim}{L}\,\underset{\sim}{U}$ factorization of $\underset{\sim}{A}$.

It is desirable that in order to minimize the computational effort, the sparsity of the matrix $\underset{\sim}{A}$ be transmitted to $\underset{\sim}{L}$ and $\underset{\sim}{U}$.

That is, the number of fills should be kept to a minimum. This can be done by a suitable permutation of rows and columns. Such an operation is known as Optimal Ordering (or pivoting). Some simple examples demonstrating the effectiveness of such ordering can be found in [6,7]. Over and above the solution efficiency and the minimization of the size of the generated code, there is another important reason for optimal ordering. Most of the equations in physical problems are nonlinear and the entries in the matrix vary from step to step of a solution process. In such cases optimal ordering prevents generation of zero-valued pivots and costly regeneration of the solution code.

The detailed discussions of the subject of optimal ordering is beyond the scope of this dissertation. They can be found in references [56,57,58,59]. Codes such as OPTORD [60,49] and MOOP [61] are two of several computer programs that can be used for optimal ordering. Although optimal ordering aims at the largest (absolute value) non-zero, row-column entry as the pivot and minimization of the number of fills, generally both cannot be achieved simultaneously and the algorithm chooses from among the larger than average pivots, the one which results in the minimum number of fills. The mode of operation for optimal ordering can be visualized from the following considerations.

Let K be an integer between 1 and n that represents the K^{th} step associated with the permutation and $\underset{\sim}{L}\ \underset{\sim}{U}$ factorization of the remaining matrix at that pivot step. Every nonzero element in the residual submatrix of dimension $(n - K + 1)$ is considered as a

candidate for the pivotal element. Associated with each nonzero element A(I,J) in the submatrix is a weighting function:

$$W(I,J) = (LROWI - 1)\ (LCOLJ - 1)$$

where LROWI is the number of nonzero elements in the row containing A(I,J) and LCOLJ is the number of nonzero elements in the column containing A(I,J). The term W(I,J) represents the number of multiplications required if A(I,J) were selected as pivotal element. For each row with I fixed, W(I,J) is determined only for elements that are numerically acceptable; i.e., if

$$|A(I,J)| \geq \varepsilon \cdot AVG(I)$$

where $0 \leq \varepsilon \leq 1$ is a user-specified tolerance,

$$AVG(I) = \frac{1}{N_I} \sum_{J=1}^{N_I} |A(I,J)|$$

and N_I is the number of nonzero elements in row I. The algorithm attempts, in a systematic manner, to locate the element with the largest absolute value and the smallest value for the weighting function.

2.3.4 Column Ordering of a Matrix

It has been mentioned in Subsection 2.3.1 that the nonzero values of the original matrix are stored in a vector G. They can be stored either in row-wise order, taking one row after another, or in column-wise order, taking one column after another. The optimal

ordering algorithms can handle both representations. However, the subroutines VMSP, VMNP, and VMBP, used in ADAMS programs for $\underset{\sim}{L}\ \underset{\sim}{U}$ factorization of matrices and numerical solution of linear equations, consider only the column order representation. The various aspects of all these subroutines are discussed in reference [62]. Many other subroutines performing similar functions can be found in [61].

In the original arbitrary "i-j" ordering, the row and column indices are stored in vectors NPOSR and NPOSC, respectively, and the nonzero elements of the matrix $\underset{\sim}{A}$ are stored in the vector G. To transform the "i-j" ordering into column ordering, a permutation vector is generated to cause the elements a_{ij} of $\underset{\sim}{A}$ to be entered into the appropriate locations in the vector G.

The process of converting the code from "i-j" ordering to column ordering is illustrated in Table 2.2. A unique number NRANK(I) is assigned to each of the NG nonzero entries in the matrix, in such a way that NRANK increases as one proceeds down a column. No value of NRANK for a certain column is greater than that for any other column to its right. After the initialization of a counter NCOUNT in Step 2 and reordering the NRANK and NCOUNT in Step 3, Step 4 generates the row indices IA(I) of the desired column ordered matrix and Step 5 generates the points JA(I) to the subscripts of the indices of the first element in each column. Step 6 generates the permutation vector NPOS for column ordering of the original G vector. This algorithm is implemented in subroutine S08000 of ADAMS 2-D [14] and SEP of ADAMS 3-D [7]. Simple numerical examples of this procedure can be found in reference [63].

Table 2.2

Column Ordering of a Random Matrix

N = dimensions of matrix
Step 1. Calculate the rank of each nonzero entry: NRANK(I) = NPOSR(I) + (N+1)*NPOSC(I), I=1,NG
Step 2. Initialize a vector NCOUNT(I) = I, I=1,NG
Step 3. Reorder NRANK from the smallest to the largest. Reorder NCOUNT along with NRANK
Step 4. Generate a vector IA(I), I=1,NG, that gives the row indices of the permuted G vector: IA(I) = NRANK(I) - (NRANK(I)/(N+1))*(N+1)
Step 5. Generate a vector JA(J), J=1,N, that points to the index I of IA(I) of the first nonzero entry in each column, and JA(N+1) = NG+1.
Step 6. Generate a vector NPOS(NCOUNT(I)) = I, I=1,NG.

In practice, one makes the assignments G(NPOS(I)) = a_{ij} to put G(I) in column order. The vector NPOS is generated only once and is never modified during the execution of the program. With the help of this vector, any element of G can be directly accessed and modified, if desired.

In the sensitivity analysis (see Chapter III of this report), the solution of adjoint equations requires column order representation of the transpose of the original matrix. To achieve this the original vectors NPOSR and NPOSC are stored in another set of vectors NPOSC1 and NPOSR1, respectively, and subroutine S08000 is called to generate the vectors NPOS and JP1 corresponding to original vectors NPOS and JA. This is done in subroutine DYNANL (see Chapter V).

2.3.5 Matrix Vectorization

Consider a column of a sparse matrix having the nonzero row positions shown in Fig. 2.8. This structure is described in the

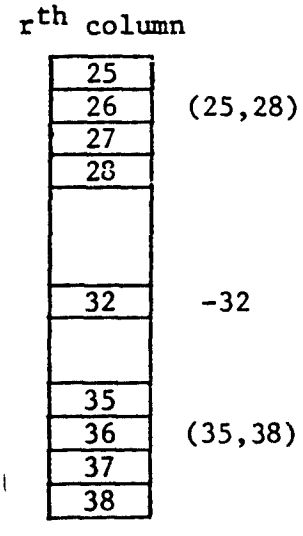


Figure 2.8 Matrix Vectorization.

conventional ordered list as

$$25, 26, 27, 28, 32, 35, 36, 37, 38 \qquad (2.64)$$

A list enumerating all the row positions in the column is called "scalar storage". For large matrices, additional savings in computer memory and execution time can be realized by further compacting the column-ordered code. This is done by a subroutine VECTOR after the optimal ordering operation. The algorithm obeys the following rules.

If in a given column, there is a set of two or more contiguous row positions, only the first and last row indices are retained in the vector IA of Table 2.2. This implies that all elements in between them, the boundary terms inclusive, are nonzero.

On the other hand, if there is an isolated element in a column, with no nonzero adjacent term, its row index is set negative.

JA of Table 2.2 is updated to reflect the changes in IA and has the same meaning.

Thus after vectorization, the r^{th} column shown in Fig. 2.8 is described in the ordered list as

$$25, 28, -32, 35, 38 \qquad (2.65)$$

Given the vectorized and permuted description of the nonsingular matrix, subroutine VMSP (Vectorized Matrix Symbolic Preprocessor) can be used to generate a symbolic description of the $\underset{\sim}{L}$ and $\underset{\sim}{U}$ matrices. The numerical $\underset{\sim}{L}$ $\underset{\sim}{U}$ factorization is performed by subroutine VMNP (using the code generated by VMSP). Subroutine VMBP then solves the

equations by forward and back substitution steps. Upon return from VMBP, the resultant solution vector is stored in the original right-hand side vector $\underline{B}$ of the equations $\underset{\sim}{A}\ \underline{x} = \underline{B}$. For further discussion, reference [62] is recommended.

CHAPTER III

FORMULATION OF THE OPTIMAL DESIGN PROBLEM AND SENSITIVITY ANALYSIS

3.1 Introduction

For constrained dynamic systems with just a few degrees of freedom, the treatment of the optimal design problem given in reference 1, in conjunction with the Newtonian equations of motion may be quite sufficient. However, for large systems, owing to nonlinearity in the system equations, the Newtonian approach is not convenient. On the other hand, it will be evident from the next sections that the Lagrangian sparse-matrix formulation of the equations of motion and implementation of a STIFF (GEAR) integration algorithm are extremely advantageous.

In the following, discussions will be confined to two dimensional systems. However, their extension to three-dimensional systems is quite straightforward, at least theoretically.

3.2 Formulation of the Optimal Design Problem

The optimal design problem for a mechanical system of n bodies, m joints, and s spring-dampers in the notation of Chapter 2 can now be stated as follows: Determine $b \in R^{\rho}$, ρ being a positive integer, to minimize the cost functional

$$\psi_0 = g_0(b,z(0),\bar{\ell}(0),z(t_1),\bar{\ell}(t_1)) + \int_0^{t_1} L_0[t,z(t),\mu(t),\ell(t),b]dt \tag{3.1}$$

where

$$\bar{\ell} = [\ell_{1+4j} \ , \ j=0,1,\ldots,(s-1)]^T = [\ell_1,\ell_5,\ldots,\ell_{(4s-3)}]^T \tag{3.2}$$

subject to the following conditions:

(a) state equations of motion of Eq. (2.33):

$$F(t,\dot{z},z,\ell,\mu,b) \equiv P(b)\dot{z} + f(t,z,\ell,\mu,b) = 0 ;$$

(b) constraints of Eq. (2.35):

$$\Phi_\alpha (z,b) = 0 \ , \quad \alpha=1,2,\ldots,2m ; \tag{3.3}$$

(c) the equations related to spring-damper pairs (Eq. (2.36));

(d) initial conditions

$$\Theta(z(0),\ell(0),b) \equiv \begin{cases} z(0) - \nu(b) = 0 & (3.4) \\ \bar{\ell}(0) - \bar{\nu}(b) = 0 & (3.5) \end{cases}$$

(e) functional equality constraints

$$\psi_\beta \equiv g_\beta(b,z(0),\bar{\ell}(0),z(t_1),\bar{\ell}(t_1)) + \int_0^{t_1} L_\beta[t,z(t),\mu(t),\ell(t),b]dt = 0 \ , \quad \beta=1,\ldots,r' ; \tag{3.6}$$

and/or functional inequality constraints

$$\psi_\beta \equiv g_\beta(b,z(0),\bar{\ell}(0),z(t_1),\bar{\ell}(t_1)) + \int_0^{t_1} L_\beta[t,z(t),\mu(t),\ell(t),b]dt \leq 0 \ , \quad \beta=r'+1,\ldots,r ; \tag{3.7}$$

(f) design parameter constraints

$$\chi_\gamma(t,b) = 0 \ , \quad 0 \leq t \leq t_1 \ , \quad \gamma = 1,2,\ldots,q' \ ,$$

$$\chi_\gamma(t,b) \leq 0 \ , \quad 0 \leq t \leq t_1 \ , \quad \gamma = q'+1,\ldots,q \tag{3.8}$$

(g) point-wise constraints of the form

$$H(t,z,\mu,\ell,b) \leq 0 \ , \quad 0 \leq t \leq t_1 \ . \tag{3.9}$$

The point-wise constraints are transformed to the equivalent functional form of Eq. (3.6) [39]:

$$\int_0^{t_1} H^2[1+\text{sgn } H]dt = 0 \tag{3.10}$$

or

$$\int_0^{t} \langle H \rangle \, dt = 0 \tag{3.11}$$

where

$$\langle H \rangle = \begin{cases} H^2 \ , & H \geq 0 \\ 0 \ , & H < 0 \end{cases} \tag{3.12}$$

3.3 Sensitivity Analysis

Before developing an optimization algorithm, it is necessary to determine how changes in design parameters change the cost functional ψ_0, the constraint functionals ψ_β, and the constraint functions χ_γ. From Eq. (3.8), it is clear that the first variations in χ_γ can easily be expressed in terms of δb. From Eqs. (3.1) and (3.6) or (3.7), it is seen that ψ_0 and ψ_β have the same form, so typical ψ, g, and L are considered.

Taking the first variation of a typical functional ψ, in terms of all of its variables, one obtains:

$$\delta\psi = \frac{\partial g}{\partial b}\,\delta b + \frac{\partial g}{\partial z(0)}\,\delta z(0) + \frac{\partial g}{\partial \bar{\ell}(0)}\,\delta\bar{\ell}(0) + \frac{\partial g}{\partial z(t_1)}\,\delta z(t_1)$$
$$+ \frac{\partial g}{\partial \bar{\ell}(t_1)}\,\delta\bar{\ell}(t_1) + \int_0^{t_1}\left[\frac{\partial L}{\partial z}\,\delta z + \frac{\partial L}{\partial \mu}\delta\mu + \frac{\partial L}{\partial \ell}\delta\ell + \frac{\partial L}{\partial b}\delta b\right]dt \tag{3.13}$$

The first variations of Eqs. (3.4) and (3.5) give:

$$\left.\begin{aligned} \delta z(0) - \frac{\partial \nu}{\partial b}\,\delta b &= 0 \\ \delta\bar{\ell}(0) - \frac{\partial \bar{\nu}}{\partial b}\,\delta b &= 0 \end{aligned}\right\} \tag{3.14}$$

To express $\delta\psi$ solely in terms of δb, an adjoint variable $\lambda(t) \equiv [\bar{\lambda}^T, \bar{\bar{\lambda}}^T, \lambda'^T]^T$ is introduced through the identities [1,70] obtained from Eqs. (2.33), (2.35) and (2.36):

$$\bar{\lambda}^T[P(b)\dot{z} + f(t,z,\mu,\ell,b)] = 0 \tag{3.15}$$

$$\bar{\bar{\lambda}}^T\,\Phi(z,b) = 0 \tag{3.16}$$

$$\lambda'^T[\bar{I}\,\dot{\ell} + \xi(z,\ell,b)] = 0 \tag{3.17}$$

where

$$\left.\begin{aligned} \bar{\lambda} &= (\lambda_1, \lambda_2, \ldots, \lambda_{6n})^T \\ \bar{\bar{\lambda}} &= (\lambda_{6n+1}, \ldots, \lambda_{6n+2m})^T \\ \lambda' &= (\lambda_{6n+2m+1}, \ldots, \lambda_{6n+2m+4s})^T \end{aligned}\right\} \tag{3.18}$$

Integrating Eqs. (3.15), (3.16), and (3.17) from 0 to t_1, one obtains

$$\int_0^{t_1} \bar{\lambda}^T[P(b)\dot{z} + f(t,z,\ell,\mu,b)]dt = 0 \tag{3.19}$$

$$\int_0^{t_1} \bar{\bar{\lambda}}^T\,\Phi(z,b)dt = 0 \tag{3.20}$$

and

$$\int_0^{t_1} \lambda'^T[\bar{I}\,\dot{\ell} + \xi(z,\ell,b)]dt = 0 \tag{3.21}$$

Integrating the first term in Eq. (3.19) by parts, one gets

$$\bar{\lambda}^T P(b)z\Big|_0^{t_1} - \int_0^{t_1} [\dot{\bar{\lambda}}^T P(b)z - \bar{\lambda}^T f(t,z,\ell,\mu,b)]dt = 0 \tag{3.22}$$

Similarly, integrating the first term in Eq. (3.21) by parts, one gets

$$\lambda'^T \bar{I}\ell\Big|_0^{t_1} - \int_0^{t_1} [\dot{\lambda}'^T \bar{I}\ell - \lambda'^T \xi(z,\ell,b)]dt = 0 \tag{3.23}$$

The first variations of Eqs. (3.22), (3.20), and (3.23) lead to

$$\bar{\lambda}^T\left[\frac{\partial(P(b)z)}{\partial b}\delta b + P(b)\delta z\right]\Big|_0^{t_1} - \int_0^{t_1}\left[\dot{\bar{\lambda}}^T\left(\frac{\partial(P(b)z)}{\partial b}\delta b + P(b)\delta z\right) - \bar{\lambda}^T\frac{\partial f}{\partial z}\delta z - \bar{\lambda}^T\frac{\partial f}{\partial \mu}\delta\mu - \bar{\lambda}^T\frac{\partial f}{\partial \ell}\delta\ell - \bar{\lambda}^T\frac{\partial f}{\partial b}\delta b\right]dt = 0 \tag{3.24}$$

$$\int_0^{t_1} \bar{\lambda}^T\left[\frac{\partial\Phi}{\partial z}\delta z + \frac{\partial\Phi}{\partial b}\delta b\right]dt = 0 \tag{3.25}$$

and

$$\lambda'^T[\bar{I}\,\delta\ell]\Big|_0^{t_1} - \int_0^{t_1}\left[\dot{\lambda}'^T(\bar{I}\,\delta\ell) - \lambda'^T\frac{\partial\xi}{\partial z}\delta z - \lambda'^T\frac{\partial\xi}{\partial\ell}\delta\ell - \lambda'^T\frac{\partial\xi}{\partial b}\delta b\right]dt = 0 \tag{3.26}$$

Integrating the first term under the integral in Eq. (3.24) by parts and making necessary adjustments, one has

$$\bar{\lambda}^T P(b)\delta z\Big|_0^{t_1} - \int_0^{t_1}\left[\dot{\bar{\lambda}}^T P(b)\delta z - \bar{\lambda}^T\frac{\partial(P(b)\dot{z})}{\partial b}\delta b - \bar{\lambda}^T\frac{\partial f}{\partial z}\delta z - \bar{\lambda}^T\frac{\partial f}{\partial\mu}\delta\mu - \bar{\lambda}^T\frac{\partial f}{\partial\ell}\delta\ell - \bar{\lambda}^T\frac{\partial f}{\partial b}\delta b\right]dt = 0 \tag{3.27}$$

It is to be noted here that in the second term under the integral of Eq. (3.27) the differentiation is of P(b) only with respect to b and -f should not be substituted for $P(b)\dot{z}$, i.e.,

$$\text{if} \quad P(b)\dot{z} = P_{ij}(b)\dot{z}_j \quad , \qquad j \text{ summed},$$

$$\frac{\partial (P(b)\dot{z})}{\partial b}\,\delta b = P_{ij,k}\dot{z}_j\,\delta b_k \quad , \qquad j,\ k \text{ summed},$$

where

$$P_{ij,k} \equiv \frac{\partial P_{ij}(b)}{\partial b_k} \quad . \tag{3.28}$$

It is further noted that $\delta z(0)$ and $\delta\bar{\ell}(0)$ are expressed in terms of δb by the Eq. (3.14).

Now to express the 4th and 5th terms of the right-hand side of Eq. (3.13) in terms of δb, the following boundary conditions are introduced:

$$\left.\begin{aligned} &\bar{\lambda}^T(t_1) = \frac{\partial g}{\partial z(t_1)}\,P(b)^{-1} \\ &\text{and} \\ &\lambda'_{2+4j}(t_1) = \frac{\partial g}{\partial \ell_{1+4j}(t_1)} \quad , \ j=0,1,2,\ldots,(s-1) \end{aligned}\right\} \tag{3.29}$$

Then, with Eqs. (3.14) and (3.29), Eqs. (3.27) and (3.26) can be rewritten as

$$\begin{aligned} &\bar{\lambda}^T(0)\,P(b)\,\frac{\partial \nu}{\partial b}\,\delta b - \frac{\partial g}{\partial z(t_1)}\,\delta z(t_1) + \int_0^{t_1}\Big[\dot{\bar{\lambda}}^T P(b)\delta z \\ &\quad - \bar{\lambda}^T \frac{\partial (P(b)\dot{z})}{\partial b}\,\delta b - \bar{\lambda}^T \frac{\partial f}{\partial z}\,\delta z - \bar{\lambda}^T \frac{\partial f}{\partial \mu}\,\delta\mu \\ &\quad - \bar{\lambda}^T \frac{\partial f}{\partial \ell}\delta\ell - \bar{\lambda}^T \frac{\partial f}{\partial b}\,\delta b\Big]dt = 0 \end{aligned} \tag{3.30}$$

and

$$\begin{aligned} \sum_{j=0}^{s-1} \lambda'_{2+4j}(0)\,\delta\ell_{1+4j}(0) &\equiv \sum_{j=0}^{s-1} \lambda'_{2+4j}(0)\frac{\partial \bar{\nu}_{j+1}}{\partial b}\,\delta b = \sum_{j=0}^{s-1} \frac{\partial g}{\partial \ell_{1+4j}(t_1)}\,\delta\ell_{1+4j}(t_1) \\ &- \int_0^{t_1}\Big[\dot{\lambda}'^T(\bar{I}\delta\ell) - \lambda'^T \frac{\partial \xi}{\partial z}\,\delta z - \lambda'^T \frac{\partial \xi}{\partial \ell}\,\delta\ell - \lambda'^T \frac{\partial \xi}{\partial b}\,\delta b\Big]dt \end{aligned} \tag{3.31}$$

Now let $\bar{\lambda}$, $\bar{\bar{\lambda}}$, and λ' satisfy the adjoint equations

$$F' \equiv -P^T \frac{d\bar{\lambda}}{dt} + \frac{\partial f^T}{\partial z} \bar{\lambda} + \frac{\partial \Phi^T}{\partial z} \bar{\bar{\lambda}} + \frac{\partial \xi^T}{\partial z} \lambda' - \frac{\partial L^T}{\partial z} = 0 \qquad (3.32)$$

$$\Phi' \equiv \frac{\partial f^T}{\partial \mu} \bar{\lambda} - \frac{\partial L^T}{\partial \mu} = 0 \qquad (3.33)$$

and

$$\bar{\xi}' \equiv -\bar{I}^T \frac{d\lambda'}{dt} + \frac{\partial f^T}{\partial \ell} \bar{\lambda} + \frac{\partial \xi^T}{\partial \ell} \lambda' - \frac{\partial L^T}{\partial \ell} = 0 \ , \qquad (3.34)$$

with initial conditions (3.29) already chosen. With Eqs. (3.32), (3.33), and (3.34), one obtains from Eq. (3.30)

$$\begin{aligned} &\bar{\lambda}^T(0)\ P(b)\ \frac{\partial \nu}{\partial b}\ \delta b - \frac{\partial g}{\partial z(t_1)}\ \delta z(t_1) + \int_0^{t_1} \Big[\bar{\bar{\lambda}}^T \frac{\partial \Phi}{\partial z} \delta z + \lambda'^T \frac{\partial \xi}{\partial z} \delta z \\ &\quad - \frac{\partial L}{\partial z}\ \delta z - \frac{\partial L}{\partial \mu}\ \delta \mu - \dot{\lambda}'^T\ \bar{I} \delta \ell + \lambda'^T \frac{\partial \xi}{\partial \ell}\ \delta \ell - \frac{\partial L}{\partial \ell}\ \delta \ell \\ &\quad - \bar{\lambda}^T \frac{\partial (P(b)\dot{z})}{\partial b}\ \delta b - \bar{\lambda}^T \frac{\partial f}{\partial b}\ \delta b \Big] dt = 0 \end{aligned} \qquad (3.35)$$

Hence from Eqs. (3.25), (3.31), and (3.35), one obtains,

$$\begin{aligned} &\bar{\lambda}^T(0)\ P(b)\ \frac{\partial \nu}{\partial b}\ \delta b + \sum_{j=0}^{s-1} \lambda'_{2+4j}(0)\ \frac{\partial \bar{\nu}_{j+1}}{\partial b} \delta b - \frac{\partial g}{\partial z(t_1)}\ \delta z(t_1) \\ &\quad - \sum_{j=0}^{s-1} \frac{\partial g}{\partial \ell(t_1)_{1+4j}}\ \delta \ell(t_1)_{1+4j} + \int_0^{t_1} \Big[\Big(- \frac{\partial L}{\partial z}\ \delta z - \frac{\partial L}{\partial \mu}\ \delta \mu - \frac{\partial L}{\partial \ell}\ \delta \ell \Big) \\ &\quad - \bar{\bar{\lambda}}^T \frac{\partial \Phi}{\partial b}\ \delta b - \lambda'^T \frac{\partial \xi}{\partial b}\ \delta b - \bar{\lambda}^T \frac{\partial (P(b)\dot{z})}{\partial b}\ \delta b \\ &\quad - \bar{\lambda}^T \frac{\partial f}{\partial b}\ \delta b \Big] dt = 0 \end{aligned} \qquad (3.36)$$

Therefore,

$$\begin{aligned} &\frac{\partial g}{\partial z(t_1)}\ \delta z(t_1) + \frac{\partial g}{\partial \bar{\ell}(t_1)}\ \delta \bar{\ell}(t_1) + \int_0^{t_1} \Big[\frac{\partial L}{\partial z}\ \delta z + \frac{\partial L}{\partial \mu}\ \delta \mu + \frac{\partial L}{\partial \ell}\ \delta \ell \Big] dt \\ &\quad = \bar{\lambda}^T(0)\ P(b)\ \frac{\partial \nu}{\partial b}\ \delta b + \sum_{j=0}^{s-1} \lambda'_{2+4j}(0)\ \frac{\partial \bar{\nu}_{j+1}}{\partial b}\ \delta b \\ &\quad - \int_0^{t_1} \Big[\bar{\lambda}^T \frac{\partial (P(b)\dot{z})}{\partial b}\ \delta b + \bar{\lambda}^T \frac{\partial f}{\partial b}\ \delta b + \bar{\bar{\lambda}}^T \frac{\partial \Phi}{\partial b}\ \delta b + \lambda'^T \frac{\partial \xi}{\partial b}\ \delta b \Big] dt \end{aligned} \qquad (3.37)$$

Eq. (3.37) can now be used to eliminate explicit dependence of $\delta\psi$ on δz, $\delta\mu$, and $\delta\ell$ and the steepest descent programming method can be used to carry out sensitivity analysis and iterative optimization. Substituting from Eq. (3.37) into Eq. (3.13), one obtains

$$\delta\psi = \frac{\partial g}{\partial b}\delta b + \left[\frac{\partial g}{\partial z(0)} + \bar{\lambda}^T(0)\ P(b)\right]\frac{\partial \nu}{\partial b}\ \delta b + \sum_{j=0}^{s-1}\left[\frac{\partial g}{\partial \ell_{1+4j}}(0) + \lambda'_{2+4j}(0)\right]\frac{\partial \bar{\nu}_{j+1}}{\partial b}\delta b + \int_0^{t_1}\left[\frac{\partial L}{\partial b} - \bar{\lambda}^T\frac{\partial(p(b)\dot{z})}{\partial b} - \bar{\lambda}^T\frac{\partial f}{\partial b} - \bar{\bar{\lambda}}^T\frac{\partial \Phi}{\partial b} - \lambda'^T\frac{\partial \xi}{\partial b}\ \delta b\right]dt \qquad (3.38)$$

As stated previously, Eq. (3.38) can be applied to all ψ_β, $\beta=0,1,\ldots,r$. For convenience, define

$$\ell^{\psi_\beta}{}^T = \frac{\partial g_\beta}{\partial b} + \left[\frac{\partial g_\beta}{\partial z(0)} + \bar{\lambda}^{\psi_\beta}{}^T(0)\ P(b)\right]\frac{\partial \nu}{\partial b} + \sum_{j=0}^{s-1}\left[\frac{\partial g_\beta}{\partial \ell_{1+4j}}(0) + \lambda'^{\psi_\beta}_{2+4j}(0)\right]\frac{\partial \bar{\nu}_{j+1}}{\partial b} + \int_0^{t_1}\left[\frac{\partial L_\beta}{\partial b} - \bar{\lambda}^{\psi_\beta}{}^T\frac{\partial(P(b)\dot{z})}{\partial b} - \bar{\lambda}^{\psi_\beta}{}^T\frac{\partial f}{\partial b} - \bar{\bar{\lambda}}^{\psi_\beta}{}^T\frac{\partial \Phi}{\partial b} - \lambda'^{\psi_\beta}{}^T\frac{\partial \xi}{\partial b}\right]dt \qquad (3.39)$$

which is the design sensitivity coefficient vector of ψ_β with respect to the design parameter b. With Eq. (3.39), one can write

$$\delta\psi_0 = \ell^{\psi_0}{}^T\delta b \qquad (3.40)$$

and

$$\delta\psi_\beta = \ell^{\psi_\beta}{}^T\delta b\ , \quad \beta=1,2,\ldots,r\ . \qquad (3.41)$$

Eqs. (3.40) and (3.41) now provide $\delta\psi_0$ and $\delta\psi_\beta$ solely in terms of δb. The generalized steepest descent or gradient projection method

of [1] may now be applied to carry out constrained sensitivity analysis and iterative optimal design.

3.4 Comparison of the Corrector Equations for the Equations of Motion and the Adjoint Equations

For system Eqs. (2.33), (2,35), and (2.36), the corrector equation can be written (see Chapter II) as:

$$\begin{bmatrix} -\frac{1}{\beta_o h} P(h) + \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \mu} & \frac{\partial f}{\partial \ell} \\ \frac{\partial \Phi}{\partial z} & 0 & 0 \\ \frac{\partial \xi}{\partial z} & 0 & -\frac{1}{\beta_0 h} \bar{I} + \frac{\partial \xi}{\partial \ell} \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \mu \\ \Delta \ell \end{bmatrix} = - \begin{bmatrix} F \\ \Phi \\ \bar{\xi} \end{bmatrix} \tag{3.42}$$

In a similar manner, the corrector equations for the adjoint Eqs. (3.32), (3.33), and (3.34) can be written as:

$$\begin{bmatrix} \frac{1}{\beta_0 h} P^T + \frac{\partial f}{\partial z}^T & \left(\frac{\partial \Phi}{\partial z}\right)^T & \left(\frac{\partial \xi}{\partial z}\right)^T \\ \frac{\partial f}{\partial \mu}^T & \underset{\sim}{0} & \underset{\sim}{0} \\ \frac{\partial f}{\partial \ell}^T & \underset{\sim}{0} & \frac{1}{\beta_0 h} \bar{I}^T + \frac{\partial \xi}{\partial \ell}^T \end{bmatrix} \begin{bmatrix} \Delta \bar{\lambda} \\ \Delta \bar{\bar{\lambda}} \\ \Delta \lambda' \end{bmatrix} = - \begin{bmatrix} F' \\ \Phi' \\ \bar{\xi}' \end{bmatrix} \tag{3.43}$$

where the following relations have been used:

$$\frac{\partial F}{\partial z} = \frac{\partial f}{\partial z}$$

$$\frac{\partial F}{\partial \ell} = \frac{\partial f}{\partial \ell}$$

$$\frac{\partial F}{\partial \mu} = \frac{\partial f}{\partial \mu} \tag{3.44}$$

From Eqs. (3.42) and (3.43) it is clear that the coefficient matrix of Eq. (3.43) is the transpose of the coefficient matrix of Eq. (3.42), except for the minus sign before $\frac{1}{\beta_0 h}$ in the diagonal terms of Eq. (3.42). However, since the adjoint equations are integrated backwards in time, the negative time-step will make the second matrix exactly the transpose of the first. Thus, essential computational advantages accrue.

3.5 The Solution of the Adjoint Equations

The system of adjoint equations (3.32), (3.33), and (3.34) are to be solved with the terminal conditions (3.29). With the substitution

$$t' = t_1 - t \tag{3.45}$$

the adjoint equations become

$$F_1' \equiv P^T \frac{d\bar{\lambda}}{dt'} + \frac{\partial f^T}{\partial z} \bar{\lambda} + \frac{\partial \Phi^T}{\partial z} \bar{\bar{\lambda}} + \frac{\partial \xi^T}{\partial z} \lambda' - \frac{\partial L^T}{\partial z} = 0 \tag{3.46}$$

$$\Phi_1' \equiv \frac{\partial f^T}{\partial \mu} \bar{\lambda} - \frac{\partial L^T}{\partial \mu} = 0 \tag{3.47}$$

$$\bar{\xi}_1' \equiv \bar{I}^T \frac{d\lambda'}{dt'} + \frac{\partial f^T}{\partial \ell} \bar{\lambda} + \frac{\partial \xi^T}{\partial \ell} \lambda' - \frac{\partial L^T}{\partial \ell} = 0 \tag{3.48}$$

with the initial conditions:

$$\left.\begin{aligned} \bar{\lambda}^{-T}(0) &= \frac{\partial g}{\partial z(t_1)}\, P(b)^{-1} \\ \lambda'_{2+4j}(0) &= \frac{\partial g}{\partial \ell_{1+4j}(t_1)} \quad , \quad j=0,1,\ldots,(s-1) \end{aligned}\right\} \tag{3.49}$$

Then the corrector coefficient matrix becomes exactly the transpose of the original coefficient matrix. To make full use of the results obtained in the process of solution of the original set of equations of motion, the time instants (indexed), Gear constant β_0, step-size, the solution variables, and the coefficient matrix elements are stored in a direct access disk at each time grid. The matrix elements or the G-vector of ADAMS 2-D [14] are originally arranged with column-wise representation.

To make use of the same subroutine as in the SPARSE-MATRIX package for LU factorization of the transposed coefficient matrix, some modifications have been made in the subroutine S08000 of ADAMS 2-D [14] to represent the transposed matrix in a column-wise manner. It is then stored in the direct access device. It is to be noted, however, that the time grid of the backward integration for the solution of the adjoint equations will not, in general, coincide with the original grid. So interpolation of the original solution variables and the second components of the Nordsieck vector [6,2] is required for calculation of the right-hand side of the corrector equations of the adjoint set and the integrands of the sensitivity

matrices. This is done by a subroutine INTERP built in the subroutine DYNANL of ADAMS 2-D (also see Chapter V). SOINEW is the subroutine for calculation of the right-hand side of the new corrector equations, which is built into S01000 of ADAMS 2-D.

The G-vector, however, is not interpolated. It is approximated by its value at the nearest original time grid point. It can actually be kept unaltered for several small time steps; in particular, during the CNTRLT and Newton iteration operations in DIFSUB of ADAMS 2-D. Finally, ADJONT is a version of ADAMS 2-D that solves the adjoint equations, with all above considerations built in. Chapter V deals with the subroutines in more detail.

3.6 Static Sensitivity Analysis for the Solution Variables

From Eqs. (3.39), (3.40), and (3.41) it is observed that for the calculation of sensitivity coefficients one needs derivatives of the initial values of some solution variables with respect to the design parameters. These are obtained as follows.

For static equilibrium the system Eqs. (2.33), (2.35), and (2.36) reduce to the following set of algebraic equations:

$$f(z,\ell,\mu,b) = 0 \tag{3.50}$$

$$\Phi(z,b) = 0 \tag{3.51}$$

$$\xi(z,\ell,b) = 0 \tag{3.52}$$

Differentiating these equations with respect to b one has

$$\frac{\partial f}{\partial z}\frac{\partial z}{\partial b} + \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial b} + \frac{\partial f}{\partial \ell}\frac{\partial \ell}{\partial b} = -\frac{\partial f}{\partial b} \tag{3.53}$$

$$\frac{\partial \Phi}{\partial z}\frac{\partial z}{\partial b} = -\frac{\partial \Phi}{\partial b} \tag{3.54}$$

$$\frac{\partial \xi}{\partial z}\frac{\partial z}{\partial b} + \frac{\partial \xi}{\partial \ell}\frac{\partial \ell}{\partial b} = -\frac{\partial \xi}{\partial b} \tag{3.55}$$

Eqs. (3.53), (3.54), and (3.55) represent a set of $(3n+2m+4s)$ linear equations for $\frac{\partial z}{\partial b}$, $\frac{\partial \mu}{\partial b}$, and $\frac{\partial \ell}{\partial b}$. In matrix form these equations are equivalent to

$$J_1 \frac{\partial \nu}{\partial b} = \frac{\partial F}{\partial b} \tag{3.56}$$

where (using Eq. (3.42)),

$$J_1 \equiv \begin{bmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \mu} & \frac{\partial f}{\partial \ell} \\ \frac{\partial \Phi}{\partial z} & 0 & 0 \\ \frac{\partial \xi}{\partial z} & 0 & \frac{\partial \xi}{\partial \ell} \end{bmatrix} \tag{3.57}$$

is the Jacobian matrix of the static analysis and

$$\frac{\partial \nu}{\partial b} \equiv \left[\frac{\partial z}{\partial b}, \frac{\partial \mu}{\partial b}, \frac{\partial \ell}{\partial b}\right]^T \tag{3.58}$$

$$\frac{\partial F}{\partial b} \equiv \left[-\frac{\partial f}{\partial b}, -\frac{\partial \Phi}{\partial b}, -\frac{\partial \xi}{\partial b}\right]^T \tag{3.59}$$

The sparse matrix codes generated during static analysis are utilized to solve for $\frac{\partial \nu}{\partial b}$, with the help of the subroutine DIFSUB. The static sensitivity analysis results for a spring-reset plowshare mechanism (see Chapter VI) are given in Tables 6.7 and 6.8 in Chapter VI.

CHAPTER IV

OPTIMAL DESIGN ALGORITHM

4.1 Steepest Descent Method with Constraint Error Compensation

The general optimal design problem formulated in Chapter III can be solved by the generalized steepest descent programming technique presented in Ref. [1]. Here the technique will only be discussed very briefly.

After the sensitivity analysis of Chapter III, the problem is reduced to finding a vector δb that minimizes $\delta\psi_0$ and corrects constraint violations. For this purpose, the following definitions are made. Define a set of indices

$$A = \{S | \psi_S + \varepsilon \geq 0\} \tag{4.1}$$

and a column vector of ε-active elements of ψ_S,

$$\tilde{\psi} = \begin{bmatrix} \psi_S \\ S \varepsilon A \end{bmatrix} \tag{4.2}$$

In order to assure the satisfaction of the constraints, the violations will be corrected by demanding that

$$\delta\psi \leq \Delta\psi_S \tag{4.3}$$

where $S \varepsilon A$ and $\Delta\psi_S$ is a desired change in the constraint function ψ_S. Generally, $\Delta\psi_S$ is chosen to be $-\psi_S$.

In the notations of Chapter III, the problem reduces to

finding δb to minimize

$$\delta\psi_0 = \ell^{\psi_0}{}^T \delta b \tag{4.4}$$

subject to the constraints

$$\delta\tilde{\psi} = \ell^{\tilde{\psi}}{}^T \delta b \leq \Delta\tilde{\psi} \tag{4.5}$$

and the quadratic step size constraint

$$\delta b^T W \delta b \leq \eta^2 \tag{4.6}$$

where η is a small number and W is a positive definite weighting matrix.

The Kuhn-Tucker necessary conditions of nonlinear programming may now be applied to solve this reduced problem. According to the Kuhn-Tucker Theorem, it is necessary that there exist a scalar multiplier $\gamma_0 \geq 0$ and a vector of multipliers, $\gamma = \{\gamma_S, S \in A\}$, $\gamma_S \geq 0$ for inequality constraints, such that

$$\gamma^T(\ell^{\tilde{\psi}}{}^T \delta b - \Delta\tilde{\psi}) = 0 \tag{4.7}$$

and

$$\frac{\partial G}{\partial(\delta b)} = 0 \tag{4.8}$$

where

$$G = \ell^{\psi_0}{}^T \delta b + \gamma^T(\ell^{\tilde{\psi}}{}^T \delta b - \Delta\tilde{\psi}) + \gamma_0(\delta b^T W \delta b - \eta^2) \tag{4.9}$$

Equations (4.8) and (4.9) give

$$\ell^{\psi_0} + \ell^{\tilde{\psi}}\gamma + 2\gamma_0 W\delta b = 0 \tag{4.10}$$

Solving for δb, one gets from Eq. (4.10)

$$\delta b = - \frac{1}{2\gamma_0} \; W^{-1}(\ell^{\psi 0} + \ell^{\tilde{\psi}} \gamma) \tag{4.11}$$

where $\frac{1}{2\gamma_0} > 0$ is to be chosen as a step-size.

In order to satisfy condition (4.7), it is first assumed that

$$\delta\tilde{\psi} - \Delta\tilde{\psi} = 0 \tag{4.12}$$

Substituting Eqs.(4.11) and (4.12) into Eq. (4.5) and putting

$$M_{\psi\psi} = \ell^{\tilde{\psi}^T} \; W^{-1} \; \ell^{\tilde{\psi}} \tag{4.13}$$

one obtains

$$\gamma = - M_{\psi\psi}^{-1} \; (\ell^{\tilde{\psi}^T} \; W^{-1} \; \ell^{\psi 0} + 2\gamma_0 \; \Delta\tilde{\psi}) \tag{4.14}$$

In order to avoid computation of $M_{\psi\psi}^{-1}$, define vectors γ^1 and γ^2 by the linear equations

$$M_{\psi\psi}\gamma^1 = - \ell^{\tilde{\psi}^T} \; W^{-1} \; \ell^{\psi 0} \tag{4.15}$$

$$M_{\psi\psi}\gamma^2 = - \Delta\tilde{\psi} \tag{4.16}$$

When γ^1 and γ^2 are solved from Eqs. (4.15) and (4.16), γ is given by the relation

$$\gamma = \gamma^1 + 2\gamma_0 \; \gamma^2 \tag{4.17}$$

One must now check the algebraic signs of the multipliers γ_S. If all $\gamma_S \geq 0$, then the assumption (4.12) is admissible. On the other hand, if some γ_S is negative for some $S \in A$, then the

corresponding constraint $\delta\psi_S - \Delta\psi_S \leq 0$ should have been strictly satisfied. Therefore, ψ_S should be removed from $\tilde{\psi}$. Then a new reduced set $\tilde{\psi}$ is formed and the multipliers of this new set are re-calculated.

From Eqs. (4.17) and (4.11) one obtains

$$\delta b = -\frac{1}{2\gamma_0} W^{-1}(\ell^{\psi_0} + \ell^{\tilde{\psi}}\ \gamma^1) - W^{-1}\ \ell^{\tilde{\psi}}\ \gamma^2 \tag{4.18}$$

Defining

$$\delta b^1 = W^{-1}(\ell^{\psi_0} + \ell^{\tilde{\psi}}\ \gamma^1) \tag{4.19}$$

$$\delta b^2 = -W^{-1}\ \ell^{\tilde{\psi}}\ \gamma^2 \tag{4.20}$$

the change δb of Eq. (4.18) can be written as

$$\delta b = -\frac{1}{2\gamma_0}\delta b^1 + \delta b^2 \tag{4.21}$$

In practice, instead of choosing η in Eq. (4.6), γ_0 is chosen directly to give the step-size $\frac{1}{2\gamma_0}$.

The following relations can be easily shown to follow as a necessary consequence from the above equations:

$$\delta b^{1^T}\ W\delta b^2 = 0 \tag{4.22}$$

$$\ell^{\tilde{\psi}^T}\ \delta b^2 = \Delta\tilde{\psi} \tag{4.23}$$

$$\ell^{\tilde{\psi}^T}\ \delta b^1 = 0 \tag{4.24}$$

$$-\ \ell^{\psi_0{}^T}\ \delta b^1 \leq 0 \tag{4.25}$$

From these relations it can be observed that:

(1) δb^1 and δb^2 are orthogonal,

(2) δb^2 provides the desired constraint correction,

(3) δb^1 has no effect on the constraint functions in $\tilde{\psi}$,

and (4) δb^1 provides a reduction in ψ_0.

A convergence criterion for this method is defined and proved in Ref. [1]. This requires that as the optimal solution is approached, δb^1 converges to 0.

The choice of step-size $\frac{1}{2\gamma_0}$ is of extreme importance. It has been observed that a very small step will result in slow convergence, while a large step may cause oscillation about the minimum point, or even divergence. Generally, the step-size is determined by assuming that the initial estimate is in the feasible region, so that $\tilde{\psi} = 0$ and a fixed percentage reduction in the cost function ψ_0 is sought. If the desired change in ψ_0 is chosen to be $\Delta\psi_0 < 0$, then from Eqs. (4.4), (4.18), and (4.19) one obtains

$$\frac{1}{2\gamma_0} = -\frac{\delta\psi_0}{{}_{\ell}\psi_0{}^T \delta b^1} = \frac{-\Delta\psi_0}{{}_{\ell}\psi_0{}^T W^{-1}{}_{\ell}\psi_0} \tag{4.26}$$

This choice is however, made at the beginning of the iterative process and subsequently the step-size is adjusted by multiplying the previous step-size by a factor to speed convergence or to avoid oscillation.

4.2 Optimal Design Algorithm

Application of the sensitivity analysis of Chapter III and the generalized steepest descent programming technique of the previous

section gives the following optimal design algorithm:

Step 1: Estimate the optimum design parameter vector $b^{(j)}$ and solve the state equations of motion (2.33), (2.35), and (2.36) for $z^{(j)}(t)$, $\mu^{(j)}(t)$, and $\ell^{(j)}(t)$ corresponding to $b^{(j)}$ with SPARSE MATRIX and STIFF INTEGRATION (GEAR) algorithms implemented through the ADAMS Program and at each time step store the solution variables, indexed time-instants, time step-size, and corrector coefficient matrix in a direct access disk.

Step 2: Check constraints of Eqs. (3.7), (3.8), and (3.10) and form the vector of constraint functions $\tilde{\psi}$, consisting of ψ_i such that $\psi_i \geq -\varepsilon$, where $\varepsilon > 0$ is small. Also choose the the constraint error correction vector $\Delta\tilde{\psi} = -\tilde{\psi}$.

Step 3: Implement a modified ADAMS program to solve the adjoint differential equations (3.46), (3.47), and (3.48) with initial conditions (3.49), to obtain $\lambda^{\psi_0}(t)$ and $\lambda^{\psi_\beta}(t)$, ψ_β corresponding to ε-active constraint functions.

Step 4: Compute

$$\ell^{\psi_0} = \frac{\partial g_0}{\partial b} + \left[\frac{\partial g_0}{\partial z(0)} + \bar{\lambda}^{\psi_0 T}(0)P(b)\right]\frac{\partial \nu}{\partial b} + \sum_{j=0}^{s-1}\left[\frac{\partial g_0}{\partial \ell_{1+4j}(0)} + \lambda'^{\psi_0}_{2+4j}(0)\right]\frac{\partial \nu_j}{\partial b} + \int_0^{t_1}\left[\frac{\partial L_0}{\partial b} - \bar{\lambda}^{\psi_0 T}\frac{\partial (P(b)\dot{z})}{\partial b} - \bar{\lambda}^{\psi_0 T}\frac{\partial f}{\partial b} - \bar{\lambda}^{\psi_0 T}\frac{\partial \Phi}{\partial b} - \lambda'^{\psi_0 T}\frac{\partial \xi}{\partial b}\right]dt \qquad (4.27)$$

$$\ell^{\psi_\beta} = \frac{\partial g_\beta}{\partial b} + \left[\frac{\partial g_\beta}{\partial z(0)} + \bar{\lambda}^{\psi_\beta}{}^T(0)P(b)\right]\frac{\partial \nu}{\partial b} + \sum_{j=0}^{s-1}\left[\frac{\partial g_\beta}{-\partial \ell_{1+4j}(0)}\right.$$

$$\left. + \lambda'^{\psi_\beta}{}^T_{2+4j}(0)\right]\frac{\partial \nu_j}{\partial b} + \int_0^{t_1}\left[\frac{\partial L_\beta}{\partial b} - \bar{\lambda}^{\psi_\beta}{}^T\frac{\partial (P(b)\dot{z})}{\partial b}\right.$$

$$\left. - \bar{\bar{\lambda}}^{\psi_\beta}{}^T\frac{\partial f}{\partial b} - \bar{\bar{\bar{\lambda}}}^{\psi_\beta}{}^T\frac{\partial \phi}{\partial b} - \lambda'^{\psi_\beta}{}^T\frac{\partial \xi}{\partial b}\right]dt \qquad (4.28)$$

and

$$M_{\psi\psi} = \begin{cases} \ell^{\psi}{}^T W^{-1}\ell^{\psi}, & \text{if } \tilde{\psi} \text{ is not empty} \\ 1, & \text{if } \tilde{\psi} \text{ is empty} \end{cases} \qquad (4.29)$$

where ℓ^{ψ} is the $(\rho \times \tilde{\beta})$ matrix constituted of the vectors ℓ^{ψ_β}, ρ is the number of design parameters and $\tilde{\beta}$ is the number of violated constraints, and W is a weighting matrix [1].

Step 5: Choose step-size $\frac{1}{2\gamma_0} > 0$ and calculate the Lagrange multiplier vector γ from

$$M_{\psi\psi}\,\gamma^1 = -\,\ell^{\psi}{}^T W^{-1}\,\ell^{\psi 0} \qquad (4.30)$$

$$M_{\psi\psi}\,\gamma^2 = -\,\Delta\tilde{\psi} \qquad (4.31)$$

and

$$\gamma = \gamma^1 + 2\gamma_0\gamma^2 \qquad (4.32)$$

Step 6: Check the algebraic sign of each component of γ associated with inequality constraints. If any components of γ are negative, redefine $\tilde{\psi}$ by removing the corresponding terms from $\tilde{\psi}$ and return to Step 4.

Step 7: Compute δb^1, δb^2, and δb from

$$\delta b^1 = W^{-1}(\ell^{\psi_0} + \ell^{\psi_\gamma 1}) \tag{4.33}$$

$$\delta b^2 = - W^{-1}\, \ell^{\psi_\gamma 2} \tag{4.34}$$

and

$$\delta b = - \frac{1}{2\gamma_0}\, \delta b^1 + \delta b^2 \tag{4.35}$$

δb gives a design improvement.

Step 8: Compute $b^{(j+1)} = b^{(j)} + \delta b$ (4.36)

Step 9: If the constraints are satisfied and $||\delta b^1||$ is sufficiently small, terminate the process. Otherwise, return to Step 1 with $b^{(j+1)}$ as the best estimate of the design.

CHAPTER V

THE DYNAMIC ANALYSIS AND DESIGN SYSTEM (DADS) PROGRAM

5.1 Introduction

The Dynamic Analysis and Design System (DADS) computer program is developed to carry out the dynamic analysis, design sensitivity analysis, and optimal design formulations described in Chapters II, III, and IV for general nonlinear mechanical systems. Provisions for regenerating sparse-matrix codes at necessary time instants of dynamic and adjoint analyses have been made so that the program can handle systems with intermittent motions (see Plow-share mechanism in Chapter VI (Section 6.3)) with sufficient ease.

The DADS program executes in two main phases: (1) The dynamic analysis phase and (2) The sensitivity analysis and optimization phase. The dynamic analysis phase of the program generates the sparse matrix code for pivoting and LU factorization of the Jacobian matrix and solves the differential-algebraic equations for the state variables during a specified time interval. It employs sparse matrix codes and the numerical integration routine presented in Chapters II and III. It then stores the Jacobian matrix and state variables on a direct access disk for use in the sensitivity analysis phase. The dynamic analysis phase also performs static sensitivity analysis for the solution variables (see Section 3.6).

The sensitivity analysis and optimization phase of the program

incorporates the techniques of Chapter IV into various subprograms and solves adjoint equations (see Section 3.3) using the data stored on disk. The same numerical integration subroutine is used to solve the adjoint equations. The program further computes sensitivity coefficients and the necessary design improvements. The process is repeated until optimum results are achieved.

5.2 Main Features of DADS Computer Program

The DADS program has the following features:

(1) The mechanical systems treated are discrete, nonlinear, and constrainted (two-dimensional at present);

(2) General nodal formulation of equations of motion for the bodies [7];

(3) Necessary data being given, formulation of the equations of motion and the Jacobian matrix is automatic;

(4) Integration of a combined set of differential and algebraic equations (DAE) is performed;

(5) The following three algorithms are used:

  (a) Newton iteration;

  (b) Sparse Matrix techniques for the solution of the linearized simultaneous equations;

  (c) Stiff Integration Algorithm (GEAR) for numerical integration.

(6) Types of analyses performed:

  (a) Static analysis;

  (b) Transient analysis;

(c) Static sensitivity analysis for solution variables;

(d) General design sensitivity analysis;

(e) Optimal design.

The static and transient analysis part of the program was mainly developed by Orlandea [7] and Wehage [14]. The corresponding references are recommended for further details. Fig. 5.1 shows an outline of DADS program capabilities, Fig. 5.2 shows the main subroutines used in the dynamic analysis phase of DADS, and Fig. 5.3 identifies the main subprograms used in the design sensitivity analysis and optimization phase, and Fig. 5.4 is a flow diagram of the DADS computer program.

5.3 Brief Description of the Dynamic Analysis Phase

The dynamic analysis phase (DYNANL) of the DADS program involves establishing the sparse matrix code description of the mechanical system and solving the differential and algebraic equations for the state variables. As shown in Fig. 5.2, this involves two major steps: (i) generation of an initial sparse matrix code (including pivoting and LU factorization code) and (ii) repetitive solution of linearized equations for the state variables during the time interval of interest.

In the first step of dynamic analysis, estimates of the initial configuration of the system are provided by INDATA and are used by VARSET to initialize a state variable vector for subsequent use by the numerical integration routine DIFSUB. A compact numbering

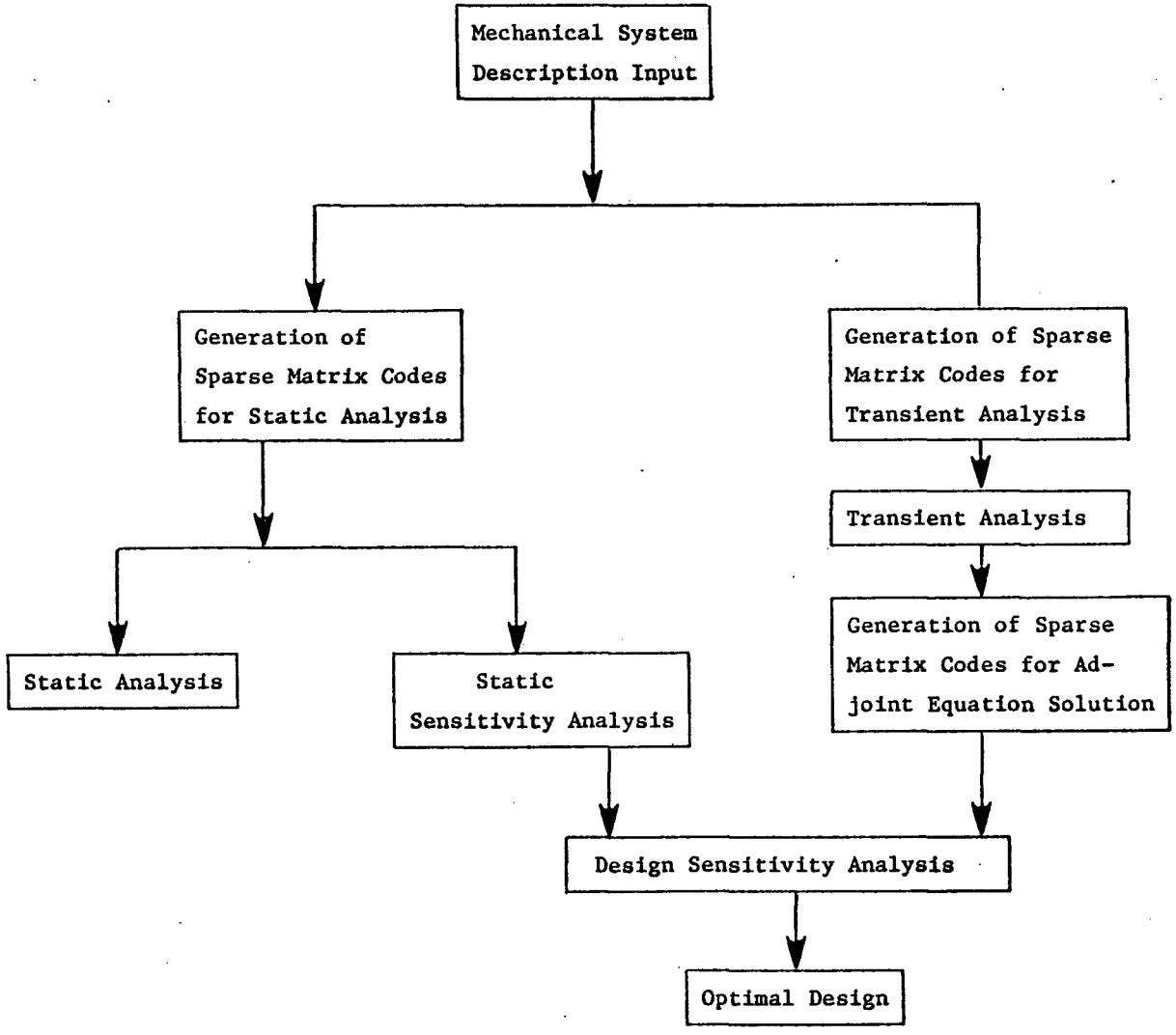


Figure 5.1 Outline of DADS Program Capabilities.

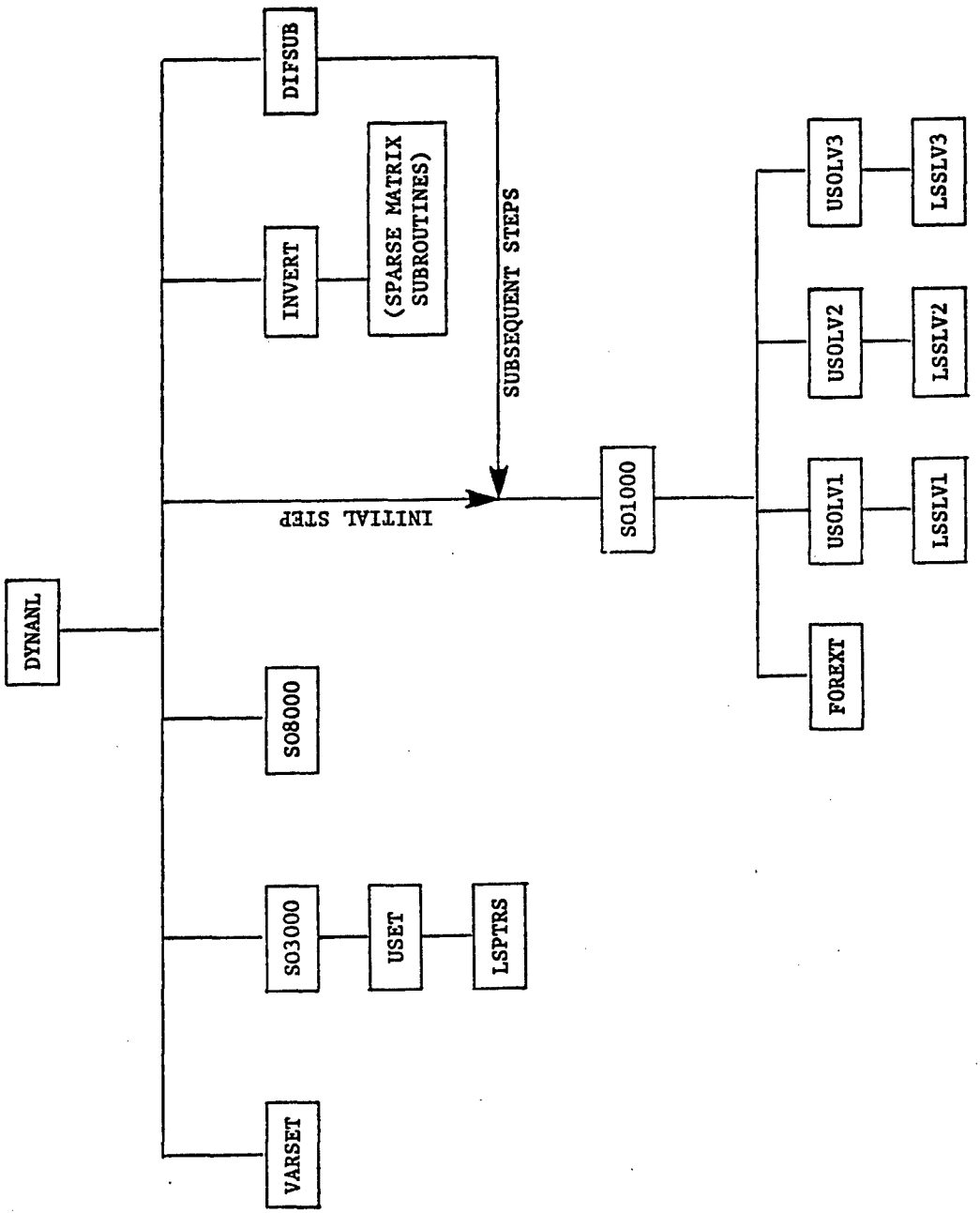


Figure 5.2 DADS - Dynamic Analysis Phase.

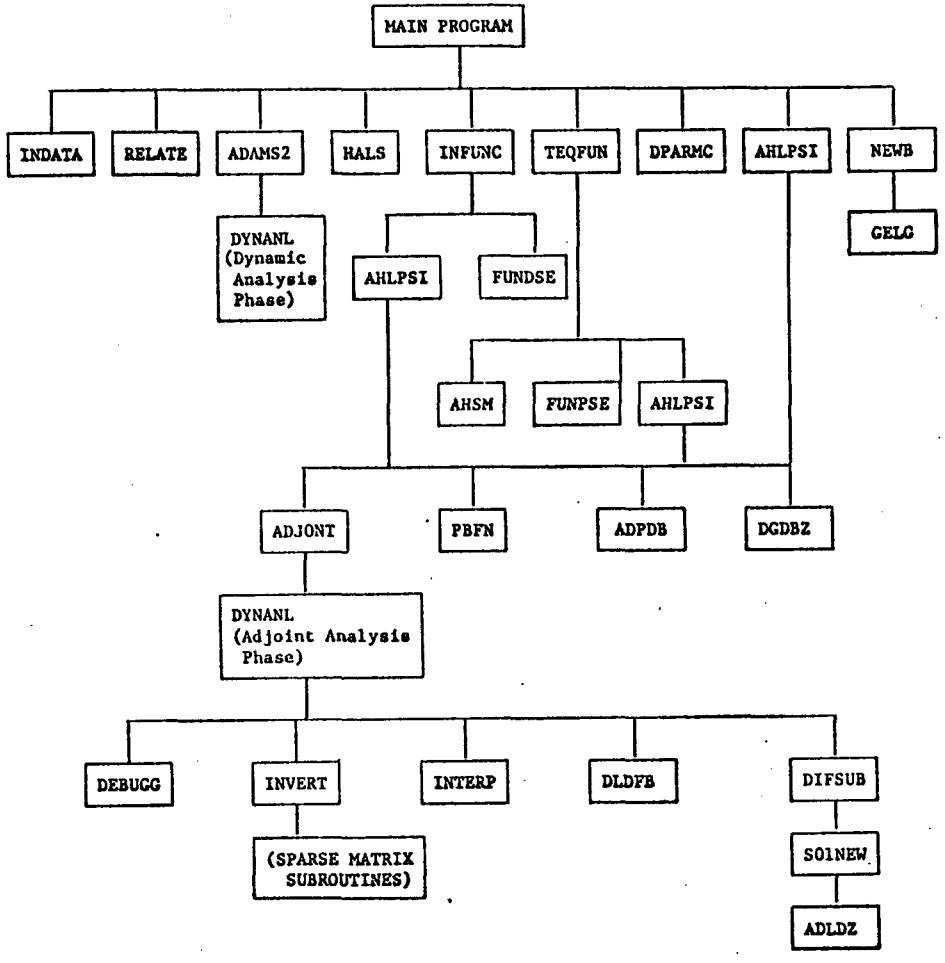


Figure 5.3 DADS - Sensitivity Analysis and Optimization Phase.

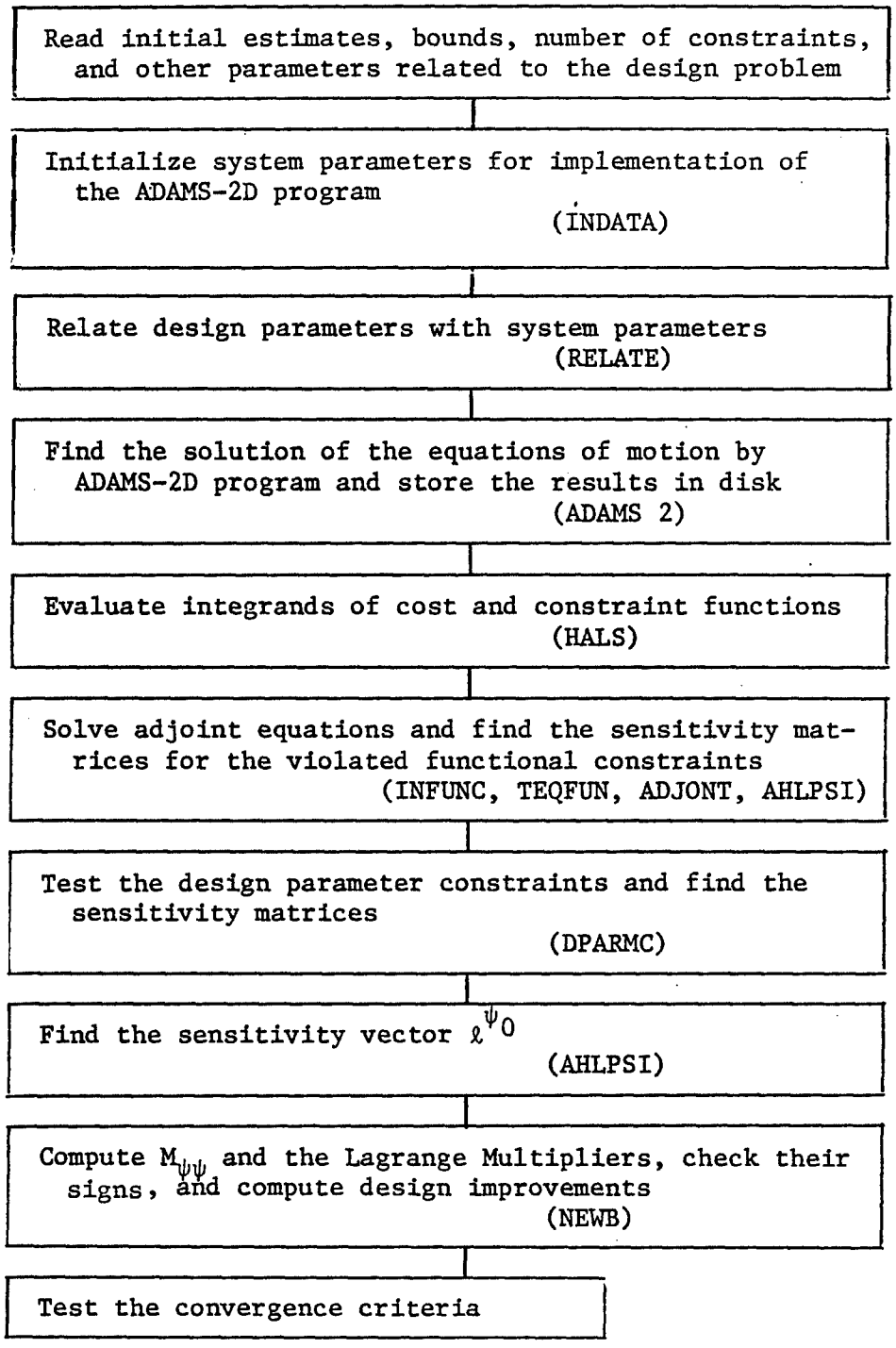


Figure 5.4. Flow Diagram of the DADS Computer Program.

system identifying bodies, joints, and spring-damper elements is used to input data through VARSET and provides the necessary description of the mechanical system configuration. These data are used to construct the Newton corrector state equation (3.42) and the system of adjoint equations (3.43) (for use in the sensitivity analysis phase). This information is used in S03000 to generate initial vectors of row and column indices for nonzero entries in the Jacobian matrix and to assemble the standard equations that are required on the right hand side of Eq. (3.42). Similarly, descriptions of user-supplied row-column positions of additional nonstandard elements are provided by incorporating the necessary code in USET. A symbolic description of the resultant matrix is printed by DEBUGG for reference purposes and a column-ordering permutation vector is generated in S08000.

Subroutine S01000 evaluates the Jacobian matrix and right hand side of Eq. (3.42). Its purpose is to (i) evaluate force and displacement functions of time that are provided by the user through subroutine FOREXT, (ii) transfer the state variables from a single vector used by the numerical integration routine to the standard variables (and user-supplied variables through USOLV1), (iii) evaluate that part of the Jacobian matrix that is associated with the standard equations (and user-supplied equations, USO1V2), using updated variables from step (ii) and the previously generated column-ordering permutation vector (S08000), and (iv) evaluate the standard equations

(and user-supplied equations, USOLV3). Subroutines LSPTRS, LSSLV1, LSSLV2, and LSSLV3 are included as user-supplied routines to incorporate nonstandard highly nonlinear springs and dampers, as described in the plow share mechanism example of Chapter VI. Finally a sparse LU factored description of the matrix is generated in subroutine INVERT.

The second step of dynamic analysis is then to numerically integrate the system of equations during the time interval of interest. This is accomplished by the numerical integration subroutine DIFSUB, which repeatedly calls SO1000 to update the Jacobian matrix and system of equations as it executes the iterative corrector formula of Eq. (3.42). The Jacobian matrix and the results of numerical integration are stored in a direct access disk.

In sensitivity analysis, DYNANL is called in ADJONT (adjoint analysis phase) (Fig. 5.4), which is again called in AHLPSI, where the sensitivity coefficients are calculated. For this case, DYNANL reads stored data from the disk and passes through two distinct steps. In the first step, the sparse matrix codes for pivoting and LU factorization are generated for the transpose of the Jacobian matrix, through INVERT. In the second step, the system of adjoint equations (3.32) to (3.34) are numerically integrated through DIFSUB, which repeatedly reads data from the disk, calls SO1NEW for the evaluation of the right-hand sides of the adjoint system, solves the adjoint variables at various time steps, and stores the solution results on a disk for use in the calculation of sensitivity coefficients.

5.4 Description of the DADS Program

As noted in Section 5.1, the computer program DADS is constituted of two main phases: (i) Dynamic Analysis phase, and, (ii) design sensitivity analysis and optimization phase. The main flow of the program is described below (see also Fig. 5.4).

In the main program, the initial estimates and bounds on design parameters, number of constraints, stepsize, percentage of cost function reduction, and other parameters related to the design problem (see Chapters III and IV) are read. Then system parameters, such as masses, moments of inertia, locations of centers of masses, applied constant forces, joint types, and spring-damper parameters are read, through the subroutine INDATA. The subroutine RELATE relates the variables and parameters of the dynamic analysis (ADAMS 2 and DYNANL; see also Section 2.1 of Chapter II) to the updated design parameters (Chapters III and IV).

For dynamic analysis, DYNANL is called through ADAMS2. The Jacobian matrix, solution variables, time step, time instant, and order of numerical integration are then stored in a direct access disk. They are first used to evaluate the integrands of the cost and constraint functionals of Eqs. (3.1), (3.6), (3.7), and (3.11), through subroutine HALS.

Next, the inequality functional constraints of Eqs. (3.7) and (3.10) are tested and the corresponding adjoint equations (3.32) to (3.34) are solved through INFUNC, TEQFUN, AHLPSI, and ADJONT. At this stage, DYNANL (Adjoint Analysis Phase) is called by subroutine

ADJONT, The matrix of sensitivity coefficients $\ell^{\psi\beta}$ (see Eq. (3.41)) is calculated for the violated functional constraints through AHLPSI.

The subroutine DPARMC then tests the design parameter constraints and calculates the corresponding design sensitivity vectors. The sensitivity vector $\ell^{\psi 0}$ of Eq. (3.40) is then calculated through AHLPSI. The subroutine NEWB then computes the matrix $M_{\psi\psi}$ of Eq. (4.29), solves Eqs. (4.30) and (4.31) for γ^1 and γ^2, and computes the design change δb given by Eqs. (4.33) through (4.35).

At this stage, convergence criteria are tested. If they are satisfied, the new design is taken as the optimum one. Otherwise, the process is repeated with the new design paramters used as the initial design estimate.

In the following sections, descriptions of the principal program variables and subroutines are presented.

5.4.1 Principal Variables

a) Dynamic Analysis

TIMPU - current time

NB - number of bodies including fixed body 1

NJ - number of joints

NSD - number of spring-damper pairs

NSDV - twice the number of nonlinear spring-damper pairs

NSDT - number of torsional spring-damper pairs

EPS - maximum one-step error in numerical integration

M - vector of masses of the bodies

JIN - vector of moments of inertia

X, Y - coordinates of the CM's with respect to inertial reference frame

PHI - angular displacements of X-axis of bodies with respect to inertial refernece frame X-axis

FX, FY - components of the force applied at CM parallel to the inertial reference frame axes

TQ - applied torques

JT - type of the joints

IB(1,I),IB(2,I) - numbers given to the two neighboring bodies connected by the i^{th} joint

X1, Y1 - co-ordinates of the revolute joint or a point on the axis of the translational joint with respect to the axes of the first body

X2, Y2 - same as above with respect to the 2^{nd} body

IBSD(1,I),IBSD(2,I) - numbers given to the two bodies connected by the i^{th} spring-damper pair

XF1,YF1 - co-ordinates of the point of attachment on the first body with respect to the body fixed axes

XF2,YF2 - same as above with respect to the 2^{nd} body

SK - spring constants

DC - damping coefficients

SDL - deformed spring lengths

SDLO - undeformed spring lengths

SDF - constant forces applied along spring-damper pairs

IBSDT(1,I),IBSDT(2,I) - numbers given to the two bodies connected by the i^{th} torsional spring-damper pairs

SKT - torsional spring constants

DCT - torsional damping coefficients

PHIO - initial difference of the angular displacements of the bodies for the undeformed torsional spring-dampers

TQO - constant couples applied at the torsional spring-damper pairs

V - the time-derivatives of SDL's

FFX,FFY - components of the spring-damper forces with respect to inertial reference frame

TQS - torsional spring-damper couples

UX,UY - components of the velocities of the CM's with respect to inertial reference frame

UP - angular velocities of the bodies with respect to inertial reference frame

LM - Lagrange multipliers of the dynamic analysis

G - vector of column-ordered nonzero elements in the Jacobian matrix

CL - right-hand sides of the corrector equations

JSIZ - size of the Jacobian matrix

NPOS - pointer to consecutive nonzero positions in the Jacobian matrix.

b) Sensitivity Analysis and Optimization Phase

The principal variables occurring in the program, over and above those mentioned above are described below.

NB1 - number of design parameters (total)

NAB - number of artificial design parameters

MS - expected size of constraint set

NFCE - number of pure equality functional constraints

NFCET - number of transformed equality functional constraints

NFCI - number of inequality functional constraints

NDCI - number of design parameter constaints

NP1 - number of time grids in stiff integration in ADAMS 2

B - design parameters

BI - starting values of the design parameters

BL - lower bounds of the design parameters

BU - upper bounds of the design parameters

COSTJ - cost function

EPE - tolerance for equality functional constraints

EPI - tolerance for inequality functional constraints

EPDI - tolerance for design parameter constraints

ER - convergence criterion

ILIM - limit of iterations in optimization program step

W - weighted matrix (generally diagonal)

NP2 - number of time grids in stiff integration in ADJONT

ALM - adjoint variable

G1 - vector of column-ordered nonzero elements in the transposed Jacobian matrix

MPOS - a vector playing the same role in adjoint analysis as NPOS in dynamic analysis (see Chapter II)

NPOSRR - a copy of original NPOSR

JP1A - a copy of JP1

} (see Chapter II)

ALS - the integrands of the cost and constraint functionals

GS - non-integral parts of the cost and constraint functional

AHSMT - maxima of the integrands of the transformed equality constraints

DLDZ - derivatives of the integrands of the cost and constraints functionals with respect to the state variables (both primary and secondary)

HLJ - sensitivity coefficients for the cost functional

HLPSI - sensitivity coefficients for the constraints functionals

PSIC - constraint functionals

NCV - number of constraint violations

NCVID - indices of the violated constraints

Some other variables are defined during the description of the subprograms in the next section.

5.4.2 Description of the Subprograms

In this subsection a brief description of each of the main subprograms of DADS is given and some of the call-list variables not appearing in the previous subsection are explained briefly.

SUBROUTINE INDATA (IECHO, ALM)

The subroutine INDATA reads input data that characterizes the mechanical system. It also reads initial values of adjoint variables and the maximum values of nonlinear spring-damper pairs into the program.

IECHO - a logic variable for writing the data on paper

= 0, data is not written,

= 1, data is written.

SUBROUTINE RELATE (YY, IMODE)

The subroutine RELATE relates the parameters and variables of the

dynamic analysis to the updated design parameters.

YY - an array which contains the dependent variables (dynamic analysis) and their derivatives. Its dimension should be at least (JSIZ,7) where JSIZ is the Jacobian size.

IMODE - a logic variable characterizing the type of analysis performed,

= 0, neither static nor dynamic analysis,

= 1, static analysis,

= 2, dynamic analysis.

SUBROUTINE ADAMS2(IFLAG1, SOLNO, JSIZ, DLDB, DFDB, DPHDB, DZIDB, YY, PB, ALM, DNDB)

The subroutine ADAMS2 feeds in some flags, time step, maximum and minimum time steps, and calls subroutine DYNANL to initiate tranient solution.

IFLAG1 - a logic variable,

= 0, for dynamic analysis,

= 1, for adjoint analysis.

SOLNO - an integer keeping a running acount of the solution numbers at different time steps during dynamic analysis.

DLDB - derivatives of the integrands of the cost and constraint functionals with respect to b (design parameters).

DFDB, DPHDB, DZIDB - derivatives of the equations of motion, constraint equations and spring-damper relations with respect to b.

PB - elements of $\underset{\sim}{P}$ matrix in equations of motion.

DNDB - derivatives of the initial solution variables with respect to b, $\left(\frac{\partial \underline{v}}{\partial \underline{b}}\right)$.

SUBROUTINE HALS(NP1,YY,JSIZ)

The subroutine HALS evaluates the integrands of the cost and constraint functionals at each time step.

SUBROUTINE INFUNC(TMAX,ITR,EPI,NFCI,NCV,JSIZ,YY,DLDB,
DPHDB,DZIDB,DPDFZT,DNDB,DGDB,DGDZ,DPDB,PB,ALM)

The subroutine INFUNC calculates, through the subroutine AHLPSI, the sensitivity coefficients for the inequality functional constraints (Eqs. (3.10)).

ITR - running number of optimiaztion iterations

DGDB - derivatives of the non-integral parts of the cost and constraint functionals with respect to $\underline{b}$

DGDZ - same with respect to initial state variables

DPDB - derivatives of $\underset{\sim}{P}$ matrix with respect to $\underline{b}$.

SUBROUTINE TEQFUN(TMAX,ITR,EPE,NFCE,NFCET,NCV,JSIZ,YY,DLDB,
DFDB,DPHDB,DZIDB,DPDFZT,DNDB,DGDB,DGDZ,DPDB,PB,ALM)

The subroutine TEQFUN calculates design sensitivity coefficients for the transformed equality functional constraints (Eqs. (3.7)).

SUBROUTINE DPARMC(TMAX,ITR,EPDI,NDCI,NCV)

The subroutine DPARMC calcualtes design sensitivity coefficients for the design parameter constraints (Eqs. (3.8)).

SUBROUTINE AHLPSI(NCV,ITR,TMAX,ISO,DLDB,DFDB,DPHDB,DZIDB,
DPDFZT,DNDB,DGDB,DGDZ,DPDB,PB,YY,JSIZ,ALM)

The subroutine AHLPSI calculates design sensitivity coefficients

for the constraints and cost functionals, with the help of subroutines ADJONT, PBFN, ADPDB, and DGDBZ.

ISO - running number of the functional constraint treated.

SUBROUTINE NEWB(EPS,NCV,STEP,ITR,NFCV)

The subroutine NEWB computes the matrix $M_{\psi\psi}$ of Eq. (4.29), solved Eqs. (4.30) and (4.31) for γ^1 and γ^2, and computes the necessary design changes δb given by Eqs. (4.33) through (4.35).

STEP - desired reuction in cost functional

NFCV - number of total functional constraint violations.

SUBROUTINE DYNANL(IS,IECHO,ITROB,TMIN, TMAX,TSTEP,HMIN,

HAMX,H,DLDB,DFDB,DPHDB,DZIDB,JSIZ,YY,PB,ALM,LIN,TLIMIT,

EPS,IFLAG1,SOLNO,SOLNO1,ISO,DNDB)

Subroutine DYNANL is described in detail in Section 5.3.

IS - a logic variable,
= 0, for static code generation,
= 1, for static solution,
= 2, for dynamic code generation,
= 3, for dynamic solution.

LIN - a user-supplied flag that limits the number of corrector iterations to 1 if all equations are linear,
= 0, for linear equations
= 1, for nonlinear equations.

SOLNO1 - an integer keeping a running account of the solution numbers at different time steps during adjoint analysis.

SUBROUTINE ADJONT(ISO,IFLAG1,SOLNO1,DLDB,DFDB,DPHDB,DZIDB,

JSIZ,YY,PB,ALM,DNDB)

The subroutine ADJONT works in a similar manner as ADAMS2 to solve the adjoint equations.

FUNCTION AHSM(I)

The function subprogram AHSM determines the maxima of the integrands of the transformed equality constraints, which are used in TEQFUN.

FUNCTION FUNPSE(I,TMAX)

The function subprogram FUNPSE evaluates the values of the functional constraints and the cost functional.

SUBROUTINE PBFN(PB)

The subroutine PBFN calculates terms of the $\underset{\sim}{P}(b)$ matrix.

SUBROUTINE ADPDB(DPDB,PB)

The subroutine ADPDB calculates derivatives of the $\underset{\sim}{P}(b)$ matrix with respect to design parameters.

SUBROUTINE DGDBZ(DGDB,DGDZ,ISO)

The subroutine DGDBZ calculates the derivatives of the non-integral parts of the cost and constraint functionals.

SUBROUTINE DLDFB(TIMPU,ISO,YY,JSIZ,DLDB,DFDB,DPHDB,DZIDB)

The subroutine DLDFB calcualtes the derivatives $\frac{\partial L}{\partial b}$, $\frac{\partial f}{\partial b}$, $\frac{\partial \phi}{\partial b}$, and $\frac{\partial \xi}{\partial b}$ for sensitivity analysis.

SUBROUTINE SOlNEW(ISO,JSIZ,HB,PB,AB,ALM,A,T,TIME,ZMAX)

The subroutine SOlNEW calculates the right-hand sides of the corrector euqations for the adjoint system.

HB - time step of dynamic analysis

AB - transformed gear coefficients of dynamic analysis used in adjoint analysis

A - current transformed gear coefficients

T - current time step

TIME - current time

ZMAX - total time interval of dynamic analysis.

SUBROUTINE ADLDZ(ISO,JSIZ,TIME,ZMAX)

The subroutine ADLDZ calculates the derivatives $\left(\frac{\partial L}{\partial z}\right)$ of the integrands of the cost and constraint functionals with repsect to solution variables of the transient analysis.

SUBROUTINE INTERP(Y,N,TA1,NP3,TMAX,JSTAT1)

The subroutine INTERP computes interpolated values of the dependent variable Y(I,1) and its time derivatives and replaces the previous values. The interpolation is to the point TOUT and uses the Nordsieck history array Y as follows:

$$Y(I,1) = \sum_{J=0}^{JSTAT1} Y(I,J+1)*S**J,$$

and

$$Y(I,2) = \sum_{J=1}^{JSTAT1} J*Y(I,J+1)*S**(J-1),$$

where

$$S = (TOUT + TA(NP3) - TMAX)/HA,$$

TA(NP3) is an old time instant during backward integration and HA is the time step of dynamic analysis.

SUBROUTINE VARSET(YY)

The subroutine VARSET sets the variable YY to be used in DIFSUB for numerical integration

SUBROUTINE SO3000(NB,NJ,IB,JT,NSD,IBSD,NPSOR,NPOSC,NG,NT,
JSIZ,ITF,NPSS,NSDT,IBSDT,IMODE)

The subroutine S03000 sets up pointers of the sparse matrix non-zero position.

NPOSR - pointer for the row number of each nonzero entry

NPOSC - pointer to the column number of each nonzero entry

NG - an integer variable keeping a running index of all the row and column pointers to the nonzero entries in the Jacobian matrix

NT - a vector giving values of NG after execution of different segments of S03000

ITF - a flag for keeping 10 nonzero positions for a revolute joint and 14 for a translational joint

NPSS - a vector storing different NG's at the beginning of each block of nonzero positions for a body in the Jacobian.

```
SUBROUTINE S01000(T,A,H,JSIZ,IMODE,YY,IFLAG1,PB,AB,ISO)
```

The subroutine S01000 mainly updates the terms of the Jacobian matrix and the right hand-side terms of the corrector equation (see Section 5.3 for further details).

```
SUBROUTINE INVERT(G,JSIZ,NP2,IP,JU,JC,IXL,IXH,IPP,ICNT,IPR,
IPC,IPRI,IVA,IVL,IVU,IC,IU ,JA,IRP,IRL,IWSR,ICP,ICL,IWSC,
IRWM,IRWC,AT,MAXS,MAXN,IB,NPOS,IMODE,ITF,NPSS,IFLAG1,IPSAV,
JASAV,IKNT,KFLAG)
```

The subroutine INVERT orders a matrix optimally and generates and stores sparse matrix codes for LU factorization. Provisions have been made in the program to call it in DYNANL whenever necessary during transient and adjoint analyses.

```
SUBROUTINE DIFSUB(N,TIME,T, TMIN,TMAX,EPS,YMAX,ERROR,KFLAG,
JSTART, MAXDER,G,IVPTR,NERR,NRR,NAL,LIN,CS,DY,TDIV,NCONV,
```

NPOS,JA,IP,IVA,IC,IU,JC,JU,DI,CC,U,IVL,IVU,CL,IPR,IPC,

IPRI,TLIMIT,IW1,TNEW,NQ,Y,YS,IFLAG1,PB,AB,ISO,A,HA,

ZMAX,TA,NP3,JSTAT1,PTEST,IKNT)

The subroutine DIFSUB is the numerical integration routine based on the gear algorithm (Chapter II) and is of extreme importance. For an original listing and explanation of the program and variables, reference [7] is recommended.

Explanations of the sparse matrix variables in subroutines INVERT and DIFSUB are available in reference [61].

CHAPTER VI

APPLICATIONS AND NUMERICAL RESULTS

6.1 Introduction

From the structure of the computer program DADS described in the previous chapter, it is clear that large classes of mechanical design problems can be handled through its implementation. To test the program, two example problems have been considered in the following sections: (1) a slider-crank mechanism (as described in Chapter II; see Figure 2.5), and (2) a trip plowshare mechanism.

The slider-crank mechanism is often used in engines and machineries [71] (see Chapter II) and thus it is quite familiar. The trip plowshare mechanism treated is a simplified version of the 2500 semi-integral spring-reset plow which is in production at John Deere (see Section 6.3). The importance of such mechanisms needs little description.

6.2 The Slider-Crank Mechanism

The slider-crank mechanism to be considered here is described in Chapter II (Subsection 2.1.2.5). Figure 2.5 shows the approximate initial position of such a mechanism with one spring-damper pair. Link 1 is ground, link 2 is the crank shaft, link 3 is the connecting rod or coupler, and link 4 is the piston or simply slider. A spring-damper pair is attached between link 4 and ground (Figure 2.5). There

is a translational joint (type 2) between bodies 4 and 1 and one revolute joint (type 1) between each of the following pairs of bodies: 1 and 2, 2 and 3, and 3 and 4. Gravitational forces are excluded from the present simulation of the mechanism.

The two-dimensional ADAMS program [14] has been implemented through the subroutine ADAMS2 to obtain a static equilibrium configuration and to determine the subsequent transient motion. The results of this analysis are used later in sensitivity analysis and optimization.

A symbolic listing of the nonzero positions of the Jacobian matrix for this example problem is given in Fig. 2.7. An explanation of the nonzero positions can be obtained in reference [14].

6.2.1 Formulation of the Optimal Design Problem

By virtue of its movement, a radial slider-crank mechanism exerts a force on ground through the crank-bearing and the wrist-pin guide (such as cylinder wall in an automotive piston-type engine). It is desirable to keep these "shaking forces" within bounds. It is also desirable to put an upper bound on the angular velocity of the crank at the final instant T of the time-interval [0,T] under consideration. The cost function is chosen to be twice the maximum energy stored in the spring during the interval of motion. The following design parameters are considered (see Chapter II):

b_1 = The spring constant K^1 of the spring

b_2 = Height of the points of attachment of the spring

b_3 = Half of the length of the uniform coupler.

With the notations of Chapters II and III, the optimal design problem is stated as follows: minimize

$$\bar{J} \equiv \max_{0 \le t \le T} b_1 (\ell_1 - \ell_0^1)^2 \tag{6.1}$$

subject to the equations of motion (2.33), the equations of constraints on motion (2.35), the spring-damper relations (2.36), the initial conditions of the form of Eqs. (3.4), and (3.5), the functional constraints

$$|\mu_2| \le \mu_2(\max) , \quad 0 \le t \le T \tag{6.2}$$

$$\dot{\phi}_2(T) \le \dot{\phi}_2(\max) \tag{6.3}$$

and the design parameter constraints

$$b_i^L \le b_i \le b_i^U , \quad i=1,2,3 \tag{6.4}$$

In Eq. (6.2) μ_2 is the y-component of reaction forces acting on body 2 at joint 1, because from Eq. (2.19) $\frac{\partial \Phi_Y}{\partial Y_2} = \frac{\partial \Phi_2}{\partial Y_2} = -1$, and so $-\frac{\partial \Phi_2}{\partial Y_2} \mu_2 = \mu_2$ in Eq. (2.24).

For the sake of simplicity, only the constraints on the vertical component of shaking forces at the crank bearing have been considered here.

After the introduction of an artificial design parameter b_4 [24,35,37] and the integral functional forms in the conventional way [24], the problem can be reformulated as: minimize

$$\psi_0 = b_4 \tag{6.5}$$

subject to Eqs. (2.33), (2.35), (2.36), (3.4) and (3.5), the constraints

$$\psi_1 \equiv \int_0^T \langle -\mu_2 - \mu_{2(\max)} \rangle dt = 0 \tag{6.6}$$

$$\psi_2 \equiv \int_0^T \langle \mu_2 - \mu_{2(\max)} \rangle dt = 0 \tag{6.7}$$

$$\psi_3 \equiv \int_0^T \langle b_1(\ell_1 - \ell_{10})^2 - b_4 \rangle dt = 0 \tag{6.8}$$

$$\psi_4 \equiv \dot{\phi}_2(T) - \dot{\phi}_{2(\max)} \leq 0 \tag{6.9}$$

and the design parameter constraints of Eq. (6.4).

The symbol $\langle \eta(t) \rangle$ used above has the following meaning (see Chapter III):

$$\langle \eta(t) \rangle = \begin{cases} (\eta(t))^2 , & \text{for } \eta(t) \geq 0 \\ 0 , & \text{for } \eta(t) \leq 0 \end{cases} \tag{6.10}$$

This formulation corresponds to the general formulation of the optimal design problem stated in Chapter III. After specification of initial numerical data, the DADS program described in Chapter V can be implemented to obtain a solution.

6.2.2 Sensitivity Analysis

In problems of optimal design, sensitivity analysis constitutes a principal part of the work to be done. When sensitivity coefficients are obtained, an iterative optimization algorithm can be applied to determine an optimal solution. In the present class of problems, subroutines RELATE, DNUDB, DGDBZ, ADLDZ, and DLDFB are the major user-supplied subprograms for sensitivity analysis. Among

them DGDBZ and ADLDZ are often relatively simpler, because they depend solely on the forms of the functional constraints. The relations entering into the subroutines RELATE, DGDBZ, ADLDZ, and DLDFB for the present problem (with 1 second interval) and the plow-share mechanism are given in the Appendix. Subroutines DGDBZ and DLDFB are used in AHLPSI to calculate sensitivity coefficients.

6.2.3 Numerical Results

Initial estimates of the parameters for the slider-crank mechanism under consideration are given in Table 6.1. The units used are inch, pound-force, and second. Two time intervals are treated: [0.0,1.0] and [0.0, 2.0]. During transient analysis a constant counter-clockwise torque of 100 in/lbf is applied to link 2. Fig. 6.2 gives the tableau of nonzero positions of the Jacobian matrix for this problem.

To examine the correctness of Eqs. (3.40) and (3.41) for sensitivity analysis, the sensitivity coefficeints ℓ^{ψ_β}, $\beta\varepsilon\{1,2,3,4\}$, and ℓ^{ψ_0} are first calculated for the 2 second interval, with initial estimates $b^I = [1.0,0.5,10.0,14.5]^T$ for the design parameters, for lower bounds $b^L = [0.8,0.2,5.0]^T$ and upper bounds $b^U = [1.5,0.8,16.0]^T$ on the first three design parameters, and with $\mu_{2(max)} = 10.0$ and $\dot{\phi}_{2(max)} = 0.3$. The cost and constraint functionals are evaluated for 0.1% and 1.0% perturbations of the first, second, and fourth design parameters (in conjunction with 0.01% and 0.1% changes in the third) and the corresponding changes in the functionals are examined. Table 6.2 gives the sensitivity coefficients ℓ^{ψ_0} and ℓ^{ψ_β}

Table 6.1

Initial Estimates of the Parameters for the Slider-Crank Mechanism (Inch, Pound-Force, Second).

LINK DESCRIPTION			
M(1) = 1.	M(2) = 4.	M(3) = 1.	M(4) = 3.
JIN(1) = 1.	JIN(2) = 10.	JIN(3) = 4.	JIN(4) = 2.
X(1) = 0.	X(2) = 1.	X(3) = 8.667	X(4) = 17.32
Y(1) = 0.	Y(2) = 0.	Y(3) = 5.25	Y(4) = .5
PHI(1) = 0.	PHI(2) = 1.5708	PHI(3) = -.5236	PHI(4) = 0.
FX(1) = 0.	FX(2) = 0.	FX(3) = 0.	FX(4) = 0.
FY(1) = 0.	FY(2) = 0.	FY(3) = 0.	FY(4) = 0.
TQ(1) = 0.	TQ(2) = 0.	TQ(3) = 0.	TQ(4) = 0.
JOINT DESCRIPTION			
IB(1,1) = 1	IB(1,2) = 2	IB(1,3) = 3	IB(1,4) = 4
X1(1) = 0.	X1(2) = 9.0	X1(3) = 10.	X1(4) = 0.
Y1(1) = 0.	Y1(2) = 0.	Y1(3) = 0.	Y1(4) = 0.5
IB(2,1) = 2	IB(2,2) = 3	IB(2,3) = 4	IB(2,4) = 1
X2(1) = -1.	X2(2) = -10.	X2(3) = 0.	X2(4) = 0.
Y2(1) = 0.	Y2(2) = 0.	Y2(3) = 0.	Y2(4) = 1.0
SPRING-DAMPER DESCRIPTION			
IBSD(1,1) = 1	XF1(1) = 35.	YF1(1) = 0.5	IBSD(2,1) = 4
XF2(1) = 2.	YF2(1) = 0.	SK(1) = 1.	DC(1) = 1.
SDL(1) = 15.68	SDL0(1) = 15.68	SDF(1) = 0.	

Table 6.2

Sensitivity Coefficients ℓ^{ψ_0} and ℓ^{ψ_β}, $\beta \in \{3,4\}$

Components	ℓ^{ψ_0}	ℓ^{ψ_3}	ℓ^{ψ_4}
1	0.0	0.6115×10^{-2}	-0.850×10^{-1}
2	0.0	0.0	0.0
3	0.0	0.1395×10^{-2}	0.1872
4	1.0	-0.4626×10^{-3}	0.0

Table 6.3

Sensitivity Analysis Results for the
Slider Crank Mechanism in Compact Form

Initial Estimate of the Design Parameters: $\underline{b}_0 = (1.0 \quad 0.5 \quad 10.0 \quad 14.5)^T$								
δb_1	δb_2	δb_3	δb_4	$\bar{J}$	$\Delta\bar{J}$	$\ell J^T \delta b$	ψ_1	ψ_2
0.0	0.0	0.0	0.0	14.5	-o-	-o-	UNVIOLATED	UNVIOLATED
0.1×10^{-2}	0.0	0.0	0.0	14.5	0.0	0.0		
0.0	0.5×10^{-3}	0.0	0.0	14.5	0.0	0.0		
0.0	0.0	0.1×10^{-2}	0.0	14.5	0.0	0.0		
0.0	0.0	0.0	0.145×10^{-1}	14.5145	0.145×10^{-1}	0.145×10^{-1}		
0.1×10^{-1}	0.0	0.0	0.0	14.5	0.0	0.0		
0.0	0.5×10^{-2}	0.0	0.0	14.5	0.0	0.0		
0.0	0.0	0.1×10^{-1}	0.0	14.5	0.0	0.0		
0.0	0.0	0.0	0.145	14.645	0.145	0.145		

Table 6.3 (Cont'd)

δb_1	δb_2	δb_3	δb_4	ψ_3	$\Delta\psi_3$	$\ell^T \psi_3 \delta b$	ψ_4	$\Delta\psi_4$	$\ell^T \psi_4 \delta b$
0.0	0.0	0.0	0.0	0.16601 x 10^{-3}	-o-	-o-	0.24144	-o-	-o-
0.1 x 10^{-2}	0.0	0.0	0.0	0.17061 x 10^{-3}	.0046 x 10^{-3}	.0061 x 10^{-3}	0.24134	-0.0001	-0.00009
0.0	0.5 x 10^{-3}	0.0	0.0	0.16601 x 10^{-3}	0.0	0.0	0.24144	0.0	0.0
0.0	0.0	0.1 x 10^{-2}	0.0	0.16717 x 10^{-3}	0.1160 x 10^{-5}	0.1395 x 10^{-5}	0.24172	0.0002	0.0002
0.0	0.0	0.0	0.145 x 10^{-1}	0.16073 x 10^{-3}	-0.0053 x 10^{-3}	-0.0064 x 10^{-3}	0.24144	0.0	0.0
0.1 x 10^{-1}	0.0	0.0	0.0	0.21534 x 10^{-3}	0.04933 x 10^{-3}	0.06115 x 10^{-3}	0.24039	-0.0010	-0.0009
0.0	0.5 x 10^{-2}	0.0	0.0	0.16601 x 10^{-3}	0.0	0.0	0.24144	0.0	0.0
0.0	0.0	0.1 x 10^{-1}	0.0	0.17783 x 10^{-3}	0.01182 x 10^{-3}	0.01395 x 10^{-3}	0.24430	0.0028	0.0020
0.0	0.0	0.0	0.145	0.11825 x 10^{-3}	-0.0478 x 10^{-3}	-0.0647 x 10^{-3}	0.24144	0.0	0.0

$\beta \varepsilon \{3,4\}$, and Table 6.3 gives other computational results in compact form. The first and second constraints, however, remain unviolated.

In these computations the integrands of ψ_1, ψ_2, and ψ_3 have been normalized by dividing by $\mu^2_{2(max)}$, $\mu^2_{2(max)}$, and b^2_4, respectively. It is observed from the Table that terms of the form $\ell^{\psi^T}\delta b$ match satisfactorily with the corresponding $\Delta\psi$'s. The slight discrepancies are attributed to various approximations and linearizations at many steps of both transient and sensitivity analysis and relatively coarse time grids for numerical integrations and transformation of pointwise constraints to functional forms. However, they need not affect the optimization process.

In carrying out the optimal design algorithm, a design reduction ratio of 3% is used to compute the stepsize in the first iteration. The bounds b^L and b^U are taken to be $[0.8,0.2,9.0]^T$ and $[1.5,0.8,12.0]^T$, respectively. The initial estimate b^I (including artificial design parameter) is taken as $[0.8244, 0.5, 9.0, 11.12]^T$ and $\mu_{2(max)}$ and $\dot{\phi}_{2(max)}$ are kept unchanged.

At the starting design, $||\delta b^1|| = 0.1248$ and $||\delta b^2|| = 0.2307$, constraints 3 and 4 are violated, and the constraint on the design parameter b_3 is tight. After 6 iterations, as the algorithm approaches the optimum design, $||\delta b^1||$ reduces to 0.8404×10^{-15}, $||\delta b^2||$ reduces to 0.2669×10^{-1}, and the constraints on b_1, b_2, and b_3 become tight. The final optimum results are given in Table 6.4.

Table 6.4

Optimum Results for the Slider-Crank Mechanism
for the Time Interval of 2 Seconds

Real cost function $\bar{J}$	=	$\max_{0<t<2} b_1(\ell_1 - \ell_0^1)^2$
Lower bounds on b	=	$[\overline{0.8}, 0.2, 9.0]^T$
Upper bounds on b	=	$[1.5, 0.8, 12.0]^T$

	Starting Values	Optimum Values
b_1	8.2440×10^{-1}	8.0000×10^{-1}
b_2	5.0000×10^{-1}	5.0000×10^{-1}
b_3	9.0000	9.0000
$\bar{J}$	1.1159×10^{1}	1.00115×10^{1}

Next, the design problem is considered for the time interval of 1 second with a different set of data. For this case a design reduction ratio of 10% is used in computing the step size in the first iteration. The bounds b^L and b^U are taken to be $[0.8, 0.2, 5.0]^T$ and $[1.5, 0.8, 16.0]^T$, respectively. The initial estimate b^I is taken as $[1.0, 0.5, 10.0, 1.0]^T$ and $\mu_{2(max)}$ and $\dot{\phi}_{2(max)}$ are taken as 5.0 and 0.3, respectively.

At the starting design, $||\delta b^1|| = 0.0$, $||\delta b^2|| = 0.1106 \times 10^{-2}$, and constraint 3 is violated. In the second iteration $||\delta b^1|| = 0.7348$, $||\delta b^2|| = 0.3814 \times 10^{-1}$, and again constraint 3 is violated. After 18 iterations, as the algorithm approaches the optimum design, $||\delta b^1|| = 5.567 \times 10^{-3}$, $||\delta b^2|| = 3.018 \times 10^{-3}$, and (lower) constraints on b_1 and b_2 remain tight. The optimum results are given in Table 6.5.

6.3 The Plow-Share Mechanism

The transient dynamic response and design sensitivity of a spring-rest plow-share mechanism is determined to illustrate the flexibility of the DADS computer program for systems with intermittent motion. Fig. 6.1 shows the approximate initial position of such a mechanism. For a system of this nature any type of closed form solution is beyond consideration.

The model consists of 6 moveable rigid bodies, identified as follows: body 2 - plow-share and standard; body 3 - lower link;

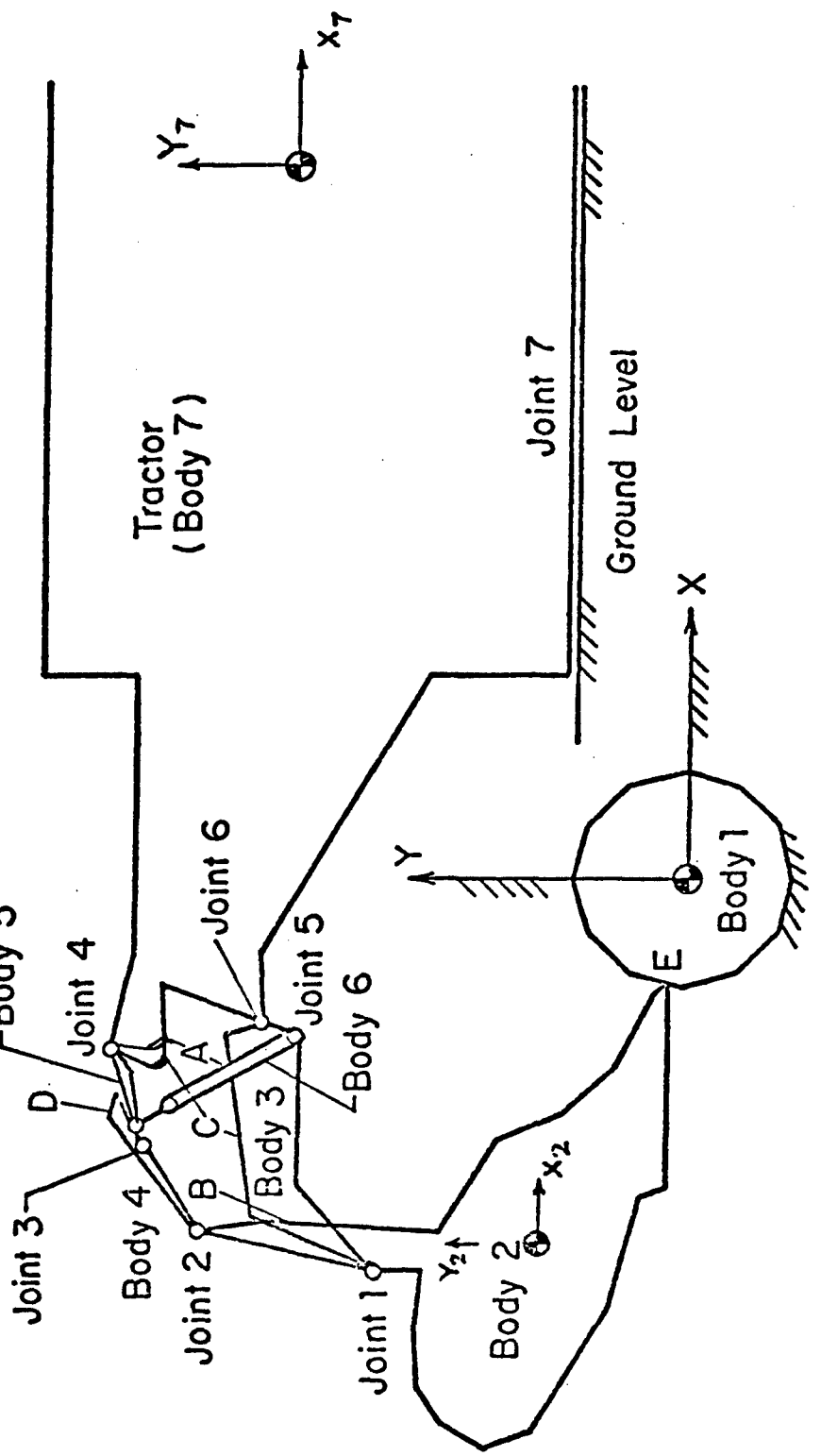


Figure 6.1 Approximate Initial Configuration of the Plow-Share Mechanism.

Table 6.5

Optimum Results for the Slider Crank Mechanism for the Time Interval of 1 Second

Real Cost Function $\bar{J} = \max_{0 \le t \le 1} b_1(\ell_1 - \ell_0^1)^2$

Lower Bounds on $b = [0.8, 0.2, 5.0]^T$

Upper Bounds on $b = [1.5, 0.8, 16.0]^T$

	Starting Values	Optimum Values
b_1	1.0000	8.0000×10^{-1}
b_2	5.0000×10^{-1}	5.0000×10^{-1}
b_3	1.0000×10^{1}	9.7566
$\bar{J}$	1.2417	9.5767×10^{-1}

body 4 - rear toggle link; body 5 - front toggle link; body 6 - U-bolt; and body 7 - combined plow frame and tractor. Body 1 (ground) is rigidly connnected to the inertial reference frame. Various pinned connections (revolute joints) are shown in Fig. 6.1. The entire system is moving to the right at approximately 2 meters per second, modeled by a horizontal translational joint between ground (body 1) and the combined plow frame and tractor (body 7). A linear spring-damper combination is connected between the U-bolt and rear toggle link. In addition, 5 contact points, identified by the letters A-E, represent: A - contact between the U-bolt and main frame; B - contact between the shank and lower link; C - contact between the lower link and main frame; D - contact between the front and rear toggle links; and E - contact between the plow-share tip and rock embedded in the ground (body 1).

Contacts are simulated by attaching modified linear spring-damper combinations between contacting bodies. The modification consists of setting the spring and damping coefficients to zero in a continuous manner as parts break contact and to their maximum values as contact is made. This is accomplished by introducing spring and damping coefficients as state variables and making them functions of spring length. A three-parameter model is chosen because of the wide range of spring and damper characteristics it is capable of representing. The equations are of the

form

$$K_{ij} = K_{ij}^{mx}(1 - EXP[-(< \pm (\ell - \ell_0) > / \theta_{ij})^{b_{ij}}])$$

$$C_{ij} = C_{ij}^{mx}(1 - EXP[-(< \pm (\ell - \ell_0) > / \theta_{ij})^{b_{ij}}]) \qquad (6.11)$$

where K_{ij}^{mx} and C_{ij}^{mx} are the maximum values of the spring and damping coefficients for the pair connecting i^{th} and j^{th} bodies. The expression $< \pm(\ell - \ell_0) >$ equals zero if $\pm(\ell - \ell_0) \leq 0$; otherwise it equals $\pm(\ell - \ell_0)$. The '+' or '-' sign is selected depending upon whether the coefficients are to increase or decrease as the spring length increases. The parameter θ_{ij} determines the interval of spring travel over which the coefficients change and b_{ij} determines the shape of the curve. Fig. 6.2 depicts various shapes for combinations of b_{ij} and θ_{ij}. For the 5 contact points (stops) in the present problem the '-' sign is chosen in equation (6.11).

It should be observed here that by varying the free spring length (undeformed), K_{21}^{mx}, C_{21}^{mx}, b_{21}, and θ_{21} for the sixth spring, various models of the underground rock can be obtained. In the present analysis the frictional forces acting between the plow tip and the rock have been taken as a continuous function of the spring forces and have been fed into the program through subroutine FOREXT (see Chapter V) after transforming them to the equivalent system at the center of mass of the second body.

Owing to the presence of the nonlinear spring-dampers, the Jacobian matrix changes very rapidly in a neighborhood of certain

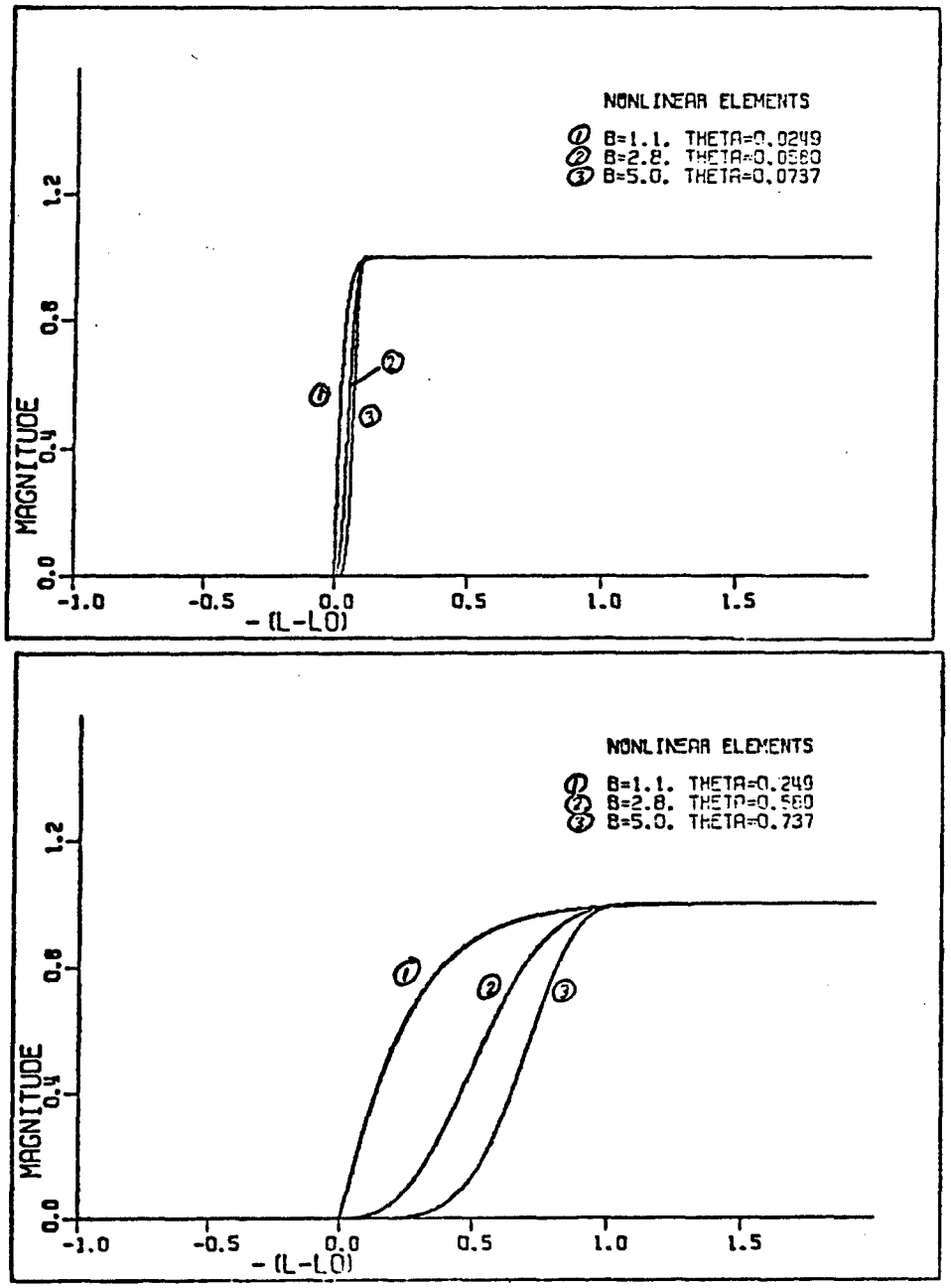


Figure 6.2 Dependence of Functions on B and Theta.

time instants. Provisions have been made in the program to regenerate the sparse matrix codes at those time instants, by introducing several error tests and flags in subroutine DIFSUB (see Chapter V). The error tests measure the ratios of maximum valued elements of Jacobian to the pivots. When a pivot tends to zero, the ratio becomes very large and a fresh code generation is demanded. For the estimation of the bound and related discussions, references [73,74] are recommended.

6.3.1 Numerical Results (Dynamic Analysis)

Initial estimates of the parameters for the spring reset plowshare mechanism are given in Table 6.6. The units used are meter, kilogram, second, and Newton. For this problem, the values of the parameters b_{ij} and θ_{ij} are the same for all five nonlinear springs. They are 1.2 and 8.617739×10^{-6}, respectively. The corresponding spring travel is approximately 1.538×10^{-5} to attain 99% of the maximum values of the spring-damper coefficients. The maximum values of the coefficients of the 5 nonlinear spring-damper pairs are $(1.0 \times 10^{6}, 6.2 \times 10^{4})$, $(1.0 \times 10^{6}, 6.0 \times 10^{4})$, $(1.0 \times 10^{6}, 6.2 \times 10^{5})$, $(1.0 \times 10^{6}, 6.2 \times 10^{4})$, and $(1.0 \times 10^{5}, 0.0)$, respectively. These are entered into the program through subroutine INDATA (see Chapter V). During transient analysis, the sparse matrix codes are regenerated ten times.

Initially the combined plow and tractor system shown in Fig. 6.1 is moving to the right at 2 meters per second, along a

Table 6.6

Initial Estimates of the Parameters for the Spring-Reset Plow-Share Mechanism (Meter, Kilogram, Second, Newton)

LINK	M	JIN	X	Y	PHI	FX	FY	TQ
1	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	60.398	3.971	-0.6595	0.249	0.0	0.0	-592.5044	0.0
3	22.0157	0.70537	-0.5015	0.6761	0.0	0.0	-215.9740	0.0
4	4.522	0.0351	-0.5336	0.8899	0.48869	0.0	- 44.3608	0.0
5	2.559	0.0122	-0.4138	0.942	0.21391	0.0	- 25.1038	0.0
6	2.155	0.0184	-0.3637	0.771	0.53232	0.0	- 21.1406	0.0
7	8237.376	12000.0	1.33	0.718	0.0	100.0	0.0	0.0

JOINT NO.	JT	IB(1,I)	X1	Y1	IB(2,I)	X2	Y2
1	1	2	-0.0845	0.276	3	-0.2425	-0.1511
2	1	2	-0.008	0.585	4	-0.1445	0.0135
3	1	4	0.046	0.0135	5	-0.08745	0.0
4	1	5	0.091	0.04503	7	-1.6475	0.2665
5	1	3	0.203	-0.0158	6	0.0	-0.1285
6	1	3	0.2315	0.042	7	-1.6	0.0
7	2	7	0.0	0.2665	1	0.0	0.9846

Table 6.6 (cont'd)

Spring-Damper #	IBSD(1,I)	XF1	YF1	IBSD(2,I)	XF2	YF2
1	4	0.0841	0.0135	6	0.0	0.127
2	7	3.0	0.1975	6	0.0	0.085
3	2	0.0	0.47	3	0.0	0.0495
4	7	-1.617	0.1975	3	0.03333	0.042
5	4	0.095	0.0135	5	-0.03845	0.0
6	2	0.4595	-0.249	1	0.0	-0.01

Spring-Damper #	SK	DC	SDL	SDLO	SDF
1	40010.08	0.0	0.05516	0.062264	0.0
2	0.0	0.0	4.7675	4.6975	0.0
3	1000000.0	60000.0	0.1572	0.1582	0.0
4	0.0	0.0	0.2791	0.08805	0.0
5	0.0	0.0	0.0128	0.0128	0.0
6	100000.0	0.0	0.199	0.2	0.0

horizontal translational joint. The tip of the plow makes contact with the rock at time = 0.0 seconds. As shown in Fig. 6.3, the tip breaks contact with the rock at about 0.1 seconds, but fails to clear it and comes to contact again between 0.22 and 0.32 seconds. The contact force imparts an angular velocity to the plow-share, causing it to move rearward and upward (see Figs. 6.4 and 6.5). This motion drives the toggle links upward, bringing spring 1 into tension (see Fig. 6.6). The U-bolt and lower link come into contact with the plow frame (contact points A and C) at 0.11795 and 0.39581 seconds, respectively. Contact at B between the standard and lower link (stop 2) is broken at 0.32384 seconds and this event coincides with the loss of contact of the tip with the rock.

Contact at C between the lower link and frame stops upward movement of the plow-share and the reset cycle begins. Stored energy in the spring rapidly collapses the toggle links. This action, shown in Figs. 6.4 and 6.5, causes a rapid change in angular displacement of the plow-share, with only a small effect on its vertical displacement. At 0.53509 seconds, the toggle links have reset (contact at D).

The lower link and U-bolt break contact with the frame at 0.55585 and 0.68854 seconds, respectively. It is interesting to note that the toggle action results in the plow-share being brought to within 20° of horizontal, while its center of mass is still 0.75 meters above ground. The plow-share therefore re-enters the

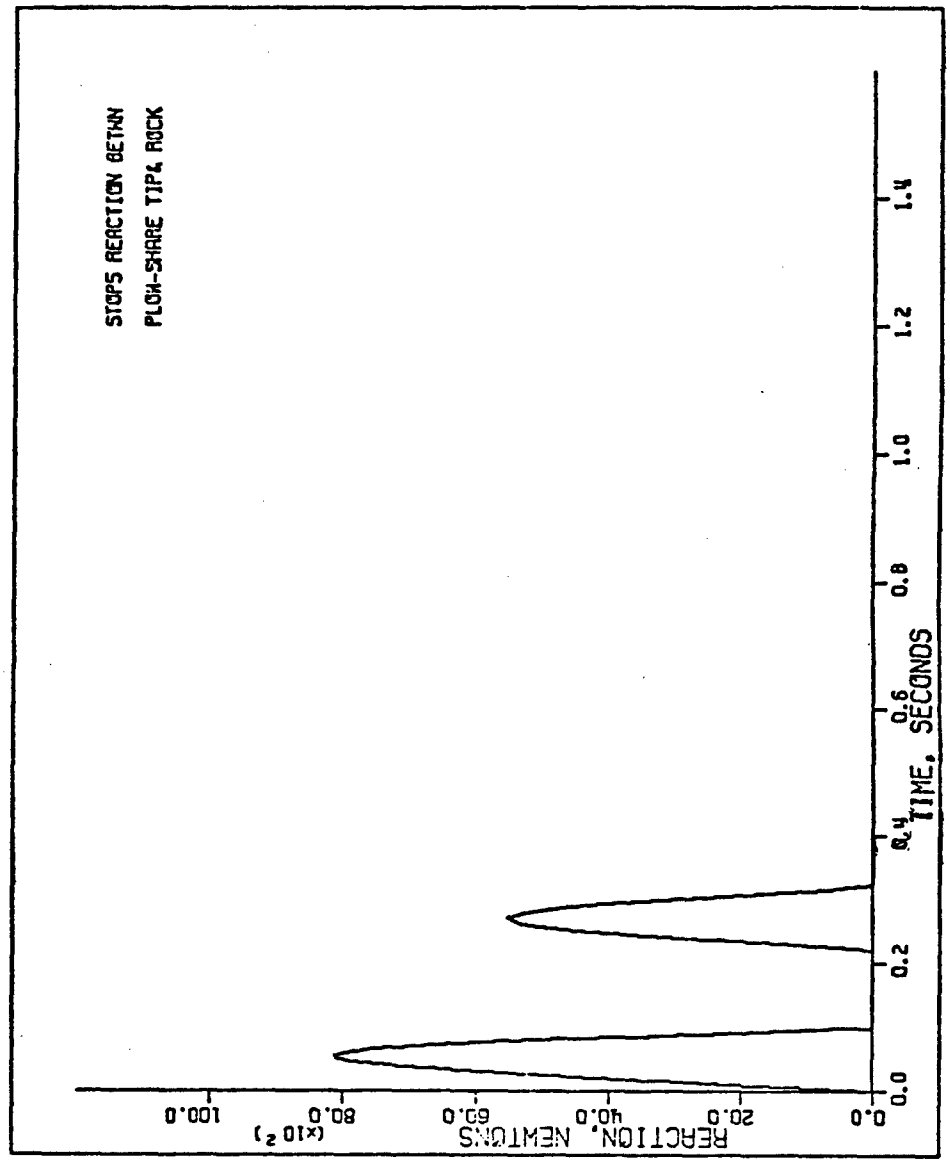


Figure 6.3 Stop 5 Reaction Between Plow-Share Tip and Rock.

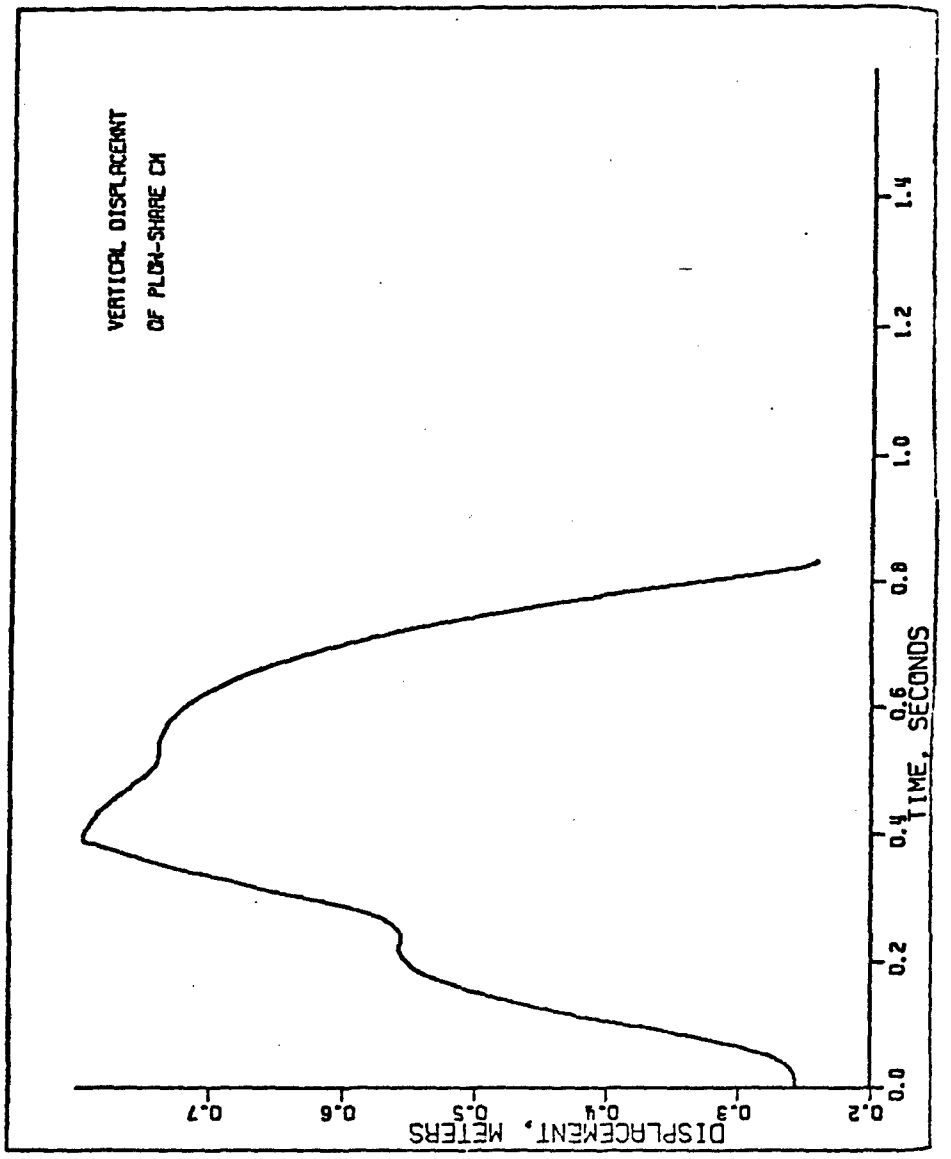


Figure 6.4 Vertical Displacement of Plow-Share CM.

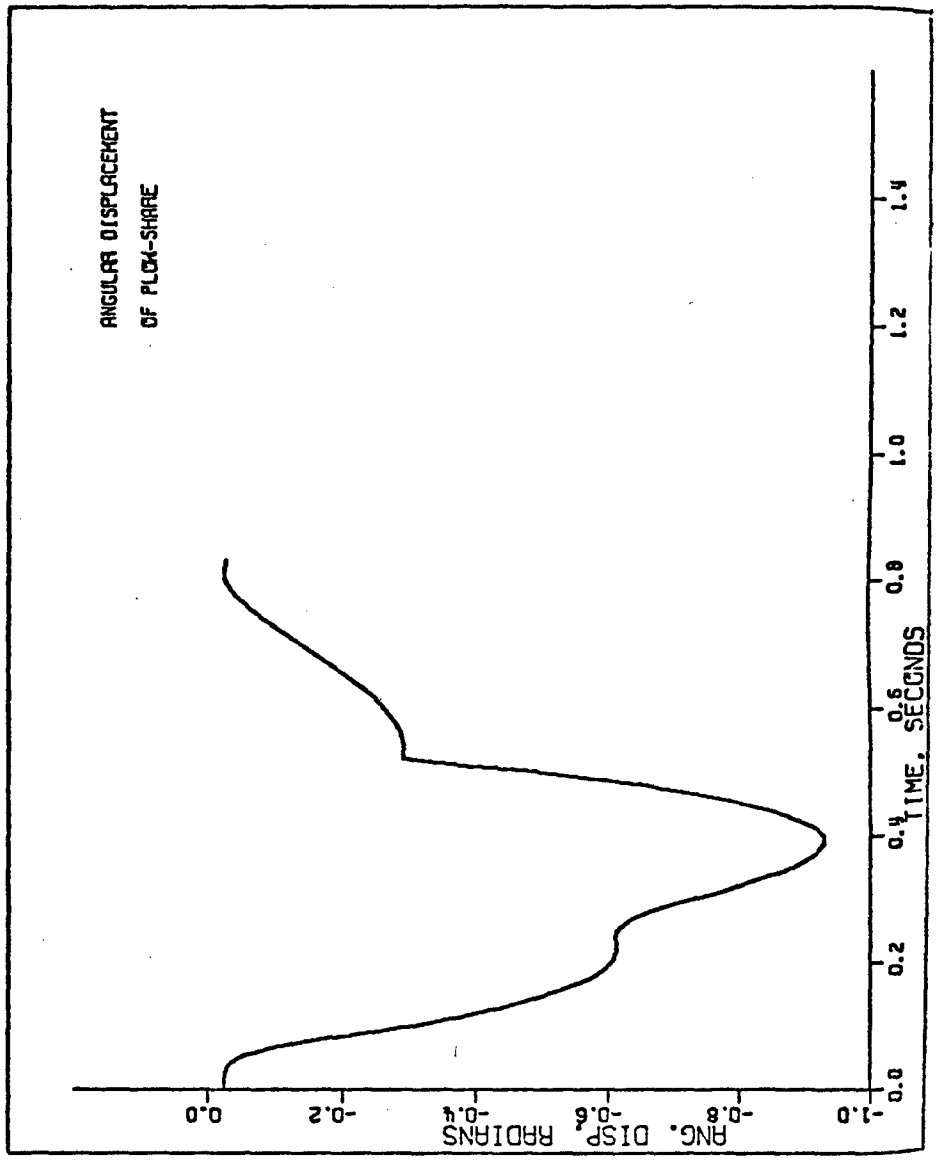


Figure 6.5 Angular Displacement of Plow Share.

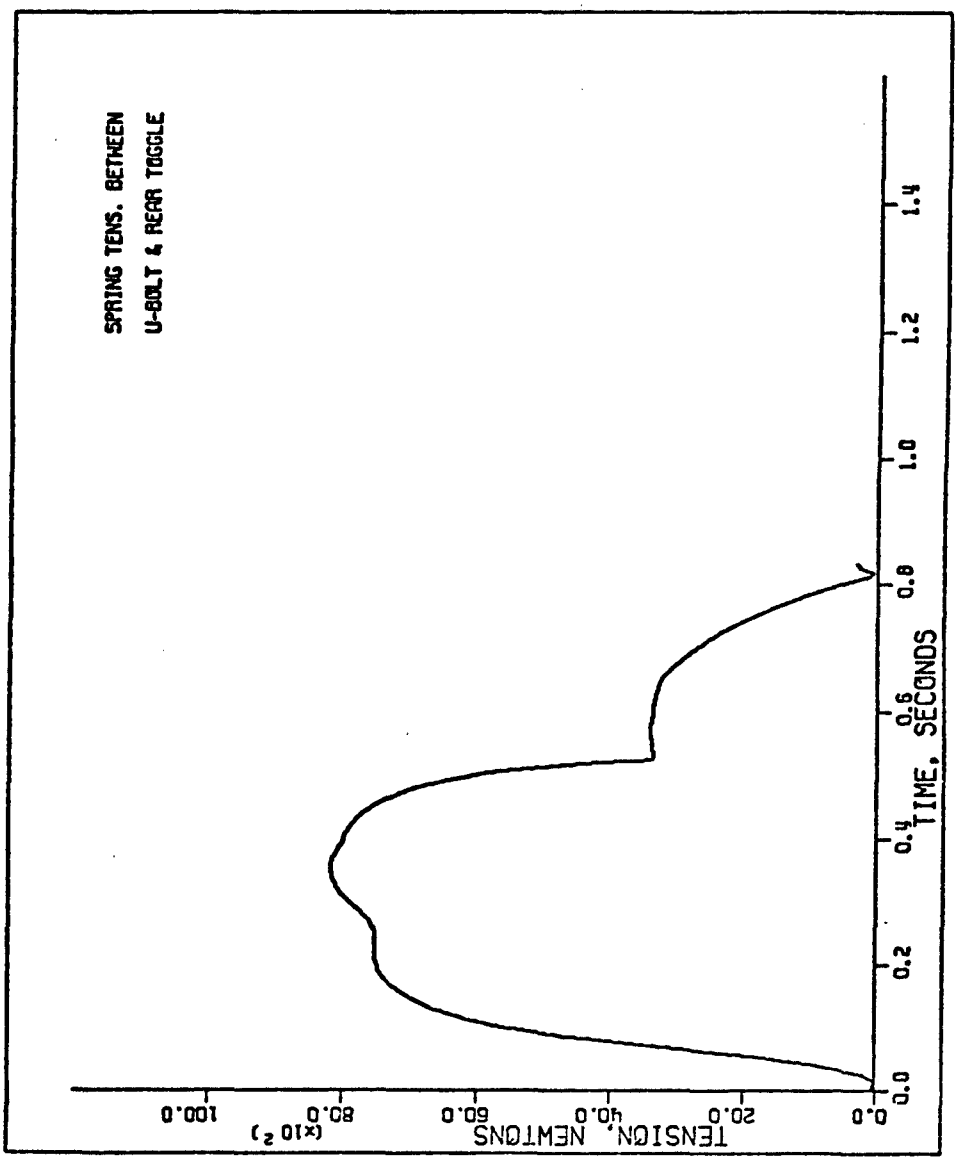


Figure 6.6 Spring Tension Between U-Bolt and Rear Toggle.

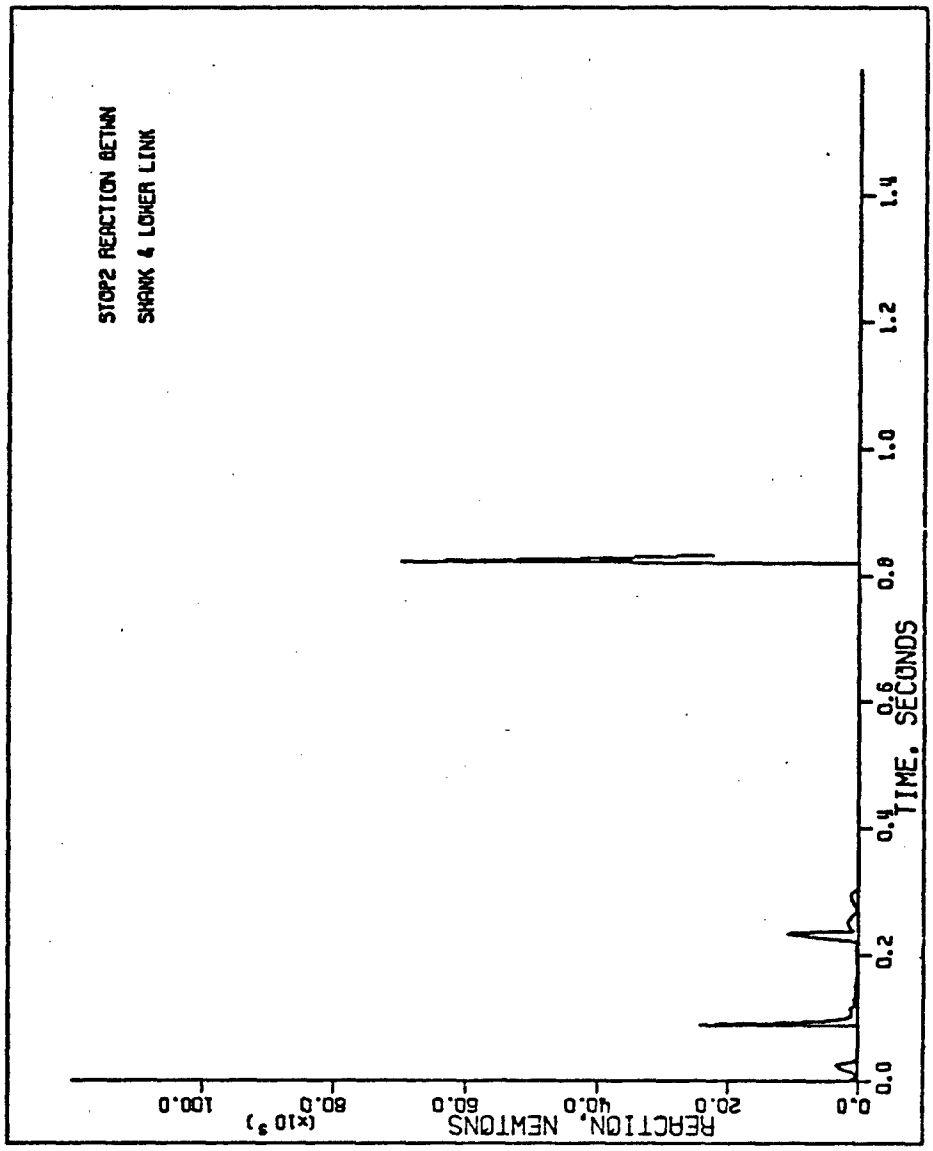


Figure 6.7 Stop 2 Reaction Between Shank and Lower Link.

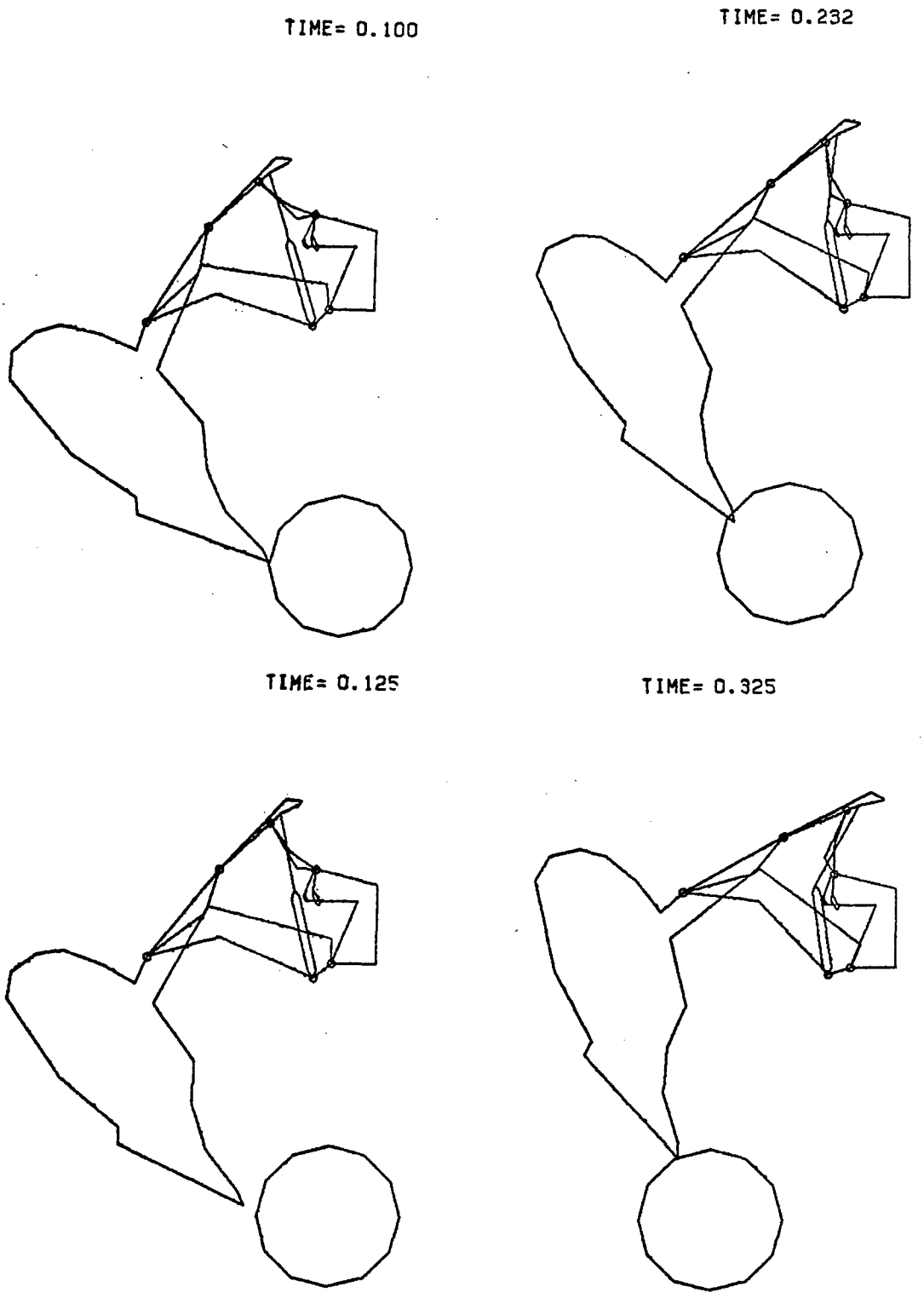
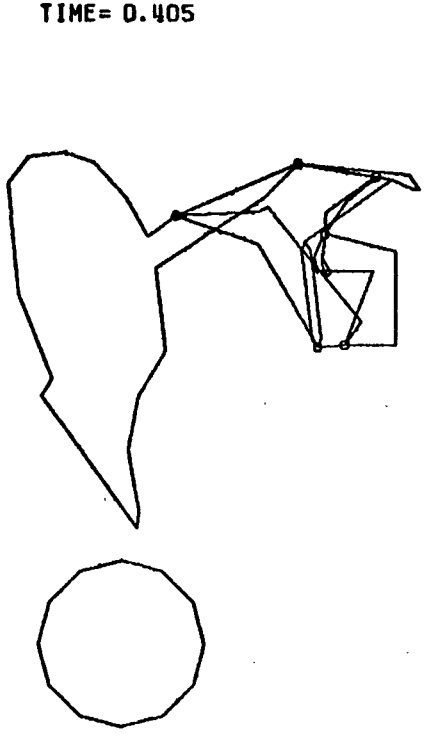


Figure 6.8 VERSATEC Plots (Snap-shot pictures) of the Plow-Share Mechanism at Selected Time Instants.



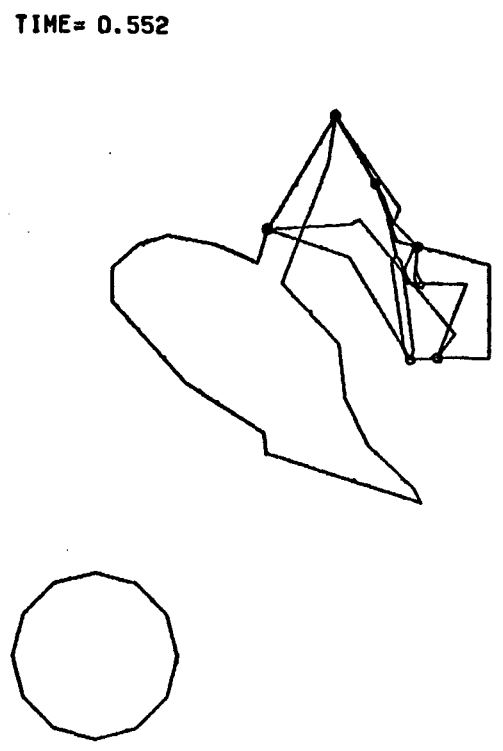


Figure 6.8 (cont'd.)

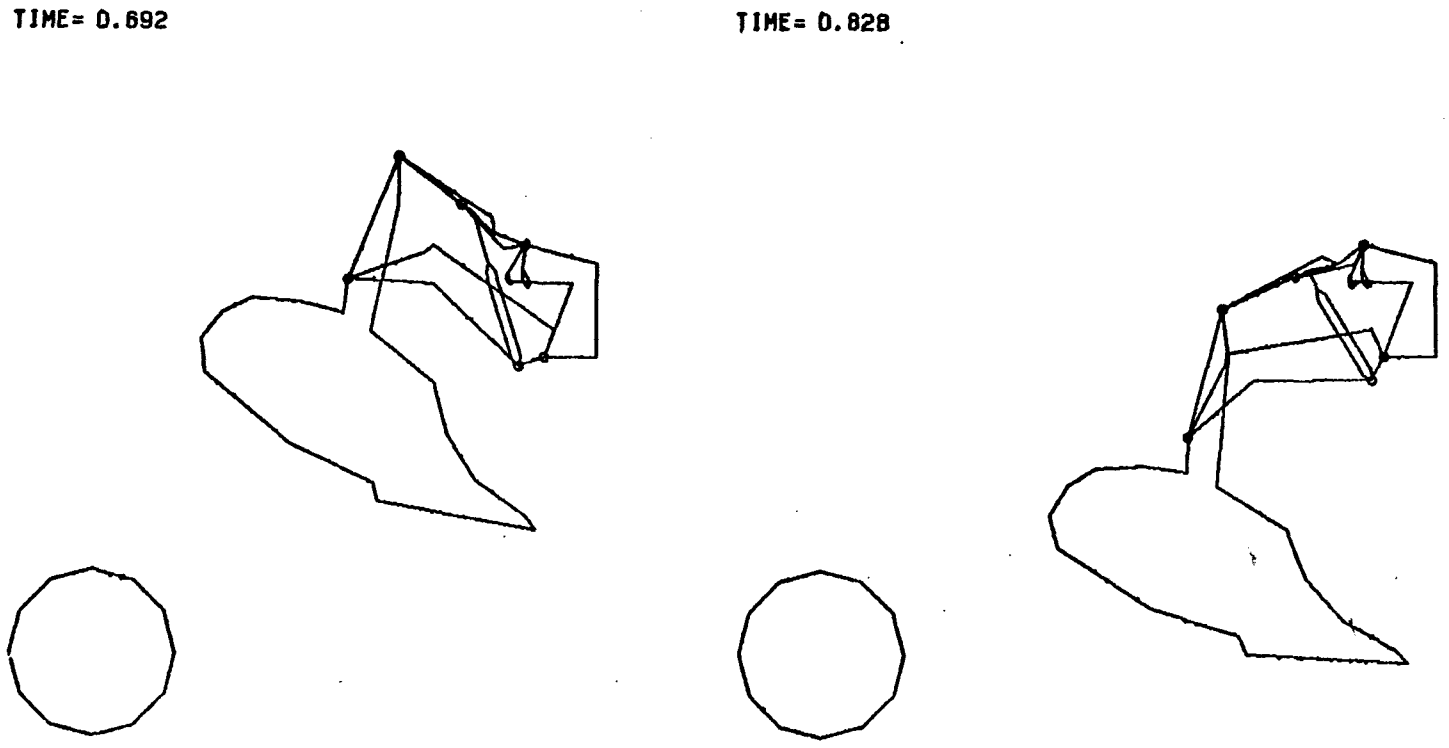


Figure 6.8 (cont'd.)

ground at a shallow angle, preventing the mechanism from being tripped again. Finally, at about 0.83174 seconds, contact occurs at stop 2 and the mechanism regains its approximate initial configuration. Fig. 6.7 gives the stop 2 reactions during the entire time interval.

Fig. 6.8 gives VERSATEC plots (snap-shot pictures) of the mechanism at selected instants of time. The whole process of the computer simulation of this dynamic analysis takes about 3 minutes 30 seconds of CPU time with an IBM-370-168 computer system.

6.3.2 Formulation of a Trip-Plow Optimal Design Problem

It was noted in the last section that while re-entering the ground the plow-share mechanism should be prevented from being tripped again. This can be achieved if a constraint is imposed such that the plow-share is brought to horizontal, to within a certain small tolerance angle, during the final phase of motion. The cost function to be minimized in the design process is chosen to be twice the maximum energy stored in the spring during the interval of motion, which is taken to be 0.832 seconds. The following design parameters are considered here:

b_1 = Diameter of the coil wire of the reset spring;

b_2 = Diameter of the coil of the reset spring;

b_3 = Number of turns in the coil.

The relation between the undeformed length ℓ_0^1 of the reset spring and design parameters is taken as

$$\ell_0^1 = b_1 b_3 + 0.56604 \times 10^{-2} \tag{6.12}$$

It should be observed here that since the undeformed length of the reset spring depends on b_1 and b_3, ℓ_0^1 is automatically a design parameter.

With the notations of Chapters II and III, the optimal design problem is stated as follows: Minimize

$$\bar{J} \equiv \max_{0 \le t \le 0.832} K^1 (\ell_1 - \ell_0^1)^2 \tag{6.13}$$

where [72]

$$K^1 \equiv K_{46} = \frac{G\, b_1^4}{8\, b_2^3\, b_3} \quad , \quad G \text{ being shear modulus,} \tag{6.14}$$

subject to the equations of motion, Eq. (2.33), the equations of constraints on motion, Eq. (2.35), the spring-damper relations, Eqs. (2.36) and (6.11), initial conditions of the form of Eqs. (3.4) and (3.5), the functional constraint

$$-\phi_2 - 0.174533 \le 0 \quad , \quad 0.75 \le t \le 0.832 \tag{6.15}$$

and the design parameter constraints

$$b_i^L \le b_i \le b_i^U \quad , \quad i=1,2,3 \tag{6.16}$$

For the functional constraint Eq. (6.15), the maximum allowable inclination of the plow-share during the interval $0.75 \le t \le 0.832$ of re-entering phase has been taken to be 10°.

As in the case of the slider-crank mechanism, with similar

notations, the problem can be reformulated as: Minimize

$$\psi_0 = b_4 \tag{6.17}$$

subject to Eqs. (2.33), (2.35), (2.36), (6.11), (3.4), and (3.5), the constraints

$$\psi_1 \equiv \int_0^{0.832} \langle\langle -\phi_2 - 0.174533 \rangle\rangle \, dt = 0 \tag{6.18}$$

$$\psi_2 \equiv \int_0^{0.832} \left\langle \frac{G b_1^4}{8 b_2^3 b_3} (\ell_1 - \ell_0^1)^2 - b_4 \right\rangle dt = 0 \tag{6.19}$$

and the design parameter constraint of Eq. (6.16).

In Eq. (6.18) the symbol $\ll H \gg$ has the following meaning:

$$\langle\langle H \rangle\rangle = \begin{cases} 0, \text{ for } 0 \leq t \leq t_1 = 0.75 \text{ and for } H \leq 0, \\ H^2, \text{ for } H \geq 0 \end{cases} \tag{6.20}$$

Since the transient analysis itself is extremely complex for this mechanism, only one regular functional constraint has been considered. However, additional constraints can be treated.

6.3.3 Modifications in Sensitivity Analysis Due to Non Standard Elements

Owing to the inclusion of nonstandard equations, Eq. (6.11), in the set of spring-damper relations, some modifications are required in the sensitivity analysis. The calculations and programming for the additional elements in NPOSR, NPOSC, and G vectors and the right-hand sides are routinely [7,14] done through the

subroutines USET, USOLV1, USOLV2, and USOLV3 together with another set of subroutines LSPTRS, LSSLV1, LSSLV2, and LSSLV3 (see Chapter V). The solution vector of (K_{ij}, C_{ij}) of Eq. (6.11) is included in the extended vector $\underline{\ell}$ and the corresponding $\bar{I}$ matrix of Eq. (2.38) becomes

$$\bar{I} \equiv \begin{bmatrix} \begin{bmatrix} 0 & 0 & & & & \\ 1 & 0 & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & 0 & 0 & \\ & & & 1 & 0 & \\ & & & & & 0 \\ & & & & & & 0 \\ & & & & & & & \ddots \\ & & & & & & & & \ddots \end{bmatrix}_{(4s \times 4s)} & & \\ & \begin{matrix} 0 \\ & 0 \\ & & \ddots \\ & & & \ddots \\ & & & & 0 \end{matrix} & \begin{matrix} (4s+NSDV) \\ \times (4s+NSDV) \end{matrix} \end{bmatrix} \tag{6.21}$$

where NSDV is twice the number of nonlinear spring-damper pairs.

For the sensitivity analysis, necessary modifications are performed in subroutine SOlNEW for calculation of the right-hand sides of the adjoint equations and additional calculations for the values of $\frac{\partial \xi}{\partial b}$ are done in DLDFB. Necessary adjustements for the dimensions of the variable DZIDB are also made. It should be remarked here

that DLDFB is called after the call of INTERP, so that the values in DLDFB are calculated with the interpolated values of the solution variables.

6.3.4 Numerical Results (Adjoint Analysis and Optimization)

In carrying out the optimal design algorithm, a design reduction ratio of 1% is used to compute the step-size in the first iteration. The bounds b^L and b^U are taken to be $[0.2 \times 10^{-2}, 0.1 \times 10^{-1}, 0.8 \times 10^{1}]^T$ and $[0.1 \times 10^{-1}, 0.1, 0.14 \times 10^{2}]^T$, respectively. The initial estimate b^I (including artificial design parameter) is taken as $[0.56604 \times 10^{-2}, 0.181361 \times 10^{-1}, 0.1 \times 10^{2}, 0.167471 \times 10^{4}]^T$. The shear modulus G has been taken to be 1.86×10^{10} N/m^2. The static analysis and static sensitivity analysis results are given in Table 6.7 and Table 6.8.

It is interesting to note that in the first iteration, all convergence criteria are satisfied with $||\delta b^1|| = 0.6209 \times 10^{-3}$ and $||\delta b^2|| = 0.2132 \times 10^{-3}$ and the final optimum results are given in Table 6.9.

In order to treat the problem in a different way, the first functional constraint is redefined such that the tip of the plowshare remains at least 2" = 0.0508 m. above the ground level during a certain interval in the final phase of motion. This time interval is taken to be [0.6, 0.832]. The corresponding functional constraint then becomes:

$$\psi_1 \equiv \int_0^{0.832} \langle\langle H \rangle\rangle \, dt = 0 \qquad (6.22)$$

Table 6.7

Static Analysis Results

Body No.I	$\dot{X}(I)$	$\dot{y}(I)$	$\dot{\phi}(I)$	X(I)	y(I)	$\phi(I)$
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	-0.65192	0.25705	-0.25425×10^{-1}
3	0.0	0.0	0.0	-0.48372	0.68102	-0.21228×10^{-1}
4	0.0	0.0	0.0	-0.50766	0.88885	0.42143
5	0.0	0.0	0.0	-0.38430	0.92972	0.11154
6	0.0	0.0	0.0	-0.34308	0.77348	0.50330
7	0.0	0.0	0.0	1.34860	0.71810	0.0

JOINT NO. I	FIRST LAGRANGE MULTIPLIER: $\mu(2I-1)$	SECOND LAGRANGE MULTIPLIER: $\mu(2I)$
1	-135.66	32.425
2	-823.83	-341.46
3	-869.47	-313.56
4	-869.47	-338.66
5	- 45.461	93.403
6	769.47	-257.18
7	-595.84	397.58

Table 6.7 (cont'd)

Spring-Damper No. I	SK(I)	DC(I)	SDL(I)	V(I)	FFX(I)	FFY(I)
1	40010.08	0.0	0.60128×10^{-1}	0.0	45.641	-72.262
2	0.0	0.0	4.7332	0.0	0.0	0.0
3	1.0×10^{6}	6.0×10^{4}	0.15734	0.0	859.49	19.771
4	0.0	0.0	0.26491	0.0	0.0	0.0
5	0.0	0.0	0.1512×10^{-1}	0.0	0.0	0.0
6	1.0×10^{5}	0.0	0.199	0.0	100.0	-3.2402

Table 6.8

Static Sensitivity Analysis Results

*DNDB(I,1), I = 1, JS1Z; (JS1Z = 90):

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.50707	0.89631	-1.9438	0.0	0.0	0.0	1.3671	0.50886	-2.1901
0.0	0.0	0.0	-0.28976	6.5612	41.333	0.0	0.0	0.0
-1.1210	3.9992	-46.817	0.0	0.0	0.0	1.0479	-0.8643×10^{-1}	2.4447
0.0	0.0	0.0	1.4484	-0.2526×10^{-15}	0.0	23453.0	18637.0	23215.0
-15546.0	-9331.4	43086.0	-9331.4	43086.0	-32546.0	58631.0	9331.4	-43078.0
7.9159	-424.75	9.8362	0.0	32546.0	-58631.0	0.58507	0.0	0.0
0.0	0.4673×10^{-1}	0.0	-46668.0	-3083.2	-0.32296	0.0	0.0	0.0
4.2676	0.0	0.0	0.0	-0.2564×10^{-5}		-0.6273×10^{-10}	-7.9159	0.0
0.0	0.0	0.0	0.0	0.0		0.0	0.0	0.0

DNDB(I,2), I = 1, JS1Z; (JS1Z = 90):

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.15568×10^{-1}	-0.27519×10^{-1}	0.5968×10^{-1}	0.0	0.0	0.0	-0.4198×10^{-1}	-0.1562×10^{-1}	0.6724×10^{-1}
0.0	0.0	0.0	0.8896×10^{-2}	-0.2015	-1.269	0.0	0.0	0.0
0.3442×10^{-1}	-0.1228	1.4374	0.0	0.0	0.0	-0.3217×10^{-1}	0.2654×10^{-2}	-0.7506×10^{-1}
0.0	0.0	0.0	-0.4447×10^{-1}	0.7725×10^{-17}	0.0	-720.06	-572.19	-712.75
477.29	286.49	-1322.8	286.49	-1322.8	999.25	-1800.1	-286.49	1322.6
-0.2430	13.041	-0.302	0.0	-999.25	1800.1	-0.1796×10^{-1}	0.0	0.0
0.0	-0.1435×10^{-2}	0.0	1432.8	94.661	0.9916×10^{-2}	0.0	0.0	0.0
-0.1310	0.0	0.0	0.0	0.7871×10^{-7}	0.0	-0.4269×10^{-11}	0.243	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

* See Chapter V for the variable names.

Table 6.8 (Cont'd.)

DNDB(1,3), I = 1, JSIZ; (JSIZ = 90):

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.24×10^{-3}	0.4242×10^{-3}	-0.92×10^{-3}	0.0	0.0	0.0	0.647×10^{-3}	0.2408×10^{-3}	-0.1037×10^{-2}
0.0	0.0	0.0	-0.1371×10^{-3}	0.3105×10^{-2}	0.1956×10^{-1}	0.0	0.0	0.0
-0.5305×10^{-3}	0.1893×10^{-2}	-0.2216×10^{-1}	0.0	0.0	0.0	0.4959×10^{-3}	-0.4090×10^{-4}	0.1157×10^{-2}
0.0	0.0	0.0	0.6854×10^{-3}	-0.1195×10^{-18}	0.0	11.099	8.8196	10.986
-7.3568	-4.4159	20.39	-4.4159	20.39	-15.402	27.747	4.4159	-20.386
0.3746×10^{-2}	-0.2010	0.4655×10^{-2}	0.0	15.402	-27.747	0.2769×10^{-3}	0.0	0.0
0.0	0.2211×10^{-4}	0.0	-22.085	-1.4591	-0.1528×10^{-3}	0.0	0.0	0.0
0.202×10^{-2}	0.0	0.0	0.0	-0.1213×10^{-8}	0.0	0.2144×10^{-13}	-0.3746×10^{-2}	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

where

$$H \equiv 0.2508 - [Y(2) + XF1(6)*DSIN(PHI(2)) + YF1(6)*D\,COS\,(PHI(2))] \tag{6.23}$$

and

$$<<H>> = \begin{cases} 0\,, & \text{for } 0 \leq t \leq t_1 = 0.6 \text{ and for } H \leq 0 \\ H^2, & \text{for } H \geq 0 \end{cases} \tag{6.24}$$

The computer variables used in Eq. (6.23) are explained in Chapter V.

For this case a design reduction ratio of 5% is used and the bounds b^L and b^U are kept unchanged. The initial estimate B^I is taken as $[0.566 \times 10^{-2},\ 0.181361 \times 10^{-1},\ 0.1 \times 10^{2},\ 0.167471 \times 10^{4}]^T$. In this case, optimum results are obtained in the first iteration, with $||\delta b^1|| = 0.0$ and $||\delta b^2|| = 0.2011 \times 10^{-4}$. The final results are given in Table 6.10.

It should be remarked here that although the results in Tables 6.9 and 6.10 are quite similar, the results of Table 6.9 are preferable, because the constraint on inclination of the plow-share is more important than the one on the height of the plow-share tip.

In order to deal with a more meaningful problem of iterative optimal design, the maximum inclination of the plow-share during the interval $0.6 \leq t \leq 0.832$ of the re-entering phase is reduced to 0.017453 radians. Then Eqs. (6.15) and (6.18) reduce to

$$-\phi_2 - 0.017453 \leq 0\,, \quad 0.6 \leq t \leq 0.832 \tag{6.25}$$

and

Table 6.9

Optimum Results for the Spring-Reset Plow-Share Mechanism for the Time Interval [0.0,0.832] (with 10° as the Maximum Allowable Inclination of the Plow-Share During Re-entering Phase)

Lower bounds on b $= [0.2 \times 10^{-2},\ 0.1 \times 10^{-1},\ 0.8 \times 10^{1}]^T$

Upper bounds on b $= [0.1 \times 10^{-1},\ 0.1,\ 0.14 \times 10^{2}]^T$

Real Cost Function: $\bar{J} = \max_{0 \le t \le 0.832} \frac{Gb_1^4}{8b_2^3 b_3} (\ell_1 - \ell_0^1)^2$

	Starting Values	Optimum Values
b_1	0.56604×10^{-2}	0.564312×10^{-2}
b_2	0.181361×10^{-1}	0.181409×10^{-1}
b_3	0.1×10^{2}	0.1×10^{2}
b_4	0.167471×10^{4}	0.165796×10^{4}

Table 6.10

Optimum Results for the Spring-Reset
Plow-Share Mechanism with
Modified Functional Constraint

Real Cost Function: $\bar{J} = \max_{0 \le t \le 0.832} \frac{G\, b_1^4}{8\, b_2^3\, b_3} (\ell_1 - \ell_0^1)^2$

Lower bounds on b = $[0.2 \times 10^{-2},\ 0.1 \times 10^{-1},\ 0.8 \times 10^{1}]^T$

Upper bounds on b = $[0.1 \times 10^{-1},\ 0.1,\ 0.14 \times 10^{2}]^T$

	Starting Values	Optimum Values
b_1	0.566×10^{-2}	0.567958×10^{-2}
b_2	0.181361×10^{-1}	0.181315×10^{-1}
b_3	0.1×10^{2}	0.1×10^{2}
b_4	0.167471×10^{4}	0.167471×10^{4} (app.)

$$\psi_1 \equiv \int_0^{0.832} << - \phi_2 - 0.017453 >> dt = 0 \qquad (6.26)$$

respectively. All other equations in the set of Eqs. (6.12) to (6.20) remain unaltered.

For this case a design reduction ratio of 0.25% is used and the bounds b^L and b^U are kept unchanged. The initial estimate b^I is taken as $[0.56604 \times 10^{-2}, 0.181361 \times 10^{-1}, 0.1 \times 10^{2}, 0.16000 \times 10^{4}]^T$. After three iterations the convergence criteria are satisfied and the optimum results are reached. Tables 6.11 and 6.12 present the pertinent numerical results of interest for various iterations.

Table 6.11

Numerical Results for the Spring-Reset Plow-Share Mechanism (with 0.017453 Radians as the Maximum Allowable Inclination of the Plow-Share During Re-entering Phase)

$b^I = [0.56604 \times 10^{-2},\ 0.18136 \times 10^{-1},\ 0.1 \times 10^{2},\ 0.1600 \times 10^{4}]^T$
Initial value of ψ_0 (actual) = 0.16029×10^4, SDLO(1) = 0.62264×10^{-1}

Iteration	ψ_1	ψ_2	ψ_0 (artificial)	ψ_0 (actual)	$\|\|\delta b^1\|\|$	$\|\|\delta b^2\|\|$	b_1	b_2	b_3	SDLO(1)
1	0.29109 x 10^{-2}	0.22046 x 10^{-6}	0.15960 x 10^4	0.15891 x 10^4	0.6252 x 10^{-3}	0.5900 x 10^{-1}	0.58703 23 x 10^{-2}	0.189700 x 10^{-1}	0.1 x 10^2	C.64364 x 10^{-1}
2	0.34400 x 10^{-2}	-o-	0.15920 x 10^4	0.15889 x 10^4	0.6266 x 10^{-3}	0.4240 x 10^{-4}	0.58700 75 x 10^{-2}	0.189698 x 10^{-1}	0.1 x 10^2 (app.)	0.64361 x 10^{-1}
3	0.25596 x 10^{-2}	-o-	0.158802 x 10^4	0.15889 x 10^4	0.6281 x 10^{-3}	0.5878 x 10^{-6}	0.58700 72 x 10^{-2}	0.189698 x 10^{-1}	0.1 x 10^2 (app.)	0.64361 x 10^{-1}

Table 6.12

Numerical Results for $\Delta\psi$ and $\ell^{\psi^T}\delta b$ for the Same Plow-Share Problem as in Table 6.11

Iteration	$\Delta\psi_1 = -\psi_1$	$\ell^{\psi_1^T}\delta b$	$\Delta\psi_2 = -\psi_2$	$\ell^{\psi_2^T}\delta b$
1	-0.29109×10^{-2}	-0.29110×10^{-2}	-0.22046×10^{-6}	-0.22046×10^{-6}
2	-0.34401×10^{-2}	-0.34401×10^{-2}	-o-	-o-
3	-0.25596×10^{-2}	-0.2560×10^{-2}	-o-	-o-

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

In this research, a systematic and unified theory for obtaining a corresponding computer program for dynamic analysis, static sensitivity analysis, general design sensitivity analysis, and optimal design of large scale constrained dynamic systems has been developed and demonstrated. Constrained mechanical systems with continuous motion, as in the case of the slider-crank mechanism of Chapter VI, and those with intermittent motion, as in the case of the plow-share mechanism discussed in Chapter VI, are handled with ease. The stiff integration (Gear) algorithm [6,2] and sparse matrix techniques [6,10,60,61,62] have been successfully employed in the DADS program that is presented here. The DADS program is presently configured for two-dimensional systems, but the extension to three-dimensional systems is theoretically straightforward.

It was noted in Chapter V that the DADS computer program is capable of dealing with mechanical systems with discrete rigid bodies only. If necessary, some amount of deformability in special elements of mechanisms can be introduced, with the help of additional spring-damper pairs. Otherwise, finite element techniques are to be introduced for the deformable elements and the Jacobian should be constructed accordingly. Then all types of analyses treated here can be performed in a routine manner. However,

for average-shaped mechanisms, introduction of deformability through spring-dampers is recommended. For structural problems, when the forcing functions depend on time and on solution variables, the conventional way of obtaining global solutions through modal analysis and Duhamel Integrals [54] fails and the techniques employed in this development may be employed. For structures such as automobiles, where finite element techniques are to be used for chassis and frames, in conjunction with the motion of rigid body mechanisms in the suspension systems, sparse matrix techniques and stiff integration (Gear) algorithms are indispensible.

Looking back into the details of the DADS computer program, it is the opinion of the author that some modifications and refinements, particularly in the field of numerical integration, are necessary to make it more efficient. Among other refinements, the following are under active consideration:

(1) Storage of the values of only those solution variables and related data in a disk in transient analysis, that are to be used in adjoint analysis. This then becomes a problem-dependent affair.

(2) Allotment of only one disk for storing the solution data set of the adjoint equations corresponding to all violated constraints in the sensitivity analysis.

APPENDIX

MATHEMATICAL RELATIONS USED IN THE SUBROUTINES RELATE, DGDBZ, ADLDZ, AND DLDFB FOR THE EXAMPLE PROBLEMS

In the following, symbols of Chapter II and V and data given in Chapter VI have been used.

The Slider-Crank Mechanism

Subroutine RELATE:

From Fig. 2.5 and the initial data (Table 6.1), one obtains

$$J_3 = \frac{1}{3} M_3 b_3^2$$

$$K^1 = b_1$$

$$YF1(1) = b_2$$

$$YF2(1) = b_2 - 0.5$$

$$X1(3) = b_3$$

$$X2(2) = -b_3$$

$$X(3) = b_3 \cos \phi_3$$

$$X(4) = 2.D0 \times X(3)$$

Subroutine DGDBZ:

From Eqs. (6.6) to (6.9)

$$DGDB(1) = DGDB(2) = DGDB(3) = DGDB(4) = 0$$

for all the four functional constraints. Only for the cost function

$$DGDB(4) = 1.0 \quad \text{and} \quad DGDB(I) = 0, \quad I = 1,2,3 \ .$$

Also for the fourth constraint, the only nonzero component for DGDZM is given by

$$DGDZM(9) = 1.0 \ .$$

Subroutine ADLDZ:

From Eq. (6.6), one obtains for the first constraint (since μ_2 is the 26-th solution variable),

```
DLDZ(26) = -A1*(1.D0 + DSIGN(1.D0,A1))/10.D0
```

where

$$A1 = -\left(\frac{\mu_2}{10.D0} + 1.D0\right) + \varepsilon \ ,$$

ε being the ε-active constant. All other compontnets are zero. D after decimal point indicates double precision. The function $DSIGN(a_1,a_2)$, a_1,a_2 being the arguments, is defined as follows:

$$DSIGN(a_1,a_2) = |a_1| \ sgn(a_2) \ ,$$

where

$$\text{sgn}(a_2) = \begin{cases} 1 & \text{for} \quad a_2 > 0 \\ 1 & \text{for} \quad a_2 = 0 \\ -1 & \text{for} \quad a_2 < 0 \end{cases}$$

From Eq. (6.7), one obtains for the second constraint the only nonzero component as

```
DLDZ(26) = A1*(1.D0 + DSIGN(1.D0,A1))/10.D0
```

where

$$A1 = \frac{\mu_2}{10.D0} - 1.D0 + \varepsilon \quad .$$

From Eq. (6.8), one obtains for the third constraint the only nonzero component as

```
DLDZ(33) = 2.D0*b1*A1*A2*(1.D0+ DSIGN(1.D0,A2))/b4
```

where

$$A1 = \ell_1 - \ell_0^1$$

$$A2 = \frac{b_1 * (A1)^2}{b_4} - 1.D0 + \varepsilon$$

From Eq. (6.9) all components of DLDZ are zero for the fourth constraint.

Subroutine DLDFB:

The nonzero components of DLDB, DFDB, DPHDB, and DZIDB are given by the following relations. For the third constraint

$$DLDB(1) = (A1)^2 * A2 * (1.D0 + DSIGN(1.D0, A2))/b_4 \quad ,$$

$$DLDB(4) = -A2 * (1.D0 + DSIGN(1.D0, A2)) * b_1 * (A1)^2/(b_4)^2 \quad ,$$

where

$$A1 = \ell_1 - \ell_0^1$$

$$A2 = \frac{b_1 * (A1)^2}{b_4} - 1.D0 + \varepsilon$$

From the equations of motion, one obtains

$$DFDB(15,3) = -(\mu_3 + \mu_5) \sin \phi_3 + (\mu_4 + \mu_6) \cos \phi_3$$

$$DFDB(21,2) = F_X^1 \cos \phi_4 + F_Y^1 \sin \phi_4$$

$$DFDB(3,2) = -F_X^1 \cos \phi_1 - F_Y^1 \sin \phi_1$$

From the equations of constraints, one obtains

$$DPHDB(3,3) = \cos \phi_3 = DPHDB(5,3)$$

$$DPHDB(4,3) = \sin \phi_3 = DPHDB(6,3)$$

From the spring-damper relations, one obtains

$$DZIDB(3,1) = \frac{\ell_1 - \ell_0^1}{\ell_1} * XI$$

$$DZIDB(4,1) = \frac{\ell_1 - \ell_0^1}{\ell_1} * YI$$

$$DZIDB(1,2) = \frac{[XI*(-\sin\phi_1 + \sin\phi_4) + YI*(\cos\phi_1 - \cos\phi_4)]}{[(XI)^2 + (YI)^2]^{1/2}}$$

$$DZIDB(3,2) = b_1 \frac{(\ell_1 - \ell_0^1)}{\ell_1} * (-\sin\phi_1 + \sin\phi_4)$$

$$DZIDB(4,2) = b_1 \frac{(\ell_1 - \ell_0^1)}{\ell_1} * (\cos\phi_1 - \cos\phi_4)$$

where

$$XI = XG1 - XG2, \quad YI = YG1 - YG2$$

$$XG1 = XF1(1) * \cos\phi_1 - YF1(1) * \sin\phi_1 + X(1)$$

$$YG1 = XF1(1) * \sin\phi_1 + YF1(1) * \cos\phi_1 + Y(1)$$

$$XG2 = XF2(1) * \cos\phi_4 - YF2(1) * \sin\phi_4 + X(4)$$

$$YG2 = XF2(1) * \sin\phi_4 + YF2(1) * \cos\phi_4 + Y(4)$$

The Plow-Share Mechanism
(with re-entering angle of 0.0174533 radians)

Subroutine RELATE:

From the initial data and definition of the design parameters (Chapter VI), one obtains (see Eqs. (6.14) and (6.12),

$$K^1 = 23.25 \times 10^8 \times \frac{b_1^4}{b_3\, b_2^3}$$

$$\ell_0^1 = b_1\, b_3 + 0.56604 \times 10^{-2}$$

Subroutine DGDBZ:

In this case the only nonzero component of DGDB is given by

DGDB(4) = 1.0

for the cost functional only.

All the components of DGDZM are zero.

Subroutine ADLDZ:

The nonzero components of DLDZ are given by the following. From Eq. (6.18), one obtains for the first constraint,

DLDZ(12) = -A1 × (1.D0 + DSIGN(1.D0,A1)) , for t > 0.6D0 ,

where

$$A1 = -\phi_2 - 0.0174533D0 + \varepsilon$$

From Eq. (6.19), one obtains for the second constraint

DLDZ(57) = 2.D0 * K^1 * A1 * A2 * (1.D0 + D SIGN(1.D0,A2))/b_4

where

$$A1 = \ell_1 - \ell_0^1$$

$$A2 = \frac{K^1 * (A1)^2}{b_4} - 1.D0 + \varepsilon$$

Subroutine DLDFB:

The nonzero components of DLDB, DFDB, DPHDB, and DZIDB are given by the following relations. From Eq. (6.14)

$$\frac{\partial K^1}{\partial b_1} = 93.D08 * \frac{b_1^3}{b_3\, b_2^3}$$

$$\frac{\partial K^1}{\partial b_2} = -69.75D08 * \frac{b_1^4}{b_3\, b_2^4}$$

$$\frac{\partial K^1}{\partial b_3} = -23.25D08 \quad \frac{b_1^4}{b_3^2\, b_2^3}$$

Then for the second constraint

$$DLDB(1) = A3 * ((A1)^2 * \frac{\partial K^1}{\partial b_1} - 2.D0 * K^1 * A1 * b_3)/b_4$$

$$DLDB(2) = A3 * (A1)^2 * \frac{\partial K^1}{\partial b_2}$$

$$DLDB(3) = A3 * ((A1)^2 * \frac{\partial K^1}{\partial b_3} - 2.D0 * K^1 * A1 * b_1)/b_4$$

$$DLDB(4) = -K^1 * A3 * \frac{(A1)^2}{b_4^2}$$

where

$$A1 = \ell_1 - \ell_0^1$$

$$A2 = \frac{K^1 * (A1)^2}{b_4} - 1.D0 + \varepsilon$$

$$A3 = A2 * (1.D0 + DSIGN(1.D0,A2))$$

From the relations corresponding to the first-spring-damper pair, one obtains

$$DZIDB(3,1) = DL1 * \frac{\partial K^1}{\partial b_1} - DL3 * b_3$$

$$DZIDB(3,2) = DL1 * \frac{\partial K^1}{\partial b_2}$$

$$DZIDB(3,3) = DL1 * \frac{\partial K^1}{\partial b_3} - DL3 * b_1$$

$$DZIDB(4,1) = DL2 * \frac{\partial K^1}{\partial b_1} - DL4 * b_3$$

$$DZIDB(4,2) = DL2 * \frac{\partial K^1}{\partial b_2}$$

$$DZIDB(4,3) = DL2 * \frac{\partial K^1}{\partial b_3} - DL4 * b_1$$

where

$$DL1 = \frac{(\ell_1 - \ell_0^1)}{\ell_1} * XI$$

$$DL2 = \frac{(\ell_1 - \ell_0^1)}{\ell_1} * YI$$

$$DL3 = K^1 * \frac{XI}{\ell_1}$$

$$DL4 = K^1 * \frac{YI}{\ell_1}$$

$$XI = XG1 - XG2 , \quad YI = YG1 - YG2$$

$$XG1 = XF1(1) * \cos \phi_4 = YF1(1) * \sin \phi_4 + X_4$$

$$YG1 = XF1(1) * \sin \phi_4 + YF1 (1) * \cos \phi_4 + Y_4$$

$$XG2 = XF2(1) * \cos \phi_6 - YF2(1) * \sin \phi_6 + X_6$$

$$YG2= XF2(1) * \sin \phi_6 + YF2(1) * \cos \phi_6 + Y_6$$

REFERENCES

1. Haug, E.J., Jr., "Computer Aided Design of Mechanical Systems", Army Materiel Command Engineering Design Handbook, AMCP 706-192, 1973.

2. Gear, C.W., "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice-Hall, Inc., 1971.

3. Lambert, J.D., "Computational Methods in Ordinary Differential Equations", John Wiley & Sons, New York, 1973.

4. Lapidus, L., and Seinfeld, J.H., "Numerical Solution of Ordinary Differential Equations", Academic Press, New York, 1971.

5. Willoughby, R.A., "Stiff Differential Systems", Plenum Press, New York and London, 1974.

6. Chua, L.O., and Lin, P.M., "Computer-Aided Analysis of Electronic Circuits", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

7. Orlandea, N., "Node Analogous, Sparsity-Oriented Methods for Simulation of Mechanical Dynamic Systems", Ph.D. Thesis, University of Michigan, Ann Arbor, Mich., 1973.

8. Orlandea, N., Chace, M.A., and Calahan, D.A., " A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems, Parts I and II", Journal of Engineering for Industry, ASME paper No. 76-DET-19/20, 1976.

9. Chace, M.A., and Sheth, P.N., "Adaptation of Computer Techniques to the Design of Mechanical Dynamic Machinery", ASME paper 73-DET-58, ASME Meeting, Cincinnati, Ohio, September 9-12, 1973.

10. Calahan, D.A., "Computer-Aided Network Design, 2nd ed., McGraw-Hill, New York, 1972.

11. Carnaham, B., Luther, H.A., and Wilkes, J.O., "Applied Numerical Methods", Wiley, New York, 1969.

12. Calahan, D.A., Grapes, T.E., and Korybalski, M.E., Private Communications.

13. ASTAP (Advanced Statistical Analysis Program), developed by IBM System Products Division, East Fishkill, New York, April 20, 1973.

14. Wehage, R.A., "ADAMS 2-D Automatic Dynamic Analysis of Mechanical Systems - Two Dimensional", M.S. Report, Iowa State University, Ames, Iowa,

15. Sheth, P.N., "A Digital Computer-Based Simulation Procedure for Multiple Degree of Freedom Mechanical Systems with Geometric Constraints", Ph.D. Thesis, University of Wisconsin, Madison, Wis., 1972.

16. Sheth, P.N., and Uicker, J.A., Jr., "IMP (Integrated Mechanisms Program): A Computer-Aided Design Analysis System for Mechanisms and Linkages", Trans., ASME 94, Ser. B, J. Eng. Indust., 454-464, 1972.

17. Chace, M.A., and Angell, J.C., "User's Guide to DRAM (Dynamic Response of Articulated Machinery)", Design Engineering Computer Aids Laboratory, Department of Mechanical Engineering, University of Michigan, 1973.

18. Chace, M.A., and Bayazitoglu, Y.O., "Development and Application of a Generalized D'Alembert Force for Multifreedom Mechanical Systems", J. Eng. Indust., 317-327, 1971.

19. Dix, R.C., and Lehman, T.J., "Simulation of the Dynamics of Machinery", J. Eng. Indust., Trans. ASME, Ser. B, 94, 433-438, 1972.

20. Andrews, G.C., and Kesavan, H.K., "The Vector Network Model: A New Approach to Vector Dynamics", Mechanism and Machine Theory, 10 (1), 57-75, 1975.

21. Paul, B., and Krajcinovic, D., "Computer Analysis of Machines with Planar Motion - 2. Dynamics", J. Appl. Mech. 37, 703-712, Trans. ASME 92, Ser. E, 1970.

22. Paul, B., "Analytical Dynamics of Mechanisms - A Computer Oriented Overview", Mechanism and Machine Theory, Vol. 10, 481-507, 1975.

23. Seireg, A., "A Survey of Optimization of Mechanical Design", J. Eng. Indust., Vol. 94, Trans. ASME, 495-499, 1972.

24. Haug, E.J., Jr., and Arora, J.S., "Optimal Mechanical Design Techniques Based on Optimal Control Methods", Proceedings of the First ASME Design Technology Transer Conference, 65-73, October 1974.

25. Sevin, E., and Pilkey, W., "Computational Approaches to the Min-Max Response of Dynamic Systems with Incompletely Prescribed Input Functions", J. Appl. Mech., Vol. 34, Trans., ASME, 87-90, March 1967.

26. Fiacco, A.V., and McCormick, G.P., "The Sequential Unconstrained Minimization Technique for Nonlinear Programming, a Primal Dual Method", Management Science, Vo. 10, pp.360-366, 1964.

27. Sevin, E., Pilkey, W.D., and Kalinowski, A.J., "Optimum Performance Bounds and Synthesis of Dynamic Systems", Computational Approaches in Applied Mechanics, ASME, New York, 107-132, 1970.

28. Sevin, E., and Pilkey, W.D., "Optimum Shock and Vibration Isolation", The Shock and Vibration Information Center, United States Department of Defense, 1971.

29. Bellman, R., and Dreyfus, S., "Applied Dynamic Programming", Princeton University Press, Princeton, N.J., 1962.

30. Brock, J.E., "A note on the Damped Vibration Absorber", J. Appl. Mech., Vol. 13, Trans. ASME, A-284, 1946.

31. Den Hartog, J.P., "Mechanical Vibrations", McGraw-Hill Book Company Inc., New York, 1956.

32. Hamad, B.M., "Optimum Design of Vibratory Systems', Ph.D. Thesis, University of Wisconsin, 1968.

33. Arora, J.S., Rim, K., and Kwak, B.M., "Optimum Design of Damped Vibration Absorber", Technical Report No. 7, Department of Mechanics and Hydraulics, College of Engineering, University of Iowa, Iowa City, Iowa, July 1973.

34. Afimiwala, K.A., and Mayne, R.W., "Optimum Design of an Impact Absorber", J. Eng. Indust., Vol. 96, Trans. ASME, 124-130, February 1974.

35. Willmert, K.D., and Fox, R.L., "Optimum Design of a Linear Multi-Degree of Freedom Shock Isolation System", J. Eng. Indust., Vol. 94, Trans. ASME, 465-47;. May 1972.

36. McMunn, J., "Multi-Parameter Optimum Damping in Linear Dynamical Systems", Ph.D. Thesis, University of Minnesota, 1967.

37. Kwak, B.M., "Parametric Optimal Design", Ph.D. Thesis, University of Iowa, December 1974.

38. Hsiao, M.H., "Mechanical Design Optimization for Transient Dynamic Response", Ph.D. Thesis, University of Iowa, May 1976.

39. Haug, E.J., Jr., and Arora, J.S., "A State-Space Method for Optimization of Weapon System Dynamics", Proceedings, First Conference on Dynamics of Precision Gun Weapons, Rock Island, Illinois, January 1977.

40. Huang, R.C., "Optimal Design of Mechanical Systems with Intermittent Motion", Ph.D. Thesis, University of Iowa, May 1976.

41. Huang, R.C., Haug, E.J., Jr., and Andrews, J.G., "Optimal Design of Mechanical Systems with Intermittent Motion, Technical Report No. 24, Division of Materials Engineering, University of Iowa, Iowa City, Iowa, 1976.

42. Chua, L.O., "Introduction to Nonlinear Network Theory", McGraw-Hill Book Company, New York, 1969.

43. Hilderbrand, F.B., "Introduction to Numerical Analysis", McGraw-Hill Book Company, New York, 1956.

44. Conte, S.D., and De Boor, Carl, "Elementary Numerical Analysis - An Algorithmic Approach", McGraw-Hill Book Company, New York, 1972.

45. Ralston, A., "A First Course in Numerical Analysis", McGraw-Hill Book Company, New York, 1965.

46. Gear, C.W., "Simultaneous Numerical Solution of Differential - Algebraic Equations", Trans. Circuit Theory, Vol. CT-18, No. 1, 89-95, January 1971.

47. Grenin, Y., "A New Approach to the Synthesis of Stiffly Stable Linear Multistep Formulas", IEEE Trans. Circuit Theory, 352-360, July 1973.

48. Gear, C.W., "The Automatic Integration of Stiff Ordinary Differential Equations", Information Processing 68, A.J.H. Morrell, Ed., North Holland Publishing Company, Amsterdam, The Netherlands, 187-193, 1969.

49. Hachtel, G.D., Brayton, R.K., and Gustavson, F.G., "The Sparse Tableau Approach to Network Analysis and Design", Trans. Circuit Theory, Vol. CT-18, No. 1, 101-113, January 1971.

50. Orlandea, N., Chace, M.A., and Calahan, D.A., "Sparsity Oriented Methods for Simulation of Mechanical Dynamic Systems", Proceedings of the sixth annual Princeton Conference on Information Sciences and Systems, Department of Electrical Engineering, Princeton University, Princeton, N.J., 32-36, 1972.

51. Calahan, D.A., and Orlandea, N., "A Program for the Analysis and Design of General Dynamic Mechanical Systems", AFIPS Conference Proceedings, Vol. 41, Part II, Fall Joint Computer Conference, Anaheim, California, 885-888, 1972.

52. Strang, W.G., and Fix, G.J., "An Analysis of the Finite Element Method", Prentice-Hall Book Company, Englewood Cliffs, N.J., 1973.

53. Zienkiewicz, O.C., "The Finite Element Method in Structural and Continuum Mechanics", McGraw-Hill Book Company, New York, 1967.

54. Clough, R.W., and Penzien, J., "Dynamics of Structures", McGraw-Hill Book Company, New York, 1975.

55. Crout, P.D., "A Short Method for Evaluating Determinants and Solving Systems of Linear Equations with Real or Complex Coefficients", AIEE Trans., Vol. 69, 1235-1241, 1941.

56. Calahan, D.A., Grapes, T.E., Donahey, J., and Orlandea, N., "Description of a Sparse Matrix Compiler with Applications", AFOSR-TR-71-2676, October 11, 1971.

57. Berry, R.D., "An Optimum Ordering of Electronic Circuit Equations for a Sparse Matrix Solution", IEEE Trans. Circuit Theory, Vol. CT-18, 40-50, January 1971.

58. Tinney, W.F., and Walker, J.W., "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization", Proc. IEEE, Vol. 55, 1801-1809, November 1967.

59. Markowitz, H.M., "The Elimination Form of the Inverse and Its Application to Linear Programming", Management Sci., Vol. 3, 255-269, 1957.

60. Proceedings, Sparse Matrix Symposium, IBM Rep. RA-1, March 1969.

61. IBM System 360 and System 370 Subroutine Library - Mathematics (SL-MATH), No. 5736-XM7, 1971. (Rental Software available from IBM).

62. Calahan, D.A., Buning, P.G., and Joy, W.N., "Vectorized General Sparsity Algorithms with Backing Store", Systems Engineering Laboratory (SEL), Report No. 96, University of Michigan, Ann Arbor, Michigan, January 1977.

63. Haug, E.J., Jr., "Computational Methods in Mechanical System Dynamics", The University of Iowa, February 1978.

64. Bunch, J.R., and Rose, D.J., "Sparse Matrix Computations", Academic Press Inc., New York, 1976.

65. Tewarson, R.P., "Sparse Matrices", Academic Press Inc., New York, 1976.

66. Goldstein, H., "Classical Mechanics", Addison-Wesley Publishing Company Inc., Cambridge, Mass., 1953.

67. Greenwood, D.T., "Classical Dynamics", Prentice-Hall Inc., Englewood Cliffs, N.J., 1977.

68. Kane, T.R., "Dynamics", Second Edition, Stanford University Press, Palo Alto, California, 1972.

69. Liusternik, L.A., and Sobolev, V.J., "Elements of Functional Analysis", Ungar, New York, 1961.

70. Lanczos, C., "Linear Differential Operators", Van Nostrand, London, 1961.

71. Mischke, C.R., "Elements of Mechanical Analysis", Addison-Wesley Publishing Company, Inc., Massachusetts, 1963.

72. "Mark's Standard Handbook for Mechanical Engineers", International Student Edition, McGraw-Hill Book Company, New York.

73. Businger, P.A., "Monitoring the Numerical Stability of Gaussian Elimination", Numer. Math. 16, 360-361, 1971.

74. Erisman, A.M., and Reid, J.K., "Monitoring the Stability of the Triangular Factorization of a Sparse Matrix", Numer. Math. 22, 183-186, 1974.

DISTRIBUTION LIST

Please notify USATACOM, DRSTA-ZSA, Warren, Michigan 48090, of corrections and/or changes in address.

Commander (25)
US Army Tk-Autmv Command
R&D Center
Warren, MI 48090

Superintendent (02)
US Military Academy
ATTN: Dept of Engineering
Course Director for
Automotive Engineering

Commander (01)
US Army Logistic Center
Fort Lee, VA 23801

US Army Research Office (02)
P.O. Box 12211
ATTN: Dr. F. Schmiedeshoff
Dr. R. Singleton
Research Triangle Park, NC 27709

HQ, DA (01)
ATTN: DAMA-AR
Dr. Herschner
Washington, D.C. 20310

HQ, DA (01)
Office of Dep Chief of Staff
for Rsch, Dev & Acquisition
ATTN: DAMA-AR
Dr. Charles Church
Washington, D.C. 20310

HQ, DARCOM
5001 Eisenhower Ave.
ATTN: DRCDE
Dr. R.L. Haley
Alexandria, VA 22333

Director (01)
Defense Advanced Research
Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Commander (01)
US Army Combined Arms Combat
Developments Activity
ATTN: ATCA-CCC-S
Fort Leavenworth, KA 66027

Commander (01)
US Army Mobility Equipment
Research and Development Command
ATTN: DRDME-RT
Fort Belvoir, VA 22060

Director (02)
US Army Corps of Engineers
Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

Commander (01)
US Army Materials and Mechanics
Research Center
ATTN: Mr. Adachi
Watertown, MA 02172

Director (03)
US Army Corps of Engineers
Waterways Experiment Station
P.O. Box 631
ATTN: Mr. Nuttall
Vicksburg, MS 39180

Director (04)
US Army Cold Regions Research
& Engineering Lab
P.O. Box 282
ATTN: Dr. Freitag, Dr. W. Harrison
Dr. Liston, Library
Hanover, NH 03755

President (02)
Army Armor and Engineer Board
Fort Knox, KY 40121

Commander (01)
US Army Arctic Test Center
APO 409
Seattle, WA 98733

Commander (02)
US Army Test & Evaluation
Command
ATTN: AMSTE-BB and AMSTE-TA
Aberdeen Proving Ground, MD
21005

Commander (01)
US Army Armament Research
and Development Command
ATTN: Mr. Rubin
Dover, NJ 07801

Commander (01)
US Army Yuma Proving Ground
ATTN: STEYP-RPT
Yuma, AZ 85364

Commander (01)
US Army Natic Laboratories
ATTN: Technical Library
Natick, MA 01760

Director (01)
US Army Human Engineering Lab
ATTN: Mr. Eckels
Aberdeen Proving Ground, MD
21005

Director (02)
US Army Ballistic Research Lab
Aberdeen Proving Ground, MD
21005

Director (02)
US Army Materiel Systems
Analysis Agency
ATTN: AMXSY-CM
Aberdeen Proving Ground, MD
21005

Director (02)
Defense Documentation Center
Cameron Station
Alexandria, VA 22314

US Marine Corps (01)
Mobility & Logistics Division
Development and Ed Command
ATTN: Mr. Hickson
Quantico, VA 22134

Keweenaw Field Station (01)
Keweenaw Research Center
Rural Route 1
P.O. Box 94-D
ATTN: Dr. Sung M. Lee
Calumet, MI 49913

Naval Ship Research & (02)
Dev Center
Aviation & Surface Effects Dept
Code 161
Washington, D.C. 20034

Director (01)
National Tillage Machinery Lab
Box 792
Auburn, AL 36830

Director (02)
USDA Forest Service Equipment
Development Center
444 East Bonita Avenue
San Dimes, CA 91773

Engineering Societies (01)
Library
345 East 47th Street
New York, NY 10017

Dr. I.R. Erlich (01)
Dean for Research
Stevens Institute of Technology
Castle Point Station
Hoboken, NJ 07030

Grumman Aerospace Corp (02)
South Oyster Bay Road
ATTN: Dr. L. Karafiath
Mr. F. Markow
M/S A08/35
Bethpage, NY 11714

Dr. Bruce Liljedahl (01)
Agricultural Engineering Dept
Purdue University
Lafayette, IN 46207

Mr. H.C. Hodges (01)
Nevada Automotive Test Center
Box 234
Carson City, NV 89701

Mr. R.S. Wismer (01)
Deere & Company
Engineering Research
3300 River Drive
Moline, IL 61265

Oregon State University (01)
Library
Corvallis, OR 97331

Southwest Research Inst (01)
8500 Culebra Road
San Antonia, TX 78228

FMC Corporation (01)
Technical Library
P.O. Box 1201
San Jose, CA 95108

Mr. J. Appelblatt (01)
Director of Engineering
Cadillac Gauge Company
P.O. Box 1027
Warren, MI 48090

Chrysler Corporation (02)
Mobility Research Laboratory,
Defense Engineering
Department 6100
P.O. Box 751
Detroit, MI 48231

CALSPAN Corporation (01)
Box 235
Library
4455 Benesse Street
Buffalo, NY 14221

SEM, (01)
Forsvaretsforskningsanstalt
Avd 2
Stockholm 80, Sweden

Mr. Hedwig (02)
RU III/6
Ministry of Defense
5300 Bonn, Germany

Foreign Science & Tech (01)
Center
220 7th Street North East
ATTN: AMXST-GEI
Mr. Tim Nix
Charlottesville, VA 22901

General Research Corp (01)
7655 Old Springhouse Road
Westgate Research Park
ATTN: Mr. A. Viilu
McLean, VA 22101

Commander (01)
US Army Developmant and
Readiness Command
5001 Eisenhower Avenue
ATTN: Dr. R.S. Wiseman
Alexandria, VA 22333